UNIVERSITÄT DER BUNDESWEHR
FAKULTÄT FÜR LUFT- UND RAUMFAHRTTECHNIK
INSTITUT FÜR FLUGSYSTEME

Deep-learning Algorithmen für die Fahrzeugdetektion auf Luftbildern: Untersuchung der Einflussfaktoren und des Trainingsverhaltens bei synthetischem Datenmaterial

Michael Krump

Vollständiger Abdruck der bei der
Fakultät für Luft- und Raumfahrttechnik
der Universität der Bundeswehr München
zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Gutachter:

1. Univ.-Prof. Dr.-Ing. Peter Stütz
2. Univ.-Prof. Dr. Dr. h.c. Wolfram Hardt

Diese Dissertation wurde am 10.11.2022 bei der Universität der Bundeswehr München eingereicht und durch die Fakultät für Luft- und Raumfahrttechnik am 20.03.2023 angenommen. Die mündliche Prüfung fand am 20.03.2023 statt.

Zusammenfassung

Eine automatisierte Auswertung von Sensordaten mit *deep-learning* basierten Algorithmen spielt in vielfältigsten Anwendungsfällen eine immer wichtiger werdende Rolle. Aufgrund der dafür nötigen und teils schwierig zu generierenden Trainings- und Testdaten wird daher der Einsatz synthetischer Simulationsumgebungen zur künstlichen Generierung der Daten erwogen. Die vorliegende Arbeit beschäftigt sich am Beispiel der luftgestützten Fahrzeugdetektion mit den dabei zu beachtenden Randbedingungen und Einflussfaktoren und liefert Ansatzpunkte für einen optimierten Einsatz synthetischer Daten.

Das YOLOv3 Netzwerk dient dabei als Testalgorithmus zur Objektdetektion. Es wird unter anderem auf die Auswahl passender realer Benchmark-Datensätze und die automatisierte Generierung entsprechender synthetischer Daten auf Basis einer virtuellen Umgebung eingegangen. Erst die gezielte Erfassung realer Aufnahmen und die Erzeugung synthetischer Duplikate ermöglicht schließlich eine umfassende Evaluierung der erzeugten Modelle. Eine im Rahmen der Arbeit entwickelte Klassifikationskette wird zudem unter Verwendung von Bildbeschreibern als Merkmalen zur detaillierten Analyse des Reality Gaps zwischen realen und synthetischen Daten eingesetzt. Insgesamt werden mit diesem Aufbau drei grundlegende Fragestellungen untersucht.

Der erste Teil befasst sich mit der Trainingsdatenzusammensetzung und evaluiert dabei reale, synthetische und gemischte Trainingsdaten und die damit erzeugten Modelle. Im zweiten Schritt wird unter Verwendung der Klassifikationskette versucht, sowohl die Ursachen für Bildunterschiede zwischen den Daten als auch die Ursachen für Leistungsunterschiede bei Verwendung synthetisch trainierter Modelle auf realen Testdaten zu identifizieren. Im letzten Teil wird abschließend die Stabilität der trainierten Modelle in Bezug auf verschiedenste Parametervariationen untersucht und durch gezielte Variation der synthetischen Trainingsdatengenerierung auch dieser Bestandteil der Detektionskette analysiert.

Insgesamt liefert die Arbeit ein umfassendes Konzept, um das Verhalten von *deep-learning* basierten Detektionsalgorithmen bei der Verwendung synthetischer Sensordaten besser verstehen zu können und evaluiert es für den Anwendungsfall der UAV basierten Fahrzeugdetektion. Es werden Empfehlungen für die Trainingsdatenzusammensetzung abgeleitet, die für einen optimierten Einsatz synthetischer Daten wichtig sind. Zudem wird der *Reality Gap* in seiner Gesamtheit analysiert und gezeigt, dass dieser eine Richtungsabhängigkeit aufweist und sich aus einem *Content Gap* und einem *Appearance Gap* zusammensetzt. Aus den mit Hilfe der statistischen Auswertung identifizierten einflussreichen Bildeigenschaften werden schließlich Gestaltungsrichtlinien im Hinblick auf die Generierung synthetischer Daten mit virtuellen Simulationsumgebungen formuliert, die schließlich durch Trainingsdatenvariationen bestätigt werden konnten.

Abstract

Automated evaluation of sensor data with *deep-learning* based algorithms plays an increasingly important role in a wide range of applications. Due to the necessary and sometimes difficult to generate training and test data, the use of synthetic simulation environments for the artificial generation of data is being considered. Using the example of airborne vehicle detection, this thesis deals with the boundary conditions and influencing factors to be considered and provides starting points for an optimized use of synthetic data.

The YOLOv3 network serves as a test algorithm for object detection. Among other things, the selection of suitable real benchmark data sets and the automated generation of corresponding synthetic data on the basis of a virtual environment are discussed. Only the targeted acquisition of real images and the generation of synthetic duplicates finally allows a comprehensive evaluation of the generated models. Furthermore, a classification chain developed in the context of the thesis is used for a detailed analysis of the *Reality Gap* between real and synthetic data using image descriptors as features. Overall, three fundamental issues are investigated with this setup.

The first part deals with the training data composition, evaluating real, synthetic and mixed training data and the models generated with them. In the second part, using the classification chain, an attempt is made to identify both the causes of image differences between the data and the causes of performance differences when synthetically trained models are used on real test data. Finally, the stability of the trained models with respect to different parameter variations is investigated in the last part and this component of the detection chain is also analyzed by selective variation of the synthetic training data generation.

Overall, the work provides a comprehensive approach to better understand the behavior of *deep-learning* based detection algorithms when using synthetic sensor data and evaluates it for the use case of UAV based vehicle detection. Recommendations for the training data composition are derived, which are important for an optimized use of synthetic data. In addition, the *Reality Gap* is analyzed in its entirety and shown to exhibit directionality and to be composed of a *Content Gap* and an *Appearance Gap*. Finally, from the influential image properties identified by the statistical evaluation, design guidelines are derived with respect to the generation of synthetic data with virtual simulation environments, which could finally be confirmed by training data variations.

Inhaltsverzeichnis

Seite

Abkürzungsverzeichnis

AA	Anti-Aliasing
ADAS	Advanced Driving Assistance System
ANOVA	Analysis of Variance
AP	Average Precision
API	Application Programming Interface
ART	Angular Radial Transform
ATO	Above Take-Off Höhe
AUC	Area Under the Curve
BB	Bounding Box
BGC	Balanced Gradient Contribution
BLC	Bildbeschreibergruppe: Helligkeit / Luminanz / Kontrast
BLOB	Binary Large Objects
BRISQUE	Blind/Referenceless Image Spatial Quality Evaluator
CAD	Computer Aided Design
CBIR	Content-Based Image Retrieval
CDB	Common Database
Chrom.	Chrominanz
CLD	Color Layout Descriptor
CNN	Convolutional Neural Network
Col	Bildbeschreibergruppe: Farbe
CSD	Color Structure Descriptor
CShD	Contour Shape Descriptor
CSS	Curvature Scale Space
CV	Computer Vision
DA	Domain Adaptation oder Data Augmentation
DBN	Bildbeschreibergruppe: Verzerrung / Unschärfe / Rauschen
DCD	Dominant Color Descriptor
DCT	Discrete Cosine Transform
DPM	Deformable Part Model
DT	Decision Tree
EHD	Edge Histogram Descriptor
Env	Bildbeschreibergruppe: Umweltbedingungen
EO	Elektro-optisch
ET	Bildbeschreibergruppe: Ecken / Texturen
FCBF	Fast Correlation Based Feature Selection
FFT	Fast Fourier Transform
FI	Feature Importance Methoden
FN	False Negative
FOV	Field of View
FP	False Positive
FPN	Feature Pyramid Network
FS	Feature Selection Methoden
FT	Fine-Tuning
FXAA	Fast Approximate Anti-Aliasing
GAN	Generative Adversarial Network
GLCM	Grey Level Co-Occurrence Matrix

GPS	Global Positioning System
GSD	Ground Sample Distance
HiL	Hardware-in-the-Loop Simulation
HMMD	Hue-Max-Min-Diff
HOG	Histogram of Oriented Gradients
HSL	Hue-Saturation-Lightness Farbraum
HSV	Hue-Saturation-Value Farbraum
HT	Hypertexturen
HTD	Homogeneous Texture Descriptor
IMU	Inertial Measurement Unit
IoU	Intersection over Union
IQM	Bildbeschreibergruppe: Bildqualitätsmetriken
IR	Infrarot
LASSO	Least Absolute Shrinkage Selection Operator
LIME	Local Interpretable Model-Agnostic Explanations
LOD	Levels of Detail
LPB	Local Binary Pattern
Lum.	Luminanz
mAP	mean Average Precision
MLR	Multiple Lineare Regression
MOSART	Moderate Spectral Atmospheric Radiance and Transmittance
MOT	Multi-Objekt Tracking
MPEG	Motion Picture Experts Group
mRMR	min Redundancy Max Relevance
MSER	Maximum Stable Extremal Regions
MUM-T	Manned-Unmanned Teaming
M&S	Modelling and Simulation
NIMA	Neural Image Assessment
NMS	Non-maximum Suppression
NST	Neural Style Transfer
POI	Point of Interest
PR	Precision-Recall
PSR	Patch Shuffle Regularization
RF	Realflug
R-FCN	Region-based Fully Convolutional Network
RGB	Rot-Grün-Blau Farbbereich
RL	Reinforcement Learning
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristic
ROI	Region of Interest
ROS	Robot Operating System
RPN	Region Proposal Netzwerk
RSD	Region Shape Descriptor
RTK	Real Time Kinematik
SCD	Scalable Color Descriptor
SDK	Software Development Kit
SDR	Structured Domain Randomization
SFFS	Sequential Floating Forward Selection
SFS	Sequential Forward Selection

SGD	Stochastic Gradient Descent
Sha	Bildbeschreibergruppe: Formen
SHAP	Shapley Additive exPlanations
SIFT	Scale-Invariant Feature Transform
SMOTE	Synthetic Minority Oversampling Technique
SSD	Single Shot Detector
SSH	Secure Shell Netzwerkprotokoll
SURF	Speeded-Up Robust Feature
SVM	Support Vector Machine
SVR	Support Vector Regressor
TBD	Texture Browsing Descriptor
TN	True Negative
TP	True Positive
UART	Universal Asynchronous Receiver / Transmitter
UAV	Unmanned Aerial Vehicle
UTM	Universal Transverse Mercator
VBS	Virtual Battlespace
VIF	Variance Inflation Factor
XAI	Explainable Artificial Intelligence
YOLO	You Only Look Once; Objektdetektor

1 Einleitung

Vor allem in den letzten Jahren führten drastische Weiterentwicklungen in den Bereichen Sensorik, Navigation, Flugführung und Antriebstechnik dazu, dass der Betrieb von unbemannten Luftfahrzeugen flexibler, einfacher und kostengünstiger wurde. Dieser Trend bewirkt, dass bis 2030 für den deutschen Drohnenmarkt ein jährliches Wachstum von 14 Prozent prognostiziert wird, das vor allem steigenden kommerziellen Anwendungen geschuldet ist [1]. Im Jahr 2019 wurden in Deutschland fast 500 000 UAV (engl.: *Unmanned Aerial Vehicle*) betrieben [1]. Insbesondere als Träger von leistungsfähigen umweltbeobachtenden Sensorsystemen kommen UAVs in den vielfältigsten Anwendungsgebieten zum Einsatz und werden sowohl zivil als auch militärisch genutzt. Sie werden bei Überwachungsaufgaben [2, 3], bei der Inspektion von Infrastruktur [4, 5], in der Land- und Forstwirtschaft [6–8], für 2-D und 3-D Vermessungsaufgaben [9, 10] und im Bereich des Katastrophenschutzes [11, 12] eingesetzt. Die Vorteile solcher unbemannter Systeme sind eine allgemein hohe Flexibilität und Mobilität [13], Kostenreduktion und ihre Einsatzmöglichkeit in sicherheitskritischen Situationen.

Dabei wird auch bei den unbemannten Systemen zwischen Drehflügel und Starrflächenplattformen unterschieden. So eigenen sich erstere aufgrund der hohen Stabilität bei Bewegungen in jede Raumrichtung besonders für die Erfassung von Sensordaten bei geringen bis mittleren Flughöhen [14] und vergleichsweise kurzen Reichweiten und Einsatzdauern. Sie werden häufig auch als Multikopter bezeichnet und unterscheiden sich in der Anzahl an verbauter Rotoren, ihrer Größe und der technischer Ausstattung. Im Verlauf der Arbeit liegt der Fokus vermehrt auf dieser Art unbemannter Systeme. Starrflächenplattformen hingegen sind für höhere Fluggeschwindigkeiten und Flughöhen ausgelegt und werden daher meist für Missionen mit größerer Einsatzdauer und Reichweite herangezogen.

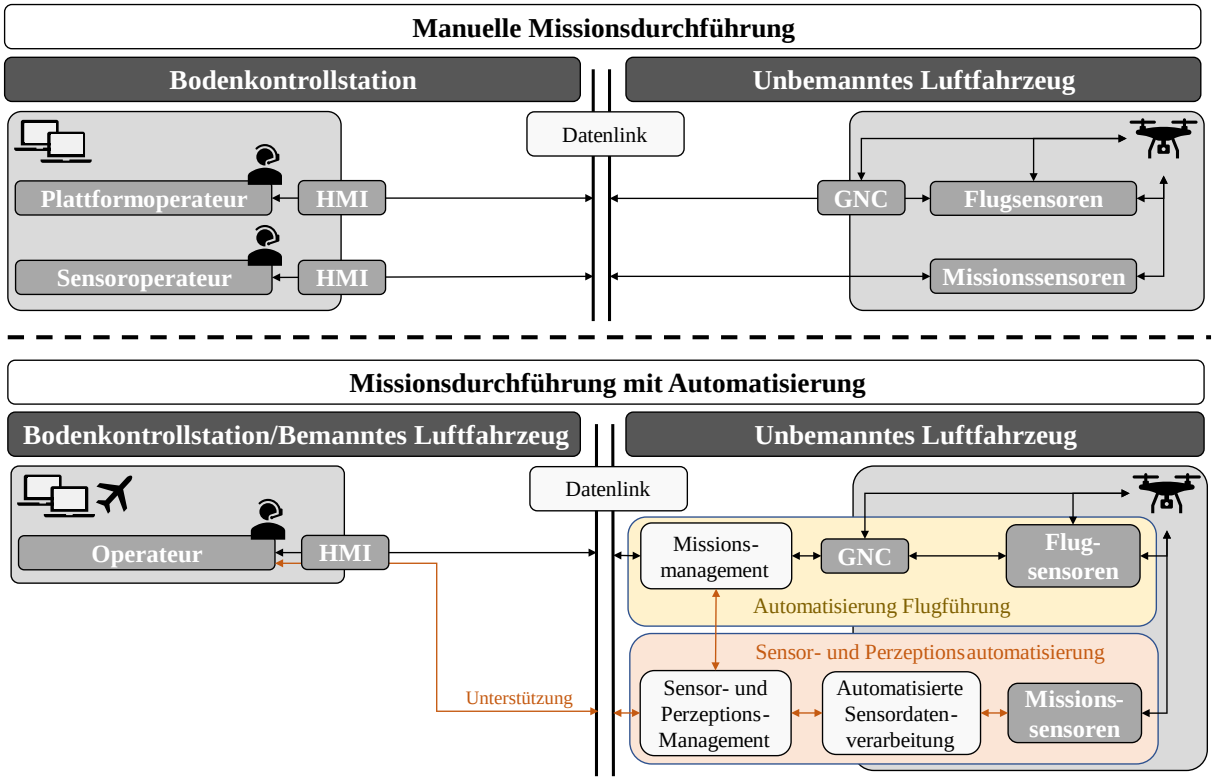


Abb. 1 Schematische Darstellung der Durchführung von UAV-Missionen. Die obere und die untere Teilgrafik zeigen einen Vergleich zwischen manueller Durchführung der Mission und der gewünschten Automatisierung bei der Missionsdurchführung. In Orange ist die angestrebte Automatisierung der Sensordatenverarbeitung eingetragen.
HMI: Human-Machine-Interface; GNC: Guidance, Navigation and Control

Abb. 1 zeigt ein typisches Führungsschema bei UAV-Missionen [15, 16]. Bei der manuellen Missionsdurchführung übernehmen üblicherweise mehrere Operateure in der Bodenkontrollstation die Steuerung und die Sensordatenauswertung. Sie haben über einen Datenlink und die entsprechenden Benutzerschnittstellen (HMI; engl.: *Human Machine Interface*) Zugriff auf das UAV. Dort befindet sich die Flugsteuerung (GNC, engl.: *Guidance, Navigation and Control*) mit den zughörigen Flugsensoren und die Missionssensorik zur Erfassung von Umgebungsinformationen. Um die Arbeitsbelastung der Operateure zu minimieren, wird vor allem bei Multi-UAV Anwendungen oder Manned-Unmanned Teaming Konzepten (*MUM-T*) [17–19] ein immer höherer Automatisierungsgrad der UAV Systeme angestrebt. Dies wird in der unteren Teilgrafik von Abb. 1 verdeutlicht und führt dazu, dass letztendlich ein Operateur für die Missionsdurchführung ausreicht bzw. sogar mehrere UAV Systeme betreibt. Die Automatisierung betrifft nicht nur die eigentliche Flugführung, sondern auch den Sensor- und Perzeptionsanteil. Dieser umfasst die Verarbeitung des von den mitgeführten Sensor-Nutzlastsystemen gewonnenen Datenmaterials, die Sensor- und Algorithmenauswahl und das Ressourcenmanagement in Bezug auf die zur Verfügung stehenden Sensoren [20]. Ziel der Sensordatenverarbeitung ist die Generierung von Informationen, die für den jeweiligen Anwendungsfall bzw. für die Erfüllung der jeweiligen Mission ausschlaggebend sind. Typische Beispiele für derartige Informationen wären z.B. die Segmentierung einer Szenerie, die Detektion bestimmter Zielobjekte oder die Klassifikation und Identifikation dieser Objekte. Dazu müssen die während dem Flug aufgezeichneten Rohdaten weiterverarbeitet und ausgewertet werden. Diese weisen jedoch meist große Datenmengen auf und umfassen je nach System zudem mehrere Sensorarten, sodass eine rein manuelle Auswertung durch den Sensoroperateur oftmals nicht zu bewerkstelligen ist. Eine Datenverarbeitung nach dem Flug, wie sie im Wesentlichen heutzutage noch vorgenommen wird, erweist sich als nachteilig, da die Sensordatenauswertung und die daraus gewonnenen Informationen erst in großem Zeitverzug zur Verfügung stehen oder auch, wie z.B. bei der Objektverfolgung, beim weiteren Verlauf der Sensoreinsatzmission berücksichtigt werden müssen.

Der aus diesen Gründen gewünschte höhere Grad an Autonomie führt dazu, dass eine hoch automatisierte, integrierte und echtzeitfähige Sensordatenverarbeitung an Bord des UAVs angestrebt wird. Abb. 1 zeigt schematisch die Verlagerung und Automatisierung dieses Aufgabenfeldes. Die Sensordaten werden im Folgenden auf die Gruppe der Bilddaten beschränkt. Der gesamte Ablauf beim Einsatz derartiger Systeme teilt sich in drei Teile [21]: Die Erfassung der Rohdaten im ersten Schritt, die unmittelbare Weiterverarbeitung und Analyse an Bord des intelligenten Systems und schließlich die Entscheidungsfindung auf Grundlage der ermittelten Informationen. Grundsätzlich ist dabei auch eine automatisierte Entscheidungsfindung an Bord der fliegenden Plattform denkbar, die dann z.B. direkten Einfluss auf die Missionsplanung und die Flugsteuerung nimmt. In diesem Kontext der automatisierten Auswertung bildgebender Sensorik ist die vorliegende Arbeit einzuordnen.

1.1 Hintergrund und Aufgabenstellung

Im Sinne der vorgenannten Automatisierungsforderung werden computergestützte Datenverarbeitungsalgorithmen zur Auswertung und Verarbeitung der vom Sensor gesammelten Bildinformationen eingesetzt. Im Bereich luftgestützter Systeme bestehen die Rohdaten meist aus Luftbildaufnahmen, die anschließend analysiert und ausgewertet werden. Dieser Prozess wird als *Computer Vision (CV)* bezeichnet und verwendet je nach Anwendungsfall verschiedene Bildverarbeitungsalgorithmen, um möglichst zuverlässige Informationen aus den Rohdaten zu extrahieren (z.B. durch Segmentierung, Objekterkennung oder -klassifikation). Eine typische Aufgabe bei Aufklärungs- und Überwachungsmissionen ist die Objekterkennung. Für die vorliegende Arbeit wurde diese als Anwendungsbeispiel ausgewählt, da eine Vielzahl von Algorithmen dafür entwickelt wurde und sie die Grundlage für viele weiterführende Aufgaben wie z.B. die Objektklassifikation oder die Objektverfolgung bildet.

UAV-basierte Luftbildaufnahmen stellen für computergestützte Detektionsverfahren besondere Herausforderungen dar, da sich die zu erkennenden Objekte aufgrund der unterschiedlichen Flughöhen und Flugsituationen in Größe, Form, Orientierung und Hintergrund unterscheiden können [22, 23]. Darüber hinaus treten bei der Erkennung unterschiedliche Umgebungs- und Lichtverhältnisse auf, die Objekte sind durch Vibrationen und Bewegungsunschärfe verzerrt und es sind viele feine, störende Strukturen und Elemente in den Bildern vorhanden. Insbesondere bei hochauflösenden Luftbildern kommt auch der Verarbeitungszeit eine entscheidende Bedeutung zu [24], da wie bereits erwähnt eine automatisierte und echtzeitfähige Sensordatenauswertung angestrebt wird.

1.2 Algorithmen zur Sensordatenauswertung

Algorithmen zur Bildverarbeitung lassen sich in zwei Gruppen einteilen (s. auch Kapitel 4.3). Die erste und ältere Gruppe enthält Algorithmen, die auf einfachen Bildmerkmalen, wie z.B. Farbunterschieden oder Kantenverläufen beruhen und keine Trainingsdaten benötigen. Die Struktur und die einzelnen Verarbeitungsschritte dieser prozeduralen Algorithmen werden bei deren Entwicklung fest vorgegeben. Durch die zum Teil festen Parameterwerte sind sie auf bestimmte Anwendungsszenarien und Randbedingungen angepasst und funktionieren nur unter definierten Umgebungszuständen [25–27]. Zur Umgehung dieses Nachteils verwendet die zweite und neuere Gruppe von Algorithmen Trainingsdatensätze, um die Erkennung relevanter Bildmerkmale zu erlernen. Diese Algorithmen zählen zu den Methoden der künstlichen Intelligenz (KI) und erreichen durch das Training eine höhere Leistungs- und vor allem Generalisierungsfähigkeit und sind dadurch universeller einsetzbar. Im Bereich der Objektdetektion kommen dabei vor allem sogenannte *deep-learning* basierte Faltungsnetze (engl.: *Convolutional Neural Network (CNN)*) zum Einsatz [28–32]. Diese extrahieren aus den Eingangsbildern durch verschiedene Faltungsoperationen und antrainierte Gewichtungen in aufeinanderfolgenden Schichten dynamisch die für die jeweilige Aufgabe relevanten Bildmerkmale. Vorrausetzung für eine erfolgreiche Anwendung ist eine große Menge passender Trainingsdaten zum Anlernen des Netzwerks. Diese Daten müssen zudem eine hohe Varianz aufweisen und auf den jeweiligen Anwendungsfall zugeschnitten sein.

1.3 Trainings- und Testdatenproblematik

Die Leistungsfähigkeit *deep-learning* basierter Verfahren wird daher in hohem Maße von der Verfügbarkeit und Menge geeigneter Datensätze beeinflusst [33, 34]. Diese werden für die prototypische Entwicklung, das Training der zugrundeliegenden Modelle und nicht zuletzt für die Evaluierung der Algorithmen unter den späteren Einsatzbedingungen benötigt. In vielen Fällen fehlen ausreichende und passende Datensätze, besonders im Bereich der luftgestützten Bildgebung. Erschwerend kommt hinzu, dass diese oft nicht frei zugänglich sind oder nur für eine spezielle Anwendung konzipiert wurden. Darüber hinaus ist es vor allem bei lernenden Algorithmen unerlässlich, Datensätze aus mehreren unterschiedlichen Quellen und Bereichen anzuwenden, um eine ausreichende Varianz in Bezug auf Inhalt, Szenarien, Vegetation, Störeffekten und Umwelt- und Wetterbedingungen zu erhalten [35]. Neben den reinen Bilddaten sind auch zugehörige Annotationen erforderlich, die sogenannte *Ground Truth*. Diese unterscheidet sich je nach Aufgabenstellung. Bei der semantischen Segmentierung entspricht sie der Zuweisung der einzelnen Bildpixel zu vorgegebenen Segmentierungsklassen. Bei der Objektdetektion hingegen enthält sie die einzelnen Begrenzungsrahmen, auch *Bounding Boxen* genannt, der im Bild vorkommenden Objekte zusammen mit der zugehörigen Objektklasse. Die händische Generierung dieser *Ground Truth* führt bei der benötigten Datenmenge zu einem enormen Arbeitsaufwand und hohen Kosten, vor allem bei pixelbasierten Annotationen oder einer Vielzahl von Objekten im Bild [33, 34].

Im Bereich der flugzeuggestützten Sensorik erfordert die Generierung der benötigten Datensätze in der Regel aufwändige und kostenintensive Flugmissionen, die zudem durch gesetzliche Restriktionen

begrenzt sind. Darüber hinaus wird nur ein Bruchteil der möglichen szenarischen Varianz (z.B. Jahreszeiten, geografische Regionen, aber auch Blickwinkel) und der im späteren Einsatz auftretenden Umwelt- und Wettereinflüsse (z.B. Schnee, Regen, Nebel, Bewölkungsgrad) erfasst. Vor allem von den Standardbedingungen abweichendes Datenmaterial ist aus zeitlichen oder auch sicherheitstechnischen Aspekten nur schwer zu erfassen. In vielen Anwendungsfällen wie z.B. bei der Objektverfolgung spielt neben den reinen Umgebungsbedingungen auch die räumliche Anordnung und das zeitliche Verhalten von statischen oder bewegten Objekten eine große Rolle [36]. Diesbezügliche Variationen müssen wiederum bei der Planung der Flugmissionen berücksichtigt werden und führen zu einer deutlich höheren Komplexität. Werden sie nicht in ausreichendem Umfang bei der Trainingsdatengenerierung miteinbezogen, bewirkt dies, dass das angelernte Modell im späteren Einsatz nicht allgemein anwendbar ist und eine geringe Generalisationsfähigkeit aufweist [37]. Derselbe Effekt tritt auf, wenn die Trainingsdaten insgesamt zu wenig Variationen enthalten. Dies führt zu einer Überanpassung des Modells auf die vorgegebenen Trainingssituationen und zu einer geringeren Leistungsfähigkeit auf unabhängigen Testdaten. Im militärischen Bereich besteht außerdem die Schwierigkeit, dass die Datenerhebung im späteren geografischen Einsatzbereich aus Sicherheitsgründen oft nicht möglich ist, für eine zuverlässige Evaluierung der Algorithmen aber zwingend nötig wäre [36].

Insgesamt gesehen kann diese Problematik zu Algorithmen mit geringer Generalisationsfähigkeit und Robustheit gegenüber schwankenden Umgebungsbedingungen führen.

1.4 Lösungsansatz: Einsatz virtueller Simulationsumgebungen zur Datengenerierung

Ein vielversprechender Ansatz, um dieses vielfältige Problem zu umgehen, stellt die Verwendung von virtuellen Simulationsumgebungen dar. Diese nutzen modellierte virtuelle Welten zur Bereitstellung von synthetischem Bild- oder Videomaterial. Ziel ist die Nachbildung realer Szenerien oder Sensoransichten mit Hilfe von Computergrafik, wobei in erster Linie die visuelle Wahrnehmung angesprochen wird. Durch stetige Weiterentwicklung und Verbesserung von Rechen- und vor allem Grafikleistung wurden im Laufe der Zeit zahlreiche Anwendungsgebiete erschlossen und es kamen Cockpitsimulatoren für das Pilotentraining [38, 39], Fahrsimulationssysteme für die Automobilindustrie, Trainingssimulatoren für militärische Zwecke und Videospiele für den privaten Gebrauch auf den Markt [38]. Moderne Simulationsumgebungen und 3D *Game-Engines* generieren dabei meist in Echtzeit photorealistisches und z.T. physikalisch modelliertes Bildmaterial und stellen darüber hinaus vielfältigste Anpassungsmöglichkeiten in Bezug auf Bilddarstellung und Sensoreffekte zur Verfügung. In der Regel zielen solche Simulationsumgebungen auf die Vermittlung visueller Sinneseindrücke ab, wobei stets die menschliche Wahrnehmung im Vordergrund steht, um so z.B. einen menschlichen Anwender in seiner Arbeitsumgebung zu trainieren.

Im Gegensatz dazu sollen nun derartige Umgebungen zur Generierung von Trainings- und Testdatensätzen für lernfähige Algorithmen herangezogen werden. Das Ziel ist in diesem Fall die Erzeugung eines Datensatzes synthetischer Sensorbilder. Dieser soll anschließend verwendet werden, um die beschriebene Datenproblematik beim Einsatz *deep-learning* basierter Bildverarbeitungsalgorithmen zu umgehen und so die Leistungsfähigkeit der Modelle durch die Optimierung der Trainingsdaten zu steigern. Dadurch liegt der Fokus in solchen Simulationen nicht mehr auf der möglichst realistischen Nachbildung aus Sicht der menschlichen Wahrnehmung. Vielmehr steht der Informationsgehalt, den das synthetische Bildmaterial für einen computergestützten Algorithmus liefert, im Vordergrund. Zudem sind Einflussfaktoren in Bezug auf Modellierung, Rendering und Datensatzzusammensetzung zu beachten, die für die Leistungsfähigkeit und das Trainingsverhalten des Algorithmus eine Rolle spielen. Eine statistisch basierte Identifikation derartiger Einflussfaktoren ist Bestandteil der in Kapitel 3.1.2 abgeleiteten Forschungsfragen.

Vorteile virtueller Simulationsumgebungen

Virtuelle Simulationsumgebungen können kostengünstig und effizient eine Vielzahl von atmosphärischen und sensorischen Effekten berücksichtigen und stellen somit eine gute Möglichkeit dar, die bereits beschriebene Trainings- und Testdatenproblematik zu umgehen, indem bestehende reale Trainingsdaten erweitert, ihre Robustheit erhöht oder diese komplett ersetzt werden. Weitere Vorteile sind die Umgehung gesetzlicher Einschränkungen im Realflug, die Möglichkeit zum vollständigen Test der Algorithmen mit dynamischem Feedback in kritischen Situationen und die Maximierung der erfassten szenarischen Varianz [40, 41]. Daraus entstehende kürzere Produktentwicklungszyklen und eine geringere Anzahl an Flugmissionen führen zu einer deutlichen Kostensenkung [42]. Darüber hinaus gewährleisten virtuelle Simulationsumgebungen eine exakte Vergleichbarkeit und Reproduzierbarkeit der Experimente [38], was eine zuverlässige Identifikation vorhandener Fehler begünstigt und eine Parameteroptimierung des Lernvorgangs ermöglicht. Des Weiteren können ideale Sensoren vor ihrer prototypischen Entwicklung in der Simulation getestet und konfiguriert werden [38]. Die Entwicklung von Methoden, deren Lernprozess auf einem *trial-and-error* basierten Verfahren in Verbindung mit Reinforcement Learning beruht, ist ausschließlich in der Simulation möglich [43–45]. Ein entscheidender Vorteil von virtuellen Umgebungen ist die schnelle Generierung simulierter Sensordaten mit der zugehörigen *Ground Truth* in Form von zusätzlichen Bildinformationen und hochgenauen Annotationen für überwachte Lernverfahren [34, 37, 38, 40, 46, 47]. Dies reduziert den ansonsten enormen manuellen Annotationsaufwand und ermöglicht die Erzeugung von Datenmengen in einer Größenordnung, wie sie für das Training von *deep-learning* basierten Netzen benötigt wird. Programmierschnittstellen erlauben dabei eine automatisierte Datensatzgenerierung mit gezielten oder zufälligen Parametervariationen.

Nachteile virtueller Simulationsumgebungen

Dennoch müssen bei der Anwendung derartiger virtueller Simulationsumgebungen zur Generierung synthetischer Sensordaten auch gewisse Einschränkungen beachtet werden [36]. So ist trotz einer Verbesserung hin zu einer immer photorealistischeren Darstellung die Anzahl, der Detailgrad und die Variation der gerenderten 3D-Objekte im Vergleich zu realen Daten vor allem in Bezug auf untergeordnete Details im Bild immer noch deutlich begrenzter. Die Leistung von Bildverarbeitungsalgorithmen ist häufig auf gerenderten Bilddaten höher als bei der späteren Anwendung, da diese klarere Strukturen und ausgeprägtere Merkmale aufweisen, die weniger stark durch Störeffekte oder optische Verzerrungen überlagert werden [48–51]. Auch in Bezug auf die verwendeten Texturen weisen synthetische Bilder deutlich homogenere Strukturen als reale Daten auf. Dies liegt zum einen daran, dass regelmäßige Muster aus Gründen der Performance häufig aus einer einzelnen, sich wiederholenden Textur zusammengesetzt werden und zum anderen die Varianz von Witterungs- und Umwelteinflüssen durch die Textur häufig nur unzureichend repräsentiert werden kann. Der Aufwand für die Modellierung der zugrundeliegenden Datenbasis (Terrain, 3D-Objekte, Vegetation, usw.) kann daher je nach geografischer Ausdehnung und Szenerie ein erhebliches Ausmaß annehmen und muss mit dem benötigten Grad an Realismus abgewogen werden [52].

Realismus und Reality Gap

Die Anpassung der Simulationsparameter und die Modellierung der atmosphärischen, physikalischen und sensorischen Effekte ist zeitaufwendig und rechenintensiv. Dennoch wird häufig nur ein Bruchteil aller komplexen realen Bedingungen und Zusammenhänge in den verwendeten Simulationsmodellen erfasst. Die daraus resultierende Differenz des Systemverhaltens zwischen den Domänen Simulation und Realität wird häufig mit dem aus der Robotik stammenden Begriff „*Reality Gap*“ beschrieben [49, 53].

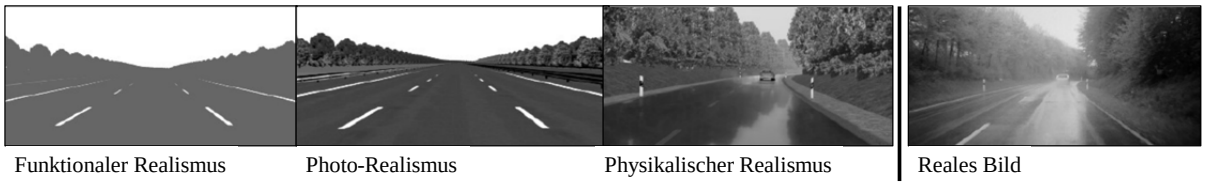


Abb. 2 Schematische Darstellung der verschiedenen Formen von Realismus mit steigender Genauigkeit und entsprechend steigendem Modellieraufwand von links nach rechts (vgl. [34]).

Bei einer systematischen und produktorientierten Nutzung virtueller Simulationsumgebungen und der daraus generierten synthetischen Daten stellt sich nun die Frage nach dem benötigten Detailgrad und den notwendigen Bildeigenschaften der gerenderten Sensordaten. Das Ziel dabei ist ein Kompromiss aus der Verringerung des *Reality Gaps* und einem gleichzeitig überschaubar bleibenden Modellieraufwand. Bemessungsgrundlage für die Ableitung von Gestaltungsrichtlinien soll dabei nicht die subjektive menschliche Wahrnehmung sein, sondern der jeweils zum Einsatz kommende Algorithmus. Ferwerda [54] stellte in diesem Zusammenhang ein Konzept vor, das die Genauigkeit von Simulationsdaten in drei hierarchisch angeordnete Stufen gliedert. In Abb. 2 sind die nachfolgend aufgelisteten Formen von Realismus schematisch für den Anwendungsfall der Fahrspurerkennung im Automobilbereich dargestellt.

1) **Physikalischer Realismus:** Dieser Zustand liegt vor, wenn die Simulation durch physikalisch korrekte Zusammenhänge die gleiche visuelle Stimulation bietet wie die reale Szenerie. Das bedeutet, dass der verwendete Bildgenerierungsprozess eine exakte Repräsentation sämtlicher realen Gegebenheiten, Materialen und Beleuchtungseigenschaften erzeugen muss. Da allein schon die Darstellung auf den heute verfügbaren Displays diese Bedingungen nicht erfüllen kann, wird dieser Zustand außer unter bestimmten Einschränkungen aktuell nicht erreicht [54].

2) **Photo-Realismus:** Um diese Bedingung zu erfüllen, dürfen die gerenderten Sensordaten optisch nicht von einer Photographie der Szenerie unterscheidbar sein. Die Simulation muss daher im visuellen System des Betrachters dieselbe visuelle Reaktion auslösen, die auch die Betrachtung der realen Szenerie hervorrufen würde. Dieser Standard ist nicht neu und wird zum Beispiel auch bei der Farbbilddarstellung durch das RGB-System oder bei der Repräsentation großer Dynamikbereiche der Helligkeit auf Displays herangezogen [54]. Entscheidender Vorteil dieser wahrnehmungsbasierten Herangehensweise ist eine gesteigerte Effizienz bei der Bildgenerierung, da visuell nicht wahrnehmbare Bildeigenschaften beim Rendern nicht berücksichtigt werden müssen [55, 56]. Grundlage bildet allerdings immer der menschliche Betrachter und daher ist dieser Ansatz für die Beurteilung der Leistungsunterschiede von Bildverarbeitungsalgorithmen nur bedingt anwendbar.

3) **Funktionaler Realismus:** Die Simulation liefert in diesem Zustand jegliche für den jeweiligen Anwendungsfall notwendige visuelle Information. Information bezeichnet in diesem Zusammenhang sämtliche Eigenschaften einer Szenerie wie z.B. Formen, Größe, Bewegung oder auch Material- oder Beleuchtungssimulation, die für eine Durchführung der jeweiligen Aufgabe mit der gleichen Leistung wie auf realem Datenmaterial nötig sind [54]. Ziel dabei ist es, die Differenzen bei vertretbarem Modellieraufwand so gering zu halten, dass sie für den jeweiligen Algorithmus vernachlässigbar sind. Diese Definition ist dabei unabhängig von der zugrundeliegenden Art des Renderings und umfasst sowohl annähernd physikalisch basierte und photorealistisch konzipierte Simulationen als auch eher abstrakt gehaltene Darstellungsformen, wie in Abb. 2 links dargestellt.

Ziel ist die Erstellung synthetischer Datensätze für Trainings- und Testzwecke *deep-learning* basierter Netze, die den funktionalen Realismus erreichen und somit sämtliche Informationen zur Verfügung stellen, die der Algorithmus für die jeweilige Aufgabenstellung benötigt.

1.5 Aufgabenstellung: Untersuchung der Einflussfaktoren und Ableitung von Gestaltungsrichtlinien für synthetische Trainingsdaten

Es stellt sich nun die Frage, inwieweit heute verfügbare virtuelle Simulationsumgebungen in der Lage sind, Sensordaten zu reproduzieren, die den funktionalen Realismus erreichen. Dies ist die Voraussetzung, um die eingangs beschriebene Trainings- und Testdatenproblematik mit Hilfe synthetischer Sensordaten zu umgehen. Es werden daher Kriterien gesucht, um abschätzen zu können, ob synthetische Daten zur Entwicklung und Evaluierung luftgestützter Bildauswertealgorithmen geeignet und im Hinblick auf deren Leistungscharakteristik mit realen Aufnahmen vergleichbar sind. Dabei ist stets zu beachten, dass die Einschätzung der Vergleichbarkeit nicht auf der menschlichen Wahrnehmung basiert, sondern lediglich auf der Anwendbarkeit der synthetischen Bilddaten in Verbindung mit einem definierten Bildverarbeitungsalgorithmus. Erschwerend kommt hinzu, dass die bei der Sensordatenauswertung eingesetzten *deep-learning* basierten Algorithmen im Allgemeinen *Black-Box* Modelle liefern und daher keine oder nur eingeschränkte Rückschlüsse über die verwendeten und relevanten Bildmerkmale zulassen.

Im Rahmen der vorliegenden Arbeit sollen daher Konzepte und Methoden entwickelt und experimentell angewandt werden, um den Einsatz synthetischer Daten in diesem Kontext zu optimieren. Dies umfasst die Untersuchung verschiedener Trainingsdatenzusammensetzungen bei der Verwendung aktueller Bildverarbeitungsalgorithmen, die anschließende Auswertung der Einflussfaktoren mit statistischen Methoden und die Analyse von Parametereinflüssen auf die Detektionsleistung. Ziel ist die Ableitung von Richtlinien zur Gestaltung der Simulationsumgebungen und die Identifikation relevanter Bildeigenschaften, die entscheidend sind für die Algorithmenleistung, um so dem Zustand des funktionalen Realismus möglichst nahe zu kommen. Damit soll vermieden werden, dass zu große Unterschiede zwischen den Domänen Simulation und Realität bei der späteren realen Anwendung der Modelle zu nicht akzeptablen Leistungsunterschieden führen.

1.6 Inhaltsübersicht

Im ersten Teil der Arbeit in Kapitel 2 werden nun die Hintergründe und der Stand der Technik genauer beleuchtet. Anhand dieser Zusammenstellung der verfügbaren Literatur werden anschließend in Kapitel 3 offene Forschungsfragen abgeleitet und ein Konzept entwickelt, mit dem eine Untersuchung dieser Fragestellungen vorgenommen werden kann. Im darauf folgenden Kapitel werden dann die Grundlagen und Methoden vorgestellt, die bei der Umsetzung des Konzepts eine Rolle spielen. Dies betrifft sowohl die verschiedenen verfügbaren Datensätze, Testalgorithmen und Simulationsumgebungen als auch die Vielzahl an Bildbeschreibermetriken und die Grundlagen der statistischen Auswerteverfahren.

Im Kapitel 5 wird genauer auf die darauffolgende Implementierung, den Experimentalaufbau und die Durchführung der Experimentalflüge zur Datensatzerzeugung eingegangen. Im ersten Schritt wird dabei die Modellierung der synthetischen Welt und die Vorgehensweise zur automatisierten synthetischen Datensatzgenerierung mit den entsprechenden Parameterabstufungen angesprochen. Auch die Inbetriebnahme des Multikoptersetups und die damit vorgenommene Erfassung der realen Sensordaten anhand eines definierten Flugplans wird beschrieben. Dies dient als Grundlage für die anschließende Erzeugung der synthetischen Bildduplikate. Abschließend wird ein Überblick über die betrachteten Parametervariation der Trainings- und Testdaten gegeben und das Detektortraining kurz angesprochen.

Ein weiteres Kapitel dient der Darstellung der Ergebnisse und der Beantwortung der aufgestellten Forschungsfragen. Es ist in drei Teile gegliedert. Der erste Teil behandelt die Zusammensetzung der Trainingsdaten, der zweite die Ergebnisse der statistischen Auswertung zur Einflussanalyse und der dritte schließlich die Einflüsse der vorgestellten Parametervariationen auf die Detektionsleistung. Es folgt eine Zusammenfassung der Ergebnisse, die auf kompakte Art und Weise die wichtigsten Erkenntnisse und

Schlussfolgerungen sammelt. Im letzten Kapitel wird schließlich ein Fazit über die Arbeit gezogen und ein Ausblick über weiterführende Untersuchungsmöglichkeiten gegeben.

2 Stand der Technik

Das nachfolgende Kapitel soll nun einen Überblick über bereits vorhandene Forschungsansätze und -ergebnisse zur Umgehung der beschriebenen Trainings- und Testdatenproblematik liefern. Der Fokus liegt dabei auf der Anwendung virtueller Simulationsumgebungen zur Generierung von synthetischem Datenmaterial für das Training tiefer neuronaler Netze in der maschinellen Bildverarbeitung. Dabei wird auch darauf eingegangen, welche Fragestellungen die bisherigen Untersuchungen noch unbeantwortet ließen und in welchen Punkten daher die vorliegende Arbeit die bisherigen Ansätze ergänzt. Abschließend wird begründet, warum die UAV gestützte Fahrzeugdetektion mit *deep-learning* basierten neuronalen Netzwerken als Anwendungsfall für das in dieser Arbeit beschriebene Konzept herangezogen wird.

2.1.1 Data Augmentation zur Trainingsdatenverbesserung

In Kapitel 1.2 wurde eine Unterscheidung in nicht trainierbare und trainierbare Algorithmen zur Sensordatenauswertung vorgenommen. Da letztere dem aktuellen Stand der Technik entsprechen und eine deutlich höhere Leistungs- und Generalisationsfähigkeit aufweisen, liegt in nahezu allen Anwendungsbereichen der Fokus auf dieser Art von Algorithmen. Im Bereich der Bildverarbeitung werden dabei meist *deep-learning* basierte CNNs eingesetzt. Das Training dieser Netzwerke erfordert eine große Menge an Trainingsdaten, deren Variation und Zusammensetzung entscheidend ist für die Güte der resultierenden Modelle. Die Problematik besteht darin, dass passende Daten nur in begrenzter Menge verfügbar sind und in der Realität nur mit großem Aufwand generiert werden können (s. Kapitel 1.3). Es gibt in der Literatur mehrere Ansätze zur Umgehung der beschriebenen Trainings- und Testdatenproblematik. *Data Augmentation* ist in diesem Zusammenhang ein Oberbegriff für unterschiedliche Methoden zur Verbesserung von Trainingsdatensätzen in Hinblick auf Datenmenge und Qualität [57]. Abb. 3 gibt einen Überblick über deren Einordnung. Auch die in dieser Arbeit betrachtete Generierung synthetischer Bilddaten mit Hilfe von virtuellen Simulationsumgebungen wird in der Übersicht im weitesten Sinne zu den *Data Augmentation* Methoden gezählt, da dabei ebenfalls ein Lösungsansatz für die Trainings- und Testdatenproblematik geliefert wird.

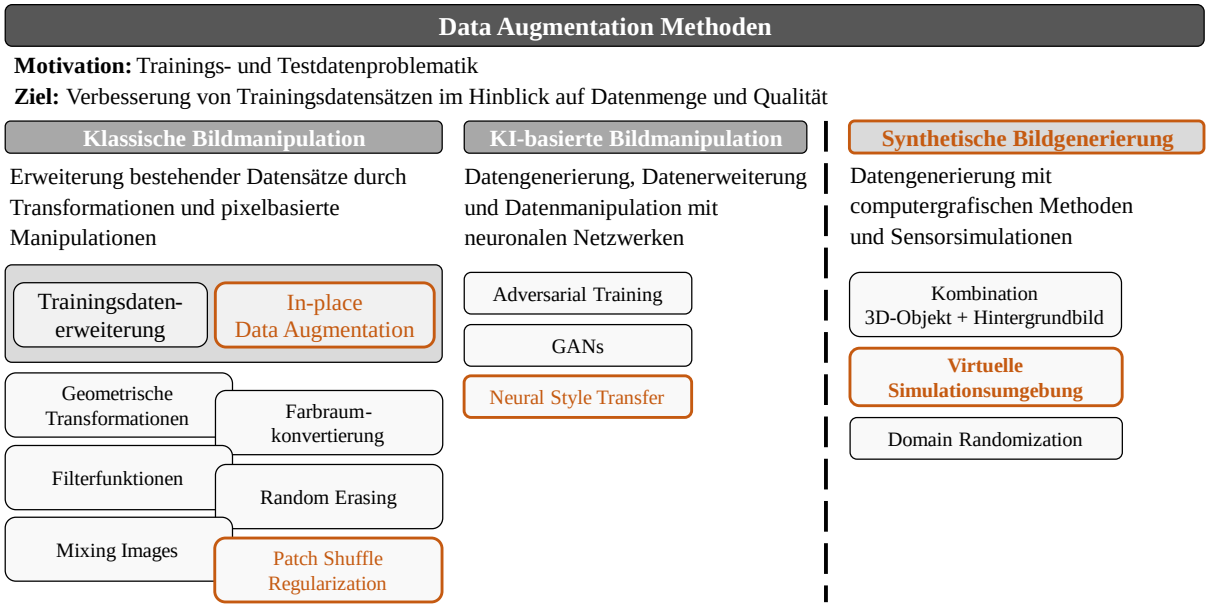


Abb. 3 Überblick und Kategorisierung gängiger *Data Augmentation* Methoden für Bilddaten zur Verbesserung von Trainingsdatensätzen. Orange markierte Einträge werden in den im Rahmen dieser Arbeit durchgeführten Analysen verwendet.
GAN: *Generative Adversarial Network*; KI: Künstliche Intelligenz

Klassische Bildmanipulation

Ein Teilbereich der *Data Augmentation* umfasst die Erweiterung eines existierenden Datensatzes im Vorfeld des Trainings. Zur Anwendung kommen dabei verschiedene Techniken der Bildmanipulation, wie zum Beispiel zufällige geometrische Transformationen (Spiegelung, Zuschneiden, Rotation, Translation, Größenänderung, …), Konvertierungen in andere Farbräume [58, 59] oder das Hinzufügen von Rauschen [60].

Neben diesen klassischen Möglichkeiten wurden auch komplexere Herangehensweisen entwickelt. So experimentierten Kang et al. [61] mit einem Filter, der die Pixelwerte in einem $n \times n$ Fenster zufällig vertauscht, nannten diese Methode Patch Shuffle Regularization und konnten damit die Fehlerrate bei Klassifikationsaufgaben verringern. Es wurde gezeigt, dass sogar das Zusammenmischen von Bildern durch Mittelung von deren Pixelwerten [62] oder durch Zusammensetzen einzelner Ausschnitte [63] eine effektive Methode zur Verbesserung der Leistungsfähigkeit darstellen kann. Darüber hinaus sollen CNNs durch das Löschen einzelner Bildregionen unempfindlicher gegenüber Objektverdeckungen werden [64]. Diese Gruppe von Methoden verbessert genau genommen nur indirekt die Generalisationsfähigkeit des Modells und generiert lediglich neue Trainingsdaten, die jedoch alle auf einer kleinen Teilmenge an Daten beruhen.

Im Gegensatz dazu zielt *In-place Data Augmentation* auf eine Transformation der Daten in jeder Epoche des Trainingsdurchlaufs. Dies stellt sicher, dass das zu trainierende Netzwerk in jeder Epoche neue Eingangsdaten erhält, die auch bei einer hohen Anzahl an Durchläufen keinesfalls mehrmals verwendet werden. Dieses Verfahren ist bei den meisten modernen Netzwerkarchitekturen bereits integriert [28–32, 65, 66] und bewirkt eine direkte Verbesserung der Generalisationsfähigkeit. Die dabei zur Anwendung kommenden Methoden unterscheiden sich je nach Architektur, beruhen aber in den meisten Fällen auf geometrischen Transformationen, Filterfunktionen oder einer Kombination mehrerer darin enthaltener Methoden.

KI-basierte Bildmanipulation

Der zweite Teilbereich betrachtet im Gegensatz zu den klassischen Methoden der Bildmanipulation ausschließlich KI-basierte Ansätze zur Bildmanipulation auf Basis neuronaler Netze [57].

Adversarial Training verwendet zwei oder mehrere rivalisierende Netzwerke. Während ein Netzwerk versucht, korrekte Ergebnisse für die jeweilige CV-Aufgabe zu generieren, lernt das konkurrierende Netzwerk iterativ gezielt diejenigen Bildänderungen in Form von Rauschüberlagerung oder gezielten Pixelmanipulationen, die auf der Gegenseite zu einem fehlerhaften Ergebnis führen. Da im Allgemeinen eine hohe Fehleranfälligkeit bei derartigen Bildmanipulation besteht [67, 68], konnte gezeigt werden, dass dieser Ansatz effektiv für die Erhöhung der Stabilität und Robustheit von *deep-learning* Modellen genutzt werden kann [69–71].

Ein ähnlicher Ansatz liegt bei der Verwendung von *GANs* (engl.: *Generative Adversarial Networks*) [72] zur Erzeugung künstlicher, synthetischer Bildinstanzen auf Basis eines vorgegebenen Datensatzes zugrunde, die als zusätzliche Trainingsdaten dienen können. *GANs* bestehen aus einem Generator und einem gegnerischen Diskriminator Netzwerk. Ziel ist die Erzeugung möglichst realitätsnaher synthetischer Daten durch das Generator Netzwerk. Der Diskriminator versucht dabei zwischen realen und synthetisch generierten Daten zu unterscheiden und verbessert somit iterativ die generierten Daten. Weiterentwicklungen dieses grundlegenden Ansatzes konnten erfolgreich zur Trainingsdatenerzeugung [73, 74] oder zur Angleichung von Daten aus verschiedenen Domänen eingesetzt werden [75, 76].

Beim *Neural Style Transfer* [77] wird innerhalb der neuronalen Schichten des Netzwerks eine Repräsentation des Eingangsbildes für die Eigenschaften „Stil“ und „Inhalt“ gebildet, was anschließend eine Transformation des Bildstils bei gleichbleibendem Inhalt ermöglicht. Im Hinblick auf den Zweck der

Data Augmentation ermöglicht dies eine Erhöhung der Variation in den Beleuchtungszuständen, den vorkommenden Texturen und sogar den Umwelt- und Umgebungszuständen (Sommer-Winter, Tag-Nacht, …).

Synthetische Bildgenerierung

Der dritte und letzte Teilbereich befasst sich mit der synthetischen Bildgenerierung, d.h. der Datengenerierung mit Hilfe computergrafischer Methoden und Sensorsimulationen. Dieses Vorgehen zählt nicht unmittelbar zu den *Data Augmentation* Methoden, wird jedoch häufig mit diesen kombiniert und hat ebenfalls die Lösung der beschriebenen Trainings- und Testdatenproblematik zum Ziel.

Jo et al. [78] verfolgten ein grundlegendes Konzept, bei dem durch verschiedene Kombinationen aus Objekt- und Hintergrundbildern ein großes Trainingsdatenset für die Objektdetektion erzeugt wurde. Sie erreichten damit ähnliche Detektionsleistungen wie bei der Verwendung real aufgenommener Trainingsdaten jedoch mit deutlich reduziertem Generierungsaufwand. In [79] wurden auf die Objektbilder im Vorfeld zusätzlich geometrische und photometrische Transformationen angewandt und Störeffekte überlagert, um die Variation im Trainingsdatenset zu erhöhen. Aufgrund der fehlenden Kontextinformation im Hintergrund waren die auf diese Weise synthetisch generierten Trainingsdaten allein nicht für die betrachtete Detektion von Logos geeignet, in Kombination mit den real aufgenommenen Sensordaten konnte jedoch wiederum eine Erhöhung der Leistungsfähigkeit erzielt werden. Rozantsev et al. [80] stellten schließlich ebenfalls ein für Objektdetektion entwickeltes Verfahren vor, das für die Überlagerung ein 3D-Modell des zu detektierenden Objekts verwendet. Die Besonderheit dabei lag in der Anpassung der Rendering Parameter durch das System zur Erzielung einer möglichst hohen Korrelation zwischen synthetischen und realen Daten in Hinblick auf die vom Detektor verwendeten Features. Die Detektionsleistung konnte dadurch signifikant gesteigert werden. Außerdem lieferte dieses Verfahren erneut eine höhere Leistung als eine reine *Data Augmentation* in Form von Überlagerung der Trainingsdaten mit Störeffekten oder eine Verwendung möglichst real wirkender synthetischer Daten, was wiederum die untergeordnete Bedeutung photorealistischer Optimierungen unterstreicht.

In diesen Teilbereich fällt auch die Anwendung von virtuellen Simulationsumgebungen zur gezielten Generierung von synthetischen Sensordaten für Trainings- und Testzwecke und zur Nutzung der damit verbundenen Vorteile (s. Kapitel 1.4). Dies bildet die Grundlage für diese Arbeit. Auf dieser Basis sollen verschiedene Trainingsdatenzusammensetzungen untersucht, Richtlinien zur Gestaltung der Simulationsumgebungen extrahiert und relevante Bildeigenschaften im Hinblick auf die Verwendung synthetischer Daten identifiziert werden. Es bleibt anzumerken, dass auch in diesem Teilbereich bei den verwendeten Testalgorithmen zur Erhöhung der Generalisationsfähigkeit dennoch Methoden der *In-place Data Augmentation* zum Einsatz kommen.

Tobin et al. [49] untersuchten in diesem Zusammenhang den Einfluss verschiedener Simulationsstile in den Trainingsdaten für den Anwendungsfall der Objektlokalisation. Dieses Vorgehen wird als *Domain Randomization* bezeichnet. Sie stellten fest, dass bei ausreichend Variation in den synthetisch generierten Trainingsdaten die Realität vom Netzwerk lediglich als weitere Variation betrachtet wird und dass eine Erhöhung der Diversität in den synthetischen Daten effektiver ist als eine Optimierung der Simulation in Hinblick auf die Erzeugung möglichst real wirkender Daten. Dieses Ergebnis lässt sich in Übereinstimmung bringen mit dem im vorigen Kapitel beschriebenen Ziel zur Erreichung des funktionalen Realismus. In [81] wurde dieser Ansatz ebenfalls erfolgreich für die 2D Fahrzeugdetektion auf dem KITTI Datensatz eingesetzt.

Analyse

Es bleibt festzuhalten, dass in der Literatur mehrere Arten der Datenaugmentierung existieren. Die synthetische Bildgenerierung mit virtuellen Simulationsumgebungen sticht dabei besonders hervor, da sie

die größte Flexibilität und das größte Potential aufweist. Vor allem in Hinblick auf eine Generierung von synthetischen Sensordatensätzen für das Training *deep-learning* basierter Bildverarbeitungsalgorithmen rückt diese Vorgehensweise immer mehr in den Mittelpunkt.

2.1.2 Einsatz von virtuellen Simulationsumgebungen bei CV-Anwendungen

In der Literatur gibt es bereits eine Vielzahl an Veröffentlichungen, in denen der Einsatz virtueller Simulationsumgebungen für die vielfältigsten Anwendungsfälle im Bereich von CV-Algorithmen betrachtet wurde. Darunter fallen zum Beispiel Methoden wie *Scene Understanding* [82–86], Fahrspurerkennung [41], Fußgängerdetektion [87–90], Fahrzeugerkennung [33, 41, 51, 91–93], Identifikation von 2D und 3D Objekten [94–98], Tracking [50, 99–102], Hindernisvermeidung [43–45], Semantische Segmentierung [34, 46] und optischer Fluss [103, 104].

Tab. 1 Schematische Eingliederung bisheriger Veröffentlichungen zum Einsatz von virtuellen Simulationsumgebungen bei CV- Anwendungen. Zum Vergleich sind die in der vorliegenden Arbeit vorgestellten Untersuchung in der letzten Zeile aufgeführt. Grün markierte Einträge stellen Merkmale derjenigen Gruppe dar, die auch in der vorliegenden Arbeit untersucht wird.
CV: *Computer Vision*; DA: *Domain Adaptation*; FT: *Fine-Tuning*; Mix: Gemischtes Trainingsdatenset, Real. Vgl. Reales Vergleichsdatenset; Scene Underst. *Scene Understanding*; CAD: 3D Simulation aus CAD Objekten; CNN: *Convolutional Neural Network*; sem.Seg.: semantische Segmentierung; UAV: *Unmanned Aerial Vehicle*; Det.: Detektion; HOG: *Histogram of Oriented Gradients*; SVM: *Support Vector Machine*; LBP: *Local Binary Pattern*; ADAS: *Advanced Driving Assistance System*; CS: Computerspiel; HiL: *Hardware-in-the-Loop* Simulator; M&S: *Modelling and Simulation* Suite; Klass. Klassifikation; Obj. Det.: Objekt Detektion; MOT: Multi-Objekt Tracking; RL: *Reinforcement Learning*; Feat.: Features; Hind.Verm.: Hindernisvermeidung

	CV-Anwendung	Algorithmus	Einsatzort	Simulation	Benchmark	DA/FT/Mix	Real. Vgl.	Bildpaare	Analyse
[82]	Scene Underst.	Intuitive Physics Engine	/	CAD	/	✗	✓	✗	✗
[83]	Scene Underst.	CNN	Indoor	Virtual Indoor	NYUv2	✓	✓	✗	✗
[84]	Scene Underst.	CNN/sem. Seg.	Indoor	Virtual Indoor	NYUv2, SUN	✓	✓	✗	✗
[85]	Scene Underst.	Inception-v3(CNN)	UAV	Bildmanipulation	/	✗	✓	✗	✗
[87]	Det. Person	HOG+SVM	ADAS	CS (Half-Life 2)	Daimler	✗	✓	✗	✗
[88]	Det. Person(IR)	Featurebasiert+SVM	ADAS	CAD	/	✗	✓	✗	✗
[89]	Det. Person	HOG/LBP+SVM	ADAS	CS (Half-Life 2)	INRIA/Daimler/Caltech	✓	✓	✗	✗
[41]	Det. Fahrzeug/Spur	Featurebasiert	ADAS	HiL	/	✗	✓	✗	✗
[33]	Det. Fahrzeug	Faster-RCNN(CNN)	ADAS	CS (GTA V)	KITTI/Cityscapes	✗	✓	✗	✗
[51]	Det. Fahrzeug	YOLOv3(CNN)	UAV	M&S (Presagis)	UAVDT	✓	✓	✗	✗
[91]	Klass. Fahrzeug	Haar Cascade	ADAS	HiL	/	✗	✓	✗	O
[92]	Det. Fahrzeug	YOLO/Faster-RCNN(CNN)	ADAS	CS (UE4)	/	✗	✗	✗	O
[93]	Det. Fahrzeug	Haar-like/LPB+Viola-Jones	UAV	M&S (VBS2)	/	✗	✗	✗	✗
[105]	Det. Fahrzeug	Inception-v3(CNN)	ADAS	M&S (VANE)	/	✓	✓	✓	✗
[106]	Det. Fahrzeug	Faster-RCNN(CNN)	ADAS	CS (UE4)	KITTI/VKITTI/…	✓	✓	✗	O
[94]	3D Obj. Det.	HOG/DPM	/	CAD	VOC/SUN	✗	✓	✗	✗
[96]	2D/3D Det. Fahrzeug	DPM	/	CAD	VOC	✓	✓	✗	✗
[97]	Obj. Det.	HOG+LDA	Indoor	CAD	ImageNet/VOC	✓	✓	✗	O
[98]	Obj. Det.	RCNN(CNN)	/	CAD	ImageNet/VOC/Office	✓	✓	✗	O
[50]	Tracking	Featurebasiert	UAV	M&S (VBS2)	/	✓	✓	✓	✗
[99, 100]	Tracking	DNN Tracker	UAV	CS (UE4)	UAV123	✗	✓	✗	✗
[101]	MOT	Fast R-CNN+Tracker	ADAS	CS (Unity)	KITTI/Virtual KITTI	✓	✓	✓	O
[43]	Hind.Verm.	Deep RL	UAV	Virtual Indoor	/	✗	✓	✗	✗
[44]	Hind.Verm.	Harris Corner Feat.	UAV	CAD	/	✗	✗	✗	✗
[34]	Sem.Seg.	CNN	ADAS	CS (GTA V)	CamVid/KITTI	✓	✓	✗	✗
[46]	Sem.Seg./Tiefen	CNN	ADAS	CS (?)	CamVid/Cityscapes	✓	✓	✗	✗

[103]	Optischer Fluss	CNNs	/	Animation	Sintel/KITTI	✓	✓	✗	✗
	Det. Fahrzeug	**YOLOv3 (CNN)**	**UAV**	**M&S (Presagis)**	**UAVDT**	✓	✓	✓	✓

Tab. 1 liefert einen Überblick über die entsprechenden Arbeiten und vergleicht die darin verwendeten Methoden und Ansätze mit den Zielen der vorliegenden Arbeit, die in Kapitel 1.5 kurz dargestellt wurden. Hierbei liegt der Fokus nicht nur auf der reinen Verwendung von synthetischem Datenmaterial, sondern vor allem auch auf der Analyse der Einflussfaktoren, die bei der Trainingsdatenzusammensetzung und der virtuellen Bildgenerierung eine Rolle spielen. Obwohl synthetisches Datenmaterial bereits sehr häufig zur Anwendung kommt, besteht bei der Analyse der Einflussfaktoren noch großer Forschungsbedarf, wie in der rechten Spalte von Tab. 1 zu sehen ist. Die Analyse ist jedoch nötig, um den Prozess der Bildgenerierung und Modellierung möglichst optimal zu gestalten und das Potential synthetischer Daten vollständig ausnutzen zu können. Auch der Bild- und Leistungsvergleich anhand realer und synthetischer Bildpaare kann diesbezüglich wertvolle Erkenntnisse liefern, da sich die Bildpaare lediglich in Erscheinungsform nicht aber in ihrem Inhalt und der dargestellten Szenerie unterscheiden. Auch dieses Vorgehen wird in bisherigen Untersuchungen nur sehr selten angewandt.

Anwendungsfall: Objektdetektion

Nentwig et al. [91] betrachteten die Anwendung virtueller Simulationsumgebungen im Kontext automobiler Fahrerassistenz- und Sicherheitssysteme (ADAS, engl.: *Advanced Driving Assistance System*). Sie untersuchten den Einfluss verschiedener Methoden der Schattengenerierung und der Simulation von Störeffekten wie Bewegungsunschärfe und Rauschen und erhielten für das Aufgabenfeld der Fahrzeugklassifizierung eine ähnliche Verteilung wahrer und falscher Hypothesen in Simulation und Realität. In [41] untersuchten sie im selben Kontext das Verhalten für die Anwendungen Fahrspurerkennung und Fahrzeugdetektion und kamen zu ähnlichen Ergebnissen. Trotz der allgemeinen Übereinstimmung bei der Betrachtung eines kompletten Szenarios, konnten dennoch beim Vergleich der Leistung auf einzelnen realen und synthetischen Bildern große Unterschiede beobachtet werden.

Pepik et al. [96] beschrieben eine Erweiterung des DPM Objektdetektors (engl.: *Deformable Part Model*) mit 3D Information z.B. zur Blickwinkelschätzung. Sie ergänzten ihre Trainingsdaten mit nicht photorealistisch gerendertem Bildmaterial, das aus 41 3D CAD Modellen von Autos und 43 Modellen von Fahrrädern generiert wurde. Während die Verwendung von ausschließlich synthetischen Trainingsdaten meist zu Leistungseinbußen führte, konnte gezeigt werden, dass mit einem gemischten Trainingsdatenset für die betrachteten Anwendungsfälle bessere Ergebnisse erzielt werden können als bei rein realem Training. Sun et al. [97] verwendeten ebenfalls 3D Modelle zur Trainingsdatengenerierung für die 2D Objektdetektion mit mehreren Klassen. Sie haben nachgewiesen, dass bei diskriminativen Detektoren, die auf Gradienten basierten Merkmalen beruhen, nicht unbedingt reale Texturen und photorealistische Trainingsdaten benötigt werden, da diese Kategorie von Detektoren ausschließlich die allgemeine Form und klassenspezifische Texturen der Objekte analysiert.

In [98] wurde für denselben Anwendungsfall aber unter Verwendung von CNNs untersucht, welche Anfälligkeit diese gegenüber fehlenden oder falschen einfachen Bildmerkmalen wie z.B. Objektfarbe, -textur und -orientierung bei synthetischen Daten aufweisen. Es stellte sich heraus, dass CNNs erstaunlich invariant gegenüber derartigen Merkmalen sind, solange im Vorfeld ein Training oder *Fine-Tuning* für die jeweilige Aufgabe stattgefunden hat. Beim Anlernen neuer Kategorien mit wenigen oder begrenzt verfügbaren Bildern ist allerdings eine Berücksichtigung dieser Faktoren in den synthetischen Daten empfehlenswert.

Hummel et al. [50] untersuchten ein merkmalsbasiertes visuelles Verfahren zum Multi-Objekt Tracking auf luftgestütztem Bildmaterial. Die realen Luftbildaufnahmen wurden in der Simulation nachgestellt und ein Leistungsvergleich ergab eine generelle Vergleichbarkeit der Domänen. Die geringfügig höhere

Detektionsgenauigkeit auf den simulierten Daten wurde auf fehlendes Rauschen, reduzierte Störeffekte und prägnantere Merkmale zurückgeführt.

In [93] wurde die Simulationsumgebung *Virtual Battlespace 2 (*VBS2*)* erfolgreich für die Generierung großer parametrisierter Datenmengen zum Training eines Klassifikators zur Fahrzeugdetektion auf Luftbildern eingesetzt. Dabei kam ein Viola-Jones Detektor mit *Haar-like* und LBP-Features (engl.: *Local Binary Patterns*) zum Einsatz und das Trainingsset enthielt Variationen in Bezug auf den Betrachtungswinkel, die Fahrzeugorientierung und die Flughöhe.

Carillo et al. [105] stellten in ihrer Studie eine physik-basierte Modellier- und Simulationsumgebung für unbemannte Fahrzeugsysteme vor, die verwendet wurde, um Bilddaten für Training, Test und Evaluierung des *Inception-v3* Netzwerks [107] zur Detektion von militärischen Fahrzeugen zu generieren. Sie zeigten, dass es damit möglich ist, Testumgebungen aufzubauen, die einerseits eine Optimierung der Trainingsparameter wie Lernrate und Anzahl der Epochen erlauben und andererseits auch eine frühe Erkennung von Überanpassung begünstigen.

Photorealistische Trainingsdatenerzeugung

Mehrere Arbeiten fokusierten in diesem Zusammenhang speziell die Erzeugung photorealistischer Trainingsdatensätze mit zugehörigen pixelgenauen Annotationen [33, 34, 46, 47, 87, 100, 101, 108–110]. Johnson-Roberson et al. [33] verwendeten die Grafik des Computerspiels *Grand Theft Auto V (GTA V)* zur Erzeugung eines synthetischen Datensatzes für das Training des Faster-RCNN Netzwerks [29] zur Fahrzeugdetektion und stellten eine automatisierte Pipeline zur gleichzeitigen Extraktion von Bilddaten und zugehörigen Annotationen in Form von *Bounding Boxen* vor. Sie demonstrierten damit, dass es möglich ist, gängige Netzwerkarchitekturen mit ausschließlich synthetischen Daten zu trainieren und erreichten bei der Evaluierung auf dem KITTI Benchmark Datenset [111] bessere Ergebnisse als mit händisch annotierten realen Sensordaten. Eine Analyse verschiedener Größen des synthetischen Datensets zeigte, dass die Variation und der Trainingseffekt für ein einzelnes synthetisches Bild aufgrund niedrigerer Komplexität in Beleuchtung, Farbe und Textur geringer ist als für ein einzelnes reales Bild. Da die annotierten synthetischen Daten kostengünstig und ohne händischen Aufwand automatisiert generiert werden können, konnte dieser Nachteil durch eine Erhöhung der Anzahl an Trainingsbildern ausgeglichen werden.

Richter et al. [34] haben am Beispiel von GTA V ein Verfahren vorgestellt, um aus der Kommunikation zwischen modernen Computerspielen und der Grafikhardware gleichzeitig Bildmaterial und pixel-genaue semantische Annotationen abzugreifen. Experimente zur Leistung von CNNs für die semantische Segmentierung haben angedeutet, dass durch die Hinzunahme synthetischer Daten die benötigte Menge an aufwendig von Hand annotierter Realdaten bei gleicher Leistung auf ein Drittel vermindert werden kann.

Marin et al. [87] konnten bestätigen, dass eine synthetisch trainierte, bild-basierte Fußgängerdetektion beim Test auf realen Benchmark Daten eine ähnliche Verteilung in Bezug auf korrekt und inkorrekt klassifizierte Testbilder aufweist wie ein auf realen Daten trainiertes Modell. Sie führten dies auf eine hohe Domäneninvarianz der HOG-Features (engl.: *Histogram of Oriented Gradients*) zurück, die in Kombination mit einem linearen SVM Klassifikator (engl.: *Support Vector Machine*) zur Detektion verwendet wurden. Das synthetische Datenset wurde auf Basis des Computerspiels Half-Life 2 generiert und enthält 81 unterschiedliche Modelle virtueller Personen. In [89] wurde der Ansatz um LBP-Features erweitert und näher auf das Problem der Datensatzverschiebung eingegangen, das im Rahmen der vorliegenden Arbeit auch oft als *Reality Gap* bezeichnet wird: Es wurde festgestellt, dass in nahezu allen Fällen Training und Test mit Bildern der gleichen Domäne zu besseren Ergebnissen führt als bei der Verwendung unterschiedlicher Domänen. Dieser Effekt tritt nicht nur zwischen realen und synthetischen Daten auf, sondern z.B. auch bei der Verwendung unterschiedlicher Kamera- oder Sensorsysteme.

Der vorgestellte *Domain Adaptation* Ansatz kombiniert wenige, aufwendig zu erzeugende Daten aus der Zieldomäne (Realität) mit vielen, leicht zu erzeugenden Daten aus der Quelldomäne (Simulation) und begünstigt durch diese gemischten Trainingsdaten ein stabileres Detektormodell, das auch in der Zieldomäne vergleichbare oder sogar höhere Leistungen erzielt. Yang et al. [92] nutzten *Unreal Engine 4* mit deren physikalisch basierter Simulation von Materialien und Reflexionen zur Erforschung des Einflusses von verschiedenen Beleuchtungsbedingungen auf die Leistung bei der Detektion von Fahrzeugen. Dabei kamen zwei gängige CNNs zur Objektdetektion, Faster-RCNN [28] und YOLO [31], zum Einsatz und es wurde gezeigt, dass beide Detektoren bei niedrigen und mittleren Beleuchtungen hohe Detektionsleistungen erzielen, die jedoch bei überstrahlten Szenarien stark absinken, wobei auch die Autofarbe einen Einfluss auf die Robustheit der Detektion hat.

Verfügbare synthetische Datensätze und Generierungsansätze

In einigen Fällen wurden mit Hilfe virtueller Simulationsumgebungen bereits synthetische Trainingsdatensätze erzeugt, die in vielen Fällen öffentlich zugänglich sind und als synthetischer Benchmark für ein oder mehrere CV-Aufgaben herangezogen werden können [46, 47, 99–101, 103, 104, 108]. Sie erlauben eine vergleichbare Analyse des Trainingsverhaltens neuronaler Netze mit synthetischen Daten und schaffen durch den automatisierten Generierungs- und Annotationsprozess eine Möglichkeit zur gezielten Parametervariation.

Mayer et al. [103] stellten ein synthetisches Video-Datenset mit über 35 000 Stereobildern vor, das für die CV-Aufgaben Disparität, optischer Fluss und Schätzung des Szenenflusses entwickelt wurde und haben dessen Eignung für das Training der entsprechenden CNNs nachgewiesen. Butler et al. [104] erzeugten mit der Open Source 3D Animation des Kurzfilms *Sintel* ein gleichnamiges synthetisches Datenset zur Evaluierung des optischen Flusses und verglichen dessen Statistik mit realen Daten. De Souza et al. [47] verwendeten die *Unity Engine* zur prozeduralen Erzeugung synthetischer Videos zum Training von *deep-learning* basierten Modellen zur Aktionserkennung. Sie nutzten die Vorteile der Simulation, um die aufwendige Erzeugung und vor allem Annotation realer Videosequenzen mit menschlichen Handlungen und Gesten zu umgehen und stellten stattdessen das synthetische Datenset PHAV (engl.: *Procedural Human Action Videos*) mit 39 982 Videos von 35 menschlichen Aktionen vor, das in Kombination mit wenigen realen Daten zu einer signifikanten Steigerung der Erkennungsleistung führte.

Ros et al. [108] generierten mit Hilfe der selben virtuellen Umgebung ein frei verfügbares Datenset namens SYNTHIA mit pixelbasierten Annotationen zur semantischen Segmentierung bodengestützter städtischer Szenen und zusätzlich verfügbarer Tiefeninformation. Dieses enthält über 213 400 synthetisch gerenderte Bilder und Variationen in Hinblick auf Jahreszeit, Blickwinkel, Wetter- und Umgebungsbedingungen. Das Datenset wurde eingesetzt, um das Lernverhalten von CNNs zur semantischen Segmentierung bei Verwendung synthetischer Daten zu analysieren, wobei auch hier bei ausschließlich synthetischen Trainingsdaten das mehrfach erwähnte Problem des *Reality Gaps* berücksichtigt werden muss. Neben der Möglichkeit von gemischten Trainingsdaten bzw. *Fine-Tuning* des Modells mit Daten der späteren Anwendungsdomäne wurde hier ein Verfahren namens BGC (engl.: *Balanced Gradient Contribution*) [112] eingesetzt, bei dem dem Netz in jedem Trainingsschritt ein Stapel mit Daten übergeben wird, der Bilder aus beiden Domänen in einem bestimmten Verhältnis enthält. Vor allem Klassen mit dynamischen Objekten wie Fußgänger oder Autos profitierten bei den Untersuchungen von dieser Form des gemischten Trainings. Shafaei et al. [46] kamen zu vergleichbaren Ergebnissen und konnten durch Versuche mit mehreren Datensätzen bei synthetisch trainierten Netzen eine ähnliche Generalisationsfähigkeit wie bei real trainierten Netzen nachweisen. Da sowohl die Leistung bei semantischer Segmentierung als auch bei Tiefenschätzung betrachtet wurde, flossen in diese Auswertung sowohl visuelle Bildmerkmale höherer als auch mittlerer Stufe mit ein.

In [111] wurde für den Anwendungsfall des autonomen Fahrens ein reales Datenset mit Namen KITTI vorgestellt, das durch Testfahrten eines mit verschiedensten Sensoren ausgestatteten Fahrzeugs in Karlsruhe erstellt wurde. Es enthält eine Vielzahl von zeitlich synchronisierten Sensordaten, unter anderem elektro-optische Kamerabilder, und eine entsprechende Variation in Bezug auf Objekte und Szenarios. Aufgrund der zur Verfügung stehenden Annotationen wurde es zu einem häufig verwendeten öffentlich verfügbaren Benchmark Datenset für verschiedenste CV-Anwendungen wie Stereosicht, optischer Fluss, Objektdetektion oder auch Tracking. Gaidon et al. [101] griffen dieses bewährte Datenset auf und präsentierten ein Verfahren zur Erstellung virtueller Duplikate der realen Daten. Sie erzeugten so wiederum mit Hilfe der *Unity Engine* ein vollständig gelabeltes, dynamisches und näherungsweise photorealistisches Datenset mit nachgestellten virtuellen Bildpaaren und nannten es Virtual KITTI (VKITTI). Die nachgestellten realen Videosequenzen resultierten in einem Datenset mit 2131 Bildpaaren. Ziel war es, den Einfluss des *Reality Gaps* und verschiedener Rendering Verfahren auf die Detektionsleistung beim Einsatz von Multi-Objekt Tracking Algorithmen zu untersuchen. Es wurde für diesen Anwendungsfall gezeigt, dass sich real trainierte Modelle in beiden Domänen ähnlich verhalten und daher eine Evaluierung von Einflussfaktoren wie Wetter und Beleuchtung in der Simulation möglich ist. Übereinstimmend mit den anderen Veröffentlichungen konnte auch hier durch Vortrainieren mit synthetischen Daten die Leistungsfähigkeit gesteigert werden.

Prakash et al. [106] stellten das Verfahren der *Structured Domain Randomization (SDR)* zur Erzeugung synthetischer Datensets vor, bei der im Gegensatz zur klassischen *Domain Randomization* die Struktur und der Kontext der Szene berücksichtigt werden. Die Objekte werden zufällig platziert, allerdings anhand einer speziellen aufgabenspezifischen Wahrscheinlichkeitsverteilung. Die Autoren evaluierten die so erzeugten synthetischen Trainingsdaten anhand der Leistung bei der Fahrzeugdetektion auf den realen KITTI Daten. Sie stellten fest, dass im Gegensatz zur *Domain Randomization* realistischere Szenen gerendert werden. Die Daten beinhalteten dennoch genügend Variation, um die Duplikate des Virtual KITTI Datensatzes bei rein synthetischem Training zu übertreffen und um eine passende Grundlage zu bilden, wenn ein späteres Fine-Tuning auf realen Daten durchgeführt wird.

Aufgrund fehlender bzw. ungeeigneter Datensätze für die Evaluierung von UAV-basiertem Tracking haben Müller et al. [99, 100] auf Basis der *Unreal Engine 4* den UAV Simulator *Sim4CV* vorgestellt. Dieser kann sowohl für luft- als auch bodengestützte überwachte Lernverfahren und Verfahren, die auf *Reinforcement Learning* basieren, eingesetzt werden und berücksichtigt in Echtzeit die Flugeigenschaften bei dynamischen und veränderlichen Umgebungsbedingungen. Auf Basis dessen wurde ein synthetischer und automatisch gelabelter Benchmark Datensatz generiert. In online und offline Evaluierungen (d.h. mit und ohne Einfluss des Trackers auf die UAV-Steuerung) wurde der Einfluss bestimmter Faktoren wie Kamerabewegungen, Blickwinkel, Verdeckungen und Größenänderungen des Ziels auf die Leistung gängiger Trackingalgorithmen untersucht.

Analyse

Zusammenfassend kann festgehalten werden, dass in der Literatur bereits mehrere Ansätze zum Einsatz synthetischer Daten und der zugehörigen automatisch erzeugten *Ground Truth* erprobt wurden. Es wurden sowohl einfachere Ansätze betrachtet, die 3D CAD Modelle in Kombination mit realen oder synthetischen Hintergründen zur Erzeugung verwenden, als auch die annähernd photorealistische Grafik moderner Computerspiele und schließlich auch physikalisch basierte Modellier- und Simulationsumgebungen mit konfigurierbaren Umgebungs- und Sensormodellen. Häufig wurde dabei anhand von Leistungsdifferenzen das Trainingsverhalten der jeweils verwendeten Algorithmen in Bezug auf reale, synthetische oder gemischte Trainingsdaten untersucht.

Insgesamt ist in diesem Kontext erneut zu betonen, dass bei der Verwendung von computergestützten Bildverarbeitungsalgorithmen eine möglichst realistische Darstellung im Sinne der menschlichen

Wahrnehmung nicht als Maßstab für die Eignung von synthetischem Bildmaterial herangezogen werden kann [49, 80, 97]. Vielmehr werden andere Kriterien benötigt, um die einflussreichen Bildmerkmale zu beschreiben und die relevanten Unterschiede zur Realität zu identifizieren. Die Literatur betrachtet neben der Anwendung synthetischer Daten nur sehr selten eine gezielte Analyse der ausschlaggebenden Einflussfaktoren. So bleibt für die meisten Anwendungsfälle offen, inwiefern die Trainingsdatenzusammensetzung optimiert werden kann und welche Einflussfaktoren, d.h. welche Bild- oder Simulationseigenschaften, für den vorhandenen *Reality Gap* verantwortlich sind. Es gibt dennoch einige wenige Quellen in der Literatur, die Ansätze zur Analyse des *Reality Gaps* und zur Ableitung von Gestaltungsrichtlinien für synthetische Daten liefern. Im folgenden Kapitel werden dieser Konzepte vorgestellt.

2.1.3 Ansätze zur Analyse der Einflussfaktoren

Peng et al. [98] untersuchten den Einfluss fehlender oder falsch dargestellter Bildeigenschaften bei synthetischen Daten aus gerenderten CAD Modellen für den Anwendungsfall der Objektdetektion. Es wurden Eigenschaften wie Objekttextur, Farbe, Pose oder Hintergrundszenerie betrachtet. Die bei der Detektion zum Einsatz kommenden *deep-learning* basierten neuronalen Netzwerke sind im Allgemeinen *Black-Box* Modelle, bei denen eine Auswertung der intern verwendeten Merkmale oder eine Fehleranalyse bei falschen Detektionen komplex und mehrdeutig ist. Um dennoch Aussagen über den Einfluss bestimmter Eigenschaften treffen zu können, wurden mehrere synthetische Datensets mit unterschiedlichen und teilweise gegensätzlichen Gestaltungsmerkmalen erstellt. Im Fall der Objekttexturen bedeutet dies z.B. die Verwendung realer Farbtexturen und einheitlicher grauer Texturen. Die Invarianz gegenüber derartigen Merkmalen wurde als Fähigkeit des Netzwerks definiert, trotz dieser fehlenden Bildmerkmale die für die jeweilige Objektklasse wichtigen Detektionsmerkmale aus den Daten extrahieren zu können. Die zu untersuchende Netzwerkarchitektur wurde daher unabhängig voneinander mit den gegensätzlichen synthetischen Daten trainiert und auf dem gleichen realen Testdatensatz evaluiert. Der Auswerteansatz bestand aus der Annahme, dass im Falle unwichtiger Bildmerkmale während des Trainings in beiden Fällen ähnliche Gewichtungen gelernt und somit bei der Evaluierung eine ähnliche Leistung erzielt wird. Dies stellt ein einfaches Verfahren dar, den Einfluss bestimmter Bildeigenschaften auf den *Reality Gap* zu bewerten. Es berücksichtigt jedoch keine miteinander korrelierten Merkmale und ist nur anwendbar, wenn die zu untersuchende Eigenschaft direkt separierbar ist und bei der synthetischen Datenerstellung beeinflusst werden kann. Außerdem ist nicht sichergestellt, ob sich je nach Datenzusammensetzung ein relevanter Einflussfaktor direkt signifikant auf die Leistung auswirkt oder durch Störeffekte überdeckt wird.

Kar et al. [113] stellten einen neuartigen Ansatz namens *„Meta-Sim“* vor, bei dem nicht die Erscheinungsform in Bezug auf bestimmte Bildeigenschaften im Fokus steht, sondern die Zusammensetzung der Inhalte und Szenerien. Um dies zu erreichen, wurde die synthetische Datenseterzeugung selbst mit einem neuronalen Netz parametrisiert. Grundlage dafür bildet eine probabilistische Szenengrammatik, wie sie z.B. auch bei Computerspielen zur Erzeugung virtueller Welten zum Einsatz kommt. Das Netzwerk versucht anschließend die Attribute des Szenengraphen derart zu modifizieren, dass die Verteilungsunterschiede zwischen den gerenderten und den realen Daten minimiert werden. Dies umfasst z.B. die Platzierung, Pose oder andere Eigenschaften der in der jeweiligen Situation vorkommenden Objekte. Dadurch hebt sich dieser Ansatz deutlich von der in Kapitel 2.1.1 vorgestellten *Domain Randomization* ab, bei der die Positionierung und Kombination der verwendeten Objekte zufällig und ohne Bezug zur Realität stattfindet. Er erweitert auch das in [106] vorgestellte Verfahren der *Structured Domain Randomization (SDR)*, das lediglich eine vordefinierte Verteilung zugrunde legt, diese aber nicht durch eine neuronales Netz an die realen Gegebenheiten anpasst. Das Vorgehen unterscheidet sich auch erheblich von klassischen Methoden der *Domain Adaptation*, bei denen eine Anpassung im Hinblick auf den Rendering-Stil oder die Darstellungsform, nicht aber in Hinblick auf die inhaltlichen Zusammenhänge der

Objekte vorgenommen wird. Verschiedene Experimente, unter anderem zur Objektdetektion auf dem KITTI Datensatz, haben gezeigt, dass damit die Qualität der synthetischen Inhaltszusammensetzung gegenüber menschlich gestalteten Szenen signifikant gesteigert werden konnte. Dies führte wiederum zu einer höheren Leistungsfähigkeit synthetisch trainierter Modelle bei der Evaluierung auf realen Testdaten und damit zu einer Minimierung des *Reality Gap*. Eine Analyse der generierten Szenarien ergab, dass die vom Netzwerk vorgeschlagenen Attribute durchaus plausibel sind. So wurden z.B. die simulierten Fahrzeuge bevorzugt in Fahrtrichtung ausgerichtet und entsprechend realitätsnahe Abstände zwischen diesen eingehalten. Die durchgeführten Untersuchungen rechtfertigen die von den Autoren gemachte Aussage, dass zur Minimierung des *Reality Gaps* nicht nur die bildbasierten Unterschiede, sondern parallel dazu auch inhaltsbasierte Differenzen beachtet werden müssen. Für ein effektives Training des vorgestellten Netzwerks muss jedoch repräsentatives reales Datenmaterial in ausreichender Menge verfügbar sein, was nicht für jeden Anwendungsfall gegeben ist. Eine weitere Einschränkung ist, dass aufgrund der Verwendung eines neuronalen Netzes keine automatisierte Auswertung möglich ist. Die Rückschlüsse bezüglich der inhaltlichen Gestaltung synthetischer Umgebungen beruhen ausschließlich auf der menschlichen Interpretation der erzeugten Daten und sind daher tendenziell vom Betrachter abhängig und nicht allgemein gültig.

Die Untersuchungen von Hummel [36] stellen im Gegensatz dazu ein Konzept vor, das bildbasierte Unterschiede zwischen den Domänen Realität und Simulation betrachtet und darüber hinaus durch eine statistische Auswertung eine reproduzierbare und wahrnehmungsunabhängige Analyse der Einflussfaktoren ermöglicht. Dies bildet auch die Grundlage für die hier vorgestellte Arbeit. Gegenstand der Untersuchungen in [36] waren reale UAV Luftbildsequenzen, die mit einer statisch am UAV befestigten und um 90 Grad Richtung Boden geneigten elektrooptischen Kamera aufgezeichnet wurden. Die geografische Umgebung des Fluggeländes wurde in der virtuellen Simulationsumgebung VBS3 nachmodelliert. Durch die während dem Flug erfassten Telemetriedaten konnten schließlich gekoppelte reale und synthetische Bildduplikate erzeugt werden. Der verwendete Ansatz ermittelt die Leistungsdifferenzen von Merkmalsdetektoren wie SIFT (engl.: *Scale-Invariant Feature Transform*), SURF (engl.: *Speeded-Up Robust Feature*) oder MSER (engl.: *Maximum Stable Extremal Regions*) zwischen diesen Bildpaaren. Die Leistungsdifferenzen werden durch die in diesem Zusammenhang gebräuchlichen Metriken der relativen und absoluten Wiederholbarkeit ausgedrückt. Merkmalsdetektoren wurden gezielt als Testalgorithmen ausgewählt, da sie, CNNs außen vor gelassen, häufig die Grundlage der Verarbeitungskette bei nicht trainierbaren CV-Anwendungen bilden. Auf der anderen Seite wurden die bildbasierten Unterschiede zwischen einem realen und synthetischen Bildpaar mit Hilfe der Differenzen sogenannter Bildbeschreiber ausgedrückt. Zur Anwendung kam dabei ein Set an MPEG7 Bildbeschreibern, das bestimmte Eigenschaften wie Farbe oder Textur repräsentiert. Es wurde angenommen, dass eine Differenz der jeweiligen Bildbeschreiber zwischen den zwei Bildpaaren auch in einem Leistungsunterschied resultiert. Mit Hilfe einer Regressionsanalyse wurde daher versucht, aus dem Verlauf der Bildbeschreiberdifferenzen für aufeinanderfolgende Bildpaare auf die dazugehörigen Leistungsunterschiede zu schließen. Ist die Güte der Zuordnung ausreichend hoch, kann die Auswertung der dabei berechneten Regressionskoeffizienten herangezogen werden, um die relevanten Bildeigenschaften zu identifizieren. Darüberhinausgehend wurde untersucht, inwiefern sich verschiedene Konfigurationen einer Simulationsumgebung in verschiedenen Szenen auf die Leistung von Merkmalsdetektoren auswirken. Einige Einflussfaktoren waren dabei Schattierungseffekte, Detailgrade der Texturen und der Satellitenaufnahmen, Anti-Aliasing-Filter und die Modellierung verschiedener Kameraeffekte. Ziel des Ganzen war es, Designvorschläge abzuleiten, die die Leistungsunterschiede zwischen den Domänen minimieren sollten.

Analyse

Die aufgeführten Veröffentlichungen über die Nutzung synthetischer Daten zeigen, dass durch die enormen Fortschritte im Bereich *deep-learning* basierter neuronale Netze eine ausschließliche Betrachtung

von Merkmalsdetektoren nicht mehr ausreicht. Zudem liegt der Fokus bei diesen Methoden auf dem Training der Netzwerkarchitektur, wodurch zusätzlich zum alleinigen Vergleich isolierter Bildpaare auch die Zusammensetzung der Trainingsdaten und die Analyse der Unterschiede kompletter Datensets in Betracht gezogen werden muss. Die hier vorliegende Arbeit soll daher ein allgemeines Konzept zur Untersuchung und Extraktion von Gestaltungsrichtlinien und relevanten Bildeigenschaften entwickeln und experimentell evaluieren, das auch für Anwendungsfälle geeignet ist, bei denen CNNs als Bildverarbeitungsalgorithmus eingesetzt werden. Da diese im Allgemeinen *Black-Box* Modelle sind, die keine oder nur eine eingeschränkte Analyse der verwendeten Merkmale erlauben, besteht hier besonderer Forschungsbedarf.

2.1.4 UAV basierte Fahrzeugdetektion als Anwendungsfall

Zur Untersuchung der praxisnahen Anwendung des Konzepts und um zu sehen, ob die ermittelten Einflussfaktoren und Gestaltungsrichtlinien im Umkehrschluss auch einen nachweislich positiven Effekt auf die jeweilige CV-Aufgabe haben, wurde in der vorliegenden Arbeit der Anwendungsfall der UAV basierten Fahrzeugdetektion auf elektro-optischen Luftbildern gewählt. Im folgenden Kapitel wird kurz erläutert, warum dieses Themengebiet für die Auswertung herangezogen wurde.

Sowohl in zivilem als auch militärischem Kontext spielt die Fahrzeugdetektion bei UAV Missionen eine große Rolle. Zudem stellt sie ein aktuelles Forschungsthema dar, für das definierbare und vergleichbare Leistungsunterschiede bestimmt werden können. Eine Vielzahl von Veröffentlichungen beschreibt zum Teil speziell für die Detektion auf Luftbildern entwickelte bzw. optimierte Methoden der Bildverarbeitung und deren Einsatz [13, 21, 23, 24, 114–118]. Objektdetektion bildet darüber hinaus die Grundlage für viele weiterführende CV-Aufgaben, wie Objektzählung, Orientierungsschätzung oder Objektverfolgung. Gerade bei Fahrzeugen als Zielobjekten spielt dabei der Einsatz synthetischer Daten eine wichtige Rolle, da sich im Gegensatz zu Menschen als Zielobjekten technische Gegenstände aufgrund der einfachen und definierten Form auch in Bewegung relativ gut modellieren und simulieren lassen.

UAV basierte Luftbildaufnahmen stellen bei der Fahrzeugdetektion besondere Herausforderungen dar, da sich die Objekte aufgrund der unterschiedlichen Flughöhen und Flugsituationen in Größe, Form, Orientierung und Hintergrund unterscheiden können [22, 23]. Bei der Detektion können zudem unterschiedliche Umgebungs- und Lichtverhältnisse auftreten und Vibrationen des Fluggerätes und Bewegungsunschärfe überlagern häufig als Störeffekte die Sensordaten. Außerdem sind viele feine und unterschiedliche Strukturen und Objekte in den Luftbildern vorhanden, die die Identifikation relevanter Merkmale erschweren und zu Fehldetektionen führen können.

Analyse

Insgesamt gesehen bietet dieser Anwendungsfall eine gute Basis zur Untersuchung der Einflussfaktoren und des Trainingsverhaltens verschiedener zeitgemäßer Detektoralgorithmen bei Verwendung synthetischer Daten. Das im folgenden Kapitel vorgestellte Untersuchungskonzept wird daher anhand der UAV-basierten Fahrzeugdetektion exemplarisch evaluiert.

3 Forschungsfragen und experimentelles Konzept

In Kapitel 1 wurden die Hintergründe dargestellt, die bei einer computergestützten Sensordatenauswertung mit CNNs zur Trainings- und Testdatenproblematik führen. Der Einsatz virtueller Simulationsumgebungen zur Datengenerierung bietet hierfür Lösungsansätze, erfordert jedoch die Untersuchung der Einflussfaktoren und die Ableitung von Gestaltungsrichtlinien, um das synthetische Datenmaterial optimal nutzen zu können. In diesem Kontext ist die vorliegende Arbeit einzuordnen. Im folgenden Kapitel werden nun dahingehend Forschungsfragen formuliert und ein Konzept zu deren Beantwortung entwickelt.

3.1 Ableitung der Forschungsfragen

Die Recherchen im Kapitel 2 haben ergeben, dass bei der Analyse der Leistungsdifferenzen von CV-Algorithmen bei Verwendung synthetischer Daten und bei der darauf aufbauenden Identifikation der dafür verantwortlichen Bild- und Simulationseigenschaften Forschungsbedarf besteht.

Daher wird ein Konzept zur Auswertung der Einflussfaktoren benötigt. Es müssen sowohl voneinander unabhängige reale und synthetische Trainings- und Testdatensätze betrachtet werden als auch gezielt erzeugte korrespondierende Bildpaare aus den beiden Domänen mit definierten Parametervariationen. Die Bildunterschiede zwischen den Domänen und die parametrisierbaren Simulationseigenschaften sollen dahingehend unterschieden werden, ob sie einen Leistungsunterschied der Algorithmen verursachen oder dafür irrelevant sind. Eine gezielte Variation einzelner Parameter der Testdaten soll in einer Art rückgekoppelten Analyse die mit Hilfe der statistischen Auswertung identifizierten Einflussfaktoren bestätigen. Die Untersuchung des Einflusses verschiedener Parametervariationen bei der Trainingsdatengenerierung schließt den Kreis und bezieht auch diesen elementaren Bestandteil des CV-Algorithmus mit ein. Das Gesamtkonzept wird am Beispiel der Fahrzeugdetektion auf UAV-Luftbildern evaluiert und soll die gezielte Generierung und den effizienten Einsatz synthetischer Sensordaten optimieren. Die dabei auftretenden Forschungsfragen lassen sich in drei Gruppen einteilen, die im Folgenden näher erläutert werden. Zudem werden die jeweiligen Forschungsfragen und die zugrundeliegende Motivation tabellarisch dargestellt.

3.1.1 Wahl der Trainingsdatenzusammensetzung und Auswirkungen auf die Detektionsleistung

Im Folgenden wird anhand des recherchierten Standes der Technik der Forschungsbedarf für die Wahl der Trainingsdatenzusammensetzung analysiert. Anschließend werden daraus offene Forschungsfragestellungen abgeleitet und die zugrundeliegende Motivation erläutert. Dies bildet die Basis für die Entwicklung des Untersuchungskonzepts und für die Gestaltung der Experimentalbeschreibung.

Forschungsbedarf

Kapitel 2.1.2 zeigt, dass in der Literatur bereits einige Veröffentlichungen existieren, die sich mit dem Einsatz synthetischer Trainingsdaten und verschiedenen Trainingsdatenkonfigurationen beschäftigen. Nur sehr wenige liefern jedoch einen Vergleich der Leistungsfähigkeit aller drei Trainingskonfigurationen (real, synthetisch, gemischt) für einen speziellen Anwendungsfall unter definierten Randbedingungen und mit den jeweils zugehörigen Testdatensätzen. Zudem werden die so erzeugten Modelle meist nicht mehr zusätzlich auf inhaltsgleichen realen und synthetischen Bildpaaren evaluiert. Jedoch ermöglicht erst diese vollständige Auswertung in ihrer Gesamtheit zuverlässige Rückschlüsse auf die Modellleistung und den auftretenden *Reality Gap*.

Ableitung der Forschungsfragen

Da nahezu alle aktuellen Detektoralgorithmen trainierbar sind, muss der Einfluss der **Trainingsdatenzusammensetzung** auf das Detektionsergebnis in Bezug auf die reale und synthetische Domäne untersucht werden. Als Grundlage dafür soll ein passender Detektor bestimmt werden, der als Testalgorithmus dient und mit entsprechend ausgewählten oder generierten realen und synthetischen Referenzdatensätzen trainiert werden kann. Ein Vergleich realer und synthetischer Trainingskonfigurationen auf Basis der resultierenden Detektionsleistung ist nötig, um eine mögliche Richtungsabhängigkeit des *Reality Gaps* zu zeigen. Dies spielt zum einen eine Rolle, wenn real trainierte Algorithmen auf synthetische Daten angewandt werden, um individuelle Einflussfaktoren auf die Detektionsleistung in der Simulation zu evaluieren, und zum anderen, wenn aufgrund fehlender Realdaten rein synthetische Trainingsdaten zur Anwendung kommen. Zudem muss untersucht werden, inwiefern sich ein hybrider Ansatz mit gemischten Trainingsdaten aus beiden Domänen auf die Detektionsleistung und die Generalisationsleistung auswirkt und welche Randbedingungen dabei eine Rolle spielen. Schließlich stellt sich in diesem Zusammenhang noch die Frage, wie sich die drei Trainingsdatenzusammensetzungen (real, synthetisch, gemischt) auf entsprechenden realen und synthetischen Bildduplikaten als Testdaten verhalten. Diese weisen einen nahezu identischen Bildinhalt auf, unterscheiden sich lediglich in den visuellen Bildeigenschaften und ermöglichen somit detailliertere Analysen. Tab. 2 listet die daraus abgeleiteten und explizit formulierten Forschungsfragen auf.

Tab. 2 Forschungsfragen zur Wahl der Trainingsdatenzusammensetzung

Trainingsdatenzusammensetzung - Forschungsfragen
1. Wie verhalten sich **real trainierte** Modelle auf synthetischen Testdaten im Vergleich zu realen Testdaten? *Motivation:* Evaluation bestimmter Parametereinflüsse in der Simulation
2. Was sind die Auswirkungen der Verwendung rein **synthetischer Trainingsdaten**? *Motivation:* Anwendung synthetisch trainierter Modelle auf reale Testdaten → Lösung der Trainings- und Testdatenproblematik Untersuchung diskreter Parameterabstufungen bei der Datengenerierung
3. Welche Leistung erreicht ein Modell, das mit **gemischten Trainingsdaten** aus beiden Domänen trainiert wurde? *Motivation:* Steigerung der Detektions- und Generalisationsleistung
4. Wie unterscheidet sich die Leistung dieser drei Trainingskonfigurationen auf inhaltsgleichen **realen und synthetischen Bildduplikaten**? *Motivation:* Untersuchung der Auswirkungen des *Reality Gaps*

3.1.2 Statistische Auswertung der Einflussfaktoren auf die Detektion

Nun wird der Forschungsbedarf in Hinblick auf die statistische Auswertung der Einflussfaktoren dargestellt und es werden wiederum offene Forschungsfragen abgeleitet.

Forschungsbedarf

In Kapitel 2.1.2 wurde gezeigt, dass bereits sehr viele Veröffentlichungen die Verwendung von synthetischem Datenmaterial untersuchen. Jedoch betrachten diese nahezu keine Auswertung der zugrundeliegenden Einflussfaktoren und liefern daher auch keine Richtlinien und Anhaltspunkte für die Gestaltung synthetischer Datensätze und Simulationsumgebungen. In Kapitel 2.1.3 wurden schließlich einige wenige Ansätze recherchiert, die im weitesten Sinne eine Auswertung der Einflussfaktoren zum Ziel haben. Diese betrachten jedoch entweder lediglich den Einfluss direkt separierbarer Eigenschaften bei der synthetischen Datengenerierung oder basieren auf der menschlichen Interpretation oder wurden nicht für den Einsatz mit *deep-learning* basierten Testalgorithmen konzipiert, bei denen auch das

Training und die Trainingsdatenzusammensetzung eine entscheidende Rolle spielt. Daher wird in dieser Arbeit ein Konzept entwickelt und exemplarisch evaluiert, das die beschriebenen Einschränkungen umgeht und eine umfassende statistisch basierte Analyse der Einflussfaktoren bei der Verwendung *deep-learning* basierter Algorithmen unter Nutzung synthetischer Sensordaten bereitstellt.

Ableitung der Forschungsfragen

Der zweite Teil bildet das Kernstück dieser Arbeit. Er ist offen, wie eine **statistische Auswertung** zur **Identifikation relevanter Bildunterschiede** zwischen realen und synthetischen Sensordaten und zur **Identifikation der Einflussfaktoren** der durch die Bildunterschiede hervorgerufenen **Leistungsunterschiede** vorgenommen werden kann. Es existiert der Ansatz, ausgewählte Bildbeschreiber als Merkmale und als Eingangsgröße für die statistische Auswertungskette zu verwenden (s. Kapitel 2.1.3). Es werden nun zwei Ziele verfolgt. Zum einen soll untersucht werden, inwieweit es möglich ist, auf Basis der ausgewählten Bildbeschreiber zwischen Daten aus der realen und synthetischen Domäne zu unterscheiden und so die für diese Unterscheidung relevanten Bildeigenschaften als Bestandteil des *Reality Gaps* zu identifizieren. Dies betrifft sowohl einzelne gekoppelte reale und synthetische Bildpaare mit identischem Bildinhalt als auch voneinander unabhängige reale und synthetische Datensätze, wie sie z.B. für das Training des ausgewählten Detektors verwendet werden. Zum anderen soll darauf aufbauend untersucht werden, welche Güte bei der Unterscheidung der Detektionsergebnisse eines synthetisch trainierten *deep-learning* basierten Fahrzeugdetektors zwischen korrekten und inkorrekten Detektionen erreicht werden kann. Dabei stellt sich unter anderem die Frage, welche Bildeigenschaften in diesem Zusammenhang entscheidend sind und inkorrekte Detektionen begünstigen. Insgesamt soll dadurch eine Möglichkeit geschaffen werden, die im ersten Teil trainierten *Black-Box* Detektormodelle und die zugehörigen Leistungsunterschiede in Abhängigkeit der verwendeten Datendomäne zu analysieren. Daraus lassen sich die in Tab. 3 gelisteten Forschungsfragen ableiten.

Tab. 3 Forschungsfragen zur statistischen Auswertung der Einflussfaktoren

Statistische Auswertung - Forschungsfragen
1. Mit welcher Zuverlässigkeit kann mit Hilfe von Bildbeschreibern zwischen **realen und synthetischen Bildpaaren** unterschieden werden und welche Einflussfaktoren sind dabei entscheidend? *Motivation:* Ermittlung der für den *Reality Gap* verantwortlichen Bildeigenschaften (Bildunterschiede)
2. Welche Zuverlässigkeit erreicht der Ansatz bei der Unterscheidung zwischen **voneinander unabhängigen realen und synthetischen Trainings- und Testdaten** und welche Einflussfaktoren sind dabei entscheidend**?** *Motivation:* Ermittlung der für den *Reality Gap* verantwortlichen Bildeigenschaften (Bildunterschiede)
3. Wie hoch ist die Güte der Klassifikation in **korrekte und inkorrekte Detektionsergebnisse** und welche Einflussfaktoren sind dabei entscheidend**?** *Motivation:* Ermittlung der für den Testalgorithmus verantwortlichen Bildeigenschaften (Leistungsunterschiede) → Verbesserung der Trainingsdatengenerierung

3.1.3 Analyse von Parametereinflüssen auf die Detektionsleistung

Abschließend wird im dritten und letzten Teil die Analyse der Parametereinflüsse auf die Detektionsleistung betrachtet.

Forschungsbedarf

Hierbei müssen zwei Teile des Untersuchungsschwerpunktes unterschieden werden. Zum einen werden Testdatensätze benötigt, die voneinander entkoppelte und annotierte Parametervariationen enthalten. Im Gegensatz zu den meisten bisherigen Veröffentlichungen (s. Kapitel 2.1.2 und 2.1.3) müssen die Modelle dabei nicht nur in Bezug auf ihre Leistung sondern auch in Bezug auf ihre Stabilität gegenüber

sich verändernden Randbedingungen hin untersucht werden. Zum anderen müssen abschließend verschieden gestaltete Trainingsdatensätze generiert und die damit trainierten Modelle erneut evaluiert werden. Somit werden die Auswirkungen der identifizierten Einflussfaktoren auch unmittelbar für den entsprechenden Anwendungsfall bewertet und untersucht, inwiefern und in welchem Maß dadurch Verbesserungen bei der Leistungsfähigkeit erzielt werden können. Diese Rückkopplung ist bei den bisherigen Untersuchungen in der Literatur nur sehr selten bis gar nicht zu finden, ist aber nötig, um die Einzelergebnisse in einen generellen Zusammenhang zu bringen.

Ableitung der Forschungsfragen

Im dritten Teil sollen nun die Auswirkungen einer gezielten **Variation einzelner entkoppelter Parameter** auf das Detektormodell und die Detektionsleistung beobachtet werden. Dabei wird untersucht, inwieweit in einer Art rückgekoppelten Analyse die mit Hilfe der statistischen Auswertung identifizierten Einflussfaktoren bestätigt werden können. Im ersten Schritt soll dies die Auswirkungen auf die Detektionsleistung verschiedener Objekt-, Umgebung-, Sensor- und Simulationsparameter in den verwendeten Testdatensätzen analysieren. Darüber hinaus ist im nächsten Schritt außerdem fraglich, wie synthetische Trainingsdatensätze in Bezug auf die Variation in den Datensetgestaltungs-, Sensor- und Simulationsparametern zusammengesetzt werden müssen, um die Anpassung auf reale Bedingungen zu verbessern. Die Betrachtung verschiedener Parametervariationen bei der Trainingsdatengenerierung schließt somit den Kreis und bezieht auch diesen elementaren Bestandteil des Detektionsalgorithmus in die Auswertung mit ein. Die abgeleiteten Forschungsfragen sind in Tab. 4 aufgeführt.

Tab. 4 Forschungsfragen zur Analyse von Parametereinflüssen auf die Detektionsleistung

Parametereinflüsse - Forschungsfragen	
1.	Welche Einflüsse haben verschiedene **Objekt-, Umgebungs-, Sensor- und Simulationsparameter** auf die Detektionsleistung? *Motivation:* Stabilitäts- und Sensitivitätsbeurteilung der Modelle
2.	Wie müssen synthetische Trainingsdatensätze in Bezug auf **Datensatz-, Sensor- und Simulationsparameter** gestaltet werden? *Motivation:* Verbesserung der Anpassung auf die realen Anwendungsbedingungen

3.1.4 Fazit

Gemeinsames Ziel aller Einzelfragen ist die Untersuchung und Zusammenstellung von Einflussfaktoren und Gestaltungsrichtlinien in Bezug auf die Trainingsdatenzusammensetzung, die relevanten Bildeigenschaften, die Parametereinflüsse und die Trainingsdatengenerierung bei der Verwendung synthetischer Daten. Dies soll einen Einblick gewähren, wie derartige Daten in den betrachteten *deep-learning* basierten Detektionsnetzen verarbeitet werden, wie sie möglichst effektiv eingesetzt werden können und welche Maßnahmen den *Reality Gap* verringern. In Kapitel 6.4 werden alle Ergebnisse in Bezug auf die gestellten Forschungsfragen zusammengefasst.

3.2 Untersuchungskonzept und Experimentalbeschreibung

Die im vorigen Kapitel definierten Forschungsfragen bilden die Grundlage für die Entwicklung eines Konzeptes zur Untersuchung der behandelten Aufgabenstellungen. Im Folgenden werden die zur Umsetzung nötigen Schritte der Reihe nach beschrieben, wobei darauf geachtet wurde, dass das resultierende Gesamtkonzept unabhängig vom konkreten Anwendungsfall bleibt und im Allgemeinen zur Identifikation relevanter Einflussfaktoren bei Verwendung eines beliebigen trainierbaren Detektionsalgorithmus anwendbar ist. Abb. 4 zeigt eine Visualisierung dieses Gesamtkonzepts. Die drei Untersuchungsschwerpunkte sind jeweils farblich hervorgehoben und sollen die Zuordnung der darin

aufgestellten Forschungsfragen zu den jeweiligen Konzeptblöcken verdeutlichen. Die folgenden Unterkapitel beschreiben jeweils diese drei Untersuchungsschwerpunkte und enthalten eine tabellarische Auflistung der zur Umsetzung notwendigen Schritte.

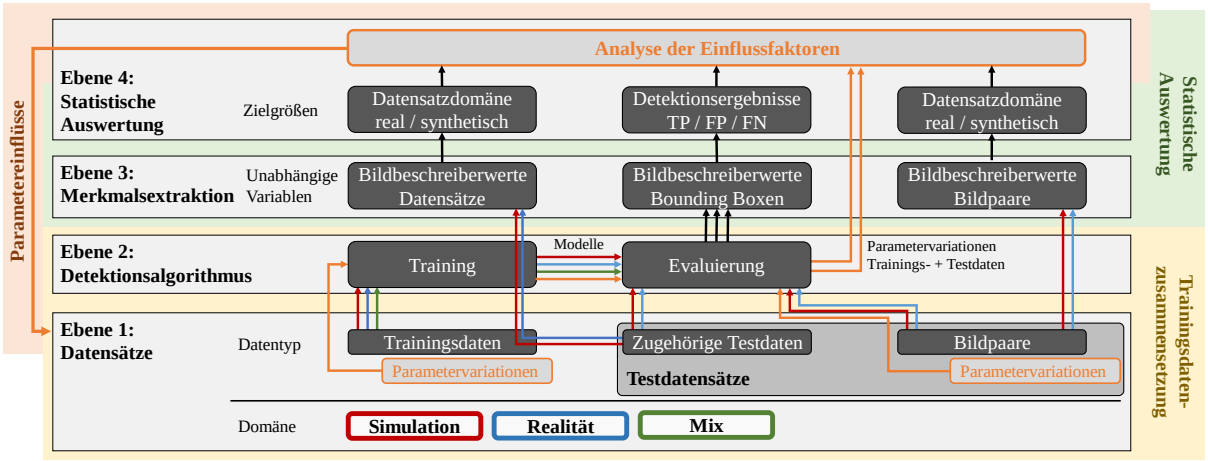


Abb. 4 Gesamtkonzept zur Untersuchung der aufgestellten Forschungsfragen und dem primären Ziel der Analyse des *Reality Gaps* durch Identifikation relevanter Einflussfaktoren bei Verwendung eines trainierbaren Detektionsalgorithmus. Die einzelnen Untersuchungsschwerpunkte (Trainingsdatengenerierung, Statistische Auswertung, Parametereinflüsse) sind farblich gekennzeichnet.
TP: *True Positive*; FP: *False Positive*; FN: *False Negative*

3.2.1 Wahl der Trainingsdatenzusammensetzung und Auswirkungen auf die Detektionsleistung

Der erste Teil der Forschungsfragen behandelt die Auswirkungen der **Trainingsdatenzusammensetzung** auf das Detektionsergebnis in Bezug auf die reale und synthetische Domäne. Ebene 1 des Konzepts (s. Abb. 4) beinhalt dabei die zur Umsetzung benötigten Typen von Datensätzen. Zum einen sollen diese die Domänen Simulation und Realität abdecken und auch gemischte Datensätze aus beiden Domänen enthalten. Zum anderen wird zwischen Trainingsdaten und Testdaten unterschieden. Die Trainingsdaten müssen in Bezug auf Menge, Bildinhalt und die enthaltene Variation zum Training eines Detektionsalgorithmus für einen bestimmten Anwendungsfall geeignet sein. Dabei sollen sowohl allgemein verfügbare Benchmark-Datensätze als Referenz betrachtet werden, als auch eigene erzeugte Datensätze mit teils unterschiedlichen Parametervariationen. Eine beim Training nicht verwendete Untermenge dieser Daten wird ebenso für Testzwecke verwendet wie auch davon unabhängige gekoppelte reale und synthetische Bildpaare. Letztere dienen der detaillierteren Analyse der durch den *Reality Gap* verursachten Leistungsunterschiede und sollten ebenfalls Untergruppen mit teils unterschiedlichen Parametervariationen aufweisen. Die Umsetzung dieses Abschnittes erfordert die Recherche passender verfügbarer Datensätze. Zudem ist eine Simulationsumgebung mit einer modellierten virtuellen Welt zur Generierung synthetischer Datensätze und Anpassung der darin enthaltenen Parametervariationen nötig. Zur Erzeugung der Bildpaare muss schließlich nach einem vordefinierten Schema eine reale Datenerfassung durchgeführt und die erfassten Bilddaten in der Simulationsumgebung nachmodelliert werden.

Die zweite Ebene des Konzepts (s. Abb. 4) umfasst Training und Evaluierung des **Detektionsalgorithmus**. Nachdem ein passender Testalgorithmus, im hier betrachteten Fall ein *deep-learning* basierter Objektdetektor zur Fahrzeugerkennung auf Luftbildern, ausgewählt wurde, wird er mit den erwähnten Trainingsdaten angelernt. Die resultierenden Modelle werden auf die beschriebenen Testdaten angewandt, sodass schließlich die Leistungsunterschiede ausgewertet und in Bezug mit den Trainingsdaten, der Domäne und der Datensatzzusammensetzung gebracht werden können.

Tab. 5 listet die zur Umsetzung beider Ebenen nötigen Schritte, auf die jeweils in Kapitel 4 und 5 näher eingegangen wird.

Tab. 5 Tabellarische Auflistung der zur Umsetzung des Konzepts in Hinblick auf die Untersuchung der Trainingsdatenzusammensetzung nötigen Einzelschritte

Trainingsdatenzusammensetzung - Umsetzung
Datensatzgenerierung
- Auswahl und Inbetriebnahme einer virtuellen Simulationsumgebung und Modellierung einer virtuellen Welt
- Generierung eines synthetischen Trainings- und Testdatensatzes mit definierten Parametervariationen
- Auswahl eines passenden realen Trainings- und Testdatensatzes für Vergleichszwecke
- Erstellung gemischter Trainingsdatensätze mit Bilddaten aus beiden Domänen
- Durchführung von Realflügen und Generierung von realen und synthetischen Bildpaaren zu Testzwecken
Detektionsalgorithmus
- Auswahl und Implementierung eines Algorithmus zur UAV-basierten Fahrzeugdetektion
- Untersuchung des Trainingsverhaltens
- Bewertung der Leistung anhand einer passenden Evaluierungsmetrik

3.2.2 Statistische Auswertung der Einflussfaktoren auf die Detektion

Der zweite Teil des Forschungsgegenstandes bezieht sich auf die **statistische Auswertung** zur **Identifikation relevanter Bildunterschiede** zwischen realen und synthetischen Sensordaten und zur **Identifikation der Einflussfaktoren** der durch die Bildunterschiede hervorgerufenen **Leistungsunterschiede.** Das Konzept aus Abb. 4 sieht dazu in Ebene 3 eine Merkmalsextraktion zur Generierung unabhängiger Variablen vor. Ziel ist eine Erfassung und Beschreibung der für Bild- und Leistungsunterschiede verantwortlichen Faktoren. Dies geschieht durch ein Set von passend ausgewählten Bildbeschreibermetriken zur Charakterisierung vielfältiger Bildeigenschaften. Zur Analyse der Bildunterschiede werden diese auf die miteinander gekoppelten realen und synthetischen Bildpaare und die voneinander unabhängigen realen und synthetischen Testdaten angewandt. Zur Analyse der Leistungsunterschiede hingegen werden damit die Bildeigenschaften der vom Detektor gelieferten *Bounding Boxen* beschrieben.

In Ebene 4 findet die eigentliche statistische Auswertung statt. Als Zielgröße für die Auswertung dient entweder die Unterscheidung der Datensatzdomäne in Realität oder Simulation oder die Beurteilung der Detektionsleistung durch die Unterscheidung in korrekte (TP: *True Positive*) oder inkorrekte (FP: *False Positive*; FN: *False Negative*) Detektionen. Der *Reality Gap* wird somit in seiner Gesamtheit, d.h. sowohl in Bezug auf die Bildunterschiede als auch in Bezug auf die Leistungsunterschiede, analysiert. Die Umsetzung erfordert in diesem Bereich die Auswahl einer passenden regressions- bzw. klassifikationsbasierten Auswertekette. Diese hat im ersten Schritt die Aufgabe, zu überprüfen, inwieweit ein Zusammenhang zwischen der Zielgröße und den unabhängigen Variablen besteht. Im Falle eines Zusammenhangs sollen dann im zweiten Schritt die für die jeweilige Zielgröße relevanten unabhängigen Variablen bzw. Bildeigenschaften mit entsprechenden Methoden der Modellinterpretation identifiziert werden. Insgesamt ermöglicht dieser Aufbau eine umfassende, unabhängige und detaillierte Analyse der Einflussfaktoren. Tab. 6 listet wiederum die zur Umsetzung nötigen Schritte.

Tab. 6 Tabellarische Auflistung der zur Umsetzung des Konzepts in Hinblick auf die statistische Auswertung nötigen Einzelschritte

Statistische Auswertung - Umsetzung
Datensätze
Zielgröße: Bildunterschiede (real / synthetisch)

- Durchführung von Realflügen und Generierung von realen und synthetischen Bildpaaren zur Unterscheidung der Domänen
- Recherche zu öffentlich verfügbaren Bildpaaren für Vergleichszwecke
- Verwendung realer und synthetischer Testdatensätze zur Unterscheidung der Domänen

Zielgröße: Leistungsunterschiede (TP / FP / FN)

- Auswertung der vom Algorithmus ermittelten Detektionen zur Unterscheidung korrekter und inkorrekter *Bounding Boxen*

Auswerteverfahren

- Bewertung von Bildbeschreibermetriken zur Generierung von Merkmalen bzw. unabhängigen Variablen für die Auswertung
- Auswahl und Implementierung einer statistischen Auswertemethode
- Recherche und Integration von Methoden der Modellinterpretation zur Bestimmung relevanter Merkmale

3.2.3 Analyse von Parametereinflüssen auf die Detektionsleistung

Der letzte Teil der Forschungsfragen behandelt die Auswirkungen einer gezielten **Variation einzelner entkoppelter Parameter** auf das Detektormodell und die Detektionsleistung. Das Konzept in Abb. 4 sieht dazu die Variation verschiedener Datensetgestaltungs-, Sensor- und Simulationsparameter bei der Generierung der Trainingsdaten und der Bildpaare vor. Die Umsetzung erfordert eine Simulationsumgebung, die die entsprechenden Parameteränderungen ermöglicht, und eine gezielte reale Datengenerierung bei der Erstellung der Bildpaare, die die jeweils im aktuellen Bild vorherrschenden Parameterwerte erfasst und eine Einflussnahme erlaubt. Im Konzept ist somit sowohl das Trainieren neuer Modelle auf Basis der modifizierten Trainingsdaten und die Evaluierung der resultierenden Leistungsdifferenzen als auch die Evaluierung modifizierter Testdaten zur Evaluierung der Anfälligkeit bestehender Modelle auf sich ändernde Bedingungen vorgesehen (s. Tab. 7). Bei der Gesamtanalyse der Einflussfaktoren werden die Erkenntnisse der Parameteranalyse mitberücksichtigt. Dies dient einerseits der parallelen Überprüfung der im zweiten Teil auf Basis der statistischen Auswertung ermittelten Faktoren. Andererseits können in einer Art Rückkopplung, die im Konzept in Orange hervorgehoben wird, ausgewählte Faktoren bei der Datensatzgenerierung berücksichtigt und deren Einflüsse auf das Trainingsverhalten evaluiert werden.

Tab. 7 Tabellarische Auflistung der zur Umsetzung des Konzepts in Hinblick auf die Analyse von Parametereinflüssen nötigen Einzelschritte

Parametereinflüsse - Umsetzung

Testdatensätze (Stabilitätsanalyse)

- Durchführung von Realflügen und Generierung von realen und synthetischen Bildpaaren mit definierten Parametervariationen → Variation der Objekt- und Umgebungsparameter
- Nachbildung bzw. Überlagerung verschiedener Sensoreffekte → Variation der Sensorparameter
- Einstellung verschiedener Modellier- und Rendering-Effekte in der Simulationsumgebung → Variation der Simulationsparameter

Trainingsdatensätze

Anpassung der automatisierten Trainingsdatengenerierung mit Hilfe der virtuellen Simulationsumgebung

→ Datensätze mit unterschiedlichen Parametern der Datensatzgenerierung

→ Datensätze mit unterschiedlichen Sensorparametern

→ Datensätze mit unterschiedlichen Simulationsparametern

3.2.4 Zusammenfassung und Unterschiede zu bisherigen Konzepten

Insgesamt deckt das vorgestellte Konzept alle geforderten Teilbereiche ab, die für eine Einflussanalyse eines *Black-Box* Detektormodells eine Rolle spielen könnten und gewährleistet Einblicke in die Funktionsweise und die Anfälligkeiten derartiger Modelle. Es weist auf den ersten Blick einige Parallelen zu den bereits in Kapitel 2.1.3 beschriebenen vorangegangenen Untersuchungen von Hummel [36]

auf, unterscheidet sich aber in einigen Punkten auch deutlich von diesem. Im Folgenden werden daher die wesentlichen Unterschiede und Erweiterungen aufgelistet und angesprochen.

In der hier vorliegenden Arbeit werden statt der grundlegenden Merkmalsdeskriptoren komplexe, trainierbare Testalgorithmen, wie z.B. ein *deep-learning* basiertes CNN zur Objektdetektion betrachtet. Dies spiegelt sich auch im Aufbau der konzeptuellen Schritte und in der Fülle der benötigten Datensätze wider. Statt aufeinanderfolgender Videobilder mit einer festen Perspektive werden umfangreiche annotierte Trainingsdatensätze mit großer Variation in Bezug auf Objekte, Blickwinkel und Umgebungsbedingungen und außerdem Einzelaufnahmen mit gezielt erfasster Parametervariation als Testdaten benötigt. Dabei werden sowohl allgemein verfügbare Benchmark-Datensätze als auch eigene generierte Bilddaten berücksichtigt.

Zudem wird der *Reality Gap* nicht nur in Bezug auf Bildeigenschaften und -unterschiede hin untersucht, die zu einem Leistungsunterschied bei Anwendung des Algorithmus führen, sondern parallel dazu auch in Bezug auf vom Algorithmus unabhängige Bildeigenschaften, die im Allgemeinen eine Unterscheidung der Domänen Realität und Simulation ermöglichen. Dies erlaubt eine umfassendere Beurteilung des *Reality Gaps*. Die Schicht der Merkmalsextraktion betrachtet nicht nur die im späteren Verlauf beschriebenen MPEG-7 Bildbeschreiber sondern ein deutlich weitreichenderes Set, das auch allgemeinere Bildeigenschaften (z.B. Helligkeit, Kontrast, …), qualitätsbasierte Eigenschaften (z.B. Rauschen, Unschärfe, …) und semantische Beschreibungen der dargestellten Szenerien bei der Auswertung berücksichtigt.

Des Weiteren werden in [36] die Bildbeschreiberdifferenzen zwischen gekoppelten Bildpaaren berücksichtigt. Das hier vorgestellte Konzept verwendet im Gegensatz dazu hauptsächlich die unverzerrten Absolutwerte und ist dadurch bei der Analyse nicht auf Bildpaare beschränkt. Auch die statistische Auswertung ist umfassender. So liegt der Fokus neben einer Regressionsanalyse mit Rückwärtselimination vor allem auf der Verwendung einer Klassifikationskette zur Analyse der Einflussfaktoren, bei der durch Berechnung mehrerer *Feature Selection* und *Feature Importance* Methoden (s. Kapitel 4.5.2) eine möglichst stabile und umfassende Modellanalyse möglich wird.

Zuletzt ist noch zu erwähnen, dass das in Abb. 4 visualisierte Konzept neben der statistischen Auswertung zum Vergleich auch den direkten Einfluss verschiedener Parametervariationen erfasst und durch die Rückkopplung auf die Trainingsdatengenerierung den Kreis schließt.

Die in Kapitel 2 aufgelisteten Recherchen zum Stand der Technik haben ergeben, dass zum aktuellen Zeitpunkt in Bezug auf die beschriebene Aufgabenstellung zwar bereits einige Teilaspekte für spezielle Anwendungsfälle untersucht wurden, ein derartiges Auswertekonzept in seiner Gesamtheit bisher aber noch nicht vorgestellt wurde. Dieses ist zudem nicht auf einen bestimmten Anwendungsfall zugeschnitten und kann dadurch im Allgemeinen zu einem besseren Verständnis der Einflussfaktoren bei der Verwendung trainierbarer Detektionsalgorithmen beitragen.

4 Methodenauswahl

Im Folgenden werden Auswahlentscheidungen hinsichtlich der verwendeten Methoden und Ausgestaltungsalternativen erläutert, die zur Umsetzung der einzelnen Bestandteile des Konzepts durchgeführt wurden. Wo es notwendig erscheint, wird dabei auf die zum Verständnis notwendigen theoretischen Grundlagen eingegangen. Außerdem werden Trainings- und Testdatensätze vorgestellt, die für den Anwendungsfall der UAV basierten Fahrzeugdetektion verfügbar sind und es wird auf die Auswahl der virtuellen Simulationsumgebung eingegangen, die für die Generierung der synthetischen Vergleichsdaten benötigt wird.

4.1 Datensätze

Zur Untersuchung der in Kapitel 3.1 beschriebenen Forschungsfragen und zur Umsetzung des daraus abgeleiteten Konzepts werden sowohl reale und synthetische Trainingsdatensätze, aber auch gekoppelte reale und synthetische Bildpaare benötigt. Der folgende Abschnitt gibt daher einen Überblick über verfügbare Datensätze für den betrachteten Anwendungsfall. Diese werden für die in Kapitel 3.1.1 und 3.1.2 beschriebenen Schritte des Konzepts benötigt und sollen helfen, den enormen Aufwand zur Erzeugung eigener realer Trainingsdatensätze zu verringern. Zudem bieten öffentlich verfügbare Benchmark Trainingsdaten den Vorteil, dass sie die Vergleichbarkeit der Ergebnisse mit bisherigen Veröffentlichungen ermöglichen und eine reproduzierbare Basis für die Experimente schaffen.

Trainingsdatensätze

Wie bereits erwähnt, stellt das Training und die Evaluierung *deep-learning* basierter Detektoren für die Fahrzeugerkennung auf Luftbildern besondere Herausforderungen an den Algorithmus. Um bei der späteren Anwendung dennoch zuverlässige Ergebnisse erzielen zu können, muss der Datensatz verschiedene Anforderungen erfüllen [117]:

1) **Größenunterschied:** Aufgrund der unterschiedlichen Flughöhen von UAVs tritt ein weiter Bereich an Objektgrößen auf, der vom Datensatz abgedeckt werden muss. Dies betrifft vor allem kleine Objekte, die infolge der geringen Anzahl an Pixeln nur wenig Information und schwer detektierbare Merkmale enthalten.

2) **Perspektive:** Bei geringen Flughöhen können vor allem in Kombination mit einem Gimbal unterschiedliche flache Blickwinkel auftreten, während bei großen Flughöhen die Objekte gewöhnlich aus der Vogelperspektive betrachtet werden. Da viele konventionelle Datensätze für bodengestützte Anwendungen ausgelegt sind, muss dies bei der Auswahl bzw. Erstellung beachtet werden.

3) **Objektorientierung:** Im Gegensatz zur Verkehrsüberwachung als Beispiel können bei UAV Anwendungen aufgrund der Flugbewegungen und der nichtstationären Datenerfassung alle Objektorientierungen auftreten. Die Vielfalt im Trainingsdatensatz muss ausreichen, um dem neuronalen Netz zu lernen, alle Orientierungen eines Objekts derselben Klasse zuzuordnen.

4) **Variation im Hintergrund:** Durch die vielfältigen Anwendungsfälle und das große Blickfeld enthält der erfasste Hintergrund komplexe Szenerien und Strukturen, die einen großen Einfluss auf die Objekterkennung haben können. Um differenzierende Merkmale ermitteln zu können, sollten außerdem verschiedene Objektklassen enthalten sein.

5) **Umgebungsbedingungen:** Dieser Punkt umfasst sowohl verschiedene Umwelt- und Wetterbedingungen als auch Sensorsysteme mit deren spezifischen Eigenschaften und Störeffekten.

In Tab. 8 sind mehrere aktuell verfügbare reale Datensätze mit ihren zugehörigen Eigenschaften aufgelistet. Diese sind öffentlich zugänglich, enthalten real erflogenes Datenmaterial und sind vollständig annotiert. Sie werden in der Literatur häufig als Benchmark-Datensätze für den Vergleich entsprechender Algorithmen herangezogen. Grundsätzlich sind alle diese Datensätze für die Fahrzeugdetektion auf Luftbildern geeignet, wobei der Großteil dennoch für speziellere oder weiterführende Anwendungen entwickelt wurde.

Tab. 8 Überblick über verfügbare Datensätze für die Fahrzeugdetektion auf Luftbildern und deren Eigenschaften. Das UAVDT Datenset wurde aufgrund seiner Eigenschaften als reales Benchmark Datenset für die durchgeführten Untersuchungen ausgewählt.
BB: Anzahl annotierter *Bounding Boxen*; GSD: Bodenpixelauflösung (engl.: *Ground Sample Distance*) in cm pro Pixel; BW: Variation der Blickwinkel; FH: Variation der Flughöhe; Umg: Variation der Umgebungsbedingungen; div.: unterschiedlich
vgl. [51]

	Jahr	Bilder	BB	Klassen	GSD	BW	FH	Umg.	Auflösung
VEDAI[119]	2015	1 250	2 950	9	12.5	✗	✗	✗	512, 1024
OIRDS[120]	2009	908	1 800	4	15.2	**O**	**O**	✗	256-640
COWC[121]	2016	53	90 963	5	15	✗	✗	✗	2K-19K
DOTA[122]	2018	2 806	188 282	15	div.	✗	✓	✗	~4000
DLR 3K[123]	2015	20	5 892	2	13	✗	✗	✗	5616x3744
EAGLE[124]	2020	8 280	215 986	2	5-45	✗	✓	✓	936x936
UAV123[99]	2016	112 578	?	?	div.	✓	✓	**O**	720p-4K
Stanford Drone Dataset[125]	2016	929 499	19 000	6	?	✗	✗	✗	1400x1904
AU-AIR[126]	2020	32 823	132 034	8	div.	✓	✓	✗	1920x1080
UA-DETRAC[127]	2018	140K	1.21M	4	div.	✓	**O**	✓	960×540
CarPK[128]	2017	1 448	89K	1	?	**O**	**O**	**O**	1280 × 720
VisDrone[129]	2018	10 209	54 200	10	div.	✓	✓	✓	2000x1500
UAVDT[22]	2018	40K	750K	3	div.	✓	✓	✓	1024x540

Viele Datensätze sind primär für die Fahrzeugdetektion auf Satellitenbildern oder auf hochauflösenden gekachelten Luftbildern vorgesehen und enthalten nicht die bei UAV Aufnahmen auftretenden Variationen in Bezug auf Blickwinkel und Objektgröße. Sie sind folglich in diesem Kontext nicht oder nur eingeschränkt für den Einsatz als Trainings- oder Testdaten geeignet. Dies trifft zum Beispiel auf die Datensätze VEDAI, COWC, DOTA, DLR 3K, EAGLE und teilweise auch OIRDS zu. Der VEDAI Datensatz enthält zusätzlich verschiedene Auflösungen und Spektralbereiche, jedoch durchgehend kleine Objekte ohne Variation im Blickwinkel. COWC setzt sich aus sechs verschiedenen Datensätzen zusammen, enthält aber nur einen Punkt als Annotation für die Fahrzeugposition und nicht die benötigten *Bounding Boxen*. DOTA ist vorrangig für den Anwendungsfall der Multi-Objekt Detektion im Rahmen der Fernerkundung vorgesehen und enthält Klassen wie „Brücke“, „Fußballfeld“ und „Schiff“. EAGLE und DLR 3K betrachten ebenfalls nur hochauflösende Luftbilder aus der Vogelperspektive, sind allerdings zusammen mit DOTA eine der wenigen Datensätze, bei denen die Labels in Fahrzeugrichtung ausgerichtet sind und die *Bounding Boxen* daher neben der Größe auch die Orientierung als Information enthalten.

Die nachfolgend vorgestellten Datensätze wurden hauptsächlich mit gängigen UAV-Systemen mit zugehöriger elektro-optischer Kamera bei Flughöhen zwischen 0 und ungefähr 100 m aufgezeichnet. Sie enthalten daher tendenziell mehr Varianz und treffen eher den in dieser Arbeit betrachteten Anwendungsfall. Das UAV123 Datenset enthält sogar einen synthetisch generierten Datenanteil, ist jedoch nur für Single Object Tracking geeignet und enthält daher pro Sequenz nur die Annotationen für ein zu verfolgendes Objekt. Das Stanford Drone Dataset enthält zwar annotierte Fahrzeuge, ist allerdings

vorrangig für Tracking und Trajektorienschätzung von Personen gedacht und besteht ausschließlich aus Videodaten, bei denen der Multikopter stationär über einer bestimmten Position auf dem Campus der Stanford Universität schwebt. In [130] wird ein Detektoralgorithmus vorgestellt, der die Metadaten der Bilder bzw. die Telemetriedaten des UAVs zum Aufnahmezeitpunkt bei der Detektion mitberücksichtigt und damit eine höhere Leistung erzielt. Das AU-AIR Datenset enthält für solche Ansätze neben den Annotationen der Objekte auch die Telemetrie wie z.B. Zeitstempel, Position, Flughöhe oder Geschwindigkeit. Es umfasst jedoch nur eine vergleichsweise geringe Variation in Bezug auf die Flughöhe mit maximalen Höhen von 30 Metern und in Bezug auf die Szenerie, da alle Daten an einer einzigen geografischen Position aufgezeichnet wurden. Das UA-DETRAC deckt in Hinblick auf Umgebungszustände, Wetterbedingungen, Objektdichte und Aufnahmeposition eine große Bandbreite an verschiedenen Zuständen ab, weist allerdings immer ähnliche Perspektiven mit einer festgelegten Höhe auf, da es vorwiegend für Detektion und Tracking beim Anwendungsfall der Verkehrsüberwachung konzipiert wurde. CarPK enthält mehrere Parkplatzszenarien mit einer großen Anzahl an Fahrzeugen im Bild und wird für Anwendungen verwendet, bei denen neben der Detektion das korrekte Zählen der vorkommenden Objektinstanzen im Vordergrund steht.

Der VisDrone und der UAVDT Datensatz erfüllen schließlich alle benötigten Anforderungen, weisen eine große Varianz in den erfassten Bilddaten auf und sind sowohl für *Single-Object* und *Multi-Object* Tracking als auch für die hier benötigte Aufgabenstellung der Objektdetektion gedacht. In [116] wird behauptet, dass CarPK, VisDrone2018-car und UAVDT aktuell die anspruchsvollsten groß angelegten drohnenbasierten Datensätze sind.

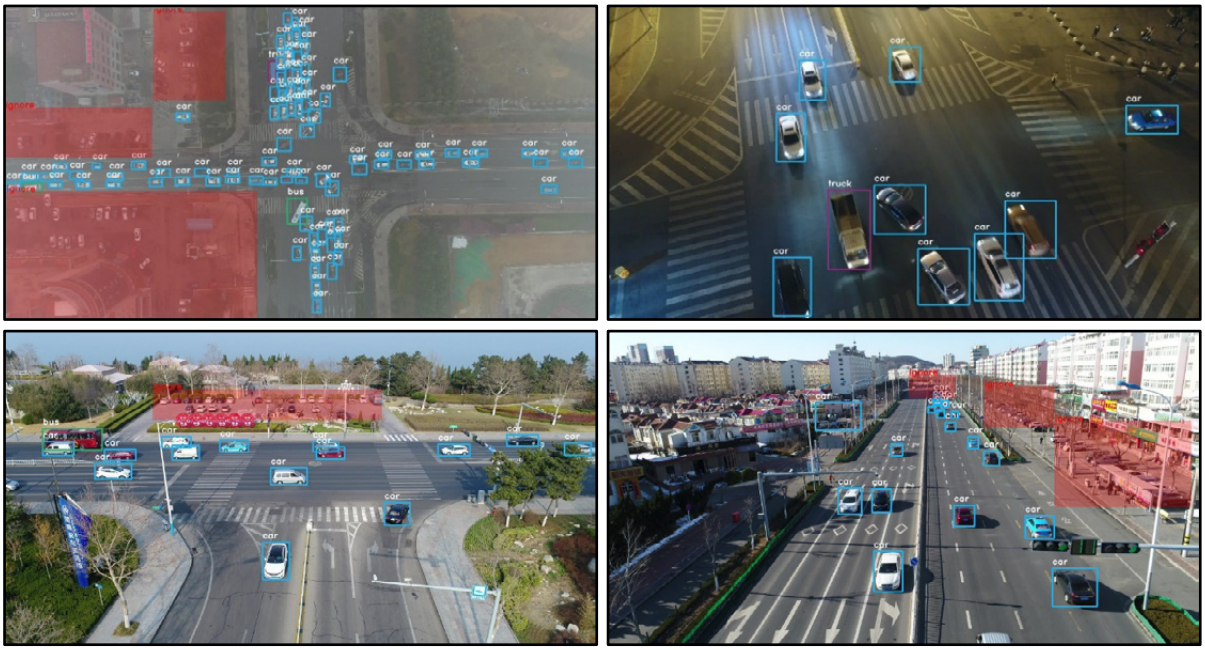


Abb. 5 Beispielbilder aus dem in dieser Arbeit unter anderem verwendeten UAVDT Datensatz [22]. Es sind verschiedene Flughöhen, Umgebungsbedingungen und auch Perspektiven enthalten. Die rot markierten Bereiche wurden bei der Annotation nicht mitberücksichtigt und werden auch bei der Evaluierung ausgegrenzt.

Aufgrund der hohen Anzahl an Bildern und vorkommender *Bounding Boxen* wurde der UAVDT Datensatz als realer Benchmark für die in dieser Arbeit durchgeführten Untersuchungen herangezogen. In Abb. 5 sind Beispielbilder mit den zugehörigen Annotationen dargestellt. Der Datensatz wurde im Jahr 2018 von Du et al. [22] vorgestellt. Aus den dynamisch während dem Flug aufgenommenen Videosequenzen wurden für die Objektdetektion über 40 000 Bilder mit über 750 000 annotierten Fahrzeugobjekten generiert. Das Datenset enthält eine empfohlene Aufteilung in Trainings- und Testdaten und betrachtet die Klassen „Auto“, „LKW“ und „Bus“. Die erfassten Szenerien variieren in ihrer Erscheinungsform, enthalten jedoch hauptsächlich verschiedene innerstädtische Hauptstraßen chinesischer

Großstädte. Zusätzlich zu den Objektannotationen sind weitere Informationen zu den enthaltenen Variationen in den Bildern hinterlegt:

- Wetterbedingungen (Tag; Nacht; Nebel)
- Flughöhe (niedrig: 10-30 m; mittel: 30-70 m; hoch: > 70 m)
- Perspektive (frontal; seitlich; Vogelperspektive)
- Verdeckung (keine; wenig: 1-30 %; mittel: 30-70 %; hoch: > 70 %)
- Aus dem Blickfeld (nicht; wenig: 1-30 %; mittel: 30-50 %; hoch: > 50 %)

Der UAVDT Datensatz erfüllt somit alle Anforderungen, die in dieser Arbeit an den zu verwendenden realen Benchmark Trainingsdatensatz gestellt wurden und dient als Grundlage für die weiterführende Auswertung. Es sei an dieser Stelle zu erwähnen, dass das vorgestellte Konzept unabhängig vom ausgewählten Datensatz anwendbar ist und somit auch in Zukunft für beliebige neue Datensätze herangezogen werden kann.

Gekoppelte reale und synthetische Bildpaare

Für eine detailliertere Analyse der Einflussfaktoren und der Trainingsdatenzusammensetzungen werden zusätzlich entsprechende reale und synthetische Bildpaare benötigt. Durch den nahezu identisch nachmodellierten Bildinhalt ermöglichen sie entkoppelte Rückschlüsse über Faktoren, die eher Inhalt und Szenerie betreffen und solchen, die sich auf die visuellen Eigenschaften oder die Bilddarstellung beziehen.

Abb. 6 Zusammengehörende reale und synthetische Bildpaare des KITTI / VKITTI Datensatzes aus drei verschiedenen Szenerien bzw. Videosequenzen [101].

In Kapitel 2.1.2 wurde bereits der KITTI Datensatz [111] erwähnt, der im Bereich des autonomen Fahrens für eine Vielzahl von CV-Anwendungen als Benchmark dient und von Gaidon et al. [101] unter Verwendung der *Unity-Engine* teilweise nachmodelliert wurde. Abb. 6 zeigt beispielhaft einige der darin vorkommenden Bildpaare. Dieser nachgestellte VKITTI Datensatz diente vorwiegend zur Untersuchung des Einflusses des *Reality Gaps* auf die Leistungsfähigkeit von Multi-Objekt Tracking Algorithmen. Er enthält fünf Videosequenzen aus unterschiedlichen Szenerien mit 2131 korrespondierenden realen und synthetischen Bildpaare mit einer Auflösung von 1242×375 Pixeln. Zusätzlich zu dieser Ausgangsbasis generierten die Autoren sieben weitere synthetische Datensätze mit einer Variation in Bezug auf die Kameraorientierung und einzelne Umgebungsbedingungen und verglichen die daraus resultierenden Leistungsunterschiede.

Trotz des unterschiedlichen Anwendungsgebietes und trotz der bodenbasierten Bilddaten anstelle von Luftbildern kann der Datensatz dennoch als Referenz für die im zweiten Block der Forschungsfragen betrachtete und vom Testalgorithmus unabhängige Unterscheidung zwischen realen und synthetischen Daten dienen. Er liefert dabei für den Fall einer unterschiedlichen zugrundeliegenden Rendering-Umgebung einen Vergleich der identifizierten Einflussfaktoren und ermöglicht eine Überprüfung der statistischen Analysemethode für allgemeinere Anwendungsfälle. Für eine detaillierte Evaluierung der im ersten Teil der Forschungsfragen trainierten Modelle auf speziellen Testdaten, die einer ähnlichen

Szenerie entstanden sind wie die Trainingsdaten und für eine Fahrzeugdetektion auf Luftbildern ausgelegt sind, wird dennoch ein weiterer Datensatz mit Bildpaaren benötigt. Dessen Generierung durch reale UAV-Flüge und die anschließende Nachbildung in der virtuellen Umgebung wird in Kapitel 5.2 detailliert beschrieben. Die gezielte Aufnahme von Sensordaten durch eigene Befliegungen ermöglicht durch die Erfassung möglichst entkoppelter Parametervariationen eine umfassendere und weiterführende Analyse zur Beantwortung der Forschungsfragen bzgl. des Einflusses von Datensetgestaltungs-, Sensor- und Simulationsparametern.

4.2 Virtuelle Simulationsumgebungen

Im folgenden Kapitel wird kurz erläutert, welche Kriterien die virtuelle Simulationsumgebung erfüllen muss, um für die zur Umsetzung des Konzepts nötige Generierung synthetischer Sensordaten geeignet zu sein. Auf Basis dieser Kriterien wurden verfügbare Simulationsumgebungen evaluiert und die *Presagis Modelling & Simulation Suite* [131] für die vorliegenden Untersuchungen ausgewählt. Zunächst wird jedoch in einem kurzen Einschub auf die Grundlagen der Computergrafik und der synthetischen Bildgenerierung eingegangen.

Computergrafik und synthetische Bildgenerierung

Als synthetische Daten sind im hier vorliegenden Fall computergenerierte Bilddaten zu verstehen, die zur Nachahmung realer Sensoraufnahmen herangezogen werden. In [36] wird sehr anschaulich das zugrunde liegende Ablaufschema der synthetischen Bildgenerierung beschrieben, das aus den einzelnen Bestandteilen Szenerie, bildgebendes System, Darstellungsform und Empfänger besteht.

Die Szenerie enthält das Terrain und die sich darauf befindlichen Objekte in einem dreidimensionalen Raum mit definierter Position, Orientierung und Größe. Sowohl Terrain als auch Objekte bestehen dabei aus sogenannten Polygonen, die in Kombination miteinander die jeweilige Form nachmodellieren und das Gittermodell des Objekts bilden. Dieses wird anschließend eingefärbt oder mit zweidimensionalen Bildern texturiert, die auf das dreidimensionale Gittermodell projiziert werden. Das Material der Oberfläche ist schließlich verantwortlich für die Lichtreflexion. Aus der Kombination von Gittermodell (Geometrie), Textur (Farbe und Detail) und Material (Reflexion) entsteht schließlich das modellierte Terrain bzw. Objekt und aus der Kombination von Terrain und Objekten schlussendlich die virtuelle dreidimensionale Szenerie. Dennoch ist anzumerken, dass diese Form der Modellierung lediglich eine vereinfachte Nachbildung der komplexen realen Gegebenheiten darstellt und daher eine Reihe von Freiheitsgraden in Bezug auf Objektgröße, -platzierung, -material und Auflösung bzw. Detailgrad der Texturierung bestehen.

Das bildgebende System berechnet im nächsten Schritt aus einer festgelegten Position das aktuelle Blickfeld auf Basis eines zugrunde liegenden Objektivmodells und konvertiert die dreidimensionale Szenerie in eine zweidimensionale Repräsentation, was einer Transformation in das Kamerakoordinatensystem entspricht. Anschließend folgt die Rasterung des vektorbasierten Bildes in eine pixelbasierte Bildmatrix. Der komplette Vorgang entspricht dem Rendern des Bildes. Objekte außerhalb des Blickfelds (*Culling*), verdeckte Objekte (*Z-Buffer*) und Objekte außerhalb einer maximalen und minimalen Rendering-Distanz (*Clipping*) werden nicht dargestellt, um den Berechnungsaufwand zu verringern. Auch hier gibt es eine Reihe von Freiheitsgraden und Vereinfachungen. Diese umfassen die verwendete Beleuchtungsmethode, das Kameramodell, Sensoreffekte, verschiedene Anti-Aliasing Methoden bei der Rasterung, Kompressionsartefakte und weitere Darstellungsparameter.

Im nächsten Schritt wird ein technisches System (z.B. Monitor) zur Darstellung des digitalen Bildes verwendet und konvertiert es somit in ein wahrnehmbares Format. Der Empfänger am Ende dieses Prozesses kann ein Mensch, ein Bildverarbeitungsalgorithmus oder auch eine objektive

Evaluierungsmethode zum Bildvergleich oder zur Berechnung von Bildeigenschaften sein. Da in dieser Arbeit der Fokus auf Letzterem liegt und die menschliche Wahrnehmung eine untergeordnete Rolle spielt, liegt der Schwerpunkt der synthetischen Bildgenerierung in diesem Fall auf der Gestaltung der Szenerie und der Optimierung des bildgebenden Systems.

Anforderungen an die virtuelle Simulationsumgebung

Kapitel 2.1.2 lieferte einen Überblick über den Einsatz virtueller Simulationsumgebungen bei CV-Anwendungen in der bisherigen Forschung. Dabei wurden verschiedene Ansätze betrachtet. Die einfachsten verwendeten eine Überlagerung von 3D-CAD Modellen mit realen oder synthetischen Hintergründen. Häufig wird auch die annähernd photorealistische Grafik moderner Computerspiele genutzt. Diese bietet jedoch meist nur beschränkte Einflussmöglichkeiten auf die zugrundeliegenden Rendering-Einstellungen und Simulationsparameter und stellt darüber hinaus teilweise nur wenige oder gar keine Möglichkeiten zur Verfügung, die Modellierung der bestehenden virtuellen Welt zu beeinflussen oder bestimmte geografische Bereiche selbst nachzumodellieren. Beides wird jedoch für das hier vorgestellte Konzept benötigt. Es sollen zum einen synthetische Sensordaten unter verschiedenen Bedingungen zum Training und zum Test von Fahrzeugdetektoren generiert werden. Zum anderen muss das Testfluggelände in der virtuellen Welt nachmodelliert werden, um gekoppelte reale und synthetische Bildduplikate zum Vergleich der Leistungsdifferenzen erzeugen zu können. Das zum Einsatz kommende Simulationsprogramm sollte daher folgende Anforderungen erfüllen [51]:

- Physikalisch basierte Sensorsimulation, bei der auch verschiedene Tages- und Jahreszeiten realistisch repräsentiert werden. Für weiterführende Untersuchungen ist es durchaus von Vorteil, wenn neben einer elektro-optischen Kamerasicht auch eine physikalisch korrekte Infrarotsimulation in verschiedenen Wellenlängenbereichen zur Verfügung gestellt wird. Dabei sollte eine Berücksichtigung der Materialeigenschaften von Terrain und Modellen zugrunde liegen, die nicht nur für die Infrarotsimulation benötigt wird, sondern auch die elektro-optische Sensoransicht verbessert.
- Simulation meteorologischer und atmosphärischer Effekte wie z.B. Regen, Nebel, Bewölkungsgrad oder Schattenwurf und Einstellbarkeit der zugehörigen Eigenschaften
- Möglichkeit der Einflussnahme auf Sensormodell und Sensoreffekte wie z.B. Rauschen, Bildunschärfe oder Farbverzerrung, die dem gerenderten Bild überlagert werden
- Programmierschnittstelle und zusätzliche grafische Benutzeroberfläche, die beide einen umfassenden Zugang zu den Einstellparametern ermöglichen
- Breites Spektrum an kompatiblen 3D-Modellen wie z.B. Vegetation und Fahrzeuge. Eine entsprechende Modellierumgebung wird zur Anpassung der Modelle, zur Platzierung in der virtuellen Welt und zur Erzeugung des synthetischen Terrains benötigt.

Auswahl der virtuellen Simulationsumgebung

Die teilweise frei verfügbaren Umgebungen wie *Unreal* [132], *Unity* [133] und *CryEngine* [134] und die drauf aufbauenden weiterentwickelten Simulatoren wie *AirSim* [109] oder *CARLA* [110] sind häufig – mit Ausnahme von *AirSim* – für bodengestützte Anwendungen und kleinflächigere Terrains entwickelt. Außerdem werden meist keine Materialeigenschaften und keine zusätzlichen darauf aufbauenden Sensoren, wie Infrarot, Nachtsicht, LIDAR oder Radar berücksichtigt. *Virtual Battlespace 3* (VBS3) von *Bohemia Simulations* [135] bietet eine vollständige, schnell lauffähige monolithische Anwendung und eine große Objektdatenbank, simuliert jedoch ebenfalls kaum Sensoreffekte und bietet nur eine sehr abstrakte Infrarotsimulation.

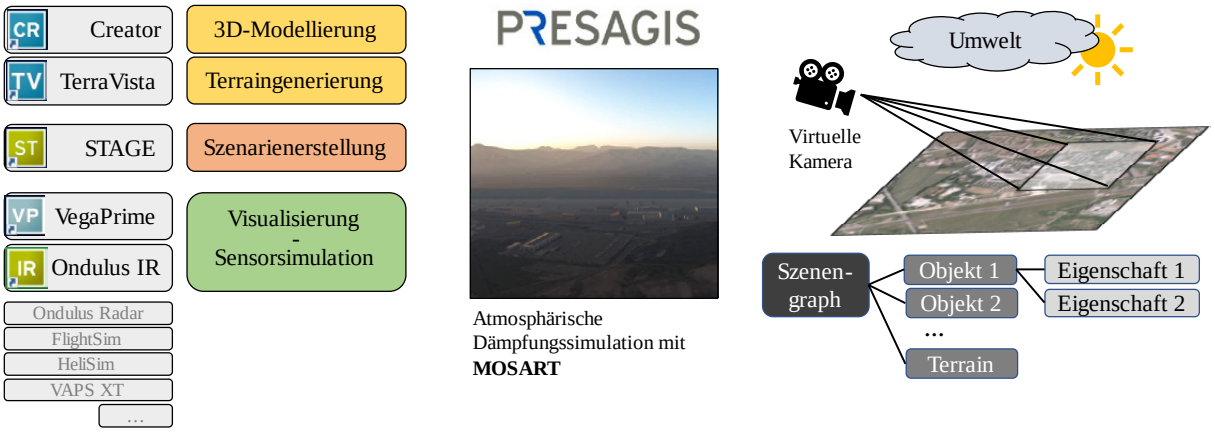


Abb. 7 Links: Modulbasierte Werkzeugkette der verwendeten *Presagis* Modellier- und Simulationsumgebung.
Mitte: Beispielhafte Darstellung der atmosphärischen Dämpfungssimulation mit MOSART.
Rechts: Schematische Darstellung der synthetischen Bildgenerierung und des zugrunde liegenden Szenengraphen.
Quelle: vgl. [131]

Für die in dieser Arbeit vorgestellten Untersuchungen wurde daher die Modellier- und Simulationsumgebung von *Presagis* [131] verwendet, da sie alle gestellten Anforderungen erfüllt und eine modulbasierte Werkzeugkette für Modellierung, Terraingenerierung und Visualisierung bzw. Sensorsimulation zur Verfügung stellt. Sie steht im Rahmen des *Presagis Academic Program* zur Verfügung. Die atmosphärische Dämpfungssimulation wird mit dem MOSART (engl.: *Moderate Spectral Atmospheric Radiance and Transmittance*) Algorithmus [136] berechnet. Trotz der physikalisch basierten Sensorsimulation mit Berücksichtigung von Materialeigenschaften ist dennoch ein annähernd echtzeitfähiges Rendering gegeben. Die gesamte Umgebung ist speziell für den Einsatz in der Luftfahrt ausgelegt und flughöhenabhängige Abstufungen in der Genauigkeit, sogenannte *Levels of Detail* (LOD), ermöglichen die Darstellung weitläufiger georeferenzierter virtueller Einsatzgebiete. Einziger Nachteil ist das Fehlen einer entsprechenden Objektdatenbank. Der zugehörige Modellierer ermöglicht jedoch eine Konvertierung sämtlicher 3D-Formate und die Integration animierter 3D-Volumenbäume von *SpeedTree* [137] führt dazu, dass dies keine Einschränkung darstellt. Abb. 7 zeigt ein Schema des zugrunde liegenden Szenengraphen, ein Beispielbild der MOSART Simulation und die zugehörigen Programmbestandteile, die im Folgenden kurz erläutert werden:

1) **Creator:** Das Programm wird zur 3D-Modellierung der benötigten Gebäude und Objekte, zur Anpassung und Konvertierung bestehender Modelle und zur Materialklassifizierung genutzt.

2) **Terra Vista:** Aus Elevationsdaten, Vektordaten, Satellitenbildern und den zuvor erstellten 3D-Modellen wird eine georeferenzierte virtuelle Umgebung nachgebildet.

3) **Stage:** Der Editor dient zur Erstellung virtueller Szenarien und Missionen. Mit Hilfe der Ground Population Funktion kann eine Vielzahl von Fahrzeugen oder Objekten eingebunden werden, wobei eine hinterlegte KI anschließend deren Bewegung und die logischen Abläufe steuert.

4) **Vega Prime:** Dieses Programm wird zum Rendern und zur Visualisierung der in der virtuellen Umgebung platzierten elektro-optischen Kamerasicht verwendet und beinhaltet die eigentliche Sensorsimulation. Es liefert außerdem die grafische Benutzeroberfläche und die API (Programmierschnittstelle, engl.: *Application Programming Interface*) zur Steuerung der Simulationsparameter.

5) **Ondulus IR:** Parallel zu *Vega Prime* generiert *Ondulus IR* die Infrarotsimulation der virtuellen Kamerasicht. Das dabei verwendete physikalisch basierte Sensormodell nutzt unter anderem die Materialklassifizierung der Objekte und des Terrains.

4.3 Auswahl des Testalgorithmus

In dem in Kapitel 3.2 beschriebenen Konzept ist zur Ursuchung der Leistungsdifferenzen zwischen realen und simulierten Sensordaten und zur Analyse der dabei relevanten Einflussfaktoren ein entsprechender Testalgorithmus als Basis nötig. Dieser sollte aus dem Bereich der Bild- oder Sensordatenverarbeitung stammen und die Eigenschaften gängiger Algorithmen aus dem aktuellen Stand der Technik möglichst gut repräsentieren. In Abschnitt 2.1.4 wurde begründet, warum in der vorliegenden Arbeit die UAV basierte Fahrzeugdetektion als Anwendungsfall dient. Im Folgenden wird nunmehr vorgestellt, welche Algorithmen für diesen Anwendungsfall in Frage kommen und warum das Objektdetektornetzwerk YOLOv3 für die durchgeführten Untersuchungen als Testalgorithmus ausgewählt wurde. Dies erfolgt auf Basis der bereits in [51] vorgestellten Recherchen.

4.3.1 Algorithmen für die UAV basierte Fahrzeugdetektion

Unabhängig von der zu detektierenden Objektklasse und der späteren Anwendung kann man alle bisher entwickelten Detektoralgorithmen auf oberster Ebene in zwei Gruppen einteilen (s. Abb. 8).

Nicht trainierbare Algorithmen

Die erste und auch ältere Gruppe kommt ohne Trainingsdaten aus und enthält Algorithmen, die bestimmte Bildeigenschaften zur Extrahierung einfacher Merkmale verwenden. Zu diesen Merkmalen zählen z.B. Farbunterschiede, Kanten im Bild und geometrische Bedingungen, aber auch Kontextinformationen über die zu detektierenden Objekte.

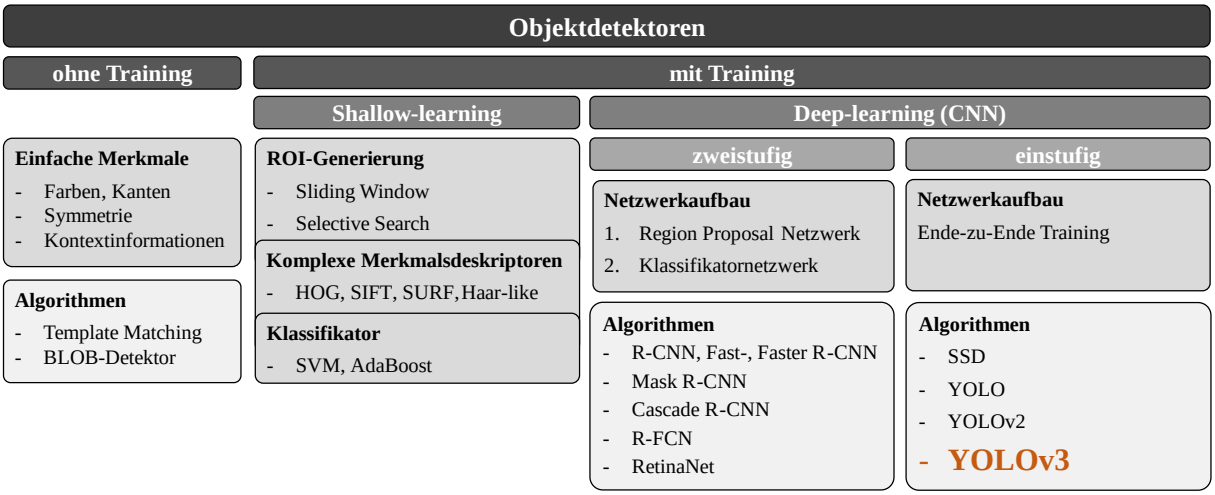


Abb. 8 Schematische Eingruppierung gängiger Algorithmen zur Objektdetektion anhand der zugrunde liegenden Konzepte.
BLOB: *Binary Large Objects*; HOG: *Histogram of Oriented Gradients*; SIFT: *Scale Invariant Feature Transform*; SURF: *Speeded-Up Robust Feature*; SVM: *Support Vector Machine*; CNN: *Convolutional Neural Network*; R-FCN: *Region-based Fully Convolutional Network*; SSD: *Single Shot Detector*; YOLO: *You Only Look Once*

Ursprüngliche Einsatzgebiete waren hauptsächlich Verkehrsüberwachung und Verkehrszählung. Cheng et al. [25] stellten schon 2009 eine Methode zur Erkennung und Zählung dynamischer Fahrzeuge aus der UAV-Perspektive vor und nutzten dabei sowohl Techniken zur Beseitigung als auch Erkennung des Hintergrunds. In [26] wurde eine median-basierte Hintergrundsubtraktion angewandt, um schließlich auf dem Vordergrund mit Hilfe einer *Binary Large Object* (BLOB) Analyse mögliche Fahrzeugpositionen identifizieren und durch Tracking deren Trajektorien ermitteln zu können. Beide Methoden sind jedoch auf sich bewegende Objekte limitiert.

Zheng et al. [27] versuchten auf hochaufgelösten Luftbildern von Autobahnen mit Hilfe einer Kombination morphologischer Operationen mögliche Fahrzeuge zu identifizieren und nutzten dabei als

Kontextinformation eine Karte des Straßennetzes. Choi et al. [138] stellten ebenfalls für hochaufgelöste Satellitenbilder ein mehrstufiges Detektionsverfahren vor, das mit Hilfe einer Mean-Shift Segmentierung und anschließender Fusion mit geometrischen Bedingungen mögliche zu untersuchende Bereiche (engl.: *Regions of Interest* (ROI)) identifiziert und deren Ähnlichkeit zu Fahrzeugen mit einer Beschreibung deren Form in Polarkoordinaten misst. Hinz et al. [139] nutzten einen BLOB-Detektor zur Erkennung von Fahrzeugen auf langwelligen Infrarotaufnahmen aus der Luft, wobei die Methode voraussetzt, dass sich die Fahrzeuge durch Parkbuchten oder Fahrspuren in einer bestimmten Anordnung befinden. Somit ist eine Detektion einzelner Fahrzeuge nicht möglich. In [140] liegt der Fokus primär auf der Extraktion des Straßennetzes, bevor anschließend durch Ausnutzung geometrischer Zusammenhänge die darauf befindlichen Fahrzeuge segmentiert und detektiert werden.

Tab. 9 Sammlung verschiedener Veröffentlichungen zur Fahrzeugdetektion auf Luftbildern und Eingruppierung der dabei verwendeten Algorithmen und Methoden. Diese Übersicht soll beispielhaft die Entwicklung in diesem Bereich zeigen und erhebt nicht den Anspruch auf Vollständigkeit.
Train.: mit Trainingsdaten; Sim.: mit synthetischen Sensordaten, nT: nicht trainierbarer Algorithmus; BLOB: *Binary Large Objects*; HOG: *Histogram of Oriented Gradients*; SVM: *Support Vector Machine*; SIFT: *Scale-Invariant Feature Transform*; RPN: *Region Proposal Netzwerk*; 1S DL: einstufiges *deep-learning* Netzwerk, 2S DL: zweistufiges *deep-learning* Netzwerk; FPN: *Feature Pyramid Network*; VEDAI: *Vehicle Detection in Aerial Imagery* Datenset; COWC: *Cars Overhead with Context* Datenset; DOTA: *Dataset for Object Detection in Aerial Images*

	Algorithmus	**Verwendung**	**Datenbasis**	**Art**	**Train.**	**Sim.**	**Jahr**
[25]	Hintergrundsubtraktion	Verkehrsanalyse	UAV-Video	nT	✗	✗	2009
[26]	Hintergrundsubtraktion + BLOB + Tracking	Verkehrsanalyse	Luftbild	nT	✗	✗	2014
[27]	Morpholog. Op. + Kontext (Straßenverlauf)	Verkehrsanalyse	Luftbild	nT	✗	✗	2012
[138]	Segmentierung + BLOB + Geom. + Formerkennung	Verkehrsanalyse	Satellitenbild	nT	✗	✗	2009
[139]	BLOB + Kontext (Anordnung)	Fahrzeugdetektion	Infrarotbild	nT	✗	✗	2006
[140]	Straßenerkennung + geometrische Bedingungen	Verkehrsanalyse	Luftbild	nT	✗	✗	2006
[141]	Viola-Jones / HOG + SVM, Kontext (Straßenorient.)	Fahrzeugdetektion	UAV	Shallow	✓	✗	2016
[142]	Asphaltsegmentierung + SIFT + SVM	Fahrzeugdetektion	UAV	Shallow	✓	✗	2014
[143]	Asphaltsegmentierung + HOG + Ähnlichkeitsmessung	Fahrzeugdetektion	UAV	Shallow	✓	✗	2014
[144]	8 versch. RPN + Fast R-CNN/ Faster R-CNN (VEDAI, DLR3K Datenset)	Fahrzeugdetektion	Satellitenbild	2S DL	✓	✗	2017
[114]	Faster R-CNN	Verkehrsanalyse	UAV	2S DL	✓	✗	2017
[145]	FPN + Cascade R-CNN (VisDrone2019)	Zählung+Tracking	UAV	2S DL	✓	✗	2021
[23]	R³ Net: orientierte *Bounding Boxen* (VEDAI, DLR3K)	Fahrzeugdetektion	Satellitenbild	2S DL	✓	✗	2019
[146]	Mask R-CNN für mehrere Objektklassen	Objektdetektion	Simulation	2S DL	✓	✓	2019
[147]	Mask R-CNN + orient. *Bounding Boxen* + Tracking	Verkehrsanalyse	UAV	2S DL	✓	✗	2020
[115]	YOLOv2 (Stanford Drone Datenset)	Fahrzeugdetektion	UAV	1S DL	✓	✗	2017
[117]	YOLOv3 (VEDAI, COWC, DOTA)	Fahrzeugdetektion	Satellitenbild	1S DL	✓	✗	2018
[118]	YOLOv2 (VEDAI Datenset)	Fahrzeugdetektion	Satellitenbild	1S DL	✓	✗	2019
[13]	Vergleich Faster R-CNN / YOLOv3	Fahrzeugdetektion	UAV	1S DL	✓	✗	2019

Wie diese Beispiele zeigen, sind die meisten der nicht trainierbaren Algorithmen und Methoden auf bestimmte Anwendungsszenarien oder Randbedingungen gebunden und funktionieren nur unter vorgegebenen und definierten Umgebungszuständen, was die generelle Anwendbarkeit stark einschränkt. Darüber hinaus sind sie nur in den wenigsten Fällen robust gegenüber Rotationen und perspektivischen Verzerrungen.

Shallow-learning basierte Algorithmen

Um diese Nachteile zu umgehen und robustere Modelle zu erhalten, verwendet die zweite große Gruppe von Algorithmen Verfahren des überwachten Lernens und benötigt daher Trainingsdaten, um die Erkennung und Klassifikation relevanter Merkmale zu lernen (s. Abb. 8). *Shallow-learning* basierte Algorithmen bilden dabei eine Untergruppe, wobei die Objektdetektion aus drei Schritten besteht. Im ersten

Schritt werden aus dem Eingangsbild je nach Methode willkürlich oder gezielt eine Vielzahl an ROIs generiert, die einen möglichst passenden Ausschnitt um das Objekt liefern sollten. Anschließend wird mit Hilfe eigens entwickelter komplexer Merkmalsdeskriptoren, wie z.B. HOG (engl.: *Histogram of Oriented Gradients*) oder SIFT (engl.: *Scale-Invariant Feature Transform*) die in diesen Ausschnitten enthaltene Information extrahiert und beschrieben. Auf Basis derer kann nun im letzten Schritt ein mit Hilfe der Trainingsdaten angelernter Klassifikator (z.B. SVM, engl. *Support Vector Machine*) unterscheiden, ob der jeweilige Ausschnitt ein Objekt enthält und wenn ja, zu welcher Klasse dieses gehört.

Xu et al. [141] stellten einen hybriden Ansatz durch Kombination von Viola-Jones Merkmalen und einem linearen SVM Klassifikator mit HOG Merkmalen vor, der zur Fahrzeugdetektion auf UAV Luftbildern dient. Durch Transformation der Straßenabschnitte in eine vorher definierte Ausrichtung wurde das Problem der Richtungsabhängigkeit der verwendeten Deskriptoren umgangen und somit auch eine Anwendung auf sich bewegenden UAVs möglich. Moranduzzo et al. [142] verwendeten für einen ähnlichen Anwendungsfall SIFT Merkmale in Kombination mit einem SVM Klassifikator, wobei die Methode im Vorfeld der Detektion asphaltierte Bereiche auswählt, um falsche Detektionen zu minimieren und die Berechnung zu beschleunigen. Zusätzlich veröffentlichten die Autoren einen weiteren Ansatz [143], der jeweils für die horizontale und vertikale Richtung HOG Merkmale berechnet und anschließend auf Basis eines Katalogs mit Referenzfahrzeugen eine Ähnlichkeitsmessung durchführt.

Deep-learning basierte Algorithmen

Eine weitere Untergruppe der trainierbaren Algorithmen sind die *deep-learning* basierten Methoden, denen im Bereich der Bildverarbeitung und Objektdetektion hauptsächlich sogenannte Faltungsnetze (CNN, engl. *Convolutional Neural Networks*) zugrunde liegen. In den ersten Schichten der CNNs werden ähnlich wie bei den *Shallow-learning* basierten Methoden möglichst aussagekräftige Merkmale aus den Eingangsdaten extrahiert. Je tiefer die Schichten, desto detaillierter sind die extrahierten Merkmale, bevor schließlich die Schichten im letzten Teil der Netzwerke die Klassifikation anhand der extrahierten Merkmale vornehmen. Den *deep-learning* basierten Netzen wurde vor allem in jüngster Vergangenheit viel Aufmerksamkeit gewidmet, was nicht zuletzt ihrer generellen Anwendbarkeit und ihrer hohen Leistungsfähigkeit geschuldet ist. Sie liefern bei der Objektdetektion im Allgemeinen bessere Ergebnisse als die bisher vorgestellten Algorithmen und sind unempfindlich gegenüber Skalierung, Rotation, perspektivischen Transformationen, sich verändernden Umweltbedingungen und Hintergründen. Diese Anpassungsfähigkeit erfordert jedoch eine große Menge an annotierten Trainingsdaten, deren Varianz und Zusammensetzung auf den jeweiligen Anwendungsfall abgestimmt sein muss. Ein weiterer Nachteil ist die schwere Interpretations- und Analysemöglichkeit dieser *Black-Box* Modelle.

Man unterscheidet in dieser Gruppe einstufige und zweistufige Detektoren. Letztere betrachten die Objektdetektion als Klassifikationsproblem, generieren im ersten Schritt mögliche ROIs mit einem Region Proposal Netzwerk (RPN), klassifizieren diese im zweiten Schritt und sind daher zwar genauer, aber auch langsamer. Die einstufigen Detektornetzwerke kombinieren beide Schritte, betrachten das Ganze als Regressionsproblem und erreichen dadurch mittlerweile echtzeitfähige Detektionsraten.

Sommer et al. [144] demonstrierten die Anwendbarkeit von Fast R-CNN [28] und Faster R-CNN [29] für die Fahrzeugdetektion auf Luftbildern aus dem DLR 3K [123] und VEDAI Datensatz [119]. Sie verglichen acht verschiedene RPN und zeigten, wie die Detektionsleistung für kleine Objekte durch Anpassung der Ankerboxen gesteigert werden kann. Xu et al. [114] verwendeten ebenfalls den Faster R-CNN Detektor und untersuchten in diesem Zusammenhang den Einfluss verschiedener Randbedingungen auf die Detektionsleistung. Es stellte sich heraus, dass der Detektor robust gegenüber Beleuchtungsschwankungen und Rotationen ist und dass die Detektionsgeschwindigkeit unabhängig von der Anzahl der vorkommenden Objekte ist. In [145] wird eine Kombination aus FPN und Cascade R-CNN sehr erfolgreich für Fahrzeugdetektion, -zählung und -tracking auf dem VisDrone2019 Datensatz

eingesetzt. Das dabei verwendete Cascade R-CNN wurde im Jahr 2021 vorgestellt [148], trainiert eine Sequenz von aufeinanderfolgenden Detektornetzwerken mit ansteigendem IoU-Schwellwert und verspricht dadurch eine geringere Anfälligkeit für Überanpassung und eine insgesamt höhere Detektionsgüte.

Li et al. [23] entwickelten ein zweistufiges Netz, das zusätzlich zur Position auch die Orientierung der Fahrzeuge bestimmt und somit orientierte *Bounding Boxen* liefert. Ein weiterer neuerer Ansatz ist die sogenannte Instanzen-Segmentierung, eine Kombination aus Objektdetektion und semantischer Segmentierung, die zusätzlich zur *Bounding Box* auch eine pixelgenaue Maske des detektierten Objekts liefert. Ein Vertreter davon ist neben dem modifizierten Cascade R-CNN auch das Mask R-CNN [65], das z.B. in [146] im militärischen Kontext für die Detektion verschiedener Objekte auf Luftbildern eingesetzt wird oder in [147] zur Verkehrsanalyse, wobei letztere die Maske wiederum zur Generierung orientierter Boundig Boxen nutzen. Da es in diesem Bereich bisher noch relativ wenig Benchmark- und Trainingsdatensätze gibt, sind beide Ansätze vorerst für die hier vorgestellten Untersuchungen nicht relevant.

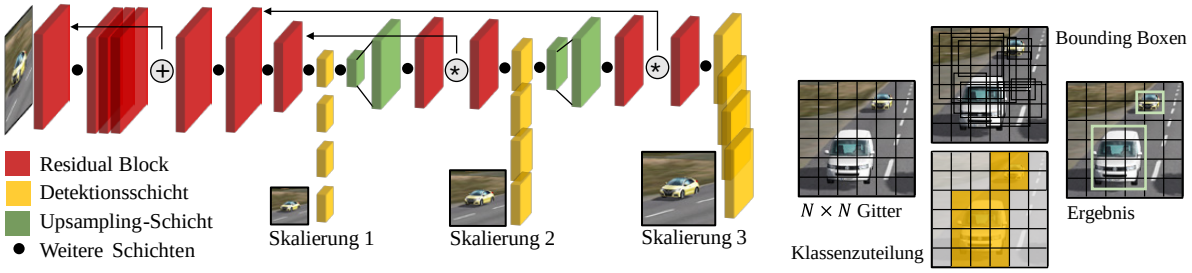
In [13, 115, 117, 118] werden schließlich verschiedene Versionen des einstufigen YOLO Detektors [30–32] für die UAV basierte Fahrzeugdetektion eingesetzt. Tang et al. [115] evaluierten YOLOv2 auf Luftbildern des Stanford Drone Datensatzes [125] und demonstrierten die Eignung des Detektors für die Echtzeit-Fahrzeugdetektion auf UAV Videodaten. Lu et al. [117] nutzten drei verschiedene Datensätze (VEDAI [119], COWC [121] und DOTA [122]) und zeigten, dass auch bei kleinen, rotierten Fahrzeugen und hohen Objektdichten die Erkennung funktioniert. Zu diesem Schluss kamen auch Lechgar et al. [118]. Interessant ist diesem Zusammenhang auch der Vergleich der ein- bzw. zweistufigen Detektoren Faster R-CNN und YOLOv3 von Benjdira et al. [13]. Es stellte sich heraus, dass YOLOv3 bei den Punkten Sensitivität und Verarbeitungsgeschwindigkeit vorne liegt und im betrachteten Anwendungsfall dennoch die gleiche *Precision* wie das Faster R-CNN Netzwerk erreicht. Bei Eingangsdaten mit einer Auflösung von 608 x 608 sind je nach verwendeter Hardware Verarbeitungszeiten von ca. 51 ms (entspricht fast 20 Hz) realistisch und ermöglichen daher z.B. die Echtzeitverarbeitung von UAV Videodaten. Das einstufige Netzwerkdesign erlaubt ein Ende-zu-Ende Training und in [30, 149] wird dem Netzwerk außerdem eine sehr hohe Generalisationsfähigkeit zwischen verschiedenen Domänen und Bildtypen bescheinigt.

Fazit

Insgesamt zeigte die vorgestellte Literaturrecherche, dass das YOLOv3 Netzwerk bereits in mehreren Veröffentlichungen erfolgreich für den auch hier betrachteten Anwendungsfall der UAV basierten Fahrzeugdetektion auf Luftbildern eingesetzt wurde und dabei mehrere Vorteile gegenüber anderen Detektoren aufweist. Es gehört außerdem zu der aktuell fortschrittlichsten und am weitesten verbreiteten Gruppe der trainierbaren *deep-learning* basierten Detektoren. Diese sind zudem weitestgehend *Black-Box* Modelle, bei denen die in dieser Arbeit vorgestellten Ansätze zur Identifikation relevanter Einflussfaktoren auf die Detektionsleistung den größten Nutzen versprechen. Der YOLOv3 Detektor wird daher in allen nachfolgend vorgestellten Untersuchungen als Testalgorithmus verwendet.

4.3.2 Funktionsweise und Aufbau des YOLOv3 Detektornetzwerkes

Zum besseren Verständnis der Ergebnisse und Auswertungsschritte wird im Folgenden kurz auf die Netzwerkarchitektur und die Funktionsweise des YOLOv3 Detektors eingegangen (vgl. [51]). YOLOv3 ist ein einstufiges CNN, das zur Gruppe der sogenannten *Single Shot Detektoren* (SSD) gehört und die Objektdetektion als Regressionsproblem betrachtet [143, 144, 149, 150].



a) Darknet-53 Netzwerkarchitektur (vgl. [151]) b) Funktionsweise YOLO (vgl. [30])

Abb. 9 Vereinfachte Darstellung der zugrunde liegenden Netzwerkarchitektur mit den drei verschiedenen Skalierungsebenen der Detektion in a) und der generellen Funktionsweise dieser Art von Netzwerken in b).

Das Eingangsbild wird bei dieser Gruppe von Detektoren im ersten Schritt in ein $N \times N$ Gitter aufgeteilt (s. Abb. 9 b)). Durch die einstufige Architektur wird das gesamte Bild anstatt speziell generierter ROIs betrachtet. Die zusätzliche Information im Kontext hilft dabei, falsche Detektionen zu reduzieren. Beim Training sind dabei die Zellen im $N \times N$ Gitter interessant, in denen sich das Zentrum einer *Ground Truth Bounding Box* befindet. Für diese Zellen wird relativ zur Gitterposition und Größe der Ankerboxen eine Regression in Bezug auf die Position und Dimension dieser *Ground Truth Bounding Box* vorhergesagt. Ankerboxen sind dabei eine Art Startpunkt, die häufig vorkommende Größen und Seitenverhältnisse der Objekte im Datenset repräsentieren. YOLOv3 verwendet neun Stück dieser Ankerboxen, die im Vorfeld des Trainings durch eine K-means Clusteranalyse aus dem Trainingsdatensatz berechnet werden. Vorteil dieses Verfahrens ist ein stabilisiertes Trainingsverhalten, da dadurch nicht eine zufällig geschätzte *Bounding Box* auf ein zu detektierendes Objekt angepasst, sondern lediglich die Abweichung einer bereits gezielt ausgewählten Startgröße optimiert werden muss. Bei der späteren Anwendung wird das Eingangsbild wiederum in $N \times N$ Zellen aufgeteilt und für jede eine Anzahl an S *Bounding Boxen* mit einem zugehörigen Zuverlässigkeitswert und einer Klassenzuteilung vorhergesagt. Durch Anwendung eines bestimmten Schwellwertes werden die unzuverlässigen Detektionen entfernt. In der Version 3 beruht die Klassenzuteilung auf einer Multi-Label Klassifikation, d.h. die Klassen sind nicht exklusiv, da für jede Klasse eine separate Zugehörigkeitswahrscheinlichkeit berechnet wird, deren Summe auch größer als Eins sein kann.

Der YOLOv3 Detektor verwendet das *Darknet-53* Netzwerk (s. Abb. 9 a)), das auf dem ImageNet Datensatz [152] vortrainiert wurde, zur Merkmalsextraktion aus den Eingangsdaten. Es folgt ebenfalls dem Trend der immer tiefer werdenden Netzwerkstrukturen und erreicht dadurch bessere Detektionsleistungen als die Vorgängerversionen. Durch die Vorhersage von nur einem Klassentyp pro Zelle wird eine gute räumliche Verteilung der Detektionen gefördert, jedoch auch eine Erkennung von kleinen und dicht verteilten Objekten erschwert. Um dies zu vermeiden, werden im *Darknet-53* Netzwerk an drei verschiedenen Positionen Merkmalsschichten mit verschiedenen Auflösungen betrachtet und diesen Schichten jeweils die von der Größe passenden Ankerboxen zugewiesen. Das Netzwerk reduziert dazu die Auflösung in diesen drei Schritten um jeweils Faktor 32, 16 und 8, was bedeutet, dass die Auflösung der Eingangsbilder auf ein Vielfaches von 32 skaliert werden muss. Gängige Auflösungen sind 320 x 320, 416 x 416 und 608 x 608, wobei eine höhere Pixelanzahl die Genauigkeit verbessert, aber auch die Verarbeitungsgeschwindigkeit reduziert. Die erste Skalierungsstufe ist für die Detektion größerer Objekte vorgesehen. Die Besonderheit ist nun, dass bei den zwei darauffolgenden Skalierungsstufen durch Verkettung auch Merkmale aus dem ersten Teil des Netzwerks verwendet werden (s. Abb. 9 a)). Dies bewirkt, dass aussagekräftige semantische Informationen und der Bildkontext nicht verloren gehen und in allen Stadien des Netzwerks mitberücksichtigt werden. Falls eine auf diese Weise vorhergesagte *Bounding Box* eine Überlappung von über 50 % zur *Ground Truth Bounding Box* aufweist, wird sie als korrekt eingestuft. Eine *Non-Maximum Suppression (NMS)* entfernt schließlich am Ende des

Netzwerks detektierte *Bounding Boxen* mit einer sehr hohen Überlappung und behält ausschließlich diejenige mit dem höchsten Zuverlässigkeitswert.

4.3.3 Metriken zum Leistungsvergleich der Objektdetektoren

Um die Leistungsfähigkeit neuronaler Netze bei der Objektdetektion objektiv beurteilen und verschiedene Algorithmen und Trainingskonfigurationen vergleichen zu können, werden entsprechende Metriken benötigt. Diese werden im Folgenden vorgestellt und im Rahmen der Arbeit auch verwendet, um die Leistungsdifferenzen zwischen realem und synthetischem Datenmaterial zu untersuchen. Grundlage für die hier betrachtete Form der Objektdetektion bilden rechteckförmige Markierungen, die sogenannten *Bounding Boxen (BB)*. Diese enthalten entweder die *Ground Truth*, d.h. die während der Annotation korrekt gelabelten Objekte, oder die vom Algorithmus gelieferten Detektionen, die zusätzlich zur Klassenbezeichnung häufig einen entsprechenden Zuverlässigkeitswert beinhalten. Die verwendeten Algorithmen und Evaluierungsmethoden berücksichtigen keine Orientierung der Objekte, weshalb im Folgenden standardmäßig nur horizontal ausgerichtete *Bounding Boxen* betrachtet werden.

Intersection over Union (IoU) ist ein gebräuchliches Maß für die Lokalisationsgenauigkeit und beschreibt das Überlappungsverhältnis zwischen einer detektierten *Bounding Box* BB_d und einer *Ground Truth Bounding Box* BB_{gt}. Überschreitet dieses einen bestimmten Schwellwert, so wird die Detektion als korrekt angesehen und als *True Positive* (TP) bezeichnet.

$$\text{IoU} = \frac{\text{Fläche}(BB_p \cap BB_{gt})}{\text{Fläche}(BB_p \cup BB_{gt})} = \quad (1)$$

Der gewählte Schwellwert ist je nach Anwendungsfall verschieden, sollte jedoch nicht zu hoch gewählt werden, um auch Ungenauigkeiten beim händischen *Labeling*-Prozess zu berücksichtigten [153]. In [154] wurde sogar festgestellt, dass es für menschliche Betrachter schwierig ist, *Bounding Boxen* mit einer IoU-Schwelle von 0,3 von denen mit einer Schwelle von 0,5 zu unterscheiden. In Abb. 10 werden verschiedene IoU-Werte zur Veranschaulichung visualisiert.

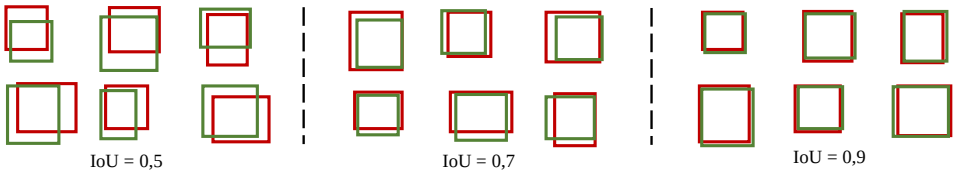


Abb. 10 Visualisierung verschiedener IoU-Werte anhand einer detektierten und einer *Ground Truth Bounding Box*. vgl. [155]

Liegt der Überlappungsgrad einer Detektion mit einer *Ground Truth Bounding Box* unter der gewählten Schwelle bzw. ist gar keine Überlappung vorhanden, dann handelt es sich um eine inkorrekte Detektion, die als *False Positive* (FP) bezeichnet wird. Ein vorkommendes, aber nicht detektiertes Objekt wird als *False Negative* (FN) definiert. *True Negatives* (TN) würden Bildausschnitte beschreiben, die keine Objekte enthalten und vom binären Klassifikator auch dieser Gruppe zugeordnet werden. Dieser Fall kommt bei den hier betrachteten *End-to-end* trainierten Algorithmen jedoch nicht vor, da diese als Eingang das gesamte Bild betrachten und ausschließlich die Position der Detektionen zurückliefern. Diese Zusammenhänge bilden die Grundlage für die eigentlichen Metriken zur Evaluierung.

Precision misst die Fähigkeit des Detektors, ausschließlich tatsächlich vorkommende Objekte zu detektieren und beschreibt den Anteil der korrekten Detektionen an allen gemachten Detektionen.

$$\text{Precision } p = \frac{\text{TP}}{\text{TP+FP}} \quad (2)$$

Recall misst die Fähigkeit des Detektors alle vorkommenden *Ground Truth* Objekte zu detektieren und beschreibt den Anteil korrekter Detektionen gegenüber allen *Ground Truth Bounding Boxen*.

$$\text{Recall}\, r = \frac{\text{TP}}{\text{TP+FN}} \tag{3}$$

***F*-Score:** Sowohl *Precision* als auch *Recall* können Werte zwischen 0 und 1 aufweisen und sind als Auswertemetriken für komplette Testdatensätze geeignet. Da beide gegensätzliche Eigenschaften des Detektors beurteilen und man häufig an einem Kompromiss zwischen beiden interessiert ist, wurde der F-Score definiert:

$$F_\beta = (1+\beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{(\beta^2 \cdot \text{Precision}) + \text{Recall}}\,, \qquad \beta\text{: Gewichtungsfaktor} \tag{4}$$

In den meisten Fällen wird dabei der Gewichtungsfaktor $\beta = 1$ gewählt. Der daraus resultierende F_1-Score beschreibt somit das harmonische Mittel aus *Precision* und *Recall*.

Average Precision (AP): Nahezu alle gängigen *deep-learning* basierten Objektdetektoren liefern neben der Position des detektierten Objektes auch einen Zuverlässigkeitswert für die gemachte Detektion. Dieser wurde bei keiner der bisherigen Metriken berücksichtigt, spielt allerdings für eine gründliche Evaluierung und einen fundierten Vergleich verschiedener Algorithmen ebenfalls eine Rolle. Daher wird im Folgenden für die Auswertung der Detektionsleistung die *Average Precision* (AP) Metrik herangezogen, die seit 2010 unter anderem auch bei der Pascal VOC Challenge zum Einsatz kommt [153].

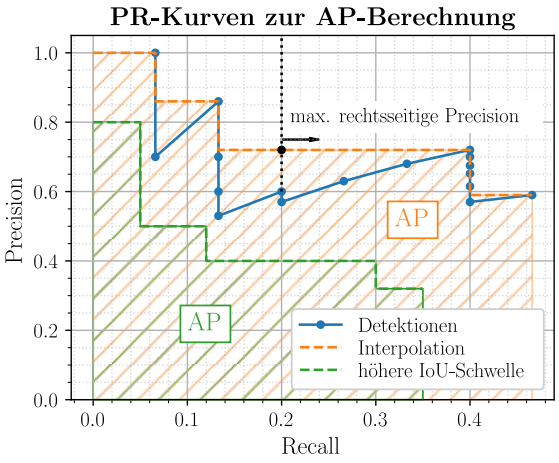


Abb. 11 Plot fiktiver PR-Kurven zur Veranschaulichung der Berechnung des AP Wertes für verschiedenen IoU-Schwellwerte.

Die AP entspricht der Fläche unter der interpolierten *Precision-Recall* (PR)-Kurve für verschiedene Zuverlässigkeitswerte und einen spezifischen IoU-Schwellwert. Die PR-Kurve visualisiert dabei das Verhältnis zwischen *Precision* und *Recall* für die verschiedenen Zuverlässigkeitswerte und bietet daher ein geeignetes Maß für die generelle Leistungsfähigkeit. Wird ein *Ground Truth* Objekt durch mehrere *Bounding Boxen* detektiert, die nicht durch eine im Detektor integrierte *Non-maximum Suppression* zusammengefasst wurden, so wird lediglich die erste als TP betrachtet und die folgenden als FP. Für die Berechnung der AP werden alle Detektionen des Testdatensets nach absteigenden Zuverlässigkeitswerten geordnet. Für jeden Eintrag in der so generierten Liste wird anschließend der aktuelle *Precision* und *Recall* Wert berechnet, wobei für die Berechnung nur Detektionen mit einem höheren Zuverlässigkeitswert als dem aktuell betrachteten berücksichtigt werden. Die berechneten Werte bilden die PR-Kurve. Um diese zu glätten, wird jeder vorkommende *Precision*-Wert durch den maximalen *Precision*-Wert ersetzt, der rechts von ihm, also bei höheren *Recall*-Werten, auftritt. Dies führt zu einer monoton abfallenden, interpolierten PR-Kurve. Dieser Vorgang ist in Abb. 11 schematisch dargestellt. Die zugehörige mathematische Berechnungsvorschrift lautet folgendermaßen:

$$\text{AP} = \int_0^1 p_{\text{interp}}(r)\, dr\,, \qquad p_{\text{interp}}(r) = \max_{\tilde{r} \geq r} p(\tilde{r}) \tag{5}$$

Die AP reduziert somit die interpolierte PR-Kurve $p_{\text{interp}}(r)$ auf einen einzigen Wert, der für den Vergleich mehrerer Detektoren oder für das Überwachen des Lernprozesses verwendet werden kann. Dieser

liegt im Bereich zwischen 0 und 1, wobei höhere Werte für eine höhere Leistung stehen. Ein Detektor weist eine hohe Leistungsfähigkeit auf, wenn die *Precision* auch bei steigenden *Recall* Werten hoch bleibt. Werden bei der Detektion mehrere Objektklassen betrachtet, so wird für jede Klasse eine separate AP berechnet und anschließend durch Mittelung die *mean Average Precision* (mAP) als Maßzahl ermittelt.

Receiver Operating Characteristic (ROC): Der Vollständigkeit halber sei an dieser Stelle auch die ROC-Kurve erwähnt. Diese entsteht ebenfalls durch Variierung des Zuverlässigkeitswertes, wobei hier der *Recall*-Wert über die Ausfallrate aufgetragen wird.

$$\text{Ausfallrate} = \frac{\text{FP}}{\text{TN+FP}} \tag{6}$$

Dies ergibt eine monoton ansteigende Kurve. Auch hier wird wieder die Fläche unterhalb der Kurve (engl.: *Area Under the Curve*, AUC) als Bewertungseinheit herangezogen. Bei dieser Metrik wird keine *Precision* berücksichtigt, sie kann jedoch verwendet werden, um in Bezug auf die Schwelle des Zuverlässigkeitswertes einen bestmöglichen Operationspunkt zu ermitteln [20]. Da bei der Berechnung der Ausfallrate *True Negatives* benötigt werden, die wie bereits erwähnt bei den hier betrachteten Ende-zu-Ende trainierten Detektoren nicht definiert sind, ist diese Metrik für die weitere Auswertung nicht anwendbar.

4.4 Bildbeschreiber

Verschiedene Methoden und Arten von Bildbeschreibern spielen ähnlich wie in [36], aber in einer ausgeprägteren und direkteren Form, eine zentrale Rolle im hier vorgestellten Konzept zur Identifikation relevanter Bildeigenschaften bei der Anwendung von Detektionsalgorithmen. Sie repräsentieren bestimmte Bildeigenschaften und sind bei Verwendung definierter Ähnlichkeitsmaße auch geeignet, um Bildpaare in Bezug auf die jeweils beschriebene Bildeigenschaft zu vergleichen. Der Grundgedanke ist, dass allein auf Basis des zur Verfügung stehenden Bildmaterials eine Klassifikation zwischen realen und synthetischen Daten und zwischen TP/FP/FN Detektionen vorgenommen werden soll. Die Bildbeschreiber dienen dabei als Merkmale für den verwendeten Klassifikationsalgorithmus und werden – vorausgesetzt die Güte der Klassifikation ist entsprechend hoch – bei der anschließenden Analyse verwendet, um einen Zusammenhang zwischen den für die Klassifikation wichtigen Merkmalen und zwischen den damit verbundenen relevanten Bildeigenschaften herzustellen.

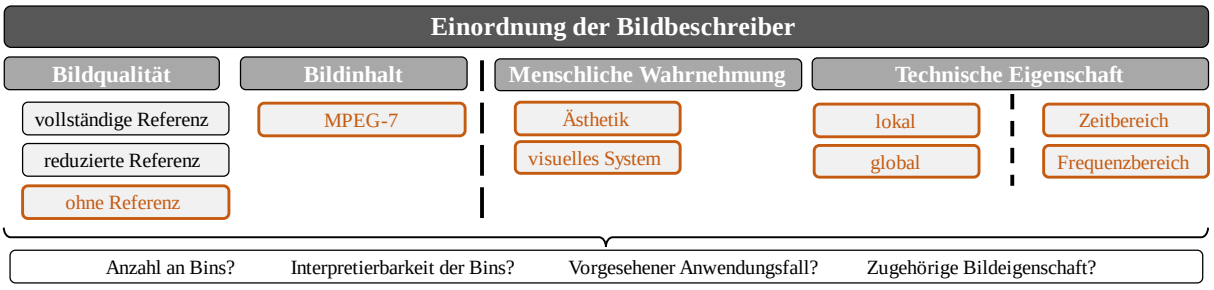


Abb. 12 Übersicht und Kategorisierung gängiger Bildbeschreiber und Auflistung einiger zusätzlicher Eigenschaften in der letzten Zeile, die für die spätere Anwendung ebenfalls von Bedeutung sind. Orange markierte Einträge werden in den im Rahmen dieser Arbeit durchgeführten Analysen verwendet.

In [156] und [157] werden allgemeine Anforderungen aufgelistet, die ein effizienter und universell anwendbarer Bildbeschreiber aufweisen sollte. Der Deskriptor sollte demnach eine surjektive Abbildung von Medienobjekten auf Punkte im Merkmalsraum liefern und hochrangig diskriminativ für die jeweilige Bildeigenschaft sein. Es wird erwartet, dass unterschiedliche Bilder zu unterschiedlichen Werten des Bildbeschreibers führen und entsprechend unterschieden werden können. Des Weiteren ist es wichtig, dass der Extraktionsprozess robust gegenüber verschiedenen Qualitätsstufen und unwichtigen

Bilddetails ist und gängige Bildtransformationen wie z.B. Rotation oder Translation nicht zu einer Verzerrung der extrahierten Werte führen. Zusätzliche positive Eigenschaften sind eine kurze Deskriptorlänge, Unempfindlichkeit gegenüber Rauschen, geringe Rechenkomplexität und Skalierbarkeit der Deskriptorgröße.

Eingruppierung der Bildbeschreiber

Die in der Literatur beschriebenen Bildbeschreiber können anhand verschiedener Kriterien gruppiert werden. In Abb. 12 ist diese Kategorisierung schematisch dargestellt. Wichtig für den im hier vorgestellten Konzept betrachteten Anwendungsfall ist eine möglichst breit gefächerte Erfassung sämtlicher Bildeigenschaften, die einen Einfluss auf die Unterschiede zwischen realen und synthetischen Daten haben und zu einem Leistungsunterschied führen könnten. Es wurde daher darauf geachtet, dass bei der Auswahl passender Bildbeschreiber alle Gruppierungen abgedeckt werden.

Auf oberster Ebene wird häufig zwischen Metriken unterschieden, die hauptsächlich die Bildqualität beurteilen und solchen, die eher den Bildinhalt beschreiben. Erstere sind insensitiv gegenüber der dargestellten Szenerie, basieren vorrangig auf der Gesamtheit der Pixelwerte und bewerten häufig anhand der menschlichen Wahrnehmung. Ziel der qualitätsbasierten Metriken ist dabei vorrangig die Beschreibung von Störeffekten, wie Rauschen, Unschärfe, Farbverzerrungen oder Kompressionsartefakten, oder die Beurteilung der ästhetischen Bildqualität. In [158] werden dabei drei Gruppen unterschieden: Metriken mit vollständiger Referenz, reduzierter Referenz und ohne Referenz. Für unsere Betrachtungen spielen ausschließlich Methoden ohne Referenz eine Rolle, da nur diese ohne ein verzerrungsfreies Referenzbild berechnet werden können. Die Schwierigkeit dabei ist die Entwicklung passender Berechnungsvorschriften, die trotz fehlender Referenz in der Lage sind, Störeffekte zu detektieren und deren Stärke zu beurteilen. Im Gegensatz zu [36] werden diese dennoch als sinnvoll für die vorliegende Auswertung betrachtet, da im hier vorgestellten Konzept nicht nur die Betrachtung der Differenzen zwischen einzelnen Bildpaaren eine Rolle spielt sondern auch die Beschreibung der Eigenschaften kompletter Trainings- und Testdatensätze und die Identifikation von Einflussfaktoren auf die Detektionsleistung. Qualitätsfaktoren wie z.B. Rauschen können durchaus Einfluss auf einzelne Detektionsergebnisse haben und sollten daher mitberücksichtigt werden.

Die zweite große Gruppe der Bildbeschreiber betrachtet Bildinhalt und -zusammensetzung. Neben Methoden der semantischen Segmentierung zur Analyse der Zusammensetzung der Szenerie kommen dabei hauptsächlich die im MPEG-7 Standard [159–161] beschriebenen Metriken zum Einsatz. Diese beschreiben spezielle Bildeigenschaften und berücksichtigen dabei sowohl lokale als auch globale Berechnungsvorschriften. Bildinhalt beschreibt in diesem Fall jedoch nicht ausschließlich das dargestellte Motiv sondern vielmehr die Form der Darstellung. In [36] wurden die Deskriptoren des MPEG-7 Standards bereits erfolgreich für die Analyse der Differenzen zwischen realen und synthetischen Bildpaaren eingesetzt. Auf weitere Anwendungsfälle und verwendete Methoden wird später noch detaillierter eingegangen.

Auf Basis dieser Aufteilung wird weiterhin zwischen Metriken unterschieden, die die menschliche Wahrnehmung berücksichtigen und welchen, die auf technischen Eigenschaften beruhen. Menschliche Wahrnehmung kann sich in diesem Zusammenhang entweder auf ästhetische Gesichtspunkte beziehen oder auch auf das menschliche visuelle System. Technische Eigenschaften werden zudem darüber unterschieden, ob sie lokal anhand von Bildausschnitten berechnet werden oder global über das gesamte Bild. Die Berechnung kann dabei je nach Metrik im Zeit- oder im Frequenzbereich stattfinden.

Tab. 10 Übersichtstabelle über die verwendeten Bildbeschreiber und deren Anzahl und Position der Merkmale, die diese zur Datenmatrix beitragen. Die restlichen Spalten beinhalten die repräsentierte Bildeigenschaft, den typischen Anwendungsfall und die folgendermaßen abgekürzten Deskriptoreigenschaften:
G/L: global/lokal; Wahrnehmung: basierend auf menschlicher Wahrnehmung; I/Q: inhaltsbasiert/qualitätsbasiert; Interpret.: Interpretierbare Werte; T/A: technisch/ästhetisch; Vertl..: Verteilung; Hist.: Histogramm; GLCM: *Grey*

Level Co-Occurrence Matrix; Sem.: semantisch; NIMA: *Neural Image Assessment*; BRISQUE: *Blind/Referenceless Image Spatial Quality Evaluator*

Gruppe	Name/ Position	Größe	G/L	Bildeigenschaft	Wahrnehmung	I/Q	Anwendung	Interpret.	T/A
MPEG Farbe	CLD	22	L	Dominante Farbe	○	I	Sketches	✗	T
	CSD	32	L	Hist. Farbverteilung	✓	I	Form, Fotografien	✗	T
	DCD	33	G	Dominante Farbe	○	I	Logos, Flaggen	✓	T
	SCD	32	G	Hist. Farbverteilung	○	I	Bildsuche (Farbe)	✗	T
MPEG Textur	EHD	80	L	Räumliche Vertl. Ecken	○	I	Cliparts, Sketches	✓	T
	HTD	32	G	Räumliche Frequenzen	✓	I	Satellitenbilder	○	T
MPEG Form	RSD	35	G	Zernike Momente	○	I	Formen	✗	T
BLC	0-5	1	G	Helligkeit	✓	I	HSL, RGB, HSP	✓	T
	6-16	11	G	Kontrast, Gamma [162]	○	Q	Kontrastverstärkung	✓	T
	17	1	G	Helligkeit [163]	✓	Q	Fotobewertung	✓	A
	18	1	G	Kontrast, Hist. [163]	✓	Q	Fotobewertung	✓	A
	19	1	G	Beleuchtung [164]	✓	Q	Fotobewertung	✓	A
Col	0	1	G	Farbigkeit [165]	✓	Q	Qualität der Farben	✓	A
	1-2	2	G	Farbstich [166]	✓	Q	Detektion Farbstich	✓	A
	3-4	2	G	Farbtemperatur [167, 168]	○	Q	Bildverarbeitung	✓	T
	5	1	G	Anzahl an Farben [163]	✓	Q	Fotobewertung	✓	A
IQM	0-1	2*1	G	NIMA [169]	✓	Q	Fotobewertung	✗	A/T
	2	1	G	BRISQUE [170]	✓	Q	Image Quality	✗	A/T
	3-27	25	L	Drittel-Regel [171]	✓	Q	Fotobewertung	✓	A
	28-29	2*1	G	Tiefenschärfe [164]	✓	Q	Fotobewertung	✓	A
DBN	0-1	1	G	Schärfe [172, 173]	✓	Q	Detektion Unschärfe	✓	T
	2-5	4*1	G	Unschärfe [163, 174, 175]	✓	Q	Detektion Unschärfe	✓	T
	6-8	2, 1	L	Rauschen [176, 177]	○	Q	Detektion Rauschen	✓	T
Sha	0-2	3	G	Breite; Höhe; Fläche	✗	I	Bildgröße	✓	T
	3-11	7+2	L	Form Vordergrund [178]	✗	I	Segmentierung	○	T
	12-19	7+1	L	Form Objekt [179]	✗	I	Segmentierung	○	T
	20-59	8*5	L	Form Szenerie [179]	✗	I	Sem. Segmentierung	○	T
Env	0	1	G	Schattenkarte [180]	○	I	Schattendetektion	✓	T
	1-6	6	G	Wetter [181]	○	I	Wetterklassifikation	✓	T
ET	0-53	54	G	GLCM [182]	○	I	Texturbeschreibung	✓	T
	54	1	L	Räumliche Kantenvertl. [163]	✓	Q	Fotobewertung	✓	A
	55	1	G	Anzahl Kanten	○	I	Kantendetektion	✓	T
	56-57	2*1	G	Glattheit [164]	✓	Q	Fotobewertung	✓	A

Tab. 10 gibt einen Überblick über die für die hier vorgestellten Untersuchungen ausgewählten Bildbeschreiber und deren Eigenschaften. Bei der Auswahl wurde sowohl die Anzahl der Bins, der ursprünglich vorgesehene Anwendungsfall und nicht zuletzt die Interpretierbarkeit einzelner Bins berücksichtigt. Es wurde versucht, alle in Frage kommenden Kategorien aus Abb. 12 zu berücksichtigen und die ausgewählten Metriken speziellen Gruppen von Bildeigenschaften zuzuweisen: Farbe, Textur, Form, Helligkeit/Luminanz/Kontrast, Kantenverteilung, Verzerrung/Unschärfe/Rauschen, Bildqualität und Umweltbedingungen. Ziel ist die Generierung von Richtlinien für das Design synthetischer Simulationsumgebungen durch Verbesserung derjenigen Bildeigenschaften, die zum einen für Bildunterschiede zwischen realen und synthetischen Daten und zum anderen für falsche Detektionen verantwortlich sind und daher beim Design synthetischer Trainingsdaten zukünftig stärker berücksichtigt werden müssen. Im Folgenden wird auf die einzelnen Gruppen aus Tab. 10 und die darin enthaltenen Deskriptoren näher eingegangen.

4.4.1 MPEG-7

Der MPEG-7 Standard [159–161] wurde 2002 verabschiedet und wird oft auch als *Multimedia Content Description Interface* bezeichnet. Er dient der Ausstattung von Multimediainhalten mit Metainformationen und enthält ein standardisiertes Set an Deskriptoren für die Bereiche Farbe, Texturen, Form, Bewegung, Audio und die Beschreibung menschlicher Gesichter. Die ausschließlich durch visuelle Kriterien generierten Metadaten liefern eine nicht textbasierte visuelle Beschreibung des Inhalts, um Bilder und Videos effizient filtern, identifizieren, kategorisieren und suchen zu können.

Anwendungsgebiete sind die inhaltsbasierte Bildabfrage (engl.: CBIR, *Content-Based Image Retrieval*) [183], Videoanalyse und *Keyframe*-Extraktion [184]. In [185] wurden die Deskriptoren auf das ImageNet Datenset angewandt. Manjunath et al. [186] beschrieben die Gruppe der Farb- und Texturdeskriptoren und empfahlen die Bildabfrage als Messmethode für die Effektivität der Deskriptoren. In [187] wurde eine semi-globale und eine globale Erweiterung der eigentlich lokalen Berechnungsvorschrift des Edge Histogram Deskriptors vorgestellt, die durch eine vollständigere Beschreibung zu einer verbesserten Leistung bei der Bildabfrage führt.

Im Folgenden werden die in dieser Arbeit verwendeten MPEG-Deskriptoren mit ihren zugehörigen Eigenschaften näher beschrieben (vgl. [188], [189]). In [190] ist eine dem Standard beigefügte Referenzsoftware mit einer C Implementierung der Deskriptoren enthalten. Um eine objektorientierte Architektur in C++ nutzen zu können, wurde für die vorliegenden Berechnungen die davon abgeleitete mpeg7FexLib Bibliothek [191] verwendet. Diese beinhaltet darüber hinaus Implementierungen der zugehörigen Ähnlichkeitsmaße und ist kompatibel mit den Datentypen der OpenCV Bibliothek.

(a) Farb-Deskriptoren: Diese sind sehr robust gegenüber Veränderungen im Hintergrund, unabhängig von der Bildgröße und Bildorientierung und spielen auch bei der menschlichen Wahrnehmung eine herausragende Rolle [160]. Sie beruhen entweder auf einem Histogramm der Farbverteilung oder beschreiben die dominanten Farben im Bild, wobei lokale und globale Deskriptoren vorhanden sind und verschiedene Farbräume berücksichtigt werden.

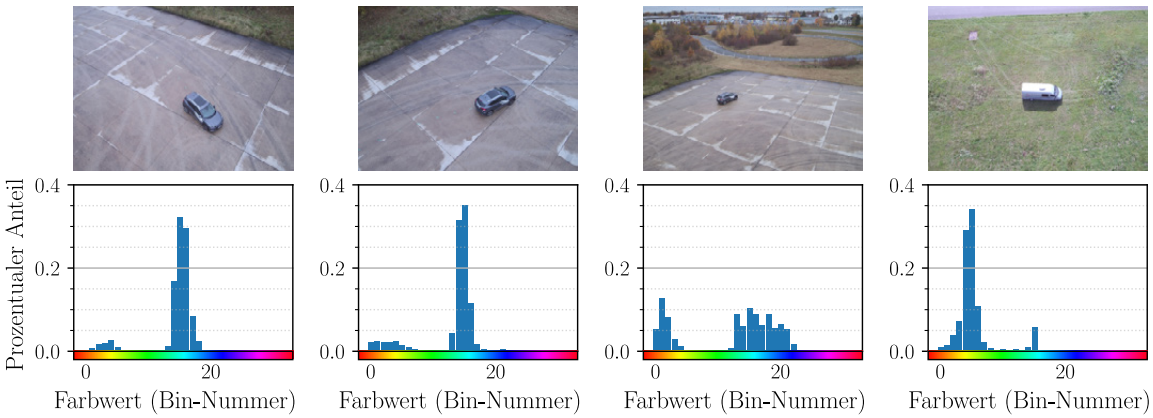


Abb. 13 Vergleich der globalen Farbverteilung für verschiedene Realflugaufnahmen anhand des Farb-Histogramms im HSV-Bereich mit 32 Bins, das auch der Berechnung des SCD zugrunde liegt.

SCD (*Scalable Color Descriptor*) beschreibt die globale Farbverteilung über das gesamte Bild mit Hilfe eines Farbhistogramms im HSV-Farbbereich. Die Werte des Histogramms werden normalisiert und nichtlinear auf eine 4-bit Repräsentation abgebildet, wobei kleinere Werte höher gewichtet werden als größere. Eine Kodierung mit Hilfe der Haar-Transformation dient der Dimensionsreduktion und ermöglicht eine Skalierbarkeit in Bezug auf Genauigkeit und Speicherbedarf des Deskriptors. Für die vorliegenden Untersuchungen wurde eine Größe von 32 Bins verwendet. Ein typischer Anwendungsbereich ist der Bild-zu-Bild Vergleich bei Farbfotografien. Abb. 13 zeigt beispielhaft die globale

Farbverteilung anhand von vier Realflugaufnahmen. Die beiden linken Bilder sind diesbezüglich sehr ähnlich und verdeutlichen die Rotationsinvarianz. Eine Veränderung in der Zusammensetzung des Bildinhalts z.B. im Hinblick auf den Untergrund führt zu einer deutlich abweichenden Verteilung. In [156] wurde gezeigt, dass der SCD wenig aussagekräftig für synthetisches Bildmaterial ist und bei monochromen Eingangsdaten eine schlechte Leistung aufweist.

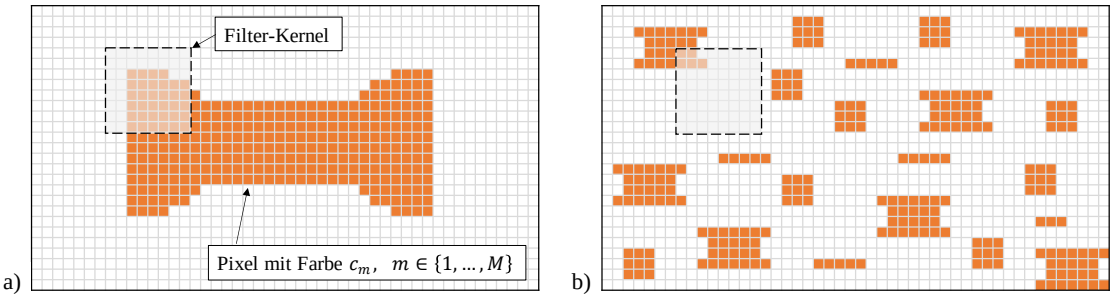


Abb. 14 Schematische Darstellung der Funktionsweise des CSD anhand zweier Bilder mit gleichem Farbhistogramm, aber unterschiedlicher lokaler räumlicher Farbstruktur: Stark strukturiertes, kohärentes Muster (a) und unstrukturiertes, inkohärentes Muster (b).
vgl. [157, 183]

CSD (*Color Structure Descriptor*) erfasst sowohl die globale Farbverteilung als auch die lokale Information bezüglich der räumlichen Struktur und Anordnung der Farben. Er dient unter anderem der Unterscheidung von Bildern, die das selbe globale Farb-Histogramm aufweisen (s. Abb. 14) und wurde ursprünglich zur Standbildabfrage bei Bildern mit rechteckig oder beliebig geformten, nicht zusammenhängenden Regionen entwickelt [161]. Die Berechnung beruht auf dem wahrnehmungsbasierten HMMD (engl.: *Hue-Max-Min-Diff*) Farbraum des MPEG7-Standards. Ein Filter-Kernel der Größe 8 x 8 Pixel scannt gemäß einem *Sliding-Window* Ansatz über das Bild und zählt die Pixelanzahl einer bestimmten Farbe im aktuell betrachteten Ausschnitt. Der Deskriptor ist skalierbar und unabhängig von der Bildgröße, da die Anzahl an Messpunkten auf 64 Stück festgesetzt ist. Bei Bildgrößen über 256 x 256 wird entsprechend unterabgetastet. Die endgültige Deskriptorgröße wird anschließend durch Requantisierung des Farbbereichs bestimmt [186]. Im Grundsatz entspricht dieses Vorgehen einer rudimentären Formerkennung und verbessert außer bei monochromen Eingangsdaten die Leistung bei der Bildsuche vor allem für natürliche Bilder [157, 161]. Für die Auswertung wurde erneut eine Deskriptorgröße von 32 Bins verwendet.

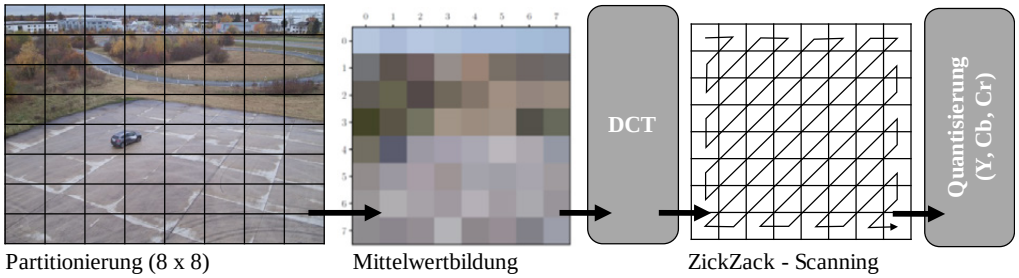


Abb. 15 Visualisierung der einzelnen Schritte bei der Berechnung des *Color Layout Descriptors* (CLD).

CLD (*Color Layout Descriptor*) beschreibt die räumliche Verteilung der dominanten Farben im YCbCr Farbraum in einer kompakten und von Bildauflösung und Skalierung unabhängigen Form [184, 185, 192]. In Abb. 15 werden die nachfolgenden Schritte anhand eines Ablaufdiagramms verdeutlicht. Das Eingangsbild wird zuerst mit einem 8 x 8 Gitter überlagert und anschließend wird für jeden Block durch Mittelung der repräsentative Farbwert bestimmt. Diese Farbwerte wiederum werden durch eine diskrete Kosinustransformation (DCT) codiert, wobei eine vorgegebene Anzahl an Koeffizienten mit niedriger Frequenz ausgewählt und durch ZickZack-Scanning quantisiert wird. Dabei werden bei

der hier betrachteten Untersuchung 10 Bins für die Luminanz im Y-Anteil und jeweils 6 Bins für jede Chrominanz im Cb- und Cr-Anteil verwendet, was zu einer Deskriptorgröße von 22 Bins führt. Jeder Anteil besteht aus dem DC-Koeffizienten, der den Grundfarbton bestimmt und weiteren AC-Komponenten. Diese beinhalten die Frequenzanteile, wobei die hohen Frequenzanteile unwichtige Details bzw. Rauschen enthalten und somit weggelassen werden können. In [192] wird eine minimale Deskriptorgröße von 12 Bins (6 für Y, 3 für Cb, 3 für Cr) empfohlen. Der CLD ist insbesondere für die schnelle Bildsuche bei skizzenähnlichem Datenmaterial, für die Inhaltsfilterung und für den Bild- bzw. Sequenzvergleich geeignet [161]. In [36] wird eine Verwendung als Messwert für mögliche Fehlanpassungen der Objekt- oder Kameraposition beim Vergleich realer und synthetischer Bildpaare ins Spiel gebracht.

Abb. 16 Visuelle Darstellung der fünf dominantesten Farben eines Beispielbildes zur Veranschaulichung des DCD.

DCD (*Dominant Color Descriptor*) wird verwendet, um eine geringe Anzahl repräsentativer Farben in einer globalen Art und Weise zu beschreiben. Im Vergleich zu den Histogramm-basierten Ansätzen ist der DCD dabei deutlich kompakter und effizienter. Durch Clusterbildung mit dem *Generalized Lloyd* Algorithmus werden je nach Bild bis zu 8 dominierende Farben ausgewählt, wobei der Deskriptor per Definition nicht auf einen bestimmten Farbraum festgelegt ist [159, 184]. Abb. 16 zeigt ein Beispielbild und die zugehörigen dominanten Farben. Zusätzlich zum Farbwert wird der prozentuale Anteil und ein optionaler Varianzfaktor bestimmt. Darüber hinaus wird für jedes Bild ein einzelner globaler räumlicher Kohärenzfaktor berechnet, der die räumliche Homogenität repräsentiert. Dies erlaubt die Unterscheidung zwischen großen zusammenhängenden einheitlichen Farbbereichen und Farben, die über das gesamte Bild verteilt sind [159]. Der in dieser Arbeit verwendete Deskriptor enthält einen Bin für die räumliche Kohärenz und zusätzlich vier Bins für den H-, S- und V-Wert und den prozentualen Anteil im Bild. Auf die Hinzunahme der Varianz wurde verzichtet, wodurch sich eine Deskriptorgröße von maximal 33 Bins ergibt, wobei diese je nach Eingangsbild schwankt. Der DCD ermöglicht einen globalen Vergleich der Farbzusammensetzung im Bild ohne Berücksichtigung der lokalen Verteilung. In [156] wurde gezeigt, dass dieser Deskriptor für jede Form von Bildinhalt angewandt werden kann, obwohl er teilweise sensitiv gegenüber Helligkeit ist.

(b) Textur-Deskriptoren: Texturen charakterisieren im Allgemeinen visuelle Muster und haben verschiedene Eigenschaften, die die strukturelle Natur einer kontinuierlichen Oberfläche beschreiben. Die Struktur hängt unter anderem von der Skalierung ab, mit der die Textur betrachtet wird und entsteht im Gegensatz zur Farbgebung durch das Zusammenspiel benachbarter Pixel. Tamura et al. [193] definierten auf Basis psychophysikalischer Studien sechs fundamentale Charakteristiken: Kontrast, Gerichtetheit, Grobkörnigkeit, Linienartigkeit, Regelmäßigkeit, Rauheit. Da Texturen unabhängig vom Farbschema sind, werden aus Effizienzgründen im Folgenden Graustufenbilder als Eingangsdaten verwendet.

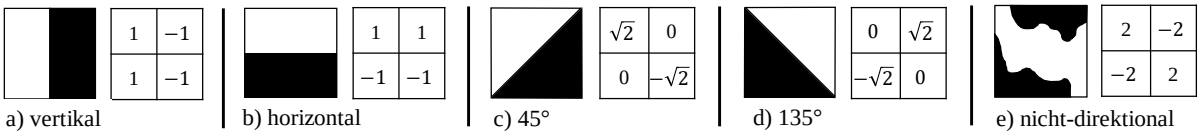


Abb. 17 Darstellung der fünf vom EHD betrachteten Arten von Kanten und der zugehörigen Kernelfunktion des Kantendetektors.
vgl. [183, 187]

EHD (*Edge Histogram Descriptor*) erfasst die räumliche Verteilung von Ecken und Kanten im Bild und wird häufig auch als nicht-homogener Texturdeskriptor bezeichnet. In [156] wurde diesem

Texturdeskriptor vor allem für Bilder mit klaren Kanten und starken Kontrasten eine sehr gute und diskriminative Leistung bei der Beschreibung nachgewiesen. Das Eingangsbild wird im ersten Schritt in 16 gleich große, nicht überlappende Blöcke eingeteilt [186]. Jeder Block wird anschließend erneut in eine feste Anzahl an Unterblöcken aufgeteilt und durch Mittelung in eine 2 x 2 Form überführt, die als Grundlage für die Kantendetektion dient. Auf diese Weise wird für jeden der 16 Blöcke ein Histogramm berechnet, das die im Teilbild vorkommenden Kanten entsprechend ihrer Richtung in fünf vorgegebene Gruppen einteilt: vertikal, horizontal, 45° diagonal, 135° diagonal und nicht-direktional. In Abb. 17 sind diese Gruppen schematisch und anhand eines zugehörigen realen Bildausschnittes dargestellt. Das Vorgehen ist hauptsächlich für die Beschreibung und den Vergleich nicht-homogener Texturen und sich nicht wiederholender Strukturen geeignet, wie sie z.B. bei Objektkanten vorkommen. Insgesamt ergibt sich eine Deskriptorgröße von $5 \cdot 16 = 80$ Bins. Won et. al [187] erweiterten den grundsätzlich lokalen EHD um globale und semi-globale Kantenhistogramme, die direkt aus den lokalen Histogrammbins berechnet werden können und konnten damit eine Verbesserung der Leistungsfähigkeit bei der Bildsuche nachweisen. Aufgrund der Vergleichbarkeit und der ohnehin schon großen Anzahl an Bins wurde diese Erweiterung in der vorliegenden Arbeit nicht verwendet.

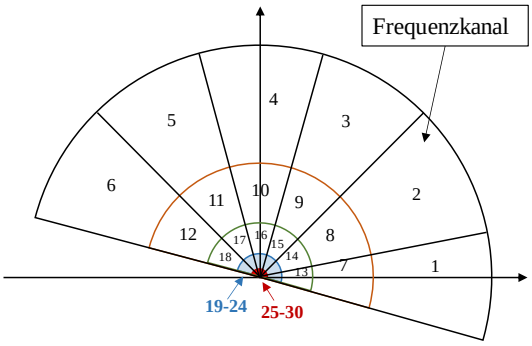


Abb. 18 Aufteilung des Frequenzbereichs in Anlehnung an die menschliche Wahrnehmung zur Berechnung des HTD.
vgl. [194]

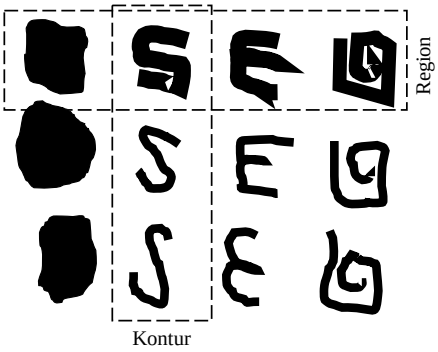


Abb. 19 Beispiele für ähnliche Formen in Bezug auf Region und Kontur.
vgl. [195]

HTD (*Homogeneous Texture Descriptor*) ahmt die menschliche visuelle Wahrnehmung nach und wird für die quantitative Charakterisierung von sich wiederholenden Strukturen und homogenen Texturen verwendet. Er beschreibt dabei globale Parameter wie die Richtung, die Grobkörnigkeit und die Regelmäßigkeit der Texturen und ist intensitäts-, skalierungs- und rotationsinvariant [160, 194]. Das Eingangsbild wird in eine Filterbank mit 30 Frequenzkanälen aufgespalten. Dazu wird der reale Frequenzbereich in 6 gleichgroße Intervalle mit jeweils 30° und in die radiale Richtung entsprechend der menschlichen Wahrnehmung in 5 Oktaven aufgeteilt (s. Abb. 18). Mit Hilfe skalierungs- und orientierungssensitiver Gabor Filter wird schließlich die Intensität und die Standardabweichung der Intensitäten für jeden Kanal durch Fourier-Transformation und Darstellung in Polarkoordinaten berechnet, wobei diese logarithmisch skaliert sind [186]. Dazu ist eine Bildgröße von mindestens 128 x 128 Pixeln nötig. Als Anwendungsfeld wird häufig die Klassifikation und Bildsuche von ähnlich aussehenden Mustern genannt, wie sie z.B. auf Luftbildern oder Satellitenbildern zu finden sind [161]. Der Deskriptor besteht im hier betrachteten Fall aus 32 Bins. Diese enthalten den Mittelwert und die Standardabweichung und 30 Bins für jeden Frequenzkanal. Die Standardabweichung der Intensitäten wurde hier nicht für die Auswertung berücksichtigt und würde die Deskriptorgröße um weitere 30 Bins erhöhen.

Der MPEG-7 Standard enthält darüber hinaus noch den TBD (Texture Browsing Descriptor). Dieser ist für schnelle Suche und Filterung in großen Texturdatenbanken gedacht, beruht aber auf sehr ähnlichen Berechnungsschritten wie der HTD und wird daher im Folgenden nicht näher betrachtet.

(c) Form-Deskriptoren: Diese dienen der Beschreibung der räumlichen Anordnung von Punkten, die zu einem bestimmten Objekt oder einer bestimmten Region gehören [195]. Es wird zwischen Konturbasierten Deskriptoren, die ausschließlich die Umrisse beschreiben und zwischen flächenbasierten Deskriptoren unterschieden, die eher verwendet werden, wenn Objekte eine ähnliche räumliche Pixelverteilung aufweisen. In Abb. 19 sind Beispiele für beide Typen dargestellt.

RSD (*Region Shape Descriptor*) repräsentiert die zweidimensionale Pixelverteilung innerhalb einer bestimmten Region und kann komplexe Objekte mit mehreren abgegrenzten Regionen ebenso beschreiben wie einfache Objekte mit oder ohne Aussparungen [195]. Der Deskriptor ist unempfindlich gegenüber Rauschen und er ist in der Lage, Objekte zu vergleichen, die aus unverbundenen Teilbereichen bestehen. Aus einem Set von ART (engl.: *Angular Radial Transform*) Koeffizienten bestehend aus zwölf Winkelfunktionen und 3 Radialfunktionen werden transformationsinvariante Momente bestimmt, die die Grundlage für die Berechnung des RSD bilden. Im hier betrachteten Anwendungsfall werden keine Masken oder segmentierten Eingangsdaten betrachtet. Außer bei der Klassifikation zwischen korrekten und inkorrekten *Bounding Boxen* bei denen hauptsächlich die Bildgröße bzw. das Seitenverhältnis beschrieben wird, spielt der RSD daher eine untergeordnete Rolle. Die Deskriptorgröße beträgt 35 Bins.

Darüber hinaus sind im MPEG-7 Standard noch weitere Form-Deskriptoren beschrieben. Der *Contour Shape Descriptor* (CShD) beschreibt den geschlossenen Umriss eines 2D Objekts oder einer Region mit Hilfe der *Curvature Scale Space* (CSS) Repräsentation. Dieser ist allerdings nur auf maskierte Eingangsdaten anwendbar und laut [156] nicht für eine statistische Analyse transformierbar. Der 3-D Shape Deskriptor berechnet ein Histogramm über die lokale Konvexität und dient der Beschreibung von 3D-Gittermodellen oder Oberflächen. Der 2D/3D Deskriptor benötigt verschiedene Ansichten eines 3D-Objekts mit unterschiedlichen Blickwinkeln und verwendet dann die beschriebenen 2D-Deskriptoren zur Beschreibung. All diese weiterführenden Form-Deskriptoren spielen allerdings für den hier betrachteten Anwendungsfall keine Rolle und werden nicht näher betrachtet. Für weiterführende Informationen sei auf [160, 161, 195] und [196] verwiesen.

Fazit

Eidenberger [156] untersuchte einen Großteil der MPEG-7 Deskriptoren in Hinblick auf deren Sensitivität, Redundanz und Vollständigkeit und die dazugehörige Parameterauswahl und generierte auf Basis dessen Richtlinien zum Gebrauch und zu einer möglichst effizienten Kombination. Er evaluierte den Einsatz auf mehreren Arten von Bildmaterial und zeigte, dass alle Deskriptoren eine gewisse Redundanz aufweisen und einige Eigenschaften visueller Medienobjekte nicht oder nur unzureichend durch die Metriken beschrieben werden, weshalb empfohlen wird, weitere Bildbeschreiber zur Auswertung hinzuzuziehen.

4.4.2 Weiterführende Bildbeschreiber

Um die ganze Bandbreite an Bildeigenschaften zu erfassen, werden zu den in Kapitel 4.4.1 beschriebenen inhaltsbasierten MPEG-7 Deskriptoren weitere z.T. qualitätsbasierte Metriken hinzugenommen. Es wurde gezielt darauf geachtet, dass im Gegensatz zu den Untersuchungen von Hummel [36] auch allgemein interpretierbare Bildeigenschaften, wie z.B. Helligkeit, Kontrast, Farbtemperatur, Rauschen oder Unschärfe enthalten sind. Viele der qualitätsbezogenen Metriken werden zum Beispiel zur allgemeinen Beurteilung von Fotografien verwendet [163, 197]. Ke et al. [163] analysierten wahrnehmungsbasierte Faktoren zur Unterscheidung zwischen professionellen Fotografien und Schnappschüssen und entwarfen auf Basis dessen globale Merkmalsbeschreiber für Bildcharakteristiken wie die Verteilung von Kanten und Farben, Farbanzahl, Unschärfe, Kontrast und Helligkeit. Sie erreichten eine

Klassifikationsgenauigkeit von 72 %. Tang et al. [197] griffen diesen Ansatz auf und erweiterten ihn um lokale Berechnungsvorschriften.

Im Folgenden sind die in dieser Arbeit betrachteten weiterführenden Bildbeschreiber kurz beschrieben und in Gruppen eingeteilt (vgl. [188]). Eine Übersicht ist wiederum in Tab. 10 zu finden, die nebenbei auch auf die zugehörigen Quellenangaben und Implementierungen verweist.

(d) Helligkeit / Luminanz / Kontrast (BLC): Dieser Merkmalsvektor enthält einige inhalts- und qualitätsbasierte Methoden für die Bewertung der Helligkeit, der Luminanz und des Kontrasts in verschiedenen Farbräumen und liefert insgesamt 20 Werte für die Beschreibung dieser Eigenschaften.

Die Helligkeit und deren Verteilung wird dabei unter anderem über den Mittelwert und die Standardabweichung der dritten Komponente im HSL-Farbraum (*Hue*: Farbton; *Saturation*: Sättigung; *Lightness*: relative Helligkeit) repräsentiert. Dieser beschreibt eine entsprechende Farbe nicht durch die Mischung bestimmter Farbanteile, sondern durch deren Eigenschaften und ist somit für die Auswahl und Beschreibung von Farben optimiert. Eine weitere Berechnungsmetrik verwendet als Maß für die Helligkeit Mittelwert und Standardabweichung der ungewichteten Summe der RGB-Werte. In [198] dagegen wird eine gewichtete Summe der RGB-Werte vorgeschlagen, da diese besser auf die menschliche Wahrnehmung der Helligkeit angepasst ist und somit auch für eine möglichst realistische Transformation von Farb- zu Graustufenbildern eingesetzt wird. Ke et al. [163] und Wang et al. [164] hingegen verwendeten für die qualitätsbasierte Beschreibung der Helligkeit bei der Fotobewertung die Summe über die erste Komponente im Lab-Farbraum. Eine ebenfalls in [163] beschriebene Methode zur Bewertung des Kontrasts berechnet das Grauwerthistogramm für jeden Kanal, summiert und normiert diese und definiert über die Breite des Histogramms eine Bewertungsmetrik. Cao et al. [162] entwickelten einen Algorithmus zur adaptiven Anpassung des Gamma-Wertes für den Anwendungsfall der automatischen Kontrastkorrektur. Durch eine leichte Abwandlung im Algorithmus wird die dafür verwendete kumulative Verteilungsfunktion der Grauwerte im Bild für die Generierung von Merkmalen abgegriffen und dem Merkmalsvektor als weitere Metrik zur Beschreibung des Kontrasts hinzugefügt. Die Vielfalt der ausgewählten Methoden ermöglicht eine umfassende Beschreibung der Bildeigenschaften Helligkeit, Luminanz und Kontrast.

(e) Farbe (Col): Da die Farbe im Allgemeinen eine sehr vielschichtige Bildeigenschaft darstellt, werden im Folgenden einige qualitätsbasierte Farbmetriken beschrieben, die die ausschließlich inhaltsbasierten und eher abstrakten MPEG7 Bildbeschreiber ergänzen und hauptsächlich auf der Wahrnehmung der Farben beruhen. Diese Farbwahrnehmung spielt neben den verwendeten Texturen und Modellen eine wesentliche Rolle für den visuellen Eindruck gerenderter Sensordaten. Bei der Auswahl wurde darauf geachtet, dass die den Metriken zugeordneten Bildeigenschaften direkt interpretierbar sind, da nur so eine unmittelbare Anpassung des Datengenerierungsprozesses möglich ist. Der resultierende Merkmalsvektor besteht aus 6 Einträgen.

Die erste Eigenschaft, die betrachtet wurde, ist die Farbigkeit des Bildes. Hasler et al. [165] entwickelten dafür auf Basis psychophysikalischer Experimente eine Berechnungsvorschrift, die durch verschiedene Kombinationen der RGB-Kanäle eine Farbigkeit angibt, die eine Korrelation von über 95 % mit der menschlichen Wahrnehmung erreicht. Für eine umfassende Beurteilung wird auch die Betrachtung von Farbstichen empfohlen [165]. Diese werden mit dem in [166] beschriebenen Farbfaktor K beschrieben. Vorteil der dabei zugrunde liegenden Berechnungsvorschrift ist die universelle Anwendbarkeit ohne die Notwendigkeit eines Referenzbildes, da eine Bewertung der Verteilung der Pixelwerte im Lab- und im Graustufenbereich vorgenommen wird. Für die Beurteilung der Farbtemperatur, die ebenfalls direkt durch das Rendering bzw. die Simulationsumgebung beeinflusst wird, wurden die Interpolation von Robertson [167] und die Berechnungsvorschrift von Hernández-Andrés et al. [168] herangezogen. Als letzter Bewertungsfaktor in dieser Gruppe dient schließlich die auf Basis eines Histogramms berechnete

Anzahl an Farben im Bild, die in [163] zur Beurteilung der Fotoqualität verwendet wurde und aufgrund ihrer direkten Proportionalität zum Detailgrad auch für die Beurteilung synthetisch erzeugter Sensorbilder interessant ist.

(f) Bildqualitätsmetriken (IQM): Diese Gruppe beinhaltet verschiedene Ansätze, die darauf ausgelegt sind, die Qualität von Fotografien in der Gesamtheit der Eigenschaften numerisch zu bewerten. Grundlage für die vorhergesagte Bewertung bildet in den meisten Fällen die menschliche Wahrnehmung. In [169] (NIMA: *Neural Image Assessment*) wurde dazu ein CNN-Modell trainiert, das zwei Bewertungen liefert, eine eher abstraktere für die ästhetische und eine eher pixel-basierte für die technische Bildqualität. Die BRISQUE (*Blind/Referenceless Image Spatial Quality Evaluator*) Methode kombiniert beides und liefert ein generelles Maß für die Natürlichkeit, das aber auch den Einfluss von Störfaktoren berücksichtigt. In [171] wurde eine Methode vorgestellt, die die Einhaltung der Drittel-Regel bei Fotografien bewertet und daher wieder eher ästhetische Gesichtspunkte und Objektanordnungen berücksichtigt. Wang et al. [164] messen durch die Verteilung und Konzentration hoher Frequenzen im Bild die Unschärfe des Hintergrunds und definierten auf diese Weise eine referenzlose Metrik zur Beschreibung der Tiefenschärfe. Insgesamt liefern diese Methoden 30 Werte zur Beurteilung der Bildqualität. Obwohl diese nur schwer in direkte Verbindung mit bestimmten Simulations- oder Bildeigenschaften gebracht werden können, ist eine Berücksichtigung in der Auswertung dennoch sinnvoll, um den Einfluss ästhetischer Gesichtspunkte und menschlicher Wahrnehmung beurteilen zu können.

(g) Verzerrung / Unschärfe / Rauschen (DBN): In dieser Gruppe sind eine Reihe von Metriken zusammengefasst, die den Anteil von Störeffekten im Bild beschreiben. Dazu zählen verschiedenste Arten von Unschärfe, aber auch Rauschen und sonstige Artefakte. Obwohl alle ausgewählten Methoden zur Gruppe der qualitätsbezogenen Bildbeschreiber gehören, beruhen sie dennoch auf einer technischen Berechnungsmethode.

Narvekar et al. [173] verwendeten zur Bestimmung der Schärfe eine kumulative Verteilungsfunktion der Kantenbreite um die im Bild vorkommenden Ecken. Kumar et al. [172] bestimmten ebenfalls die Schärfe durch Betrachtung der Veränderung der Luminanz an den im Bild vorkommenden Ecken. Im Gegensatz zur Schärfe wird in vielen Ansätzen zur Beschreibung der gleichen Bildeigenschaft auch häufig die Unschärfe bestimmt. Eine einfache und gleichzeitig effiziente Möglichkeit dies zu tun, ist die Anwendung des Laplace-Operators auf das Eingangsbild, was der Berechnung der zweiten Ableitung gleichkommt. Die Varianz des Ergebnisses ist ein Maß für die Stärke der Unschärfe. Tong et al. [175] verwendeten die Haar-Wavelet-Transformation zur Beschreibung der Unschärfe und können dadurch Bewegungsunschärfe und Unschärfe durch Defokussierung erfassen. In [163] wird ebenfalls das Verfahren von Tong et al. [175] verwendet und mit einer weiteren Methode kombiniert, die mit Hilfe der Fouriertransformation (FFT: *Fast Fourier Transform*) das Frequenzspektrum analysiert. Der letzte hier betrachtete Ansatz [174] verwendet schließlich eine Singulärwertzerlegung zur Erzeugung einer Unschärfekarte, bei der nun auch regionale Unschärfebereiche berücksichtigt werden.

Ein weiterer Störfaktor in Bildern ist Rauschen. Chen et al. [177] leiteten aus der Analyse statistischer Zusammenhänge in verrauschten Bildern eine Metrik zu Identifikation von weißem Gauß´schem Rauschen ab. In [176] wurden durch Klassifikation möglichst homogene Bereiche im Bild gesucht, da darin die Stärke des vorkommenden Rauschens am besten abgeschätzt werden kann. Die zugrunde liegende Schätzfunktion berücksichtigte dabei jedoch nicht nur weißes Gauß´sches Rauschen, sondern auch Poisson-Rauschen und farbiges Rauschen.

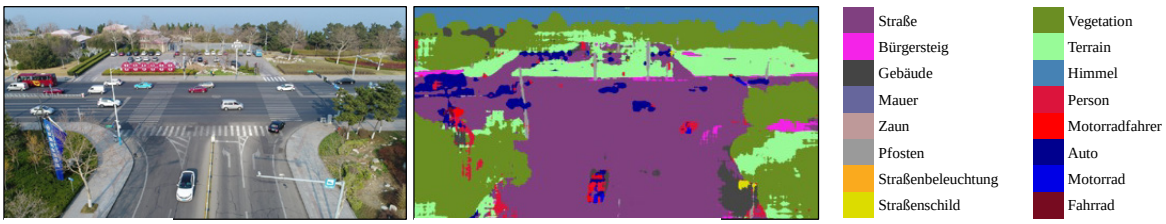
Insgesamt enthält der Vektor neun Werte. Die betrachteten Störeffekte werden durch die MPEG-7 Bildbeschreiber nicht erfasst, spielen aber dennoch eine entscheidende Rolle für die Auswertung. Durch Hinzunahme der beschriebenen Metriken kann nun untersucht werden, inwiefern vorhandene oder auch in der Simulation fehlende Störeffekte den *Reality Gap* und die Detektionsleistung beeinflussen.

(h) Formen (Sha): Die ersten drei grundlegenden Eigenschaften in diesem Vektor sind die Breite, Höhe und Fläche der Eingangsdaten in Pixeln. Bei Trainingsbildern sind diese immer gleich und beinhalten daher keine Information. Die hier vorgestellten Metriken dienen jedoch auch als Datenbasis für die Klassifikation von *Bounding Boxen* in korrekte und inkorrekte Detektionsergebnisse anhand der darin enthaltenen Bildinformation, wobei auf diese Weise der Einfluss der Größe untersucht werden soll. Allgemeines Ziel der in dieser Gruppe gesammelten Metriken, die 60 Werte liefern, ist die Beschreibung der Szenerie. Die Grundlage dafür bilden binäre Segmentierungsmasken. Hu-Momente werden zur Beschreibung und Klassifikation der darin enthaltenen Formen und Strukturen verwendet [178]. Diese sieben skalierungs- und rotationsinvarianten Momente dienen dabei zur Extraktion der Information. Die benötigten binären Segmentierungsmasken werden wie folgt generiert:

Vorder- / Hintergrundsegmentierung: Die erste Maske berechnet eine Vorder- und Hintergrundsegmentierung, wobei das Verfahren von Otsu zur Berechnung des Schwellwertes verwendet wird [199]. Morphologische Verfahren glätten anschließend das Ergebnis, bevor die Hu-Momente zur Beschreibung des resultierenden Binärbildes bestimmt und dem Datenvektor hinzugefügt werden.

Semantische Segmentierung beschreibt im Bereich der Bildverarbeitung die Zuweisung einer semantischen Klasse, wie z.B. Vegetation, Gebäude oder Fahrzeug zu jedem Pixel des Eingangsbildes. Ziel ist dabei eine vereinfachte Darstellung des Bildes, die eine Analyse der dargestellten Objekte, Inhalte und Szenerien ermöglicht. Tiefe neuronale Netze sind prädestiniert für derartige Aufgaben und erreichen eine hohe Güte, weshalb mehrere Architekturen für derartige Anwendungen zur Verfügung stehen. Für den hier betrachteten Vektor werden auf diese Weise zwei weitere Segmentierungsmasken erstellt:

Objektsegmentierung: Für diesen Zweck kam eine *Pytorch* Implementierung des DeepLabv3 Netzwerks [179, 200] zum Einsatz, die auf der *ResNet-101* Architektur beruht und mit dem COCO train2017 Datensatz [201] trainiert wurde. Alle 20 PascalVOC Klassen [153, 202] zählen bei dieser Klassifizierung zur Kategorie „Objekt“, während der Rest des Bildes als „Hintergrund“ betrachtet wird. Mit Hilfe der Hu-Momente wird anschließend die Form und Verteilung der Objekte in der Szenerie beschrieben, um Rückschlüsse auf diesbezügliche Einflüsse auf die Detektionsleistung analysieren zu können.



a) Ausgangsbild b) Semantische Segmentierung c) Legende

Abb. 20 Beispielhafte semantische Segmentierung eines Bildes aus dem UAVDT Datensatz zur Analyse der Zusammensetzung der dargestellten Szenerie

Semantische Segmentierung: Bei dieser Auswertung wird eine Maske für jede der Klassen „Straße“, „Objekt“, „Gebäude“, „Vegetation“ und „Himmel“ berechnet. Verwendet wurde dafür eine *Tensorflow* Implementierung des DeepLabv3 Netzwerks [203] auf Basis der *MobileNetV2* Architektur, das auf dem Cityscapes Datenset [204] trainiert wurde. Abb. 20 zeigt die resultierende Segmentierung, bei der zur Veranschaulichung alle Masken der einzelnen Klassen überlagert dargestellt sind. Die Genauigkeit der Segmentierung spielt dabei eine untergeordnete Rolle und wurde daher nicht speziell optimiert. Ziel ist vielmehr die numerische Beschreibung der Anteile und Formen der im Bild vorkommenden Szenerie, um daraus Rückschlüsse auf die Detektionsleistung ziehen zu können. Mögliche Fragestellungen wären dabei, inwiefern der Anteil der Straße oder Vegetation im Bild Einfluss auf die korrekten Detektionen hat.

(i) Umweltbedingungen (Env): Diese Gruppe enthält verschiedene Ansätze zur Beschreibung der vorherrschenden Umweltbedingungen. In [180] wurde ein neuronales Netzwerk entworfen, trainiert und zur Verfügung gestellt, das unter anderem durch automatisierte Berücksichtigung des Bildkontexts sehr zuverlässige Schattenkarten generiert. Der daraus ermittelte Anteil an Schattierungen im Bild bildet das erste Merkmal in dieser Gruppe. Um zu bestimmen ob und welche Wetterphänomene einen Einfluss auf die Detektionsleistung haben, wurden die Klassen „Bewölkt", „Nebelig", „Regnerisch", „Schnee", „Sonnig" und „Sonnenaufgang" definiert. Anschließend werden verschiedene fertig trainierte neuronale Klassifikationsnetzwerke [181] verwendet, um die Wettersituation im Eingangsbild vorherzusagen. Insgesamt enthält der Vektor somit sieben Werte.

(k) Ecken / Texturen (ET): Hier werden über die MPEG7-Deskriptoren hinausgehende Metriken zur Beschreibung von Texturen und Strukturen im Bild zusammengefasst. Die Grauwertematrix (engl.: *Gray Level Co-Occurrence Matrix* (GLCM)) [182] ist ein interessanter Ansatz, bei dem durch die Berechnung eines Histogramms die Lage benachbarter Grauwerte zueinander bei einem gewissen Offset beschrieben wird. Von der resultierenden Matrix können nun bestimmte Merkmale, wie z.B. Kontrast, Unähnlichkeit, Homogenität, Energie und Korrelation berechnet werden, die wiederum zur Beschreibung und Klassifikation von Texturen dienen. Eine weitere Methode [163] analysiert die räumliche Kantenverteilung, beschreibt damit die Komplexität im Bild und wird zur Unterscheidung von Bilder mit überladenem Hintergrund von Bildern mit fokussierten Objekten verwendet. Außerdem werden mit dem Canny-Algorithmus Kanten im Bild extrahiert und die Summe über die dadurch detektierten Kantenpixel dient anschließend als Merkmal für die Bildbeschreibung. Als letztes Merkmal wird schließlich im Frequenzbereich die Glattheit im Bild durch zwei verschiedene Ansätze bestimmt [164] und der Vektor somit auf 58 Werte erweitert.

4.5 Statistische Auswertemethoden

Auf oberster Ebene des in Kapitel 3.2 vorgestellten Konzepts steht die Auswertung und Analyse des *Reality Gaps* zwischen den Domänen Realität und Simulation. Grundlage dafür bildet zum einen die Detektionsleistung des auf verschiedene Weise trainierten Testalgorithmus (s. Kapitel 4.3) und zum anderen die aus den Testdaten durch Bildbeschreibermetriken (s. Kapitel 4.4) extrahierte Bildinformation. Statistische Verfahren sollen nun einen kausalen Zusammenhang zwischen beiden herstellen, um ableiten zu können, welche Bildeigenschaften einen Einfluss auf die Leistung des Testalgorithmus haben und welche Simulations- und Rendering-Parameter bei der Verwendung synthetischer Sensordaten ausschlaggebend sind. Als Methoden kommen dabei sowohl Regressionsverfahren als auch Klassifikationsalgorithmen in Frage, die beide zu den Verfahren des Überwachten Lernens zählen, da die Zielvariable bereits im Vorfeld bekannt ist. Im Folgenden werden beide Gruppen vorgestellt und auch die jeweiligen Vor- und Nachteile beschrieben. In beiden Fällen besteht die Auswertung dabei aus drei Schritten. Im ersten Schritt wird anhand von Trainingsdaten das Modell bestimmt. Anschließend wird mit Hilfe von Testdaten validiert, wie gut das Modell die Zusammenhänge beschreibt, bevor schließlich im letzten Schritt die für die Modellbildung relevanten Merkmale analysiert werden.

4.5.1 Regressionsanalyse

Bei der Regression wird ein Modell berechnet, dass versucht, mit Hilfe einer oder mehrerer unabhängiger Variablen eine bestimmte kontinuierliche Zielgröße vorherzusagen. In Abb. 21 sind die dazu gehörenden mathematischen Zusammenhänge dargestellt. Grundlage bildet dabei die Datenmatrix $\boldsymbol{X}$, in der jedem der n Objekte eine Zeile und jeder der p Eigenschaften, auch als unabhängige Variablen bezeichnet, eine Spalte zugeordnet wird. Ist $p > 1$, d.h. es wird der Einfluss mehrerer unabhängiger Variablen betrachtet, spricht man von multivariater Regression. Im hier betrachteten Fall repräsentiert jede Zeile eines der n realen und synthetischen Bildpaare und jede Spalte die dazugehörigen Differenzen der p

Bildbeschreibermetriken. Der Zielgrößenvektor $\boldsymbol{y}$ enthält diejenigen Werte, die durch Gewichtung der in der Datenmatrix $\boldsymbol{X}$ enthaltenen Informationen vorhergesagt werden sollen. Im hier betrachteten Fall sind das die Leistungsunterschiede des Testalgorithmus bezogen auf ein bestimmtes reales und synthetisches Bildpaar. Die zur Messung des Leistungsunterschieds verwendete Metrik spielt dabei im ersten Schritt keine Rolle. Der Regressionsvektor $\boldsymbol{b}$ wird je nach Regressionsalgorithmus auf verschiedene Art und Weise berechnet und beschreibt die Gewichtung der Datenmatrix $\boldsymbol{X}$ d.h. der Bildbeschreiberdifferenzen zur bestmöglichen Vorhersage der Zielgröße $\boldsymbol{y}$ d.h. der Leistungssdifferenzen. Dieser Vektor bildet die Grundlage für die spätere Auswertung, da er zur Identifikation der für die Vorhersage der Zielgröße relevanten Einflussfaktoren verwendet werden kann und somit eine Interpretation des berechneten Modells ermöglicht. Da in den meisten Fällen $n \gg p$ gilt, ist das beschriebene Gleichungssystem überbestimmt und es existiert keine eindeutige Lösung. Der Fehlerterm $\boldsymbol{e}$ enthält daher den nicht durch das Modell erklärten Anteil der Zielgröße und wird durch den Regressionsalgorithmus bei der Berechnung des Regressionsvektors $\boldsymbol{b}$ minimiert. Der Vektor $\hat{\boldsymbol{y}}$ besteht aus den Vorhersagewerten bei der späteren Anwendung oder dem Test des Modells, wobei gilt $\hat{\boldsymbol{y}} = \boldsymbol{Xb}$.

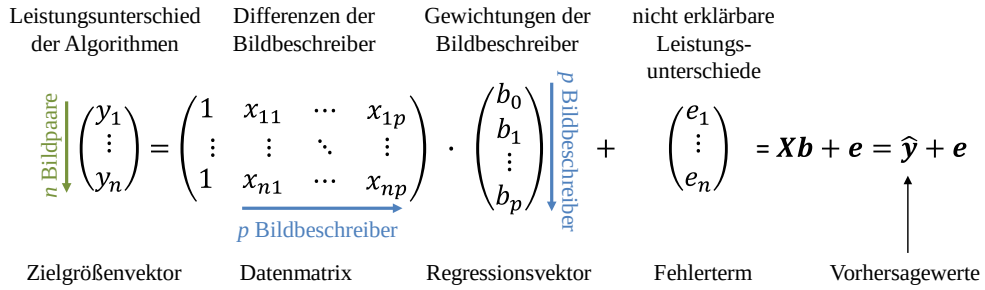


Abb. 21 Formaler Zusammenhang bei der Aufstellung der multivariaten linearen Regressionsgleichung.

In Abb. 23 werden diese Zusammenhänge grafisch für den vereinfachten univariaten d.h. zweidimensionalen Fall beschrieben. Ein linearer Regressionsalgorithmus liefert dabei eine Regressionsgerade $\hat{\boldsymbol{y}} = b_0 + b_1\boldsymbol{x}$ als Ergebnis, die die vorhandenen Datenpunkte bestmöglich interpolieren soll. $\hat{y}_k$ beschreibt den Vorhersagewert für einen bestimmten Datenpunkt und e_k den dazugehörigen Fehler. Wie bei allen Methoden des maschinellen Lernens ist zu beachten, dass das Modell nur für den während des Trainings betrachteten Wertebereichs der Eingangsdaten zuverlässige Ergebnisse liefert. Insgesamt lässt sich zusammenfassen, dass bei der Regression durch die Differenzen der Bildbeschreiber die Leistungsunterschiede zwischen realen und synthetischen Sensordaten vorhergesagt und durch die Auswertung des dazu benötigten Regressionsvektors die relevanten Bildeigenschaften identifiziert werden sollen.

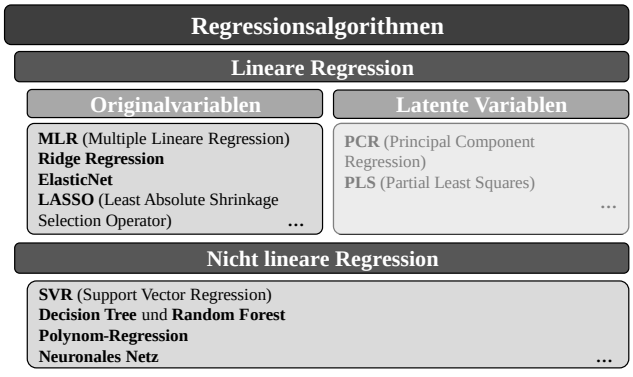


Abb. 22 Auflistung und Eingruppierung verschiedener Regressionsalgorithmen (vgl. [205]).

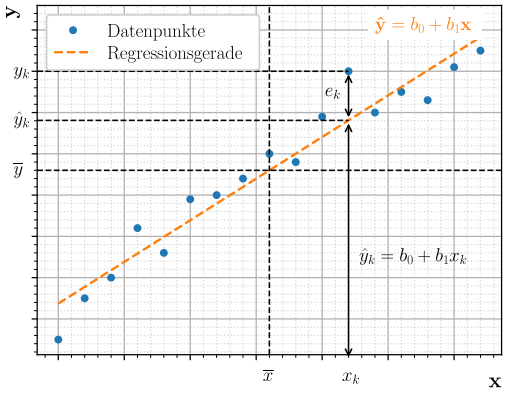


Abb. 23 Grafische Darstellung wichtiger Größen bei der univariaten Regression als zweidimensionale Vereinfachung der multivariaten Regression.

Abb. 22 zeigt eine systematische Einteilung gängiger Regressionsalgorithmen. Es wird zwischen linearen und nichtlinearen Methoden unterschieden. Da eine Interpretation des Modells für die Auswertung entscheidend ist, kommen Verfahren, die auf latenten Variablen beruhen, nicht in Frage. Die MLR (*Multiple Lineare Regression*) bildet die Basis der in der Gruppe der linearen Regressionsalgorithmen aufgelisteten Algorithmen und bestimmt den Regressionsvektor $\boldsymbol{b}$ aus der in Abb. 21 dargestellten Gleichung $\boldsymbol{y} = \boldsymbol{Xb} + \boldsymbol{e}$ mit der Methode der kleinsten Quadrate gemäß der Formel $\boldsymbol{b} = \left(\boldsymbol{X^T X}\right)^{-1} \boldsymbol{X^T y}$ [206]. Es gibt mehrere Voraussetzungen für deren Anwendbarkeit. Unter anderem soll keine Multikollinearität zwischen den unabhängigen Variablen vorliegen, da diese zu Instabilitäten bei der Matrixinversion führt, weshalb ein Verhältnis von $n{:}\,p$ von mindestens 3:1 empfohlen wird. Um diese Anfälligkeit zu reduzieren, beziehen darauf aufbauende Methoden wie z.B. Ridge Regression, ElasticNet oder LASSO (*Least Absolute Shrinkage Selection Operator*) Bestrafungs- und Regularisierungsterme bei der Berechnung mit ein. Eine weitere Gruppe sind nichtlineare Regressionsmethoden, die wiederum andere Minimierungsterme zu Grunde legen. Die SVR (*Support Vector Regressor*) ist dadurch z.B. vergleichsweise robust gegenüber Ausreißern. *Decision Trees* bilden auf Basis des Informationsgehalts in einem Knoten Entscheidungsbäume und sind dadurch leicht zu interpretieren. *Random Forest Regression* berechnet mehrere *Decision Trees* auf einer Teilmenge des Datensatzes und reduziert durch Mittelung die Gefahr für Überanpassung.

Allen Methoden gemeinsam ist jedoch die Tatsache, dass diese Form der Auswertung nur für Bildduplikate anwendbar ist, da nur hier die Berechnung von Bildbeschreiberdifferenzen zwischen realen und synthetischen Daten sinnvoll ist. Im Umfeld der Objektdetektion muss als Zielgröße der Leistungsunterschied zwischen den Domänen pro Bild berechnet werden, was vor allem bei wenigen Objekten im Bild nur begrenzt aussagekräftig ist und zu Verzerrungen führen kann. Darüber hinaus ist es ausschließlich möglich, die Bildunterschiede für das gesamte Bild zu betrachten im Gegensatz zur Analyse charakteristischer lokaler Unterschiede für eine bestimmte Gruppe von Detektionen (z.B. korrekte/inkorrekte *Bounding Boxen*). Für erste grundlegende Analysen zur allgemeinen Eignung von regressionsbasierten Verfahren wird die MLR als Algorithmus ausgewählt. Aufgrund der genannten Nachteile wird neben der Regression in dieser Arbeit zusätzlich ein weiterer alternativer Ansatz auf der Grundlage einer Klassifikation betrachtet.

4.5.1.1 Gütekriterien zur Bewertung der Regressionsanalyse

Die Auswertung eines Modells zur Ableitung relevanter Einflussfaktoren ist nur dann sinnvoll und aussagekräftig, wenn im Vorfeld sichergestellt werden kann, dass das gefundene Modell die vorherrschenden Zusammenhänge in den Daten zuverlässig beschreibt und eine entsprechende Güte aufweist. Zur Beurteilung und zum Vergleich dieser Güte werden im Folgenden die für die jeweilige Aufgabenstellung des maschinellen Lernens verfügbaren Metriken vorgestellt. Grundlage für die Beurteilung der Qualität einer Regression bilden hauptsächlich die im Fehlerterm **e** enthaltenen Abweichungen zwischen den Vorhersagewerten des Modells $\hat{y}_i$ und den tatsächlichen Werten der Zielgröße y_i.

Mittlerer Fehler (RMSE): Der RMSE (engl.: *Root Mean Square Error*) ist die Wurzel aus dem mittleren quadratischen Fehler und entspricht der nicht durch das Modell erklärten Streuung der Zielgröße [206, 207]:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}} \tag{7}$$

Er besitzt die gleiche Einheit wie die Referenzwerte und ermöglicht somit im betrachteten Messbereich eine gute Abschätzung der zu erwartenden Abweichungen, ist jedoch ungeeignet, um Modelle zu vergleichen, die auf unterschiedlichen Datensätzen und Zielgrößen beruhen.

Bestimmtheitsmaß (R^2): Dieses beschreibt den Anteil der durch das Modell erklärten Streuung an der Gesamtstreuung der betrachteten Zielgröße [206, 207] und nimmt Werte zwischen 0 und 1 an, wobei höhere Werte für eine bessere Beschreibung der Daten durch das Modell stehen:

$$R^2 = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (8)$$

Es ist unabhängig von der Maßeinheit der Daten und wird daher häufig für einen allgemeinen Vergleich von Modellen verwendet.

Grafische Überprüfung des Kalibrationsmodells: Ein sogenannter *Predicted-vs.-Measured*-Plot (vgl. Abb. 23), der auf der x-Achse die Werte der Zielgröße und auf der y-Achse die Vorhersagewerte des Modells enthält, bietet ebenfalls eine sehr vielseitige grafische Möglichkeit der Qualitätseinschätzung. Eine über die Datenpunkte gefittete Regressionsgerade sollte im Idealfall möglichst nahe an der Einheitsgerade verlaufen und repräsentiert das systematische Verhalten der Residuen. Der Vorteil dieser grafischen Analyse ist, dass eventuelle Nichtlinearitäten und vor allem auch einflussreiche Ausreißer in den Datenpunkten gut erkannt werden können.

4.5.2 Klassifikationsanalyse

Im Gegensatz zur Regression ist die Klassifikation die Berechnung eines Modells zur Vorhersage einer Zugehörigkeit zu einer bestimmten vorher definierten Gruppe anhand verschiedener Merkmale und der daraus gelernten Klassifikationsregeln. Im Vergleich zum Clustering sind die Gruppen im Vorfeld bekannt. Abb. 24 zeigt die mathematischen Zusammenhänge bei der Klassifikation. Grundlage bildet wiederum die Datenmatrix $\boldsymbol{X}$. Jede der n Spalten repräsentiert einen Datenpunkt, der im hier betrachteten Fall entweder ein Eingangsbild oder eine *Bounding Box* enthält. Im Gegensatz zur Regression sind in den p Spalten jedoch direkt die Werte der zugehörigen Bildbeschreibermetriken enthalten, was den entscheidenden Vorteil hat, dass keine Bildpaare zur Differenzbildung benötigt werden und auch lokale Bildeigenschaften in den *Bounding Boxen* beschrieben und untersucht werden können. Der Zielgrößenvektor $\boldsymbol{y}$ beschreibt für jeden Datenpunkt die jeweilige Klassenzugehörigkeit. Der verwendete Klassifikationsalgorithmus bestimmt bzw. lernt nun anhand der Trainingsdaten eine bestimmte Abbildungsfunktion $f(\boldsymbol{X})$, die die in der Datenmatrix $\boldsymbol{X}$ enthaltenen Informationen zur Bestimmung der Klassenzugehörigkeit nutzt.

Je nach Randbedingungen unterscheidet man die in Abb. 25 aufgeführten Arten der Klassifikation, die wiederum die Auswahl eines passenden Klassifikationsalgorithmus beeinflussen. Der einfachste Fall ist die binäre Klassifikation, bei der lediglich zwischen zwei Klassen unterschieden wird. Ist die Anzahl der Klassen $k > 2$ so spricht man von Multi-Klassen Klassifikation, wobei dabei jeder Datenpunkt ausschließlich einer bestimmten Zielklasse zugeordnet werden kann. In Abb. 27 wird dieser Fall für einen linearen (*Linear SVM*) und einen nichtlinearen (*Decision Tree*) Klassifikationsalgorithmus grafisch veranschaulicht, wobei die Klassenzugehörigkeit farblich gekennzeichnet ist. Die beiden Achsen x_1 und x_2 enthalten die Information aus der Datenmatrix, die verwendet wird, um ein möglichst gutes Klassifikationsmodell zu bestimmen. Bei der Multi-Label Klassifikation hingegen kann nun ein Datenpunkt zu mehreren Klassen gehören, die teilweise untereinander hierarchische Abhängigkeiten aufweisen. Eine starke Ungleichheit bei der Verteilung der Klassen im Datenset bezeichnet man schließlich als Imbalanced-Klassifikation, wobei diese meist im Rahmen der Vorverarbeitung behoben wird.

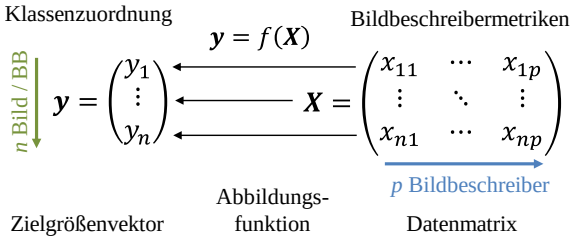


Abb. 24 Datenstruktur bei der Berechnung eines Klassifikationsmodells.

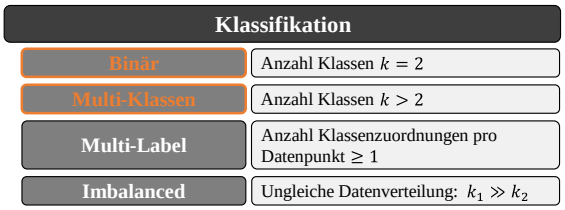


Abb. 25 Unterscheidung verschiedener Arten von Klassifikation. Die Orange hervorgehobenen Gruppen spielen für die hier betrachtete Auswertung eine Rolle.

Für die hier betrachtete Aufgabenstellung werden zwei Ziele definiert. Auf Basis der durch die Bildbeschreiber extrahierten Informationen soll zum einen zwischen realen und synthetischen Sensorbildern (= binäre Klassifikation) und zum anderen zwischen korrekten (TP), falschen (FP) und nicht erkannten (FN) Detektionen (= Multi-Klassen Klassifikation) unterschieden werden. Eine Interpretation des Klassifikationsmodells dient schließlich der Identifikation der für die jeweilige Zuordnung relevanten Einflussfaktoren. Abb. 26 zeigt eine Übersicht über verschiedene Klassifikationsalgorithmen, die im Folgenden anhand der eben definierten Rahmenbedingungen auf ihre Eignung hin untersucht werden sollen.

4.5.2.1 Klassifikationsalgorithmus

Die erste Gruppe von Klassifikatoren unterstützt ausschließlich lineare Klassifikationsgrenzen. Die Logistische Regression ist ein Vertreter dieser Gruppe und liefert ein interpretierbares Modell, ist jedoch anfällig gegenüber Multikollinearität in den Daten. Der SGD (engl.: *Stochastic Gradient Descent*) ist sehr effizient bei großen Datenmengen, erfordert aber eine Optimierung der Hyperparameter und ist sensitiv gegenüber eine Skalierung im Merkmalsraum. Die Lineare SVM (engl.: *Support Vector Machine*) erzeugt sehr kompakte und genaue Modelle, erfordert dabei aber unter Umständen lange Trainingszeiten und erlaubt vor allem keine Interpretation der relevanten Einflussfaktoren. Darüber hinaus sind die in dieser Gruppe enthaltenen Algorithmen vorwiegen für den binären Fall entwickelt und erlauben nur über Umwege eine Anwendung bei den für uns ebenfalls nötigen Multi-Klassen Klassifikationen.

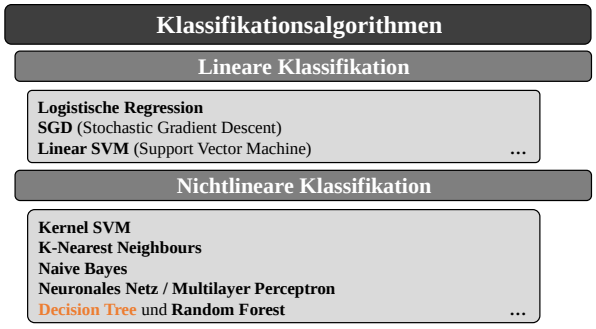


Abb. 26 Auflistung und Eingruppierung verschiedener Klassifikationsalgorithmen. Aufgrund seiner Eigenschaften wurde für die hier betrachteten Untersuchungen der *Decision Tree* Klassifikator gewählt.

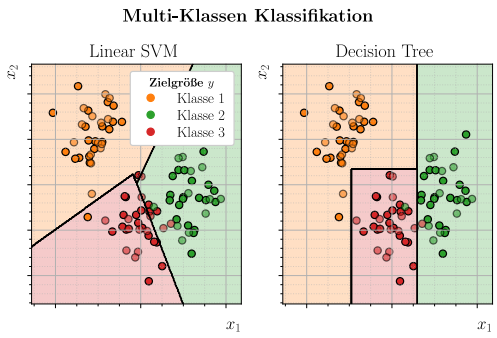


Abb. 27 Grafische Veranschaulichung der Klassifikationsgrenzen für zwei verschiedene Klassifikatoren.

Eine weitere Gruppe sind nichtlineare Klassifikatoren. Die *Kernel SVM* erlaubt durch die Verwendung spezieller Kernel Funktionen nun die Beschreibung nichtlinearer Zusammenhänge, betrachtet aber wiederum vorwiegend den binären Fall. Der *K-Nearest Neighbours* Algorithmus klassifiziert neue Daten auf Basis der Distanz zu den vorhandenen Trainingsdaten, erreicht dadurch im Allgemeinen eine hohe Klassifikationsgüte und ist robust gegenüber Rauschen, allerdings eher ineffizient bei der späteren Anwendung. Der wahrscheinlichkeitsbasierte Bayes-Klassifikator nutzt den Satz von Bayes zur Klassenzuteilung der Datenpunkte und kann einfach an neue Trainingsobjekte adaptiert werden. Die

erforderlichen bedingten Wahrscheinlichkeiten sind jedoch oft unbekannt. Auch bei komplexen Problemstellungen erreichen neuronale Klassifikationsnetze aufgrund ihrer hohen Anpassungsfähigkeit teils sehr hohe Klassifikationsgenauigkeiten, erfordern jedoch die passende Auswahl mehrere Hyperparameter und sind anfällig für Überanpassung und Rauschen. Trotz ihrer universellen Anwendbarkeit und hohen Klassifikationsgüte haben die bisher in der Gruppe der nichtlinearen Algorithmen vorgestellten Methoden alle den entscheidenden Nachteil fehlender Interpretierbarkeit. Für die in dieser Arbeit beschriebene Anwendung ist eine Interpretation des Modells zur Identifikation relevanter Einflussfaktoren jedoch zwingend notwendig.

Auswahl des Decision Tree als Klassifikationsalgorithmus

Aus diesen Gründen und da er die erforderliche Interpretierbarkeit erfüllt, wird im Folgenden der *Decision Tree (DT)* Algorithmus näher betrachtet [208]. Er ist ein nicht parametrischer Klassifikator, der aus den Trainingsmerkmalen einfache Entscheidungsregeln ableitet und daraus einen Entscheidungsbaum zur Eingruppierung neuer Datenpunkte erstellt. Das Set an Trainingsdaten wird dabei auf Basis einer Teilungsregel in zwei oder mehrere Untergruppen aufgeteilt. Für die Aufspaltung wird unter Verwendung heuristischer Methoden dasjenige Merkmal bestimmt, dass im aktuellen Fall je verwendetem Kriterium die beste Bewertung erhält. Dieser Vorgang wird iterativ für jeden Kindknoten wiederholt und auf diese Weise der Entscheidungsbaum erstellt. Jeder Knoten repräsentiert dabei eine Entscheidung basierend auf einem bestimmten Merkmal und jeder Zweig die entsprechende Aufspaltung der Datenpunkte. Das Durchschreiten der verschiedenen Pfade führt schließlich zu den Blättern, die die resultierende Klasse enthalten und im Idealfall auch nurmehr aus Trainingsdatenpunkten dieser Klasse bestehen.

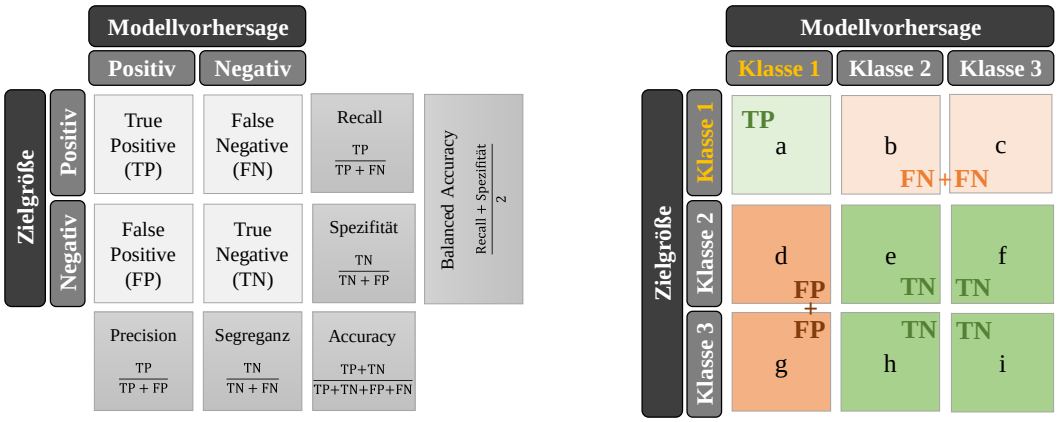
Der *Decision Tree* beruht somit nicht auf Annahmen zur Wahrscheinlichkeitsverteilung und es können binäre und Multi-Klassen Probleme gleichermaßen behandelt werden. Darüberhinaus werden nichtlineare Zusammenhänge abgebildet und es ist sowohl die Verarbeitung numerischer als auch kategorischer Merkmale möglich. Der Algorithmus ist effizient bei großen Datenmengen, erfordert nur wenig Aufbereitung der Eingangsdaten, besitzt aufgrund der Baumstruktur, die visualisiert werden kann, eine sehr gute Interpretierbarkeit und erlaubt die Validierung des Modells mit statistischen Tests.

Nachteilig ist, dass die Berechnung eines Entscheidungsbaums exponentiell mit der Anzahl an Merkmalen steigt und die verwendeten heuristischen Methoden nicht garantieren können, das globale Optimum zu finden. Außerdem können kleine Variationen in den Daten bei erneuter Berechnung zu komplett unterschiedlichen Bäumen führen. Um zu komplexe Baumstrukturen und dadurch die Anfälligkeit für Überanpassung zu minimieren, kann als Gegenmaßnahme eine minimale Anzahl an Datenpunkten pro Blatt und eine maximale Baumtiefe festgesetzt werden. Dominiert eine Klasse zahlenmäßig das Datenset, tendiert der Algorithmus zur Generierung verzerrter Modelle. Dies kann durch eine vorangehende Angleichung der Datenverteilung vermieden werden.

Da die aufgeführten Nachteile für die hier betrachtete Auswertung nur eine untergeordnete Rolle spielen bzw. durch Gegenmaßnahmen umgangen werden können, wird aufgrund der genannten Vorteile der *Decision Tree* Algorithmus für die in dieser Arbeit verwendete Klassifikationskette als Klassifikator eingesetzt.

4.5.2.2 Gütekriterien zur Bewertung der Klassifikationsanalyse

Auch hier ist eine weiterführende Analyse nur dann sinnvoll, wenn eine Klassenzuweisung der Datenpunkte mit Hilfe des trainierten Modells mit ausreichend hoher Zuverlässigkeit möglich ist. Zur Einschätzung dieser Güte stehen wiederum verschiedene Kriterien zur Verfügung. Diese entsprechen teilweise den bereits in Kapitel 4.3.3 vorgestellten Metriken, da die Objektdetektion im letzten Schritt einer Klassifikation der ROI entspricht.



a) Binäre Klassifikation b) Multi-Klassen Klassifikation

Abb. 28 Schematische Darstellung der Konfusionsmatrix zur Bewertung eines Klassifikationsmodells. Links für eine binäre Klassifikation mit der entsprechenden Zuordnung der Beurteilungen und der daraus abgeleiteten Bewertungsmetriken. Rechts für den Fall der Multi-Klassen Klassifikation und der Zuordnung der Beurteilungen (TP, TN, FP, FN) bei Betrachtung der Klasse 1.

Grundlage für die Beurteilung bildet die Konfusionsmatrix, die in Abb. 28 dargestellt ist und die Häufigkeiten des Auftretens für alle möglichen Klassenkombinationen aus Zielgröße und Modellvorhersage enthält. Sie setzt sich aus den Werten für die grundlegenden Beurteilungen TP, TN, FP und FN zusammen und ermöglicht damit eine vollständige Bewertung aller Aspekte des Modells. Abb. 28 a) zeigt den Aufbau für den Fall der binären Klassifikation und einige der daraus abgeleiteten Gütekriterien.

Precision und ***Recall*** wurden bereits in Kapitel 4.3.3 behandelt. **Accuracy** ist ein weiteres gängiges Maß und beschreibt den Anteil korrekter Klassifikationen im Verhältnis zu allen durchgeführten Klassifikationen. Liegt jedoch ein nicht ausbalanciertes Datenset vor, bei dem eine oder mehrere Klassen über- oder unterrepräsentiert sind, führt dies zu einer starken Verzerrung dieser Metrik. Aus diesem Grund wird häufig die ***Balanced Accuracy*** [209] verwendete, die diesen Nachteil durch Mittelung von *Recall* (Anteil der korrekt positiv klassifizierten Datenpunkte an allen positiven Datenpunkten) und Sensitivität (Anteil der korrekt negativ klassifizierten Datenpunkte an allen negativen Datenpunkten) umgeht. Bei ausbalancierten Datensets ist der Wert der Metrik identisch mit dem der Accuracy.

Der häufig verwendete ***F_1*-Score** ist ebenfalls eine zusammengesetzte Metrik und beschreibt das harmonische Mittel aus Precision und *Recall* (s. Kapitel 4.3.3). In einigen Fällen wird auch die **ROC-AUC** (s. Kapitel 4.3.3) als Bewertungskriterium herangezogen. Diese hat jedoch den Nachteil, dass sie durch nicht ausbalancierte Datensätze verzerrt wird, nur für Klassifikatoren anwendbar ist, die Zuverlässigkeitswerte für die einzelnen Klassifikationen liefern und keine Multi-Klassen Probleme beschreiben können. Letzteres kommt bei der hier betrachteten statistischen Auswertung jedoch vor, da auf Basis der Bildbeschreiber nicht nur zwischen realen und synthetischen Sensordaten sondern ebenso zwischen korrekten (TP) und inkorrekten (FP, FN) Detektionen unterschieden werden soll.

Abb. 28 b) zeigt die Konfusionsmatrix für diesen Fall der Multi-Klassen Klassifikation und die dabei verwendete Zuordnung zur Berechnung der Beurteilungen (TP, FP, FN, FP). Diese werden nun, anders als beim binären Fall, für jede Klasse separat berechnet. Bei der Bestimmung darauf aufbauender weiterführender Metriken wird nun zwischen **Mikro(μ)- und Makro(M)-Mittelung** unterschieden. Letztere berechnet für jede Klasse separat die jeweils betrachtete weiterführende Metrik und mittelt anschließend zur Beurteilung des Gesamtmodells die Einzelmetriken über alle Klassen. Mikro-Mittelung hingegen summiert die Beurteilungen (TP, FP, FN, FP) aller Klassen und berechnet dann aus diesen Werten die weiterführende Metrik, was vor allem bei nicht ausbalancierten Datensätzen zu verlässlicheren Ergebnissen führt. Für die hier betrachtete Auswertung des Klassifikationsergebnisses wird der $F_{1,\mu}$-Score mit Mikro-Mittelung verwendet, da er eine robuste und allgemeine Beschreibung der Gesamtleistung

ermöglicht, *Precision* und *Recall* berücksichtigt und eine gute Vergleichbarkeit über alle Modelle gewährleistet.

4.5.2.3 Klassifikationskette

Die statistische Auswertung bildet das Kernstück des in Kapitel 3.2 vorgestellten Konzepts. Durch die Analyse der Klassifikation von Sensordaten in die Domänen real und synthetisch und von Detektionen in die Klassen TP, FP und FN sollen Einflussfaktoren bzw. Bildeigenschaften identifiziert werden, die für die Generierung von und das Training mit synthetischen Sensordaten von Bedeutung sind. In Abb. 29 ist der gesamte Prozess als Klassifikationskette grafisch dargestellt. Der Fokus liegt dabei auf der Identifikation der einflussreichsten Merkmale bzw. Bildeigenschaften durch *Feature Selection* (FS) vor und *Feature Importance* (FI) Methoden nach der Klassifikation.

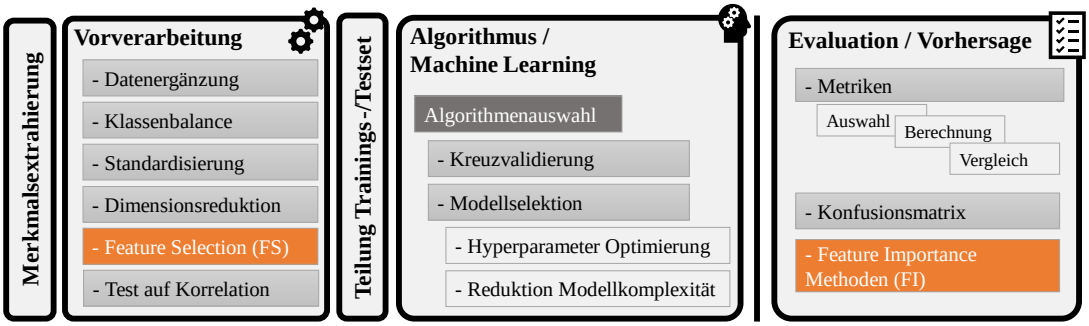


Abb. 29 Blockdiagramm zur grafischen Veranschaulichung des Ablaufs und der einzelnen Prozessschritte bei einer typischen Klassifikation (vgl. [188]).

Die Merkmalsextrahierung wurde bereits in Kapitel 4.4 näher beschrieben und entspricht der Berechnung der Bildbeschreibermetriken. Die dadurch beschriebenen Bildeigenschaften dienen als Merkmale für die Klassifikation und liefern eine Datenmatrix $\boldsymbol{X}$ mit insgesamt 496 Spalten. Die Anzahl der Datenpunkte und somit die Anzahl der Zeilen ist je nach Datensatz variabel, wurde aber auf 10 000 zufällig ausgewählte Bilder begrenzt, da sonst die Berechnung der Bildbeschreiber in Bezug auf die Rechenzeit nicht durchführbar wäre.

Vorverarbeitung

Der nächste Block enthält eine Reihe von Vorverarbeitungsschritten. Als erstes werden leere Einträge in der Datenmatrix infolge von variablen Deskriptorgrößen oder ungültigen Berechnungen durch den Mittelwert der jeweiligen Merkmalsspalte ersetzt. Dadurch soll deren Einfluss auf die Klassifikation möglichst gering gehalten werden.

Im nächsten Schritt wird die zahlenmäßige Verteilung der Klassen im Datensatz analysiert und gegebenenfalls korrigiert, da unterrepräsentierte Klassen bei der Verwendung von *Decision Trees* einen negativen Einfluss auf die Klassifikationsgüte haben, da sie zu verzerrten Entscheidungsbäumen führen. Zur Korrektur wird der SMOTE Algorithmus (engl.: *Synthetic Minority Oversampling Technique*) [210] verwendet. Dieser hat den Vorteil, dass er nicht nur Datenpunkte aus der unterrepräsentierten Klasse dupliziert, sondern aus der Verteilung der Datenpunkte im Merkmalsraum durch lineare Interpolation und Auswahl der nächsten Nachbarn neue künstlich generierte Datenpunkte erzeugt und somit zu stabileren Modellen führt.

Bei nahezu allen Algorithmen des maschinellen Lernens ist im Vorfeld des Trainings eine Standardisierung der Datenmatrix sinnvoll, um Einflüsse verschiedener Größenordnungen der Merkmale zu vermeiden. Dazu wird jeder Merkmalsvektor $\boldsymbol{x}$ mittenzentriert und anschließend durch seine Standardabweichung geteilt und besitzt somit Mittelwert $\overline{x} = 0$ und Standardabweichung $\sigma(\boldsymbol{x}) = 1$. Formelmäßig gilt folgender Zusammenhang für die Berechnung des standardisierten Merkmalsvektors $\boldsymbol{x}^s$:

$$\boldsymbol{x}^s = \frac{\boldsymbol{x} - \overline{x}}{\sigma(\boldsymbol{x})} \tag{9}$$

Zur Erhöhung der Stabilität bei der Berechnung des Modells werden im nächsten Schritt häufig Methoden zur Reduktion der Dimensionalität des Merkmalsraums angewandt. Durch Projektion in einen neuen Unterraum mit neuen Achsen wird der Einfluss von Rauschen reduziert und der Informationsgehalt erhöht. Allerdings sind bei Verwendung der projizierten Daten keine Rückschlüsse mehr auf den Einfluss einzelner spezieller Merkmalsvektoren bzw. Bildschreiberwerten möglich, weshalb dieser Schritt für die hier betrachtete Analyse bewusst vermieden wurde.

Während der Vorverarbeitung werden einige FS-Methoden verwendet, um konstante, quasi-konstante oder doppelt enthaltene Merkmalsvektoren aus der Datenmatrix zu entfernen. Details zu weiteren FS-Methoden, die entkoppelt von der Vorverarbeitung zur späteren Analyse und Identifikation der relevanten Einflussfaktoren dienen, werden in Kapitel 4.5.2.4 erläutert.

Abschließend werden nun die in der Datenmatrix auftretenden Korrelationen analysiert, da zu hohe Korrelationen je nach Klassifikator einen mehr oder weniger starken negativen Einfluss auf die Interpretierbarkeit des Modells haben. Multikollinearität tritt auf, wenn zwei oder mehr Merkmalsvektoren untereinander stark korrelieren, d.h. ein bestimmter Merkmalsvektor kann durch die Linearkombination von zwei oder mehr anderen Merkmalsvektoren ausgedrückt werden [206]. Hohe Werte in der Korrelationsmatrix liefern erste Anzeichen, sind jedoch bivariate Indikatoren und beschreiben nur paarweise Abhängigkeiten. Um auch die Stärke multivariater Korrelationen aufzudecken, wird eine Regression jedes Merkmalsvektors $\boldsymbol{x}_j$ auf die verbleibenden Merkmalsvektoren durchgeführt. Der dadurch erhaltene multiple Korrelationskoeffizient R_j beziehungsweise das entsprechende Bestimmtheitsmaß ${R_j}^2$ sind ein Maß dafür, wie gut sich $\boldsymbol{x}_j$ durch die restlichen Merkmale darstellen lässt. Häufig wird auch der daraus abgeleitete Variance Inflation Factor (VIF) für diesen Zweck verwendet [206, 211]:

$$VIF_j = \frac{1}{1 - {R_j}^2} \tag{10}$$

Klassifikation

Der nächste Block enthält eine zufällige Aufteilung der Ausgangsdaten in Trainings- und Testdaten mit einem Verhältnis von 7:3. Die Trainingsdaten werden für den Lernprozess des Klassifikationsalgorithmus verwendet. In Kapitel 4.5.2.1 wurde näher erläutert, warum der *Decision Tree* Algorithmus hier zum Einsatz kommt. Zum Zwecke der Modellselektion wird dabei die Methode der Kreuzvalidierung verwendet [212]. Diese teilt im ersten Schritt das Datenset in Gruppen bestimmter Größe ein. In einem iterativen Prozess wird jede der Gruppen einmal ausgelassen, während aus den Daten der verbleibenden Gruppen das Klassifikationsmodell erstellt wird, welches anschließend anhand der ausgelassenen Datenpunkte validiert werden kann. Auf diesem Verfahren beruht die nachfolgende Modellselektion, die zur Optimierung der Hyperparameter dient. Betrachtet werden dabei je nach Datensatz verschiedene Baumtiefen und zwei unterschiedliche Aufspaltungskriterien (*Gini*, *Entropy*) [213]. Es wird schließlich dasjenige Modell mit der niedrigsten Baumtiefe ausgewählt, dessen Klassifikationsgüte aber dennoch innerhalb der ersten Standardabweichung der bestmöglichen Güte liegt. Auf diese Weise soll ein Kompromiss zwischen Modellkomplexität und Modellgüte hergestellt werden, um die Anfälligkeit für Überanpassung auf die Trainingsdaten zu reduzieren.

Evaluierung

Der letzte Block beschreibt schließlich die Evaluierung des ausgewählten Modells auf den unbekannten Testdaten. Ebenso wie bei der Kreuzvalidierung wird zur Bewertung der F1-Score mit Mikro-Mittelung verwendet, da dieser das harmonische Mittel aus *Recall* und *Precision* darstellt und auch für Multi-

Klassen Probleme geeignet ist. Des Weiteren dient die Konfusionsmatrix zur detaillierteren Analyse der Klassifikationsergebnisse in Bezug auf einzelne Klassen. Abschließend findet unter Verwendung des trainierten und selektierten Klassifikationsmodells die Berechnung der FI-Methoden statt. Durch Kombination und Vergleich mit den FS-Methoden, die ausschließlich die Datenmatrix betrachten, wird durch Mittelung eine stabile Identifikation derjenigen Merkmale bzw. Bildeigenschaften erreicht, die für die Klassifikationsaufgabe von Bedeutung sind. Aufgrund ihrer zentralen Bedeutung für die Auswertung werden daher im Folgenden die dabei verwendeten FS- und FI-Methoden genauer beschrieben.

4.5.2.4 Feature Selection Methoden

Feature Selection Methoden werden zur Identifikation relevanter Merkmale in der Datenmatrix herangezogen. Sie zählen zur Gruppe der Datenvorverarbeitungsalgorithmen mit dem Ziel einer Maximierung der Relevanz und einer Minimierung der Redundanz in der Datenmatrix [214, 215]. Sie wählen aus der ursprünglichen Anzahl an Merkmalsvektoren eine kleinere repräsentative Untergruppe aus und werden verwendet, um ausschließlich auf Basis der Ausgangsdaten und unabhängig vom verwendeten Klassifikator die relevanten Merkmale zu selektieren. Im Gegensatz zu Methoden der Dimensionsreduktion werden dabei keine Transformationen angewandt, wodurch die Interpretierbarkeit der Modelle erhalten bleibt. Vorteile sind kürzere Trainingszeiten und kompaktere Modelle durch einen höheren Informationsgehalt in der Datenmatrix, eine Verringerung von Korrelationen und ein niedrigeres Risiko für Überanpassung durch die Reduktion von Rauschen und Redundanzen. Im hier vorgestellten Ansatz verändert lediglich die Entfernung konstanter, quasi-konstanter und doppelt vorhandener Merkmale im Zuge der Vorverarbeitung direkt die für die Klassifikation verwendete Datenmatrix. Alle anderen im Folgenden vorgestellten Methoden dienen ausschließlich der Identifikation relevanter Merkmale und nehmen keinen Einfluss auf die Datenmatrix.

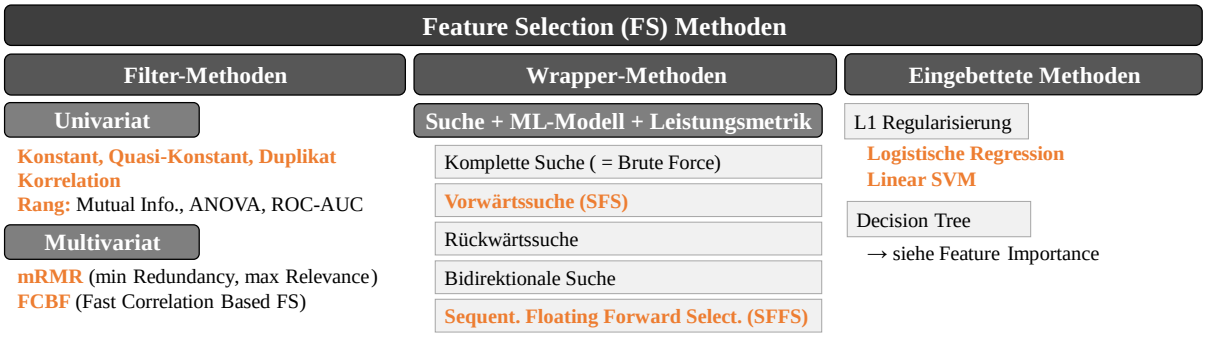


Abb. 30 Überblick und Einordnung gängiger *Feature Selection* Methoden, die auf Basis der Datenmatrix relevante Merkmale identifizieren. Die Orange hervorgehobenen Methoden werden für die spätere Auswertung verwendet (vgl. [188]).
ANOVA: *Analysis of Variance*; ROC: *Receiver Operator Characteristics*; AUC: *Area Under Curve Metric*; ML: *Machine Learning*, SVM: *Support Vector Machine*

In Abb. 30 ist eine Eingruppierung der verschiedenen FS-Methoden dargestellt. In Orange hervorgehobene Methoden werden für das vorgestellte Analyseverfahren verwendet. Auf oberster Ebene wird zwischen Filter, Wrapper und eingebetteten Methoden unterschieden [214–216].

Filter-Methoden

Filter-Methoden bestimmen mit Hilfe statistischer Tests den Zusammenhang zwischen Merkmalsvektor und Zielgrößenvektor und liefern als Ergebnis dieser Charakterisierung der Daten eine Rangfolge der Merkmalsvektoren zurück, die deren Einfluss auf die Zielgröße widerspiegelt. Vorteile dieser Gruppe sind die Einfachheit und Effizienz der Berechnung und somit auch die Anwendbarkeit für große Datensätze. Es ist jedoch zu beachten, dass die mathematischen Annahmen für die statistischen Tests erfüllt

sein müssen und dass derartige Methoden kein Maß für die optimale Größe der zu erstellenden Untergruppe an Merkmalen liefern.

Im univariaten Fall wird nun für jeden Merkmalsvektor separat ein entsprechendes Bewertungsmaß berechnet. Dies hat den Nachteil, dass keine Zusammenhänge zwischen den Merkmalen erfasst werden und vor allem bei hoher Korrelation in den Daten eine Auswahl redundanter Merkmale möglich ist. Als Bewertungsmaße kommen hierbei *Mutual Information*, eine Varianzanalyse (ANOVA, engl.: *Analysis of Variance*) oder die Fläche unter der ROC-Kurve (ROC-AUC, engl.: *Area Under Curve*) in Frage. *Mutual Information* misst eine beliebige Form statistischer Abhängigkeit zweier Variablen, in unserem Fall Merkmalsvektor und Zielgröße, und erfasst dabei im Gegensatz zur Korrelation auch nicht-lineare Beziehungen. Die Varianzanalyse schätzt mit Hilfe eines F-Tests anhand des Grades der linearen Abhängigkeit den Beitrag des Merkmals, erwartet dabei allerdings eine Normalverteilung innerhalb der Variablen. Desweitern kann für jedes Merkmal auf Basis des *Decision Tree* Klassifikators ein separates Modell berechnet und die resultierende ROC-AUC Metrik als Maß für die Festlegung der Rangfolge verwendet werden. Dies hat den Vorteil, dass keine statistischen Annahmen nötig sind.

Im Gegensatz zum univariaten Fall beruhen die multivariaten Methoden auf einer Suchstrategie zur Generierung von Untergruppen, berücksichtigen auch die Zusammenhänge zwischen den Merkmalen und analysieren dadurch den gesamten Merkmalsraum. Ein Vertreter dieser Gruppe ist der heuristische mRMR Algorithmus (minimale Redundanz – maximale Relevanz) [217]. Er beruht auf der Vorwärtssuche und versucht durch Betrachtung des Einflusses (*Mutual Information*) aber auch der Korrelation eine möglichst optimale Untermenge an Merkmalen zu finden, ohne dabei redundante Merkmale mit aufzunehmen. Der FCBF Algorithmus (engl.: *Fast Correlation Based* FS) [218] hingegen verwendet eine Rückwärtssuche, ist effizient bei großen Datenmengen und wählt Merkmale aus, die eine hohe Korrelation mit der Zielgröße, aber eine geringe Korrelation mit den anderen Merkmalen aufweisen. Die dafür verwendete Metrik wird als „*Symmetrical Uncertainty*“ bezeichnet.

Wrapper-Methoden

Die zweite große Gruppe unten den FS-Algorithmen sind Wrapper-Methoden. Sie bestehen aus einer Suchmethode, die in einer definierten Art und Weise Untergruppen an Merkmalen aus dem gesamten Merkmalsraum auswählt und einem Algorithmus des maschinellen Lernens, der in jedem Schritt auf Basis dieser Untergruppen trainiert wird. In Verbindung mit einer Kreuzvalidierung und einer geeigneten Leistungsmetrik wird nun das Modell und somit die Güte der Untergruppen evaluiert und diejenige mit der höchsten Güte als Ausgangspunkt für den nächsten Schritt verwendet. Dieser Vorgang wird iterativ so lange wiederholt, bis eine bestimmte Anzahl an Merkmalen oder eine bestimmte Güte erreicht ist. Nachteilig ist, dass sich aus dieser Vorgehensweise bei großen Datensätzen recht hohe Rechenzeiten ergeben. Außerdem ist die Auswahl der Merkmale in gewissem Maß für den jeweils trainierten Algorithmus optimiert, was jedoch auch von Vorteil sein kann, wenn dieser Algorithmus beim späteren Einsatz zur Anwendung kommt. Weitere Vorteile sind die im Allgemeinen sehr hohe Genauigkeit und die Berücksichtigung von Zusammenhängen zwischen den Merkmalen.

Man unterscheidet dabei mehrere Suchstrategien. Eine komplette Suche mit Berücksichtigung aller Kombinationen an Merkmalen (= Brute Force) findet zwar das globale Optimum, ist jedoch sehr rechenintensiv, bei großen Datensätzen nicht anwendbar und sehr anfällig für Überanpassung. Eine Alternative ist die Rückwärtssuche. Sie startet mit einem Modell, dass alle verfügbaren Merkmale berücksichtigt und entfernt in jedem Schritt dasjenige Merkmal, ohne das die Leistungsmetrik am geringsten abfällt oder sogar steigt. Im Gegensatz dazu startet die Vorwärtssuche mit einem einzelnen Merkmal und fügt iterativ diejenigen Merkmale hinzu, die die Leistung des Modells am deutlichsten verbessern. Nachteil beider Methoden ist, dass ein einmal hinzugefügtes bzw. entferntes Merkmal nicht mehr

betrachtet wird, auch wenn sich dessen Einfluss durch Hinzunahme bzw. Entfernung anderer Merkmale geändert hat, was vor allem bei hoher Multikollinearität in den Daten vorkommen kann.

Weitere auf diesen grundlegenden Strategien aufbauende Verfahren umgehen die Problematik. Die Bidirektionale Suche berechnet zum Beispiel parallel eine Vorwärts- und Rückwärtssuche und erreicht durch vorgegebene Randbedingungen, dass diese zur selben Lösung konvergieren. Die SFFS (engl.: *Sequential Floating Forward Selection*) hingegen erweitert eine Gruppe ähnlich wie die Vorwärtssuche iterativ um das Merkmal mit der höchsten Leistungssteigerung, fügt dann jedoch einen Schritt hinzu, bei dem getestet wird, ob nun das Entfernen eines beliebigen Merkmals aus dieser Gruppe zu einer Verbesserung führt. Dieser Vorgang des Hinzufügens und des optionalen Entfernens wird so lange wiederholt, bis ein Stoppkriterium erreicht wird.

Für das hier vorgestellte Auswerteverfahren wird zum einen die einfache Vorwärtsselektion betrachtet, da diese mit der schrittweisen Identifikation der einflussreichsten Merkmale ein ähnliches Ziel verfolgt und zum anderen die darauf aufbauende SFFS, um durch Berücksichtigung von Korrelationen innerhalb der Daten ein möglichst aussagekräftiges gemitteltes Endergebnis zu erzielen.

Eingebettete Methoden

Die letzte Gruppe beinhaltet schließlich die eingebetteten Methoden. Diese erhalten durch Analyse eines fertig trainierten Modells Einblick über den Einfluss eines bestimmten Merkmals für die Vorhersage, wobei nur bestimmte Algorithmen des maschinellen Lernens ein *White-Box* Modell generieren, das für diese Beurteilung geeignet ist. Eingebettete Methoden betrachten damit die FS als Teil der Modellgenerierung bzw. des Trainingsverhaltens und sind aus diesem Grund weniger rechenaufwendig als die Wrapper Methoden und weniger anfällig für Überanpassung. Sie erfassen aber dennoch die Zusammenhänge der Merkmale und sind dadurch genauer als Filter Methoden.

Die erste Untergruppe aus diesem Bereich (s. Abb. 30) beinhaltet vorwiegend lineare Algorithmen und verwendet beim Training einen Bestrafungsterm, wie z.B. die L1-Regularisierung. Dieser setzt gezielt die Koeffizienten einiger Merkmale zu Null und unterdrückt damit deren Einfluss bei der Vorhersage. Die Koeffizienten sind daher ein Maß zur Gewichtung der Merkmale. Dies führt zu kleineren und kompakteren Modellen, zu einer höheren Generalisierungsfähigkeit und zur Entfernung von Rauschen, das vorwiegend in einflusslosen Merkmalen vorhanden ist. Logistische Regression und Linear SVM sind zwei Methoden aus dieser Gruppe, die beim vorgestellten Auswerteverfahren zum Einsatz kommen. Die zweite Untergruppe analysiert baumbasierte Methoden, wie z.B. den auch in dieser Arbeit für die Klassifikation verwendeten *Decision Tree* Algorithmus. Er kann jedoch auch zu den FI Methoden gezählt werden und wird daher im nachfolgenden Kapitel genauer behandelt.

4.5.2.5 Feature Importance Methoden

Feature Importance Methoden kommen zum Einsatz, um ein bereits bestehendes Modell zu interpretieren und diejenigen Merkmale zu identifizieren, die bei der Anwendung des Modells einen entscheidenden Beitrag zur Vorhersage der Zielgröße liefern [219]. Dieses Vorgehen zählt im Allgemeinen zum Themenkomplex der „erklärbaren künstlichen Intelligenz“ (engl.: *Explainable Artificial Intelligence*, XAI). Tab. 11 liefert eine Übersicht über gängige Methoden und deren Eigenschaften. Es wird dabei zwischen Modell-spezifischen und Modell-agnostischen Methoden unterschieden, wobei letztere für jeden beliebigen Algorithmus anwendbar sind, während Modell-spezifische Methoden nur auf einen bestimmten Typ von Algorithmus zugeschnitten sind. Lokale Methoden betrachten die Verteilung und den Einfluss der Merkmale lediglich für eine bestimmte Vorhersage bzw. für einen bestimmten Datenpunkt, während globale Methoden den gesamten Datensatz berücksichtigen und somit eine umfassendere Analyse ermöglichen.

Tab. 11 Übersicht über gängige Methoden zur Bestimmung der *Feature Importance* und deren Eigenschaften. Die in Orange markierten Methoden werden für die vorliegende Auswertung herangezogen.
LIME: *Local Interpretable Model-Agnostic Explanations*; Mod.: mit Rauschen modifizierte Testdaten; SHAP: *Shapley Additive exPlanations*

FI Methode	Modell-spezifisch	Modell-agnostisch	Lokal	Global	Neues Training	Datensatz	Eigenschaften
Decision Tree FI	✓	✗	✗	✓	✗	**Training**	Eingebettet
Permutation FI	✗	✓	✗	✓	✗	Training + **Test**	
Drop-Out FI	✗	✓	✗	✓	✓	**Training + Test**	
LIME [220]	✗	✓	✓	✗	✓	Mod. Test	Binäre Klassen
SHAP [221], **TreeSHAP** [222]	✗	✓	✓	✓	✗	**Training**	Pro Klasse

Die *Decision Tree* FI spielt bereits bei der Berechnung des Entscheidungsbaumes eine Rolle (Eingebettete Methode, s. Kapitel 4.5.2.4) und zählt daher zu den globalen Modell-spezifischen Methoden. Sie sagt aus, um welches Maß die Reinheit in den Kindknoten steigt, wenn der Baum anhand eines speziellen Merkmals aufgesplittet wird und erzeugt somit eine Rangfolge der Merkmale. Merkmale, die in der Baumstruktur näher an der Wurzel zu einer Aufteilung führen, haben daher einen höheren FI Wert, da sie bereits einen signifikanten Beitrag zur Unterscheidung der Klassen liefern. Die *Decision Tree* FI kann sehr schnell aus einem bestehenden Modell extrahiert werden, ist auf den zur Anwendung kommenden Klassifikator zugeschnitten und erfordert kein erneutes Training. Hohe Korrelationen in den Daten können jedoch zur Verfälschung der Ergebnisse führen und die Methode tendiert zur Bevorzugung von Merkmalen, die eine hohe Anzahl an möglichen Werten aufweisen [223]. Darüber hinaus werden die Zusammenhänge lediglich in Bezug auf die Trainingsdaten analysiert.

Um dennoch eine möglichst aussagekräftige und stabile Analyse des Modells zu erhalten, werden daher in dieser Arbeit parallel dazu weitere FI Methoden betrachtet. Die Permutation FI evaluiert im ersten Schritt das bestehende Modell. Anschließend werden für einen bestimmten Merkmalsvektor die Einträge willkürlich vertauscht, um sicherzustellen, dass kein Zusammenhang mehr zum Zielgrößenvektor besteht, bevor das Modell erneut mit den modifizierten Daten evaluiert wird. Dieser Prozess wird für jeden Merkmalsvektor separat durchgeführt und der jeweils beobachtete Leistungsabfall ist ein Maß für den Informationsgehalt bzw. die FI des Merkmals. Von Vorteil dabei ist, dass die Evaluierung sowohl auf den Trainings- als auch auf den Testdaten durchgeführt werden kann, um z.B. die Ursachen für Überanpassung zu identifizieren und dass diese globale und Modell-agnostische Methode sehr zuverlässige Ergebnisse liefert. Es besteht jedoch die Gefahr, dass korrelierte Merkmale unterbewertet werden und zudem ist der Rechenaufwand vor allem bei hochdimensionalem Merkmalsraum nicht unerheblich.

Drop-out FI ist ebenfalls eine globale und Modell-agnostische Methode, die das mit allen Merkmalen trainierte Modell mit einem Modell vergleicht, das ohne dem zu bewertenden Merkmal erstellt wurde. Der Leistungsabfall bei Evaluierung auf Trainings- oder Testdaten ist ein Maß für die FI des Merkmals. Drop-out FI ist die genaueste FI Methode, weshalb sie in dieser Arbeit sowohl für die Trainings- als auch für die Testdaten berechnet wird und somit einen höheren Einfluss bei der Gesamtbetrachtung hat. Nachteilig ist, dass die Methode vergleichsweise rechenintensiv ist, da für jedes Merkmal ein neues Modell trainiert werden muss.

Ein eher spezielleres Verfahren ist LIME (*Local Interpretable Model-Agnostic Explanations*) [220]. Es analysiert, welche Merkmale in einem speziellen Bereich des Merkmalsraums den größten Einfluss haben und was eine Änderung der Merkmalswerte in Bezug auf die Vorhersage bewirken würde. Da dabei nur eine lokale Auswertung eines speziellen Datenpunktes möglich ist und das Verfahren zudem auf binäre Klassifikationen beschränkt ist, wird es in der vorgestellten Auswertung nicht berücksichtigt.

SHAP (*Shapley Additive exPlanations*) [221] ist ein Modell-agnostisches Verfahren, das auf den aus der Spieltheorie stammenden *Shapley* Werten [224] beruht. Grundsätzlich ist das Verfahren für jede Art von Modell anwendbar, für baumbasierte Algorithmen wie den hier verwendeten *Decision Tree* Klassifikator wurde von Lundberg et al. [222] jedoch eine spezielle Erweiterung namens TreeSHAP vorgestellt, die besonders effizient und im Gegensatz zur *Decision Tree* FI robust gegenüber Verzerrungen ist und dadurch eine sehr zuverlässige Modellinterpretation ermöglicht. Ziel des Verfahrens ist die Beurteilung des Beitrags, den ein bestimmtes Merkmal zur Vorhersage einer bestimmten Klasse leistet. Für jede Klasse wird somit eine Rangfolge der Merkmale erstellt, wobei die ermittelten Beiträge im Gegensatz zu den bisher vorgestellten Verfahren sowohl positiv als auch negativ sein können, was die Identifikation zusätzlicher Zusammenhänge ermöglicht. Sowohl SHAP als auch TreeSHAP weisen dabei folgende von den Shapley Werten stammenden Eigenschaften auf:

- *Effizienz*: Summe der Beiträge entspricht Unterschied zwischen Vorhersage und Mittelwert
- *Symmetrie*: Zwei Merkmale mit gleichem Wert liefern auch gleichen Beitrag
- *Dummy-Sensitivität*: Merkmal ohne Beitrag zur Vorhersage erhält den Shapley Wert null
- *Additivität*: Eine Kombination der Shapley Werte über mehrere Modelle ist möglich

Diese Eigenschaften ermöglichen auch eine Zusammenfassung der ursprünglich lokalen Werte und somit eine globale Aussage für das ganze Datenset, die konsistent ist mit der lokalen Bewertung einzelner Datenpunkte. Aufgrund der aufgeführten positiven Eigenschaften wurden die TreeSHAP Werte ebenfalls in die Auswertung integriert, wodurch je nach Anzahl der Klassen k schließlich $4 + k$ FI Methoden für die Identifikation der einflussreichsten Bildeigenschaften herangezogen werden.

5 Implementierung, Experimentalaufbau und Durchführung der Experimentalflüge

Im Anschluss an die Beschreibung des Standes der Technik und der grundlegenden Methoden soll nun genauer auf den Experimentalaufbau eingegangen werden, der zur Umsetzung des in Kapitel 3.2 beschriebenen Konzepts und zur Beantwortung der aufgestellten Forschungsfragen verwendet werden soll. Dabei wird insbesondere auch auf die 3D-Modellierung und Darstellung der virtuellen Szenerie eingegangen und die Zusammensetzung und Variationen der damit erzeugten synthetischen Trainings- und Testdatensätze vorgestellt. Anschließend folgt eine Beschreibung des Hardware- und Softwareaufbaus zur Durchführung der Realflüge, die zur Generierung von realen und synthetischen Bildpaaren mit Szenarien aus dem späteren Anwendungsfall nötig sind. Außerdem wird in diesem Abschnitt auch näher auf das Trainingsverhalten und die Bestimmung und Auswahl der Trainingsparameter für das als Testalgorithmus ausgewählte YOLOv3 Detektornetzwerk eingegangen,

5.1 Generierung und Beschreibung der synthetischen Datensätze

Aus den in Kapitel 1 dargestellten Gründen spielt die Verwendung synthetischer Sensordaten eine immer größer werdende Rolle. Daher muss im Allgemeinen und für jeden speziellen Anwendungsfall untersucht werden, inwiefern diese das Trainingsverhalten und die Detektionsleistung beeinflussen, welche Einflussfaktoren dabei die ausschlaggebende Rolle spielen und was daraus für Schlüsse zur Trainingsdatengenerierung mit synthetischen Daten abgeleitet werden können. Zur Untersuchung aller dieser Teilaspekte wird eine entsprechend modellierte virtuelle Szenerie benötigt, aus der anschließend mit Hilfe eines entsprechend parametrisierten Generierungsprozesses synthetische Trainings- und Testdatensätze abgeleitet werden können.

5.1.1 Aufbau und Modellierung der virtuellen 3D-Szene

Bei der Gestaltung der verwendeten virtuellen Szene müssen bei den vorliegenden Untersuchungen zwei Ziele als Rahmenbedingungen berücksichtigt werden. Zum einen soll die virtuelle Szene zur Generierung eines synthetischen Trainingsdatensatzes geeignet sein, wobei zur Einflussanalyse auch eine Variation der dabei verwendeten Datensatzgestaltungs-, Sensor- und Simulationsparameter nötig ist. Zum anderen werden zur weiterführenden und detaillierteren Analyse des *Reality Gaps* auch inhaltsgleiche reale und synthetische Bildpaare benötigt. Die virtuelle Welt muss daher ebenfalls geeignet sein, die während realer Flugversuche aufgezeichneten Sensordaten und deren Eigenschaften entsprechend nachzubilden. Dies setzt voraus, dass das Fluggelände und dessen Zustand zum Aufnahmezeitpunkt in der virtuellen Szene nachmodelliert wird.

Um diese beiden Kriterien erfüllen zu können, wurde das Gelände der Bundeswehr Universität München mit dem im Süden angrenzenden Testflugbereich als Ausgangspunkt für die Modellierung ausgewählt. Es erlaubt die Durchführung von Realflügen, weist die nötige Variabilität bezüglich verschiedener Szenarien auf und ermöglichte durch die örtliche Nähe die Aufnahme der benötigten Texturen und die Nachmodellierung der in der Realität vorhandenen Details. Außerdem steht es in keinem Bezug zu dem in Kapitel 4.1 beschriebenen realen Benchmark Trainingsdatensätzen. Die damit trainierten Modelle können somit auf realen und nachgebildeten synthetischen Testdaten evaluiert werden, die komplett unabhängig von den verwendeten Trainingsdaten sind. Dies ist besonders vorteilhaft, da das Gelände somit eine sehr aussagekräftige Testumgebung zur Repräsentation des späteren Anwendungsfalls darstellt und die Form der Evaluierung auf unabhängigen Testdaten die zuverlässigste Art und Weise darstellt, um die tatsächliche Leistungsfähigkeit der Modelle zu beurteilen. Abb. 31 gibt einen Überblick

über das Gelände und zeigt einen Vergleich zwischen realer Aufnahme und nachmodellierter virtueller Welt.

Im Folgenden wird näher beschrieben, wie diese mit Hilfe der in Kapitel 4.2 ausgewählten *Presagis* Simulationsumgebung aufgebaut wurde. Als Datenbank zur Speicherung diente das CDB Format (engl.: *Common Database*), da dieses im Vergleich zu anderen Formaten die höchste Flexibilität aufweist und nativ von der Presagis Umgebung unterstützt wird. Das Format basiert auf der Kombination mehrerer Modellierungsschichten. In Abb. 32 sind einige dieser Schichten dargestellt. Die Geodaten stammen dabei von der Bayerischen Vermessungsverwaltung [225]. Alle Schichten sind georeferenziert und zeigen als Ausschnitt einen ca. 2 km × 2 km großen Bereich um das Gelände der Bundeswehr Universität München, der in der Simulation nachmodelliert wurde. Grundlage dieses Schichtaufbaus bilden die Elevationsdaten mit einer Gitterweite von 1 m. Sie enthalten ausschließlich die Geländeoberfläche und wurden von Störeinflüssen wie z.B. Gebäuden oder Vegetation bereinigt. Das resultierende Geländemodell wird schließlich mit dem Luftbild mit einer Bodenauflösung von 20 cm pro Pixel (cpp) überlagert, welches die Basis für die Darstellung der Bodentextur bildet.

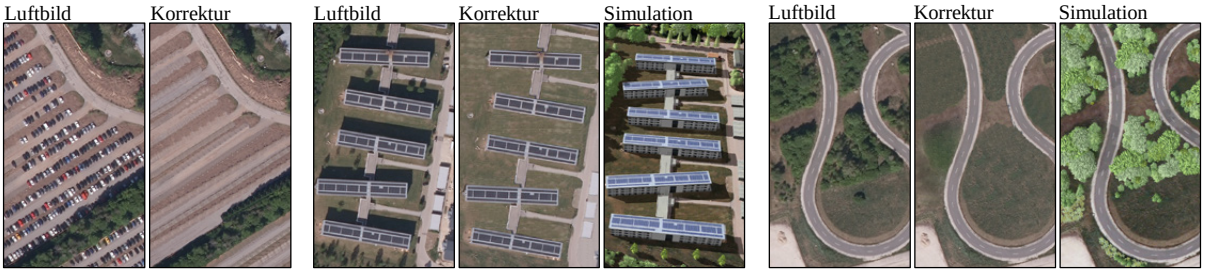
Abb. 31 Gegenüberstellung eines real aufgenommenen Luftbildes (links) und einer mit Hilfe der *Presagis Modelling and Simulation Suite* nachgestellten synthetischen Aufnahme (rechts). Der Ausschnitt zeigt das Gelände der Bundeswehr Universität München mit dem im oberen Teil angrenzenden Testfluggelände. Beide dienen als Basis für die Erstellung der in dieser Arbeit verwendeten Bilddatensätze.

a) Elevationsdaten b) Luftbild c) Materialklassifizierung d) Vektordaten

Abb. 32 Überblick über verschiedene Schichten, die im CDB Format als Grundlage für die Erstellung und spätere Visualisierung der virtuellen Welt dienen. (Geobasisdaten: Bayerische Vermessungsverwaltung)

Bei einem derartigen Aufbau der Terrainmodellierung ist je nach Anwendungsfall und Flughöhe zur Steigerung der Modellierungsqualität und -genauigkeit eine Nachbearbeitung des Luftbildes sinnvoll. Abb. 33 zeigt einen Überblick über die in der vorliegenden Arbeit angewandten Verbesserungen. Im ersten Schritt wurden sämtliche auf dem Luftbild sichtbaren Fahrzeuge retuschiert. Da die virtuelle Welt

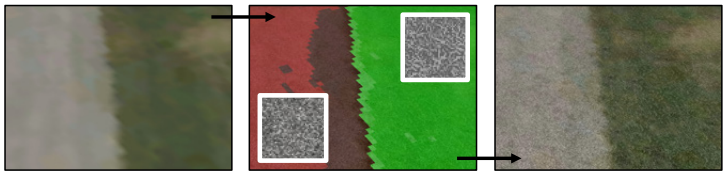
zur Generierung von Trainings- und Testdaten für den Anwendungsfall der UAV-basierten Fahrzeugdetektion dient, ist dies unbedingt nötig, um Störeinflüsse durch nicht gelabelte und perspektivisch nicht korrekt dargestellte Fahrzeuge ausschließen zu können. Außerdem sollen real erflogene Sensoraufnahmen möglichst detailgetreu in der Simulation nachgebildet werden. Aus diesem Grund wurde im Bereich des Testfluggeländes der Schattenwurf der Gebäude aus dem Luftbild retuschiert. Dies gewährleistet, dass in den synthetischen Bilddaten ausschließlich der durch die Simulationsumgebung erzeugte und an die Tageszeit angepasste Schattenwurf sichtbar ist. Die weiterhin sichtbaren Gebäudeumrisse stellen keine Beeinträchtigung dar, da sie durch die im weiteren Verlauf hinzugefügten 3D-Modelle überdeckt werden und Anhaltspunkte für die korrekte Positionierung von diesen liefern. Eine Nachbearbeitung in Bezug auf die im Luftbild sichtbare Vegetation ist hingegen vor allem im Bereich des Testfluggeländes durchaus sinnvoll, da es ansonsten bei bodennahen Aufnahmen aufgrund der Perspektive vorkommen kann, dass die vom Luftbild stammende Vegetation unter den 3D-Volumenbäumen durchscheint.



a) Nachbearbeitung Fahrzeuge b) Nachbearbeitung Schattenwurf c) Nachbearbeitung Vegetation

Abb. 33 Darstellung der verschiedenen Nachbearbeitungsstufen und Retuschen des Luftbildes vor seiner Verwendung als Bodentextur bei der Erstellung der virtuellen Welt

Eine weitere Schicht in der CDB Struktur enthält die Materialklassifizierung des Untergrunds (s. Abb. 32 c)). Die verwendete Softwareumgebung bietet eine Reihe von vordefinierten Materialen, wobei auch Schichtdicken und Materialzusammensetzungen berücksichtigt werden. Anhand des Luftbildes bzw. vorverarbeiteter Versionen von diesem wird in *Terra Vista* eine farbbasierte Klassenzuordnung erstellt, die im Anschluss weiter verfeinert werden kann. Die fertige Materialklassifizierung bildet einerseits die Grundlage für die physikalisch basierte Infrarotsimulation. Andererseits wird sie aber auch bei der Simulation elektro-optischer Sensordaten verwendet, um das bei geringen Flughöhen tendenziell niedrig aufgelöste Luftbild mit feinen, halbtransparenten Strukturen, den sogenannten *Hypertexturen (HT)*, zu überlagern. Diese bilden strukturelle Feinheiten der Oberfläche im Bereich von Asphalt oder Grasflächen nach und verbessern so vor allem bodennahe Aufnahmen. In Abb. 34 ist dieser Vorgang mit entsprechenden Beispielbildern visualisiert.



Luftbild Materialklassifizierung Luftbild mit HT

Abb. 34 Überlagerung des Luftbildes mit *Hypertexturen* (HT) zur verbesserten Simulation feiner Bodenstrukturen

Die CDB Struktur sieht außerdem eine Schicht zur Platzierung von Lichtpunkten z.B. zur Nachbildung der Start- und Landebahn-Befeuerung vor, welche aber für den vorliegenden Untersuchungen keine Rolle spielt. Ein weiterer Bestandteil des Schichtaufbaus sind Vektordaten. Diese können punkt-, linien- oder flächenförmige Vektoren und die dazugehörigen Eigenschaften enthalten. Sie repräsentieren dabei unter anderem die vorkommenden 3D-Modelle, wie z.B. Gebäude, Fahrzeuge oder Vegetation, aber auch die Nachbildung von Straßenverläufen, Parkplätzen oder ähnlichen Untergründen durch künstliche Texturen an Stelle des Luftbildes. Eine dieser Vektordateien enthält die im Kartenausschnitt vorkommende Vegetation, die durch 3D-Volumenbäume von *SpeedTree* [137] simuliert wird. Liegt der Fokus

auf einer möglichst realistischen Nachmodellierung, wie beispielsweise bei der Generierung realer und synthetischer Bildpaare, können diese einzeln mit Punktvektoren in der virtuellen Welt platziert werden. Zur Generierung ganzer Waldstücke bzw. zur Nachbildung dicht bewachsener Wiesenstreifen oder Gebüsch, bietet *Terra Vista* ein Streuwerkzeug, das innerhalb eines definierten Polygons ausgewählte Modelle mit einer vorgegebenen Dichte automatisch platziert. Abb. 35 zeigt beispielhaft zwei verschiedene Baummodelle und eine Auswahl an Gras und Strauchtypen. Verschiedene Versionen ermöglichen es dabei, die Vegetation an die vorherrschende Jahreszeit anzupassen.

Abb. 35 Beispielhafte Darstellung der verwendeten 3D-Volumenbäume von *SpeedTree* zu verschiedenen Jahreszeiten und Auswahl an Gras und Strauchmodellen von *SpeedGrass* zur Simulation der Bodenvegetation (vgl. [137]).

Eine weitere Vektordatei beinhaltet die verschiedenen Gebäudemodelle. Die Verwendung generischer Gebäudemodelle war in dieser Arbeit nicht möglich, da gekoppelte reale und synthetische Bilddaten benötigt werden. Aus diesem Grund wurden sämtliche im beschriebenen Kartenausschnitt vorkommenden Gebäude realitätsgetreu nachmodelliert. Als Ausgangspunkt dafür diente die Modelldatenbasis aus [36], die aktualisiert und weiterentwickelt wurde. Die *Presagis* Umgebung bietet für den Zweck der Modellierung das Softwaretool *Creator*. Um eine möglichst hohe Kompatibilität mit der verwendeten Simulationsumgebung und der CDB Datenbasis zu erreichen, wurden alle Modelle ins *OpenFlight* (.flt) Format konvertiert. Jedes 3D-Modell besteht dabei aus einem Gittermodell, das die Form und die Dimension abbildet und einer oder mehrerer Texturen, die auf dieses Gittermodell projiziert werden. Sämtliche Seitentexturen der Gebäude wurden vor Ort aufgenommen, entzerrt und mit Hilfe eines Bildverarbeitungsprogrammes von Störobjekten und Störeinflüssen, wie z.B. Schattenwurf, befreit. Die Dachflächen bestehen aus generischen Texturen, die die Realität möglichst gut widerspiegeln. Alle Texturen wurden im Nachgang in *Creator* materialklassifiziert. Die fertigen 3D-Gebäude werden anschließend in der Vektordatei verlinkt und mit Hilfe des Luftbildes ausgerichtet und orientiert. Durch entsprechende direkte, ambiente und diffuse Beleuchtung und Schattierung wird seitens der Simulationsumgebung eine harmonische Eingliederung der Modelle in die vorherrschenden szenarischen Umgebungsbedingungen sichergestellt.

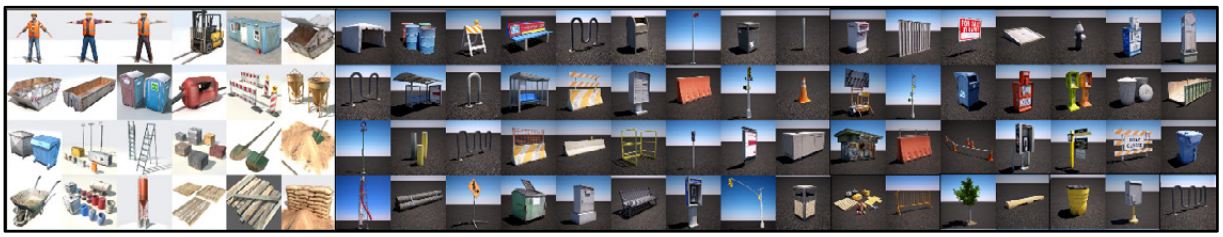


Abb. 36 Übersichtsdarstellung des Modellkatalogs mit Zusatzmodellen und Kleinteilen zur detaillierteren Ausgestaltung der virtuellen Welt und zur Erstellung von Trainingssatzvariationen mit Störobjekten in Fahrzeugnähe (vgl. [226]).

Abb. 37 Ausschnitt aus dem Modellkatalog zur Simulation verschiedener Fahrzeugmodelle (vgl. [227]).

Neben den Gebäudemodellen werden zur detaillierten Nachbildung der realen Gegebenheiten weitere Zusatzmodelle und Kleinteile benötigt. Abb. 36 gibt einen Überblick über die verwendeten Modelle aus [226], die ebenfalls ins *OpenFlight* Format konvertiert wurden und ähnlich wie die Gebäude statisch über eine Vektordatei platziert werden können. Des Weiteren wird diese Art von Modellen im späteren Verlauf der Arbeit zur Generierung von Trainingsdatensatzvariationen (s. Kapitel 5.3.2) benötigt, um den Einfluss von zufällig in Fahrzeugnähe verteilter Störobjekte untersuchen zu können. Insgesamt stehen in dieser Kategorie 154 verschiedene Modelle zur Verfügung.

Zur realistischen Simulation von Fahrzeugen für die Trainings- und Testdatengenerierung ist ebenfalls ein entsprechendes 3D-Modelldatenset mit der benötigten Variation in Bezug auf Farbe, Form und Fahrzeugmodell nötig. Abb. 37 zeigt einen Ausschnitt der insgesamt 148 Modelle aus [227], die dabei zum Einsatz kamen. Sie bestehen aus einem Gittermodell mit durchschnittlich rund 3000 Polygonen und einer UV-gemappten Textur. Dieser Aufbau bietet durch Umfärben der Textur in einem Bildverarbeitungsprogramm eine effektive Möglichkeit zur Generierung weiterer Farbvariationen für jedes Modell, wodurch das Datenset auf insgesamt 466 Modelle erweitert werden konnte. Neben der statischen Platzierung über Vektordateien bei der Generierung der virtuellen Welt, werden über die Programmierschnittstelle von *Vega Prime*, das zur späteren Visualisierung der fertigen CDB Datenbasis dient, verschiedene Objekte bzw. Fahrzeugmodelle während der Laufzeit dynamisch platziert. Im nächsten Abschnitt wird beschrieben, wie dabei auch eine automatisierte Generierung der *Ground Truth* in Form von *Bounding Boxen* um das Fahrzeug implementiert werden kann.

5.1.2 Generierung der Bounding Boxen und Ground Truth in der virtuellen Szene

Um Aufnahmen aus der beschriebenen virtuellen Welt als Trainings- und Testdaten für den Anwendungsfall der UAV-basierten Fahrzeugdetektion nutzen zu können, wird für jedes im Bild enthaltene Fahrzeug die zugehörige Annotation in Form einer *Bounding Box* benötigt. Vorteil der Verwendung einer virtuellen Umgebung ist unter anderem, dass diese *Bounding Box* und weitere Annotationen automatisiert und reproduzierbar erzeugt werden können und somit das zeitaufwendige händische Labeln entfällt. Da nicht jede Simulationsumgebung bereits von vornherein diese Information zur Verfügung stellt, wird im Folgenden eine allgemeine Methode zur Implementierung dieser Funktionalität beschrieben. Wie bereits in Kapitel 4.3.1 erwähnt, werden dabei für die hier vorgestellten Untersuchungen vorerst keine orientierten *Bounding Boxen* oder semantische Maskierungen der Objekte betrachtet.

Abb. 38 Visualisierung des Prozesses zur automatisierten *Bounding Box* Generierung auf Basis der 3D Hülle um das zu detektierende Objekt und einer Transformation der Eckpunkte von Welt- in Kamerakoordinaten (vgl. [51]).
Lat: Latitude; Long: Longitude; h: Höhe

Eine der in der Literatur beschriebenen Methoden für *Game-Engines* [33] nutzt die Kommunikation mit der Grafikhardware und generiert im ersten Schritt aus dem *Stencilbuffer* eine semantische Aufteilung der Objektklassen im Bild. Mit Hilfe des *Tiefenbuffers* wird anschließend zwischen den individuellen Objekten unterschieden und aus dem jeweiligen Umriss die gesuchte *Bounding Box* berechnet. Im Gegensatz dazu werden bei der hier beschriebenen Methode, ähnlich wie in [101], lediglich direkte Informationen wie Objektposition und -größe benötigt. Dadurch ist sie für nahezu alle gängigen Simulationsumgebungen anwendbar und spart die Implementierung einer Schnittstelle zur Grafikhardware.

In Abb. 38 ist der Annotierungsprozess visualisiert dargestellt. Nach dem *Culling* und *Rendering* wird am Ende der Bildgenerierungspipeline eine Liste an Objekten abgefragt, die sich zum aktuellen Zeitschritt im Blickfeld (engl.: *Field of View*, FOV) der Kamera befinden. Da Verdeckungen (*Occlusion Culling*) dabei nicht von jeder Simulationsumgebung berücksichtigt werden, wird auf das Zentrum eines jeden in der Liste enthaltenen Objekts der Reihe nach ein virtueller Laserstrahl ausgerichtet, der häufig auch als *Isector* bezeichnet wird. Durch Auslesen seines Schnittpunktes kann zum einen überprüft werden, ob der Objektmittelpunkt durch Hindernisse verdeckt wird und zum anderen ist eine Bestimmung der Entfernung zwischen Objekt und virtueller Kamera möglich, ohne dass dafür eine Tiefenkarte zur Verfügung stehen muss. Über die aktuelle Position in Weltkoordinaten, die zugehörige Orientierung und die Objektdimensionen ist nun die Berechnung einer dreidimensionalen Begrenzungsbox möglich. Mit Hilfe der Kameramatrix, die in jedem Simulationsschritt neu berechnet wird, können nun die Eckpunkte der Begrenzungsbox vom dreidimensionalen Weltkoordinatensystem ins zweidimensionale Kamerakoordinatensystem transformiert werden. Das kleinste, diese acht transformierten Eckpunkte einschließende Rechteck bildet eine ausreichend genaue und reproduzierbare *Bounding Box* und kann als *Ground Truth* für die Objektdetektion verwendet werden. Um darüber hinaus bei der späteren Analyse auch den Einfluss weiterer Parameter auf die Detektionsleistung untersuchen zu können, werden für jedes Bild zusätzliche Informationen aus der Simulationsumgebung abgespeichert. Nachfolgende Auflistung zeigt den Aufbau der Annotationsdatei, die für jedes Bild automatisiert generiert wird.

Tab. 12 Aufbau der Annotationsdatei, die für jedes synthetisch generierte Bild automatisiert erzeugt wird und neben den *Bounding Boxen* der vorkommenden Objekte auch allgemeine Annotationen enthält.
UTM (engl.: *Universal Transverse Mercator*) Koordinatensystem; BB: *Bounding Box*; l.o. linke obere Ecke als Ausgangspunkt

Allgemeine Annotationen																
Geometrische Eigenschaften							Umgebungsbedingungen									
x (UTM)	*y (UTM)*	*Höhe (UTM)*	*Orientierung*	*Radius*	*Blickwinkel*	*Auflösung x / y*	*Uhrzeit*	*Jahr*	*Monat*	*Tag*	*Schatten I/O*	*Schattenqualität*	*Sichtweite*	*Bewölkung*	*Rauschanteil*	
Objektannotationen (für jedes im Bild vorkommende Fahrzeug)																
Objektdaten							Objektkontext									
Klasse	*Farbe*	*BB x / y (l. o.)*	*BB Breite*	*BB Höhe*	*Fahrzeugmodell*	*Orientierung*	*x (UTM)*	*y (UTM)*	*Höhe (UTM)*	*Verdeckung*	*Distanz*	*Untergrund*				

Die ursprüngliche Annotationsform beschreibt die *Bounding Boxen* anhand der (x, y) Pixelkoordinaten der linken oberen Ecke und der zugehörigen Breite und Höhe in Pixeln. Um für Trainings- und Evaluierungszwecke konform mit der vom YOLOv3 Detektor geforderten Notation zu sein, wird für jedes synthetisch generierte Bild im Nachgang aus der ursprünglichen Annotationsdatei eine Textdatei generiert, die für alle im Bild vorkommenden Fahrzeuge folgende Beschreibung der *Bounding Boxen* auflistet:

<Objektklasse> <x> <y> <Breite> <Höhe>

Das YOLO-Format beschreibt dabei als (x, y) Koordinaten das Zentrum der *Bounding Box* und enthält für x, y, Breite und Höhe relative Werte im Gegensatz zu absoluten Pixelangaben. Dies hat den Vorteil, dass die Angaben unabhängig von der Bildauflösung sind.

Fazit

Insgesamt ermöglicht dieses Vorgehen die schnelle Generierung einer großen Anzahl synthetischer Daten mit automatisch erzeugten Annotationen, welche nicht nur *Bounding Boxen* um die vorkommenden Fahrzeuge beinhalten, sondern auch weitere allgemeine Bild-, Umgebungs- und Objektparameter aus der Simulationsumgebung extrahieren.

5.1.3 Schema und Implementierung zur synthetischen Datengenerierung

Nach den Beschreibungen zur Modellierung der virtuellen Szene und zur Implementierung der automatisierten Annotationsgenerierung wird in diesem Kapitel auf die Erzeugung synthetischer Trainings- und Testdatensätze und davon abgeleiteter Variationen mit unterschiedlichen Parameterverteilungen zur Untersuchung des Trainingsverhaltens eingegangen. Für diesen Zweck wird die Programmierschnittstelle der *Presagis* Umgebung genutzt, die über das *Vega Prime SDK* (engl.: *Software Development Kit*) Zugriff auf die Gestaltung der Visualisierungsparameter und des Szenengraphen bietet. Sämtliche dafür erstellte *Plug-Ins* und Softwarebestandteile sind aufgrund der Kompatibilität und der Geschwindigkeitsanforderungen in C++ erstellt. Abb. 39 zeigt die einzelnen Bestandteile der Implementierung.

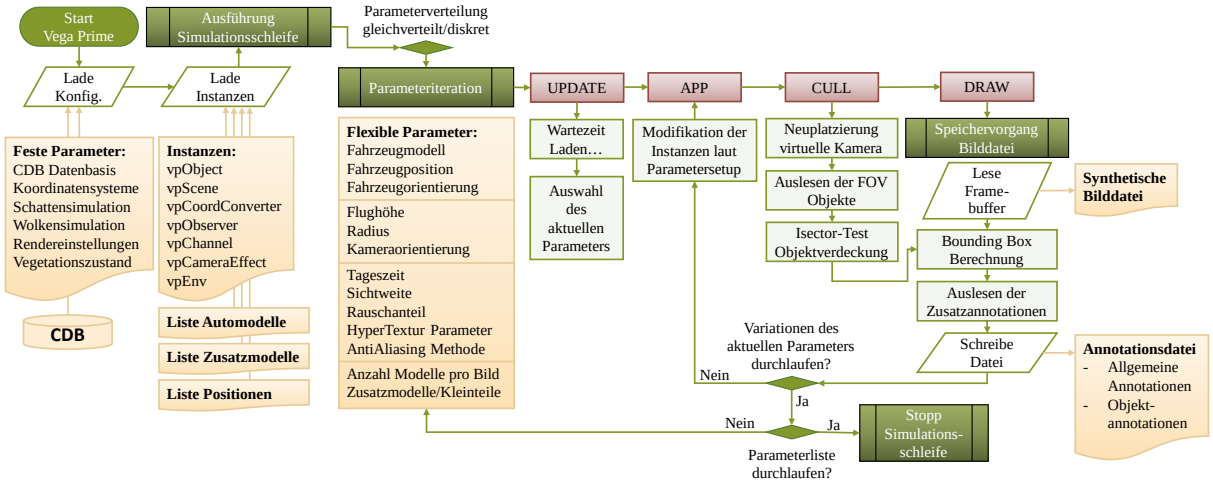


Abb. 39 Ablaufdiagramm zur Visualisierung des Programmablaufs bei der Erzeugung eines synthetischen Datensets über die Programmierschnittstelle der Simulationsumgebung

Im ersten Schritt wird dabei eine *Vega Prime* Instanz zur Visualisierung ausgeführt, welche eine initiale Startkonfiguration lädt. Diese Startkonfiguration kann im Vorfeld über eine grafische Benutzeroberfläche erstellt werden und enthält feste Parameterwerte zur Initialisierung der virtuellen Umgebung, wie z.B. die zu betrachtende CDB Datenbasis oder verschiedene vordefinierte Einstellungen bzgl. Schatten-, Wolken- oder Vegetationssimulation. Um während der Laufzeit Einfluss auf weitere veränderbare

Parameterwerte nehmen zu können, werden in nächsten Schritt die zugehörigen Instanzen der Simulationsobjekte geladen, bevor schließlich die Simulationsschleife gestartet wird.

Die Simulationsschleife besteht aus mehreren Unterprozessen, die in Abb. 39 rot dargestellt sind. Das Hauptaugenmerk liegt dabei auf der schrittweisen Iteration über bestimmte Parameter zur Generierung einer Variation im Datensatz. Bei der Auswahl der möglichen Parametervariationen wurden die in Kapitel 4.1 beschriebenen speziellen Anforderungen an das Datenmaterial für die Fahrzeugdetektion auf Luftbildern berücksichtigt. In Kapitel 5.1.4 wird genauer auf die verwendeten Parameter und die Wertebereiche eingegangen, die als Ausgangspunkt zur Erzeugung eines synthetischen Referenzdatensatzes in dieser Arbeit verwendet werden. Kapitel 5.3.2 beschreibt darauf aufbauend weitere Parametervariationen zur Untersuchung des Trainingsverhaltens in Abhängigkeit der Datensatzerzeugung. Die Verteilung der einzelnen Parameterwerte kann dabei sowohl gleichverteilt oder diskret sein. Eine diskrete Parameterverteilung wirkt sich multiplikativ auf die Größe des generierten Datensets aus, da sowohl die einzelnen Parameter als auch die zugehörigen diskreten Parameterwerte der Reihe nach durchlaufen werden. Parameter mit gleichverteilten Werten können entweder parallel zur beschriebenen Iteration in jedem Schritt einen zufälligen Wert annehmen oder es wird ihnen wiederum eine bestimmte Anzahl an Durchläufen zugewiesen, bei der sie ebenfalls einen zufälligen Wert annehmen, dabei dann aber wiederum Einfluss auf die Datensatzgröße nehmen.

Über die zugewiesenen Objektinstanzen werden am Anfang der Simulationsschleife die jeweiligen Parameter in der Simulation eingestellt. Beim nachfolgenden Cull-Prozess wird die virtuelle Kamera neu ausgerichtet, bevor als Basis für die *Bounding Box* Generierung alle Objekte im aktuellen FOV ausgelesen werden. Der Draw-Prozess rendert die aktuelle Szenerie, die anschließend durch Abgreifen des Framebuffers als synthetische Bilddatei abgespeichert werden kann. Nach der Berechnung der *Bounding Boxen* gemäß dem in Kapitel 5.1.2 vorgestellten Schema wird abschließend die zugehörige Annotationsdatei generiert. Um Interferenzen zu vermeiden und da die Bildwiederholrate für diese Anwendung eine untergeordnete Rolle spielt, werden die einzelnen Prozesse der Simulationsschleife anders als sonst üblich nicht parallel, sondern sequenziell ausgeführt. Nach jedem Durchlauf wird abschließend entweder der Wert des aktuellen Parameters geändert oder mit dem nächsten Parameter begonnen, bevor schließlich nach Durchlauf aller Variationen die Simulationsschleife und damit das Programm beendet wird.

Fazit

Dieser Ablauf kann auf ein beliebiges Simulationsprogramm zur Generierung synthetischer Datensätze übertragen werden und ermöglicht durch Anpassung der Parameterzusammensetzung den erforderlichen Gestaltungsspielraum. Eine gleichverteilte oder diskrete Wahl der Parameterwerte erlaubt zusammen mit der Festlegung der Stufenanzahl eine Einflussnahme auf die Größe des Datensatzes und vor allem auf den Anteil, den ein bestimmter Parameter an der Gesamtvariation des Datensatzes hat.

5.1.4 Parameterverteilung und explorative Datenanalyse

Im Folgenden wird nun auf Basis des beschriebenen Generierungsschemas diejenige Parameterverteilung vorgestellt, die als Ausgangspunkt für die Erzeugung eines synthetischen Referenzdatensatzes dient.

In verschachtelten Schleifen werden jeweils die Parameter Fahrzeugmodell, Fahrzeugposition, Flughöhe, Radius und Orientierung der virtuellen Kamera geändert, um beim Training ein möglichst breites Spektrum der später vorkommenden Bedingungen zu erfassen. Tab. 13 gibt einen Überblick über die Spanne der Parameterwerte und deren Verteilungen für Trainings- und Testdatenset, wobei im Folgenden als erstes die Trainingsdatengenerierung beschrieben wird. Die Auswirkungen der einzelnen Variationen sind in Abb. 40 anhand von Beispielbildern dargestellt.

Tab. 13 Tabellarische Übersicht über die Parameterverteilung bei der Generierung des synthetischen Referenzdatensatzes für Trainings- und Testzwecke. Der Trainingsdatensatz enthält 86 640 *Bounding Boxen* in 93 312 Bildern, der Testdatensatz 9527 *Bounding Boxen* in insgesamt 9719 Bildern. Das Verhältnis von Trainings- zu Testdaten beträgt dabei 10:1.
Orient.: Fahrzeugorientierung; p.B.: pro Bild; p.S.: pro Schritt; gl.vert.: gleichverteilte Parameter

	Fahrzeugparameter			**Geometrische Parameter**			**Umgebungsparameter**		
	Modelle	Positionen	Orient.	Flughöhe	Radius	Orient.	Uhrzeit	Sichtweite	Rauschen
Train	*80 (38); 1 p. B.* diskret	*6* diskret	*0°* diskret	*15, 30, 50, 90 m* diskret	*0, 20, 40, 80 m* diskret	*30° p.S.* diskret	*6 - 18 Uhr* gl.vert.	*0,3 - 30 km* gl.vert.	*0 - 0,15* gl.vert.
Test	*80 (38); 1 p. B.* diskret	*4* diskret	*0 - 359°* gl.vert.	*15 - 100m* gl.vert.	*0 - 80m* gl.vert.	*0 - 359°* gl.vert.	*6 - 18 Uhr* gl.vert.	*0,3 - 30 km* gl.vert.	*0 - 0,15* gl.vert.

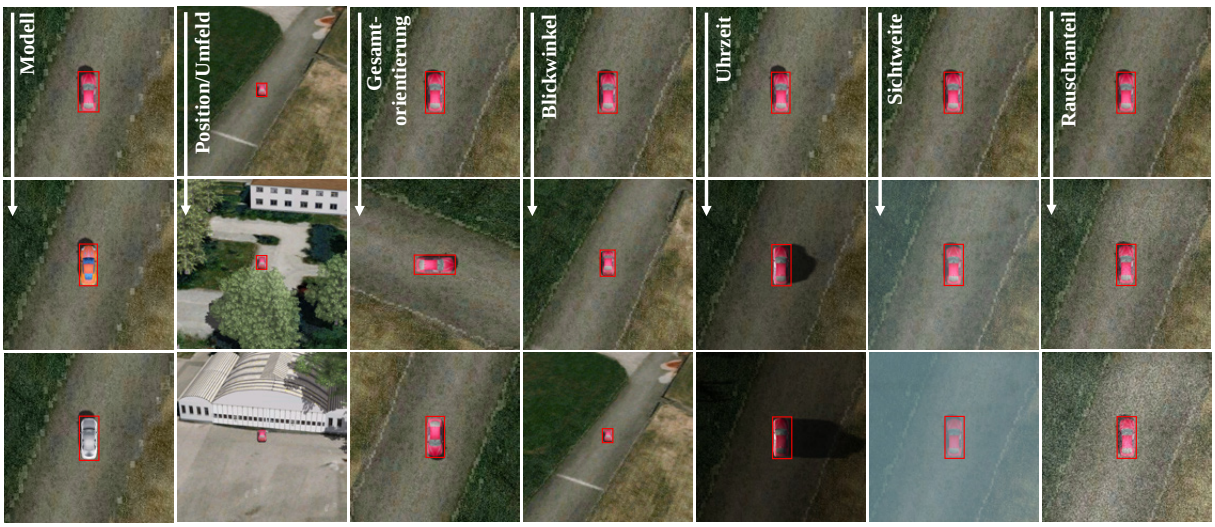


Abb. 40 Verschiedene Beispielbilder aus dem Generierungsprozess des synthetischen Referenzdatensatzes. In den einzelnen Spalten sind exemplarisch die Auswirkungen der jeweiligen Parametervariationen zu sehen.

Im ersten Schritt wird über die 3D-Fahrzeugmodelle iteriert. Für das Referenzdatenset wurden 38 verschiedene 3D-Modelle ausgewählt. Auch wenn in diesem Zusammenhang häufig von Fahrzeugen gesprochen wird, wird hier und auch in der späteren Auswertung ausschließlich die Klasse „Auto“ betrachtet, da diese im Vergleich zu den Klassen „Bus“ oder „Lastwagen“ auch bei realen Daten stark dominiert und somit repräsentative Ergebnisse liefert. Um eine möglichst hohe allgemeine Anwendbarkeit des synthetisch trainierten Detektormodells zu erhalten, wurden die 38 3D-Modelle entsprechend der weltweiten Farbverteilung bei Autos umgefärbt (s. Abb. 41). Dies resultierte in einer Gesamtanzahl von 80 Automodellen, über die schrittweise iteriert wird, wobei jeweils nur ein Auto im Bild platziert und ein Durchlauf ohne Automodell ergänzt wird, um sicherzustellen, dass bei einer gewissen Anzahl an Bildern im Datensatz keine Fahrzeuge sichtbar sind.

Im nächsten Schleifendurchlauf wird für jedes Modell die Position und damit das Umfeld des Fahrzeugs variiert. Der Trainingsdatensatz betrachtet sechs verschiedene Positionen, wobei darauf geachtet wurde, dass diese so gewählt werden, dass sich sowohl der Untergrund (helles/dunkles Grass, Asphalt mit/ohne Straßenmarkierungen) als auch die Umgebung (Landstraße, Parkplatzfläche, Industriegebiet) verändern. Ein weiterer wichtiger Punkt sind Variationen in Bezug auf verschiedene Objektansichten. Während die Orientierung des Fahrzeugmodells beim Trainingsdatensatz vorerst immer 0° beträgt, werden bei der virtuellen Kamera die Parameter Flughöhe, Radius zum Objekt und Kameraorientierung nacheinander in diskreten Schritten variiert und so quasi die halbkugelförmige Sphäre über dem Objekt abgetastet. Die Schrittweiten sind in Tab. 13 zusammengefasst. Aus der Verrechnung von Fahrzeugorientierung und Kameraorientierung ergibt sich die Gesamtorientierung des Objekts (s. Abb. 40), die in 30 Grad Schritten variiert. Die Kombination aus verschiedenen Flughöhen und Radien zum Objekt führt schließlich zu verschiedenen Blickwinkeln auf das Objekt, die aufgrund des

verschachtelten Aufbaus wiederum für alle Gesamtorientierungen durchlaufen werden (s. Abb. 40). Insgesamt ergeben sich aus diesem Aufbau $81 \cdot 6 \cdot 1 \cdot 4 \cdot 4 \cdot 12 = 93\,312$ Trainingsbilder.

Zur Erhöhung der Variation in Bezug auf sich verändernde Umgebungsbedingungen werden vor dem Abspeichern von diesen die Parameter Uhrzeit, Sichtweite und Rauschanteil in der Simulationsumgebung zufällig für jedes Bild zwischen bestimmten vorgegebenen Werten variiert (s. Tab. 13). Eine Änderung der Uhrzeit in der Simulation beeinflusst dabei nicht nur die Helligkeit im Bild, sondern auch den Schattenwurf und die gesamte Beleuchtungssituation, wie in Abb. 40 zu sehen. Als Rauschen wird weißes Gauß'sches Rauschen verwendet, das den synthetisch gerenderten Bilddaten mit Gewichtungsfaktoren zwischen 0 und 0,15 additiv überlagert wird und ausschließlich die Helligkeitswerte beeinflusst. Abb. 42 zeigt schließlich eine grafische Übersicht über die Parameterverteilung der synthetischen Referenzdatensätze. Es wird deutlich, dass trotz der diskreten Werte für Flughöhe, Kameraorientierung und Radius zum Objekt alle relevanten Blickwinkel und Objektdistanzen abgedeckt werden.

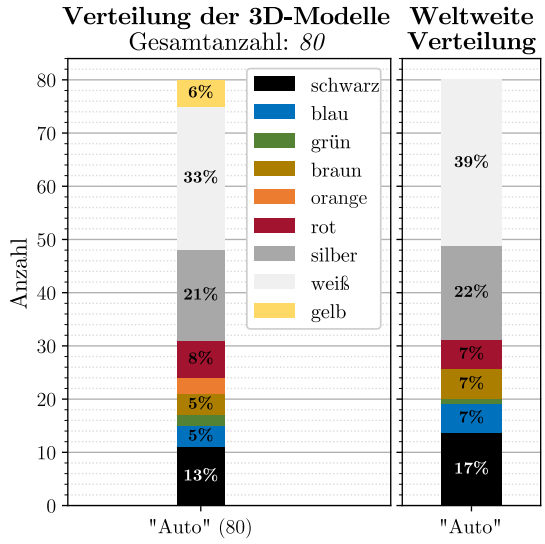


Abb. 41 Farbverteilung der zur Datensatzgenerierung verwendeten 3D-Modelle. Auf der rechten Seite ist als Vergleich dazu die weltweite Verteilung der Autofarben aufgetragen

Der synthetische Referenz-Testdatensatz wird auf ähnliche Art und Weise jedoch mit einer anderen Parameterverteilung erzeugt. Das Modelldatenset bleibt identisch. Um eine aussagekräftige Evaluierung zu ermöglichen und identische Bildpaare bei Training und Test zu vermeiden, wurden bei der Generierung der Testdaten jedoch vier andere Fahrzeugpositionen ausgewählt. Im Gegensatz zum Trainingsdatensatz sind beim Testdatensatz nicht nur die Umgebungsparameter gleichverteilt, sondern auch die Parameter zur Kameraplatzierung. Dies führt dazu, dass sämtliche Objektansichten bei der Evaluierung vorkommen und untersucht werden kann, inwiefern das mit diskreter Schrittweite trainierte Detektormodell auf diese unbekannten Ansichten generalisieren kann. Abb. 42 zeigt in Grün die Verteilung für diese kontinuierliche Parameterwahl. Die daraus resultierenden Werte für die Blickwinkel sind relativ homogen über die gesamte Fläche verteilt und zeigen einen leichten Abfall ab 40°. Die Verteilung der Objektdistanzen zeigt eine klare Häufung bei Werten um 80 m. Beide Charakteristiken sind durchaus realistisch und können auch bei realen Daten in dieser Form auftreten. Der Testdatensatz enthält insgesamt 9719 Bilder, was ca. 10 % der Größe des Trainingsdatensatzes entspricht.

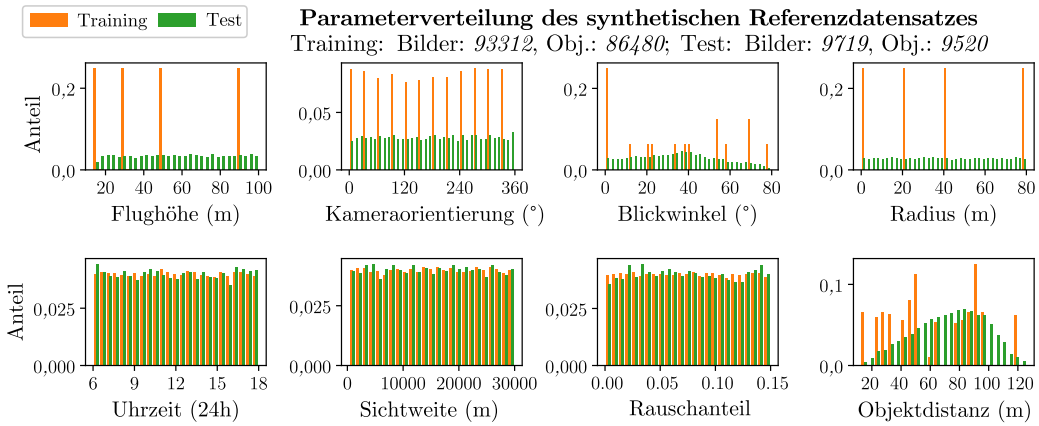


Abb. 42 Grafische Analyse der Verteilung der verschiedenen geometrischen Parameter und der Fahrzeug-, und Umgebungsparameter bei der Erzeugung des synthetischen Referenzdatensatzes für Training und Test. Der Trainingsdatensatz enthält 86 640 *Bounding Boxen* in 93 312 Bildern, der Testdatensatz 9527 *Bounding Boxen* in insgesamt 9719 Bildern. Das Verhältnis von Trainings- zu Testdaten beträgt dabei 10:1 (vgl. [51]).

In beiden Fällen ist für erste Untersuchungen lediglich ein Auto pro Bild vorhanden. Dieses wird aufgrund der einfacheren Umsetzbarkeit stets in Bildmitte platziert. Ein gezielt um Faktor 2 in x- und y-Richtung größer gewählter FOV der virtuellen Kamera erlaubt es nun, dass ein zufällig ausgewählter Bildausschnitt um das Fahrzeug gewählt wird und das Auto somit eine zufällige Position innerhalb des Bildausschnitts einnimmt. Positiver Nebeneffekt dieser Vorgehensweise ist, dass trotz der lediglich sechs bzw. vier festgelegten Positionen des Fahrzeugs das Umfeld und die im Bild enthaltenen Objekte variieren, die Szenerie jedoch die gleiche bleibt. Die finalen synthetischen Sensordaten haben in Anlehnung an das für den realen Vergleich ausgewählte UAVDT Datenset eine Auflösung von 1024 x 540 Pixeln. Abb. 43 zeigt einige repräsentative Beispiele und soll die Zusammensetzung des synthetischen Referenzdatensatzes und der beschriebenen Parametervariationen visualisieren.

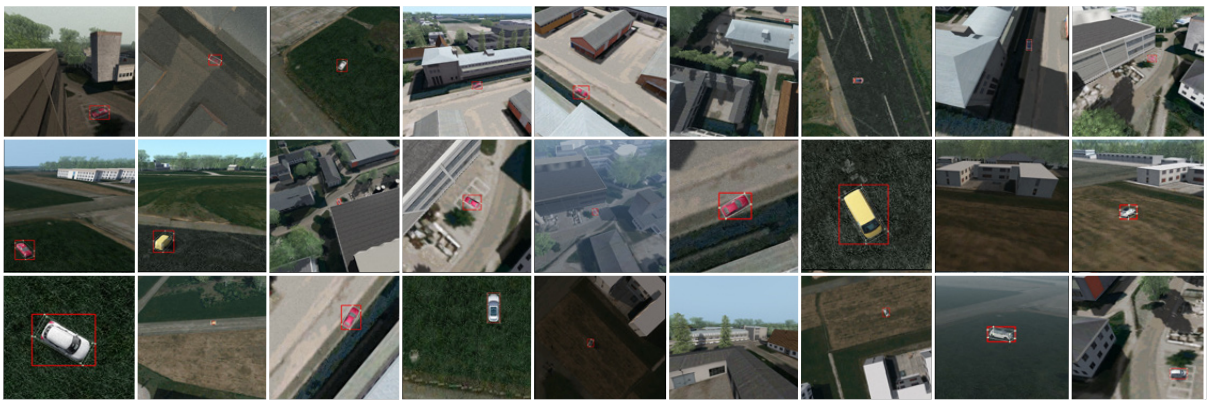


Abb. 43 Repräsentative Darstellung einiger Beispielbilder aus dem synthetischen Referenz-Trainingsdatensatz mit den zugehörigen automatisch generierten Objektannotationen. Die Auswahl spiegelt den Effekt der zugrunde liegenden Parameterverteilung wider und soll einen Eindruck vom visuellen Erscheinungsbild der synthetischen Daten vermitteln.

Zusammenfassung

Die vorgestellten Parametervariationen dienen als Ausgangsbasis und Referenz für die Untersuchungen zum Einfluss synthetischer Sensordaten und werden in Kapitel 5.3 gezielt um zusätzliche Varianten erweitert. Zum Schluss ist erneut die diskrete Verteilung der Parameterwerte im Trainingsdatensatz hervorzuheben, die sowohl bei der realen als auch der synthetischen Trainingsdatengenerierung bisher nur selten angewandt wird. Sie erlaubt jedoch einerseits die relativ entkoppelte Einflussanalyse einzelner Parameter und hilft andererseits, die Variation bzgl. einzelner Parameter zu beschränken und somit eine Überanpassung dahingehend unter Umständen zu vermeiden. Ob bei passender Schrittweite im Sinne

von schrittweisem Lernen der Detektor trotzdem auf alle vorkommenden Parameterwerte generalisiert, kann anhand des Testdatensatzes mit seiner kontinuierlichen Parameterverteilung überprüft werden.

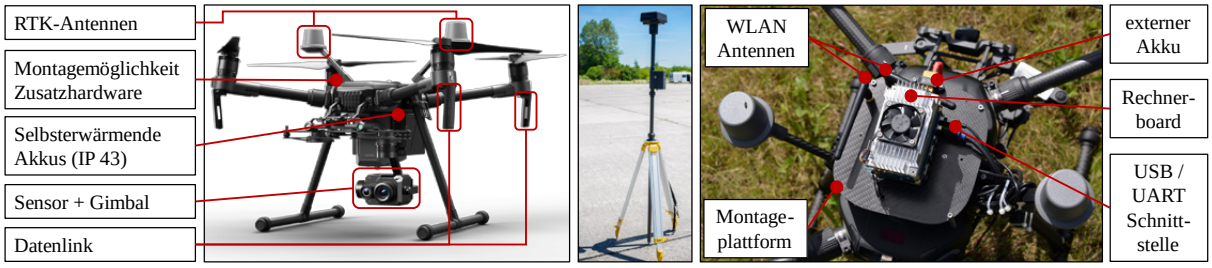
5.2 Durchführung von Drohnenflügen zur Generierung realer und synthetischer Bildpaare

Für eine gezieltere und detailliertere Analyse des *Reality Gaps* sind neben den bereits vorgestellten realen und synthetischen Trainings- und Testdatensätzen vor allem auch inhaltsgleiche reale und synthetische Bildpaare notwendig. Durch die direkte Nachbildung in der Simulationsumgebung und den dadurch nahezu identischen Bildinhalt können sie aufschlussreiche Einblicke über die Leistungsunterschiede und die Einflussfaktoren bei der Verwendung synthetischer Sensordaten liefern. Aus diesem Grund wird im Folgenden der experimentelle Aufbau zur Durchführung der Realflüge, die Parameterbeschreibung im dadurch real erflogenen Datensatz und schließlich die Nachbildung der synthetischen Duplikate beschrieben. Die Generierung eigener realer Flugaufnahmen war notwendig, da für den Anwendungsfall der UAV-basierten Fahrzeugdetektion zum jetzigen Zeitpunkt nur wenig passendes derartiges Bildmaterial zur Verfügung steht. Außerdem stammen die auf diese Weise generierten Bildpaare somit aus derselben geografischen Umgebung wie der bereits beschriebene synthetische Referenz-Datensatz und repräsentieren damit einen typischen späteren Anwendungsfall. Aufgrund dieser Eigenschaft können sie weitere Erkenntnisse liefern, inwiefern eine Erweiterung realer Benchmark-Trainingsdaten mit synthetischen Daten aus derselben Szenerie und Umgebung zu einer höheren Anpassungsfähigkeit der Detektionsmodelle auf die späteren Einsatzbedingungen führt.

5.2.1 Multikopter: Hardware- und Softwaresetup

Zur Erreichung der eben beschriebenen Ziele und zur Erstellung eines darauf angepassten Datensatzes wird ein entsprechendes Multikoptersystem benötigt, mit dem die Erfliegung der realen Luftbildaufnahmen möglich ist. Multikopter eigenen sich aufgrund der hohen Stabilität bei Bewegungen in jede Raumrichtung besonders für die Erfassung von Sensordaten bei geringen bis mittleren Flughöhen [14] und vergleichsweise kurzen Reichweiten und Einsatzdauern. Sie sind daher für den hier vorgestellten Einsatzzweck sehr gut geeignet.

Darüber hinaus werden noch verschiedene weitere Anforderungen an das System gestellt. So muss es zum Beispiel über eine gut dokumentierte Programmierschnittstelle verfügen, die eine Steuerung und optional auch eine Automatisierung des Flugablaufs ermöglicht und darüber hinaus Zugriff auf die aktuellen Telemetriedaten gewährt. Zudem wird eine Integrationsmöglichkeit für ein Sensorsystem mit Gimbal benötigt, um die Kamera auf das Fahrzeugobjekt ausrichten und die entsprechenden Luftbildaufnahmen erstellen zu können. Die dabei verwendete Schnittstelle sollte eine Steuerung der Gimbal- und Kameraparameter erlauben und mit der Programmierschnittstelle des Multikopters kompatibel sein. Zur Ansteuerung dieser Schnittstelle und zur Abspeicherung der Daten wird ein externes Rechnerboard benötigt. Daher ist eine entsprechende Montagemöglichkeit am Kopter erforderlich und die maximal mögliche Zuladung muss so dimensioniert sein, dass sowohl Sensorsystem als auch Hardwareerweiterungen problemlos transportiert werden können. Um Flüge unter verschiedenen Witterungsbedingungen wie z.B. Nebel zu ermöglichen und damit die Varianz im Datensatz zu erhöhen, ist eine gewisse Schutzart bzw. IP-Kennzahl ebenfalls positiv zu bewerten. Als letzte Anforderung ist ein eingebauter GPS-Empfänger mit Echtzeitkinematik (RTK, engl.: *Real Time Kinematik*) zu nennen, der für eine genaue Erfassung der aktuellen Position und Telemetrie nötig ist, welche wiederum eine Voraussetzung für eine entsprechende genaue Erzeugung synthetischer Duplikate darstellt.



a) Multikoptersystem b) RTK c) Hardwareerweiterungen

Abb. 44 Übersicht über die Eigenschaften und Bestandteile des verwendeten Multikoptersystems (DJI M210 RTK V2 mit DJI Zenmuse XT2, für a) vgl. [228]). Rechts sind außerdem die RTK-Bodenstation und die eigenen Hardwareerweiterungen (Nvidia Jetson TX2) zur Durchführung der Flugversuche und zur automatisierten Steuerung des Flugpfades zu sehen.
RTK: *Real Time Kinematik*, UART: *Universal Asynchronous Receiver / Transmitter*

Auf Basis dieser Anforderungen wurden mehrere auf dem Markt verfügbare Multikopter evaluiert. Kleinere Modelle weisen oft nicht die nötigen Integrationsmöglichen und Zuladungen auf. Sehr große Multikopter mit sechs und mehr Rotoren sind hingegen häufig nicht flexibel genug einsetzbar und unterliegen in vielen Fällen aufgrund des höheren Gewichts strikteren gesetzlichen Einschränkungen. Daher wurde der Quadrokopter Matrice M210 RTK V2 vom chinesischen Hersteller DJI für den in dieser Arbeit betrachteten Anwendungsfall ausgewählt. Er besitzt ein vollständig integriertes RTK und kann mit dem Sensorgimbal DJI Zenmuse XT2 erweitert werden. In Tab. 14 sind die zugehörigen technischen Daten aufgeführt. Abb. 44 zeigt die Bestandteile dieses Multikoptersystems, die zugehörige RTK-Bodenstation und die eigenen Hardwareerweiterungen zur reproduzierbaren Missionsplanung und Datenspeicherung.

Tab. 14 Technische Daten des verwendeten Multikopters DJI M210 RTK V2 in Kombination mit dem Sensorgimbal DJI Zenmuse XT2

DJI M210 RTK V2		**DJI Zenmuse XT2**		**DJI RTK Bodenstation**	
Größe	*88 × 88 × 43 cm*	Gewicht	*588 g*	Updaterate	*Bis 20 Hz*
Gewicht	*4,91 kg*	**IR-Kamera**		Einschwingdauer	*< 45 s*
Max. Ladung	*1,23 kg*	Auflösung (IR)	*640 × 64, 30 Hz*	Schutzart	*IP65*
Schwebegenauigkeit (RTK)	*H/V ± 10cm*	Linse, FOV (IR)	*19 mm, 32° × 26°*	Positionierungsgenauigkeit	*H/V ~ 1 - 2 cm*
Max. Steigrate	*5 m/s*	Spektralband (IR)	*7,5 - 13,5 µm*		
Geschwindigkeit	*81 km/h*	**EO-Kamera**			
Flugzeit	*ca. 24 min*	Auflösung (EO)	*4K, 30 Hz*		
Schutzart	*IP43*	Linse, FOV (EO)	*8 mm, 57° × 42°*		
Detektoren	*FPV, IR, US*				

Die Hardwareerweiterungen bestehen aus einer Montageplattform, dem Rechnerboard Nvidia Jetson TX2 und einem von der Stromversorgung des Multikopters unabhängigen externen Akku. Abb. 45 zeigt eine schematische Visualisierung des Gesamtaufbaus mit den jeweiligen Schnittstellen. Das Rechnerboard ist dabei über USB und UART mit dem Multikopter verbunden. Eine ROS (*Robot Operating System*) basierte Interprozesskomunikation ermöglicht nun den Zugriff auf das Onboard-SDK. So können in C++ grundlegende High-Level Funktionalitäten für Kamera- und Gimbalsteuerung, Start- und Landevorgang oder die lokale Positionierung um einen POI (engl.: *Point of Interest*) implementiert werden, was eine reproduzierbare Flugpfadplanung ermöglicht. Das Onboard-SDK dient darüber hinaus zum Abgreifen der Telemetriedaten, wobei hierbei besonderes Augenmerk auf die Synchronisation der Daten gelegt werden muss. Das Rechnerboard wiederum ist über WLAN und das SSH Protokoll mit der Bodenkontrollstation verbunden, die auf diese Weise verschiedene Konfigurationsparameter übermittelt und die Ausführung der Funktionen überwacht.

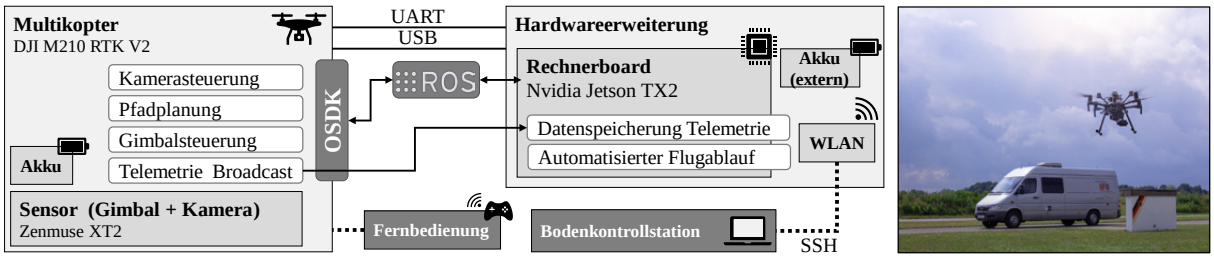


Abb. 45 Schematische Darstellung der Softwareschnittstellen und Hardwarebestandteile, die beim Betrieb und bei der automatisierten Durchführung der Datenerfassungsflüge zum Einsatz gekommen sind (vgl. [229]). Das rechte Bild zeigt den Multikopter bei einem Flug um ein reales Testfahrzeug.
OSDK: Programmierschnittstelle (engl.: *Onboard Software Development Kit*); UART: *Universal Asynchronous Receiver / Transmitter*; SSH: *Secure Shell* Netzwerkprotokoll

Zeitgleich mit der Aufnahme eines jeden Bildes werden auch sämtliche zugehörigen und verfügbaren Telemetriedaten abgespeichert. Sie sind zum einen zur Erstellung der synthetischen Duplikate, aber auch zur Untersuchung der Parametereinflüsse nötig. Tab. 15 gibt einen Überblick über deren Zusammensetzung. Attitude und IMU (Inertiale Messeinheit; engl.: *Inertial Measurement Unit*) beinhalten beide die internen Flugwinkel des Multikopters, wobei letztere bei der Angabe auch eine Messung der Beschleunigungen miteinbeziehen. ATO Höhe bezeichnet die mit Hilfe des Barometers ermittelte Höhe über dem Startpunkt. Außerdem wird die Ausrichtung des Gimbals erfassst. Die nächsten beiden großen Blöcke beschreiben die GPS Daten, bestehend aus Anzahl der verfügbaren Satelliten und Positionierungsangaben, und die ähnlichen, aber genaueren RTK Daten. In Bezug auf den im folgenden Kapitel vorgestellten Flugpfad werden des Weiteren der aktuelle Radius zum POI bzw. zum Testfahrzeug und die Drehrichtung, mit der der halbkreisförmige Flugpfad abgeflogen wird, abgespeichert. Die in Orange markierten Einträge bilden die Grundlage für spätere Positionierung und Ausrichtung der virtuellen Kamera in der synthetisch nachmodellierten Welt zur Generierung der gekoppelten Bildpaare, die auf diese Weise den gleichen inhaltlichen Ausschnitt wie die realen Aufnahmen zeigen.

Tab. 15 Auflistung aller Telemetriedaten, die während der Durchführung der realen Flüge zur Datenerfassung über die Programmierschnittstelle des Multikopters abgegriffen und für jedes aufgenommene Bild abgespeichert werden. In Orange sind diejenigen Daten hervorgehoben, die zur Nachbildung der synthetischen Duplikate verwendet werden.
IMU: Inertiale Messeinheit (engl.: *Inertial Measurement Unit*); ATO: Höhe über Startpunkt (engl.: *Above Take-Off*); GPS: *Global Positioning System*; RTK: *Real Time Kinematik*

Telemetriedaten											
Intern				GPS			RTK			Flugpfad	
Attitude $\varphi / \theta / \psi$	*IMU* $\varphi / \theta / \psi$	*ATO Höhe*	***Gimbal*** $\varphi / \theta / \psi$	*GPS Satelliten*	*GPS Lat/Long*	*GPS Höhe*	***RTK Lat/Long***	***RTK Höhe***	*RTK* ψ	*Radius*	*Drehrichtung*

Fazit

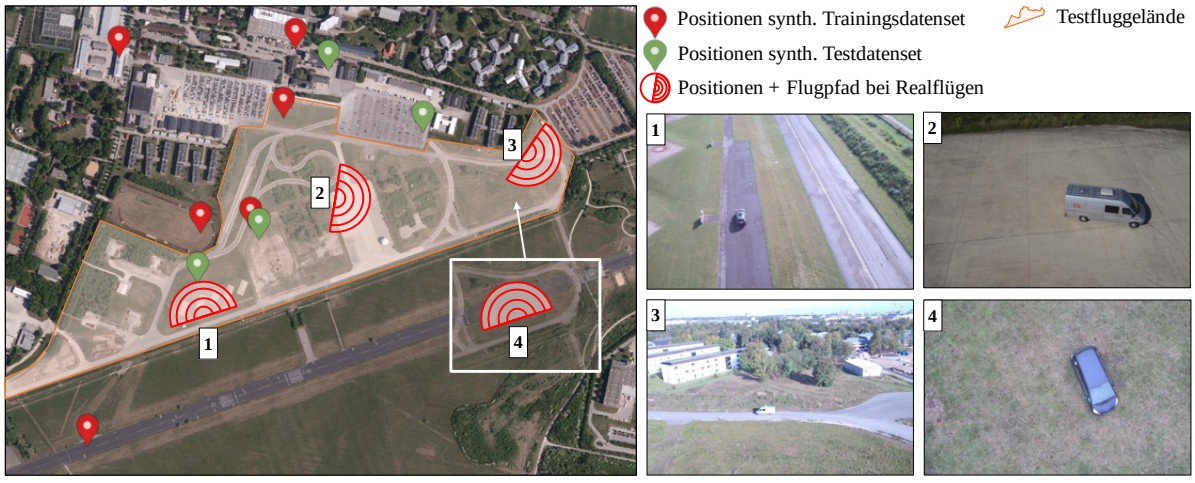
Insgesamt erlaubt das Multikoptersystem zusammen mit den beschriebenen Erweiterungen die reproduzierbare Ausführung eines bestimmten Flugpfades zur Erfassung realer Luftbildaufnahmen von Fahrzeugen mit der nötigen Varianz. Das externe Rechnerboard erhöht in Verbindung mit der Programmierschnittstelle der Drohne und der Steuerung über High-level Funktionen die Flexibilität des Systems. Der Zugriff auf die zugehörigen Annotationen und Telemetriedaten ermöglicht in diesem Zusammenhang die Generierung synthetischer Bildduplikate und die Analyse von Parametereinflüssen auf die Detektionsleistung.

5.2.2 Flugplanung und Flugdurchführung

Im ersten Schritt wurde evaluiert, welcher Aufbau, welcher Flugpfad und welche Randbedingungen zur Analyse der in Kapitel 3.1 beschriebenen Forschungsfragen geeignet sind. Im Allgemeinen soll damit untersucht werden, inwiefern sich höherwertige Bildverarbeitungsroutinen bzw. in unserem Fall *deep-learning* basierte Fahrzeugdetektoren auf realen und synthetischen Daten verhalten und welche Bildeigenschaften dabei die Detektionsleistung beeinflussen und einen *Reality Gap* verursachen. Ziel ist die systematische Erfassung von Szenarien mit möglichst entkoppelten Parametern. Eine automatisierte Steuerung soll dabei helfen reproduzierbare Aufnahmen der gleichen Szenerie zu verschiedenen Umgebungsbedingungen und mit verschiedenen Parametern zu erfassen und dadurch die Entkopplung der Parameter zu verstärken.

Auswahl der Aufnahmepositionen

Da die bildbasierte Fahrzeugdetektion im Vordergrund steht und keine Tracking-Algorithmen untersucht werden sollen, wird im Folgenden davon ausgegangen, dass die Testfahrzeuge statisch positioniert sind. Außerdem trägt dies dazu bei, dass für jedes Bild definierbare Parameterzustände vorliegen. Im ersten Schritt wurden nun passende Positionen ausgewählt und ausgemessen, die bei der Durchführung der Flüge als POI für die Ausrichtung des Gimbals dienen.



a) Übersichtskarte b) Reale Beispielbilder

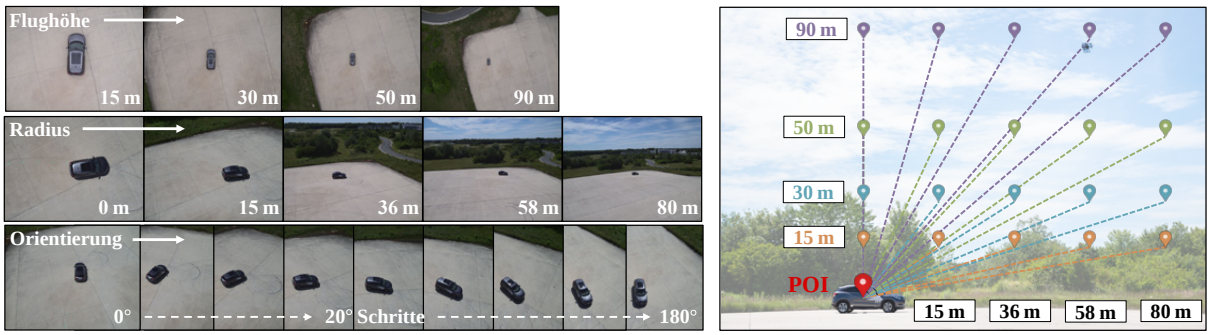
Abb. 46 Vergleich der in Kapitel 5.1.3 zur Erstellung der synthetischen Datensätze verwendeten Positionen mit den Fahrzeugpositionen bei den hier beschriebenen Realflugaufnahmen. Zusätzlich wird der reale Flugpfad durch die maßstabsgetreuen roten Halbkreise repräsentiert. Rechts sind Beispielbilder des Fahrzeugumfelds bei den vier verwendeten Positionen zu sehen. (Geobasisdaten: Bayerische Vermessungsverwaltung)

Abb. 46 zeigt einen Überblick über die Gegebenheiten, wobei das Testfluggelände orange hinterlegt ist. Vier Standorte sind dabei hervorgehoben und zusammen mit dem halbkreisförmigen Flugpfad eingezeichnet. Bei dem Auswahlprozess wurde darauf geachtet, dass sowohl unterschiedliche Untergründe und Straßenformen als auch verschiedene Hintergrundszenarien erfasst werden. In Abb. 46 sind auf der rechten Seite zugehörige Beispielbilder dargestellt. Die Szenen 1-3 sind dadurch charakterisiert, dass sich das Fahrzeug auf befestigtem Untergrund befindet. Dennoch unterscheiden sie sich bzgl. der Art und Farbe des Untergrunds (Teer / Beton), der vorkommenden Straßenform (zweispurig / Parkplatz bzw. mit / ohne Straßenmarkierungen) und dem erfassten Hintergrund (mit / ohne Gebäude bzw. homogener / strukturierter Hintergrund). Bei Szene 4 hingegen liegt der Fokus auf der Erfassung von möglichst wenig Infrastruktur und das Fahrzeug wird auf einer Wiese platziert. Dies ist vor allem deshalb interessant, da dieser Zustand in den gängigen realen Benchmark-Trainingsdaten nicht oder nur sehr unterrepräsentiert abgedeckt wird.

Zusätzlich sind in der Karte von Abb. 46 neben den vier Positionen für die reale Bilderfassung auch diejenigen Positionen eingezeichnet, die bei der synthetischen Trainings- und Testdatengenerierung in der nachmodellierten virtuellen Umgebung verwendet wurden. Diese sind über das gesamte Gelände verteilt, wobei ähnliche oder identische Aufnahmepositionen zur realen Bilderfassung vermieden wurden. Dadurch enthalten die synthetischen Datensätze ähnliche Szenerien aus der gleichen geografischen Umgebung jedoch ohne unmittelbar gleiche Bilddaten und können somit, wie bereits anfangs erwähnt, verwendet werden, um die gezielte synthetische Anpassung von Detektormodellen auf die späteren Einsatzbedingungen zu untersuchen.

Festlegung des Flugmusters

Im zweiten Schritt wurde evaluiert, welches Flugmuster und welche Abstufung in Bezug auf die zu betrachtenden Parametervariationen sinnvoll und zielführend ist. Als Grundlage wurden dazu auch die Überlegungen aus Kapitel 5.1.4 zur Generierung des synthetischen Datensatzes herangezogen. Im Gegensatz dazu wurde als Flugmuster bei den realen Aufnahmen lediglich ein Halbkreissektor gewählt, da dieser dennoch alle Objektansichten erfasst, platztechnisch leichter umgesetzt werden kann und zudem die bei der realen Datenerfassung begrenzten Ressourcen besser ausnutzt. Außerdem gewährleistet dies, dass alle Objektorientierungen in gleicher Anzahl vorkommen.



a) Parameterabstufungen b) Blickwinkel

Abb. 47 Visualisierung der Parameterabstufungen anhand realer Beispielaufnahmen für die Parameter Flughöhe, Radius des geflogenen Halbkreises und Orientierung zum Testfahrzeug. Die rechte Seite zeigt die daraus resultierende Abdeckung der halbkreisförmigen Sphäre um das Testfahrzeug.
(vgl. [229])

Abb. 47 a) zeigt zur Veranschaulichung Beispielbilder der verschiedenen Abstufungen. Ähnlich wie bei der synthetischen Datengenerierung werden Flughöhen bis 90 m betrachtet, wobei der Abstand zwischen den Abstufungen bei zunehmender Höhe größer wird, um gleichmäßig verteilte Objektgrößen zu erfassen. Die Radien bei der halbkreisförmigen Umfliegung des Testfahrzeugs liegen gleichmäßig verteilt zwischen 0 m und 80 m, wobei ersteres einer Drehung des Multikopters an einer festen Position direkt über dem Fahrzeug entspricht.

Abb. 47 b) zeigt die aus diesen Kombinationen resultierenden Blickwinkel und die dadurch erreichte Abtastung der halbkreisförmigen Sphäre über dem Testfahrzeug, wobei das Kamerasystem an jedem Abtastpunkt auf den POI bzw. das zu erfassende Fahrzeug ausgerichtet wird. Um die vorkommenden Fahrzeugorientierungen zu erfassen, wird während der halbkreisförmigen Umfliegungen des statisch platzierten Testfahrzeugs alle 20° ein Kamerabild aufgenommen. Der Kopter schwebt dabei kurzzeitig an der jeweiligen Position, um externe Einflüsse wie z.B. Bewegungsunschärfe oder zeitliche Verzögerungen der Telemetriedaten ausschließen zu können. Neben der Erfassung der verschiedenen Objektorientierungen führt dieses Vorgehen besonders bei niedrigen Blickwinkeln auch zu einer Variation des Hintergrunds. Der so gestaltete Flugablauf wurde auf die Flugdauer des Multikopters angepasst und erfasst 200 Bilder pro Szenerie.

Implementierung

Im letzten Schritt wird nun auf die C++ Implementierung zur automatisierten Durchführung dieses Flugmusters eingegangen. Abb. 48 visualisiert den Ablauf.

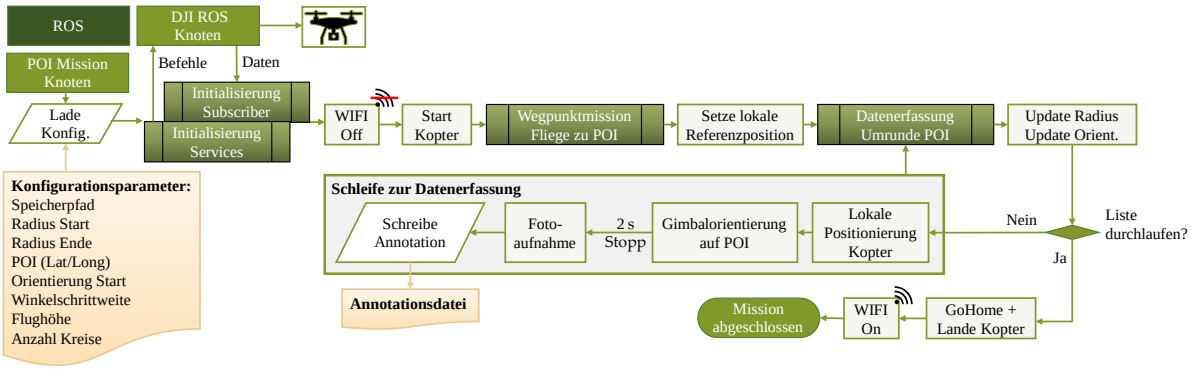


Abb. 48 Ablaufdiagramm zur Visualisierung des Programmablaufs bei der Generierung des real erflogenen Datensets über die Programmierschnittstelle des Multikopters.
ROS: Interprozesskommunikation (*Robot Operating System*)

Die ROS Interprozesskommunikation beinhaltet den DJI ROS Knoten zur Kommunikation mit dem Multikopter und den POI Mission Knoten zur Durchführung des Flugmusters. Nach dem Einlesen der Konfigurationsdatei werden *Subscriber* und *Services* initialisiert, die während dem Flug zum Austausch von Befehlen und Daten genutzt werden können. Sämtliche Programmbestandteile laufen auf dem mit dem Kopter verbundenen Rechnerboard. Um Interferenzen zu vermeiden, wird die WLAN-Verbindung zur Bodenkontrollstation vor dem Start gestoppt. Ein davon unabhängiger Eingriff in die Steuerung des Multikopter durch die Fernbedienung ist dabei aus Sicherheitsgründen jederzeit möglich.

Nach dem automatisierten Start fliegt das System wegpunktbasiert in vorgegebener Flughöhe zum konfigurierten POI. Dort wird die lokale Referenzposition zurückgesetzt und die Schleife zur Datenerfassung startet. In jedem Durchlauf wird dabei der Radius und die Orientierung zur Referenzposition neu bestimmt, der Kopter anhand dieser Daten lokal positioniert und der Kameragimbal auf das Fahrzeug ausgerichtet. Ein Schwebeflug mit einer Dauer von 2 s vor jeder Bildaufnahme garantiert statische Verhältnisse. Zum Auslösezeitpunkt der Kamera werden nun die in Kapitel 5.2.1 beschriebenen Annotationen über die Interprozesskommunikation des Multikopters abgefragt und abgespeichert. Wurden alle Parameter durchlaufen, so fliegt das System automatisch zur Startposition zurück und leitet den Landevorgang ein. Hindernisdetektoren helfen dabei bei der Erfassung der Umgebung. Insgesamt ermöglicht diese Implementierung eine effiziente und vor allem reproduzierbare Datengenerierung an verschiedenen Standorten, mit verschiedenen Testfahrzeugen und unter verschiedenen Umgebungsbedingungen.

Flugdurchführung

Dieser Abschnitt gibt einen kurzen Überblick und Eindruck über die mit dem beschriebenen Setup durchgeführten Realflüge. Die Flugdauer pro automatisiert abgeflogenem Flugmuster beträgt circa 7 Minuten. Tab. 16 listet die einzelnen Flüge mit den zugehörigen Randbedingungen, wie z.B. Position, Testfahrzeug, Datum, Flughöhe und Wetterbedingungen. Insgesamt wurden 91 Flugmissionen durchgeführt, wodurch 4516 Sensoraufnahmen generiert wurden. Dabei wurden die vier beschriebenen Positionen in Kombination mit drei verschiedenen Testfahrzeugen betrachtet. An dieser Stelle ist hervorzuheben, dass identische Kombinationen mehrmals abgeflogen wurden, um dieselben Aufnahmen mit Variationen in Bezug auf Jahreszeiten und Wetterbedingungen zu erfassen. Diese entkoppelte Erfassung verschiedener Parameter bietet weiterführende Möglichkeiten bei der späteren Auswertung. Eine detaillierte Analyse der Parameterverteilung folgt im nächsten Kapitel.

Tab. 16 Übersicht über die durchgeführten Realflüge zur Datenerfassung mit den zugehörigen Randbedingungen. Die Tabelle zeigt jeweils die Position, den Fahrzeugtyp des Testfahrzeugs, das Aufnahmedatum, die Flughöhen, Wetterbedingungen und Aufnahmezeitpunkte und einige relevante Zusatzbemerkungen.
Pos.: Position; Frühjahr; Sommer; Herbst

Pos.	Fahrzeug	Datum	Flughöhen / Wetterbedingungen				Bemerkungen
			15 m	30 m	50 m	90 m	
1	Sprinter	02.09	☁18°; 11:50	☀/☁18°; 12:40	☀/☁18°; 12:15	☀/☁18°; 11:14	
	Sprinter	04.11	☁9°; 09:30	☁9°; 09:55	☁9°; 09:45	☁9°; 09:15	Nasse Fahrbahn
	Sprinter	04.11	☁🌙9°; 16:45	☁🌙9°; 16:15	☁🌙9°; 16:30	☁🌙9°; 16:00	Dämmerung, Fahrzeuglicht
	SUV	02.09	☁12-15°; 9:40	☀18°; 9:50	☀18°; 10:14	☀/☁18°; 10:30	Nasse Fahrbahn, Tau
	SUV	03.11	☁15°; 14:00	☁13°; 13:30	☁13°; 13:45	☁13°; 14:15	Nasse Fahrbahn
	KW	04.11	☁9°; 13:15	☁9°; 12:50	☁9°; 13:05	☁9°; 12:35	Abtrocknende Fahrbahn
	KW	30.03	☀21°; 15:25	☀21°; 15:40	☀21°; 15:50	☀21°; 15:10	
2	Sprinter	03.09	☀20°; 10:30	☀20°; 11:00	☀20°; 10:50	☀20°; 10:13	Teilweise Überblendung
	Sprinter	04.11	☁9°; 11:25	☁9°; 10:55	☁9°; 11:10	☁9°; 10:45	Fleckige Fahrbahn
	SUV	08.07	☀30°; 15:00	☀30°; 15:00	☀26°; 15:00	☀30°; 15:00	Teilweise Überblendung
	SUV	29.09		☁10°; 16:30		☁10°; 15:30	Fleckige Fahrbahn
	SUV	03.11	☁13°; 12:35		☁13°; 12:55		Fleckige Fahrbahn
	KW	04.11	☁9°; 14:40	☁9°; 14:15	☁9°; 14:25	☁9°; 14:00	Abtrocknende Fahrbahn
	KW	06.11			≋7°; 10:55	≋7°; 10:45	Trocken, Hochnebel
	KW	30.03	☀21°; 13:35	☀21°; 13:55	☀21°; 14:05	☀21°; 14:25	
3	Sprinter	01.10	☀16°; 9:45	☀16°; 9:30	☀17°; 10:15	☀17°; 10:40	Tiefstehende Sonne
	SUV	01.10	☀22°; 12:05	☀22°; 11:50	☀22°; 11:40	☀22°; 11:15	
	SUV	31.03	☀15°; 09:00	☀15°; 09:15	☀15°; 09:25	☀15°, 09:40	Tiefstehende Sonne
	KW	30.03	☀17°; 10:20	☀17°; 10:40	☀17°; 10:55	☀17°; 11:10	
4	Sprinter	04.09	☀23°; 12:15	☀23°; 12:05	☀23°; 13:00	☀/☁23°; 11:50	
	SUV	04.09	☀19°; 11:20	☀19°; 11:05	☀19°; 10:30	☀/☁19°; 10:50	
	SUV	31.03	☀17°; 10:15	☀17°; 10:30	☀17°; 10:50	☀17°; 11:05	
	KW	06.11	≋7°; 09:45	≋7°; 09:30	≋7°; 10:00	≋7°; 09:10	Trocken, Hochnebel
	KW	30.03	☀19°; 12:00	☀19°; 12:15	☀19°; 12:30	☀19°; 12:50	

5.2.3 Parametervariationen und Datensatzbeschreibung

Ähnlich wie bereits bei der synthetischen Datengenerierung wird nun auch die Zusammensetzung und die Parameterverteilung im real erflogenen Datensatz, der im weiteren Verlauf der Arbeit häufig mit R-UAV bezeichnet wird, näher beschrieben. Dabei ist hervorzuheben, dass dieser im Gegensatz zu den synthetischen Daten nicht als Trainingsdatensatz konzipiert wurde und daher auch nicht die für das Training notwendige Größe und Gesamtvarianz aufweist. Vielmehr ist das Ziel reale und synthetische Bildpaare mit definierten Parameterzuständen und definierten Szenarien als Testdaten zu generieren, um den Einfluss einzelner Parameter auf die Detektionsleistung für die reale und für die synthetische Domäne zu analysieren und zu vergleichen.

Die mit dem beschriebenen Multikoptersystem generierten Sensordaten haben im Original eine Auflösung von 4000×3000 Pixeln, welche aufgrund einer höheren Vergleichbarkeit mit anderen Datensätzen und aufgrund einer besseren Kompatibilität mit dem Detektornetzwerk auf 1000×750 Pixel reduziert wird. Um Kameraverzerrungen vor allem in den äußeren Bildbereichen zu vermeiden, werden die Bilddaten vor der Weiterverwendung mit den im Vorfeld berechneten Kameraparametern kalibriert. Abb. 49 visualisiert die Vorgehensweise und zeigt, wie der infolge der Kameraverzerrung gebogene Straßenverlauf im oberen Bildteil durch die Kalibrierung korrigiert wird.

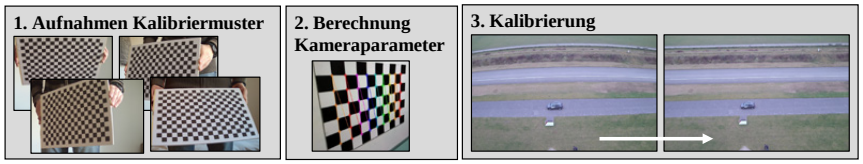


Abb. 49 Vorgehensweise bei der Kalibrierung des Kamerasystems zur Entfernung von Verzerrungen in den Bildrandbereichen

Da bei real aufgenommenen Bildern eine automatische Annotation der Objekte nur schwer realisiert werden kann, muss jedes Fahrzeug händisch mit der entsprechenden *Bounding Box* annotiert werden. Die Annotationsdatei enthält ebenso wie bei den automatisch generierten synthetischen Annotationen die Objektklasse (ausschließlich „Auto“) und die (x, y) Koordinaten des Zentrums der *Bounding Box* mit der zugehörigen Breite und Höhe im YOLO-Format. Um den zeitaufwendigen händischen Annotationsprozess möglichst effektiv zu gestalten, wurden mehrere dafür vorgesehene Programme evaluiert und schließlich das *MakeSense Label Tool* [230] ausgewählt. Ähnlich wie beim UAVDT Datensatz (Abb. 5) werden unbeteiligte Fahrzeuge, die sich zufällig ebenfalls auf dem Testgelände befanden oder im Hintergrund parkten und in der Aufnahme zu sehen sind, mit Hilfe eines händisch positionierten Ausschlussbereichs markiert. Sie werden dadurch bei den Detektionen nicht berücksichtigt und verhindern, dass durch zufällig sichtbare Fahrzeuge die Einflussanalyse verzerrt wird.

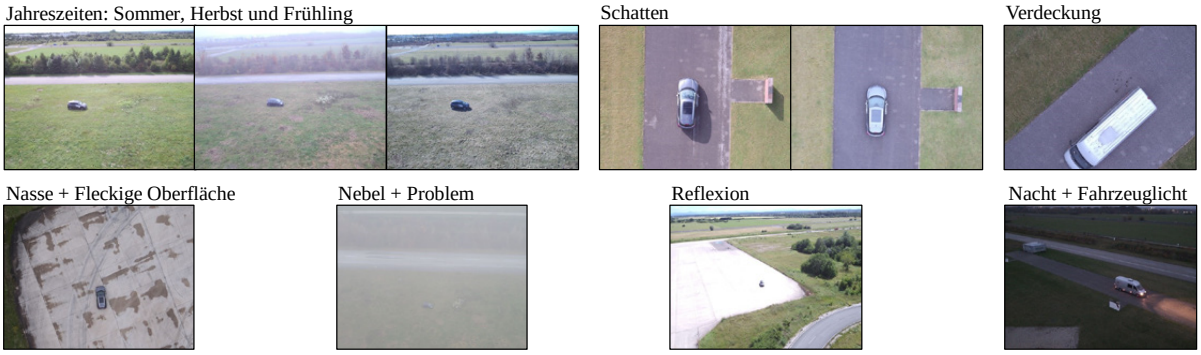


Abb. 50 Übersicht über die manuell zugewiesenen Umgebungsparameter anhand real erflogener Beispielaufnahmen.

Neben den Objektannotationen werden auch die verschiedenen Umgebungszustände auf den real erflogenen Bilddaten händisch annotiert. Abb. 50 zeigt die dabei verwendeten Klassen und zugehörige Beispielbilder. Durch das reproduzierbare Flugmuster konnten identische Szenerien zum Beispiel zu verschiedenen Jahreszeiten aufgenommen werden. Wird einem Bild außerdem die Klasse Schatten zugewiesen, so ist der Fahrzeugschatten für den menschlichen Betrachter sichtbar. Verdeckungen kommen sehr selten vor und wenn dann nur bei großen Testfahrzeugen und niedrigen Flughöhen. Nach Regenschauern wird teilweise eine nasse und dadurch dunklere Straßenoberfläche wahrgenommen, die durch ungleichmäßiges Abtrocknen häufig auch zur Gruppe „Fleckige Oberfläche“ gezählt werden kann. Vor allem bei starkem Nebel wird das Bild oftmals zusätzlich zur Kategorie „Problem“ gezählt, wenn das Fahrzeug auch vom menschlichen Betrachter nur schwer bzw. unzuverlässig erkannt werden kann. Weitere Kategorien sind „Reflexion“ im Umfeld des Fahrzeugs, ein eingeschaltetes „Fahrzeuglicht“ oder Aufnahmen bei „Nacht“ oder in der Dämmerung.

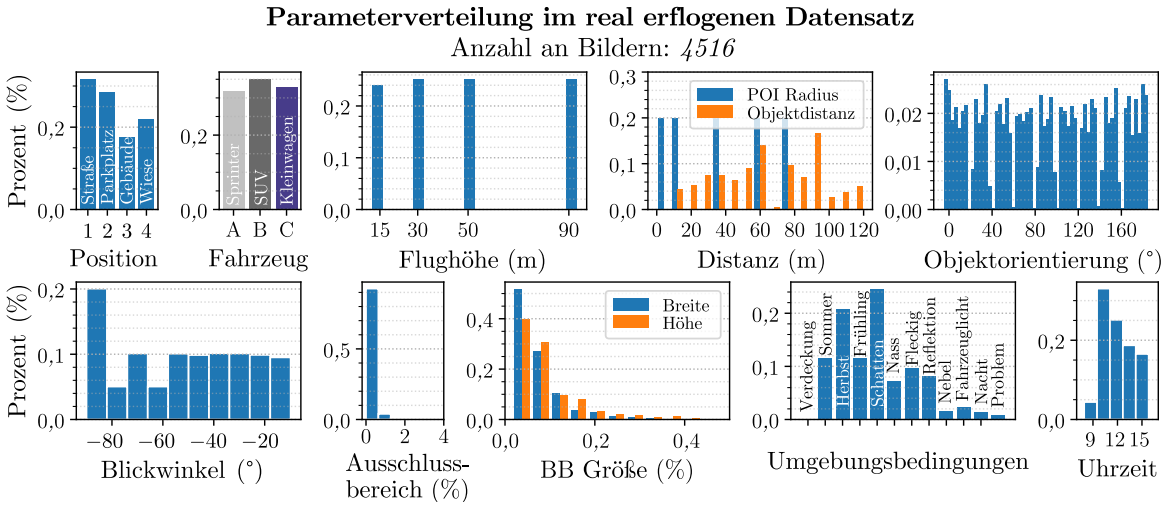


Abb. 51 Grafische Analyse der Verteilung der verschiedenen geometrischen Parameter bzw. Fahrzeug- und Umgebungsparameter beim real erflogenen Datensatz (R-UAV).

Insgesamt enthält der Datensatz 4516 annotierte UAV-basierte Luftbildaufnahmen von Fahrzeugen. Abb. 51 gibt einen Überblick über die Verteilung der Parameter. Viele davon wurden bereits bei der Auswahl und Beschreibung des Flugpfades bestimmt und näher erläutert. An den vier ausgewählten Standorten kamen drei verschiedene Testfahrzeuge zum Einsatz, die verschiedenen Fahrzeugklassen zugehören und in Kapitel 0 näher beschrieben werden. Die dargestellten Messwerte für die Flughöhe liegen ebenso wie die POI Radien relativ exakt auf den theoretisch vorgegebenen Werten. Insgesamt ergeben sich dadurch Distanzen zum Fahrzeug zwischen 15 m und 120 m, was für UAV Aufnahmen durchaus realistisch ist. Im Gegensatz zu Flughöhe und POI Radien können die Vorgaben zur Orientierung des Multikoptersystems hardwarebedingt nicht exakt eingehalten werden. Eine Häufung der Orientierungen bei den vorgegebenen 20° Schritten ist aber deutlich zu erkennen. Die Blickwinkel sind durch die Gestaltung des Flugmusters näherungsweise gleichverteilt. Die Überhöhung bei -90 Grad kommt dadurch zustande, dass vertikal nach unten gerichtete Aufnahmen vom Fahrzeug für jede der vier Flughöhen erfasst werden, die darauf folgenden Aufnahmen bei verschiedenen Abständen zum Fahrzeug jedoch bei jeder Flughöhe zu unterschiedlichen Blickwinkeln führen (vgl. Abb. 47 b)). Die beschriebenen Ausschlussbereiche decken in nahezu allen Fällen weniger als 1 % der Bildfläche ab und können daher bei der Auswertung vernachlässigt werden. Der Anteil, den die *Bounding Box* vom Gesamtbild einnimmt, ist durch den Versuchsaufbau näherungsweise exponentiell abfallend, wodurch kleinere und dadurch schwerer zu detektierende Fahrzeuge häufiger im Datensatz enthalten sind. Abb. 51 zeigt darüber hinaus auch die Verteilung der händisch zugewiesenen Umgebungsbedingungen und die Uhrzeit der Aufnahmen.

Tab. 17 Parameterliste mit den für das Realflugdatenset erfassten Parametern, die später auch zur Einflussanalyse verwendet werden. Die Orange hervorgehobenen Parameter stammen aus einer manuellen Klasseneinteilung der Bilder und wurden nicht automatisch generiert bzw. zugewiesen.
Geom.: Geometrische Parameter bei der Platzierung des Multikopters bzw. der Kamera

	Objektparameter	**Kontextparameter**	**Geom. Parameter**	**Umgebungsparameter**
Erfasste Parameter (RF) im Realflugdatensatz	Fahrzeugtyp Fahrzeugorientierung Fahrzeuglicht BB (%): x, y, gesamt	Fahrzeugposition / Umfeld Ausschlussbereich Verdeckung Distanz zum Objekt	Flughöhe POI Radius Blickwinkel	Uhrzeit Jahreszeit: Sommer/Herbst/Frühling Schatten Nasse Oberfläche Fleckige Oberfläche Reflexion am Fahrzeug Nebel Nacht Problem (menschliche Detektion)

Übersicht

Eine zusammenfassende Auflistung aller erfassten Parameter ist in Tab. 17 dargestellt und in die Untergruppen Objekt-, Kontext, Umgebungs- und geometrische Parameter aufgeteilt. Alle diese Parameter, die im Folgenden auch mit RF abgekürzt werden, dienen ebenso wie die in Kapitel 4.4 aufgelisteten Bildbeschreiber als Merkmale für die Klassifikationskette und tragen damit zur Identifikation der für die Detektion relevanten Einflussfaktoren bei.

5.2.4 Erzeugung synthetischer Duplikate

Um die Auswirkungen des *Reality Gaps* detaillierter untersuchen zu können, war ein Ziel der realen Datenerfassung die Nachbildung der realen Aufnahmen durch detailgetreue synthetisch generierte Duplikate mit nahezu identischem Bildinhalt. Voraussetzung dafür ist die Nachmodellierung des realen Testfluggeländes in der virtuellen Simulationsumgebung, wie bereits in Kapitel 5.1.1 beschrieben. Des Weiteren werden entsprechende 3D-Modelle der real verwendeten Testfahrzeuge benötigt, die in der Simulation an der entsprechenden POI-Position platziert werden können. Abb. 52 zeigt einen Überblick über die drei vorkommenden Modelle. Um eine möglichst identische Nachbildung zu erhalten und Nebeneffekte in der späteren Analyse zu vermeiden, wurde versucht, neben Größe, Baujahr und Autofarbe, auch sämtliche benutzerdefinierten Erweiterungen, wie z.B. Dachboxen oder Aufkleber, zu modellieren.

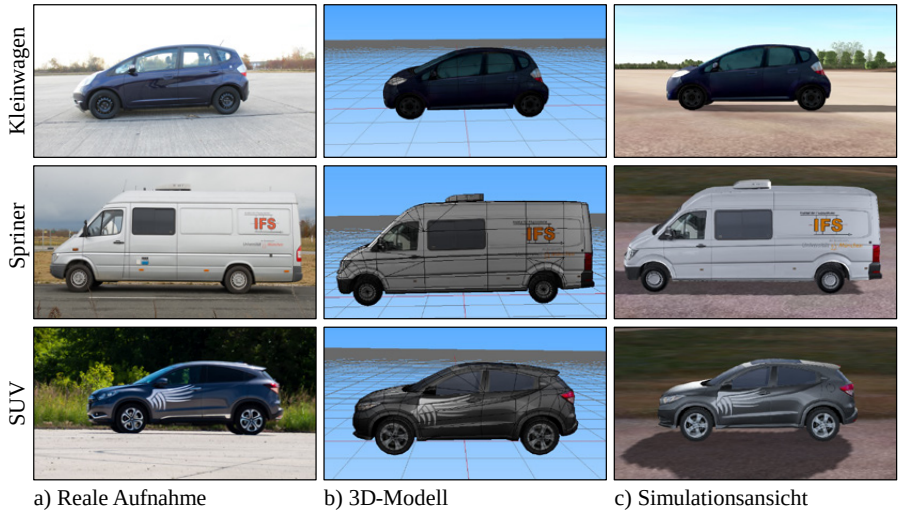


Abb. 52 Übersicht über die verwendeten Testfahrzeuge zusammen mit dem nachmodellierten 3D-Modell und der resultierenden Ansicht in der Simulationsumgebung

Zur automatisierten Generierung der synthetischen Duplikate wird wiederum die Programmierschnittstelle der *Presagis* Umgebung genutzt. Abb. 53 visualisiert den Ablauf, der sehr ähnlich zur Generierung des synthetischen Trainings- und Testdatensatzes ist (vgl. Abb. 39), jedoch einige Besonderheiten aufweist. So wird im Vorfeld für jeden Flug in Vega Prime eine separate Konfigurationsdatei erstellt, die die bei den realen Flügen vorherrschenden Bedingungen in der Simulationsumgebung nachbildet. Dies umfasst zum einen Fahrzeugmodell, -position und -orientierung, aber auch Umgebungsparameter wie Tageszeit, Sichtweite, Schattensimulation oder die vorherrschende diffuse und ambiente Umgebungshelligkeit zur Simulation von Reflexionen und ähnlichen Effekten. Zusätzlich wird die Bewölkung mit der vollständig integrierten *Silver-Lining* Wolkensimulation [231] nachgebildet und der Vegetationszustand mit Hilfe der in Abb. 35 dargestellten Variationen der *SpeedTree* Modelle [137] an die vorherrschende Jahreszeit angepasst. Nach diesen Voreinstellungen kann die Simulationsschleife starten, die über die während dem Flug für jede Aufnahme synchronisiert abgespeicherten Telemetriedaten iteriert. Diese enthalten die in Tab. 15 orange markierten Daten zur Positionierung und Ausrichtung der virtuellen Kamera und ermöglichen so die Generierung identischer synthetischer Duplikate. Anschließend folgt wie bisher die in Kapitel 5.1.2 beschriebene automatisierte *Bounding Box* Generierung, das

Rendern und Abspeichern der synthetischen Bilddatei und die Generierung der zugehörigen Annotationen. Dieser Vorgang wird so lange wiederholt, bis für den aktuellen Flug alle Aufnahmen durchlaufen wurden.

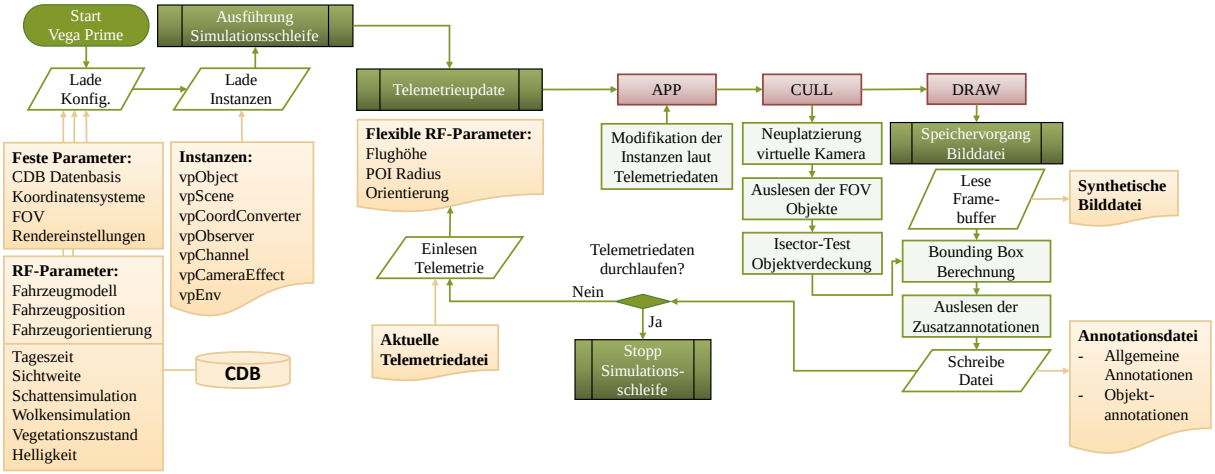


Abb. 53 Ablaufdiagramm zur Visualisierung des Programmablaufs bei der Erzeugung der gekoppelten synthetischen Daten aus den aufgezeichneten Telemetriedaten über die Programmierschnittstelle der Simulationsumgebung

Abschließend werden alle auf diese Weise generierten Bildpaare zur Qualitätskontrolle manuell verglichen. Dadurch wird unter anderem sichergestellt, dass keine fehlerbehafteten Telemetriedaten verwendet werden und dass die händisch und automatisiert generierten Annotationen korrekt zugewiesen sind. Darüber hinaus wird außerdem überprüft, ob die synthetische Modellierung den realen Gegebenheiten entspricht. Der finale Datensatz umfasst ebenso wie die realen Daten (R-UAV) 4516 Bilder und wird im Folgenden mit S-UAV bezeichnet. Abb. 54 zeigt fertige Bildpaare der vier verwendeten Standorte.

Abb. 54 Gegenüberstellung der realen Sensoraufnahmen (R-UAV) und der zugehörigen synthetischen Duplikate (S-UAV) für die vier betrachteten Standorte

5.2.5 Zusammenfassung

Insgesamt wurde in diesem Kapitel ein flexibles und universell anwendbares Multikoptersystem zur gezielten Erfassung realer Sensoraufnahmen für den Anwendungsfall der UAV-basierten Fahrzeugdetektion vorgestellt. Der spezielle Experimentalaufbau und der automatisiert abgeflogene Flugpfad ermöglichen reproduzierbare Aufnahmen zu unterschiedlichen Umgebungsbedingungen. Der generierte Datensatz ist dabei vorrangig für Testzwecke gedacht und erlaubt durch die möglichst entkoppelt erfassten Parametervariationen und die zugehörigen Annotationen eine Einflussanalyse der RF-Parameter auf die Detektionsleistung. Die darauf aufbauende Generierung synthetischer Duplikate liefert reale und

synthetische Bildpaare mit nahezu identischem Bildinhalt und erweitert damit den Anwendungsbereich des Datensatzes für eine gezielte Analyse des *Reality Gaps*.

5.3 Parametervariationen zur Einflussanalyse

Neben den durch die reale Datenerfassung vorgegebenen RF-Parametern werden für eine umfassende Auswertung weitere entkoppelte Parametervariationen in der Analyse mitberücksichtigt. Im ersten Schritt werden dazu die erfassten Testdatensätze (R-UAV und S-UAV) im Nachhinein oder bei ihrer Erzeugung mit den entsprechenden Sensor- und Simulationseffekten versehen. Die Anwendung bereits trainierter Detektormodelle auf diese modifizierten Datensätze lässt Rückschlüsse auf die Stabilität bzw. Anfälligkeit der Detektoren gegenüber den untersuchten Parametern zu. Diese Zusammenhänge können in Verbindung mit den durch die Klassifikationsanalyse identifizierten relevanten Bildeigenschaften gebracht werden und dienen im zweiten Schritt als Grundlage für die Erstellung von Variationsmöglichkeiten bei der Trainingsdatengenerierung. Dieser Schritt ist im Konzept aus Kapitel 3.2 als Rückkopplungszweig aufgeführt. Aus Änderungen in der Detektionsleistung der neu trainierten Modelle können schließlich Rückschlüsse für eine sinnvolle Trainingsdatengenerierung abgeleitet und die zuvor gefundenen Einflussfaktoren verifiziert werden. Tab. 19 gibt einen Überblick über alle untersuchten Trainings- und Testdatenvariationen.

5.3.1 Variationen der Testdatensätze

Im Folgenden werden zunächst die Variationen der Testdatensätze betrachtet. Dabei wird zwischen Sensor- und Simulationsparametern unterschieden.

Sensorparameter

Die Gruppe der Sensorparameter beinhaltet dabei diejenigen Effekte, die sich auf die Bilddarstellung und bestimmte Bildeigenschaften auswirken (s. Abb. 55, vgl. [229]). Sie werden im Nachhinein sowohl auf die real erfassten Bilddaten (R-UAV) als auch auf die synthetischen Duplikate angewandt und in verschiedenen Untergruppen zusammengefasst.

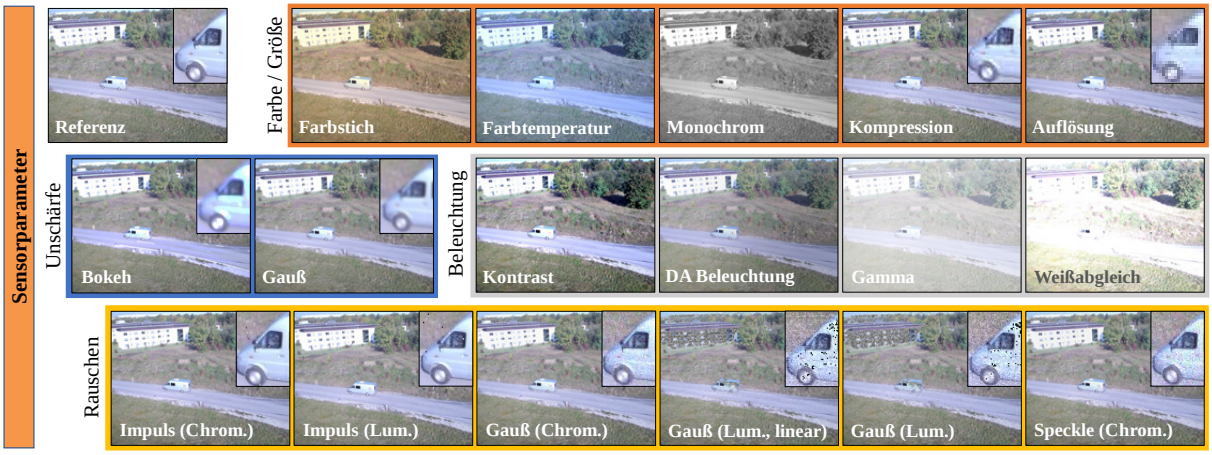


Abb. 55 Überblick über die untersuchten Sensoreffekte und Visualisierung anhand von Beispielbildern. Diese Effekte werden sowohl auf die realen als auch auf die synthetischen Testdaten angewandt (R-UAV + S-UAV).
DA: *Data Augmentation*; Chrom.: Chrominanz; Lum.: Luminanz

Die Gruppe „Farbe / Größe“ enthält beispielsweise einen Datensatz, bei dem durch zufallsbasierte Änderung des V-Wertes im HSV-Farbbereich ein Farbstich verursacht wird. Des Weiteren wird ein Satz mit zufällig veränderter Farbtemperatur [232] zwischen 4500 und 12 000 K erstellt oder ein monochromer Datensatz betrachtet, um den generellen Einfluss der Farbinformationen beurteilen zu können. Ebenfalls enthalten ist eine Reduktion in der JPEG-Kompressionsqualität auf 25 von 100 und eine

niedrigere Auflösung von 400 × 300 px, wobei die Eingangsgröße des Detektornetzwerks 608 × 608 px beträgt.

Als Unschärfe wird zum einen die Gaußsche Unschärfe mit einer Kernelgröße von 5 × 5 px betrachtet, da diese Form häufig als Vorverarbeitungsschritt gängiger Bildverarbeitungsalgorithmen eingesetzt wird. Zum anderen aber auch eine Nachbildung des Bokeh-Effekts [233], der eine Art Hintergrundunschärfe verursacht und in der verwendeten Implementierung zusätzlich leichte Farbfehler verursacht, wie sie auch bei realen Linsen auftreten können.

Die Untergruppe „Beleuchtung“ enthält eine Erhöhung der Parameter Kontrast und Gamma und eine eigentlich für den Zweck der *Data Augmentation* entwickelte Methode [234], die durch zufällige Überlagerung der Ausgangsbilder mit parallel verlaufenden Helligkeitsgradienten oder Spots verschiedene Beleuchtungssituationen simuliert. Beim Weißabgleich werden die Farbwerte durch den über das Bild gemittelten Intensitätswert geteilt und anschließend normalisiert, was teilweise zu einem lokalen Anstieg der Helligkeit führt und dadurch Reflexionen simuliert.

Bei der Untergruppe „Rauschen“ wird zwischen Luminanz-Rauschen, das ausschließlich Helligkeitswerte beeinflusst, und Chrominanz-Rauschen, das alle Farbkanäle betrifft, unterschieden. Zudem werden verschiedene Arten betrachtet. Impuls-Rauschen, auch als *Salt-and-Pepper* Rauschen bezeichnet, manipuliert einzelne Pixel im Bild. Diese Art ist vor allem im Kontext der Forschungsbereiche „*One Pixel Attack*“ und „*Adversarial Attack*“ [68, 235, 236] interessant, die eine Täuschung des Detektors durch gezielte Pixelmanipulationen untersuchen. Weitere Arten sind Gaußsches Rauschen, das einer additiven Überlagerung von Bild und Rauschmuster entspricht und in einem Fall auch lineare Strukturen im Rauschen enthält, und Speckle Rauschen als multiplikative Überlagerung, die dadurch hellere Bereiche stärker beeinflusst.

Simulationsparameter

Die Gruppe der Simulationsparameter wird ebenfalls zur Variation des Testdatensatzes genutzt, kann jedoch im Gegensatz zu den bisher vorgestellten Sensorparametern ausschließlich auf die synthetischen Duplikate (S-UAV) angewandt werden, da diese Effekte den Rendering-Prozess bzw. die allgemeine synthetische Darstellung beeinflussen.

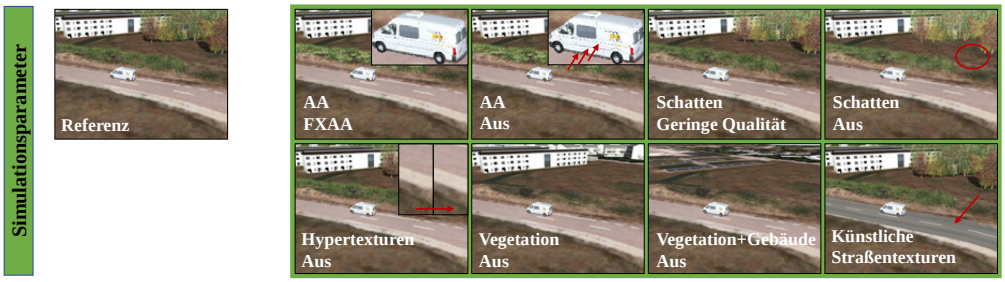


Abb. 56 Überblick über die untersuchten Simulationseffekte und Visualisierung anhand von Beispielbildern. Diese Effekte können nur für die Generierung der synthetischen Testdaten untersucht werden (S-UAV).
AA: Anti-Aliasing; FXAA: *Fast Approximate* Anti-Aliasing

Abb. 56 gibt einen Überblick über die untersuchten Effekte. Dabei wird unter anderem der Einfluss von Anti-Aliasing (AA) untersucht, wobei die FXAA (*Fast Approximate* Anti-Aliasing) Methode zur Anwendung kommt bzw. im anderen Fall keine AA-Reduktion verwendet wird, was zu einer schlechteren Kantenglättung führt. Die Simulationsumgebung bietet außerdem die Möglichkeit, die Qualität der synthetisch berechneten Schattendarstellung zu reduzieren bzw. diese komplett zu deaktivieren. Durch Weglassen der in Kapitel 5.1.1 beschriebenen *Hypertexturen* zur Verbesserung der Bodenstruktur soll untersucht werden, inwiefern die Detektormodelle Abhängigkeiten gegenüber diesen Strukturen beim Training angelernt haben. Des Weiteren wurden Datensätze ohne Modellierung der Vegetation und

zusätzlich ohne Modellierung der Gebäude erstellt, um die Auswirkungen beider Modelltypen auf die Detektionsleistung zu analysieren. Als letztes werden schließlich synthetische Bilddaten betrachtet, bei denen Straßen nicht wie bisher lediglich durch das Luftbild texturiert wurden, sondern durch künstlich generierte Straßentexturen und Straßenmarkierungen, die anhand des in der Simulation nachmodellierten Straßennetzes platziert werden. Insgesamt wurden damit bezüglich der Testdatengenerierung 25 Parametervariationen implementiert und in die Analyse miteinbezogen.

5.3.2 Variationen der Trainingsdatensätze

Im zweiten Schritt werden nun diejenigen Parameter betrachtet, die bei der Trainingsdatengenerierung variiert werden. Durch Analyse der dadurch erreichten Änderungen der Detektionsleistung können Rückschlüsse auf eine sinnvolle Datensatzzusammensetzung bei der Verwendung synthetischer Trainingsdaten gezogen werden. Tab. 18 gibt einen Überblick über die dazu verwendeten Parametervariationen und -werte.

Parameter der Datensatzgenerierung

Die erste Untergruppe betrachtet die bereits in Kapitel 5.1.4 vorgestellten Fahrzeug-, Umgebungs- und geometrischen Parameter der Datensatzgenerierung. Tab. 18 listet in der ersten Zeile erneut die Parameterwerte aus Kapitel 5.1.4, die als Referenz bei der synthetischen Generierung dienen. Blau hinterlegte Felder der nachfolgenden Variationen unterscheiden sich in Bezug auf den jeweiligen Parameter nicht vom Referenzdatensatz.

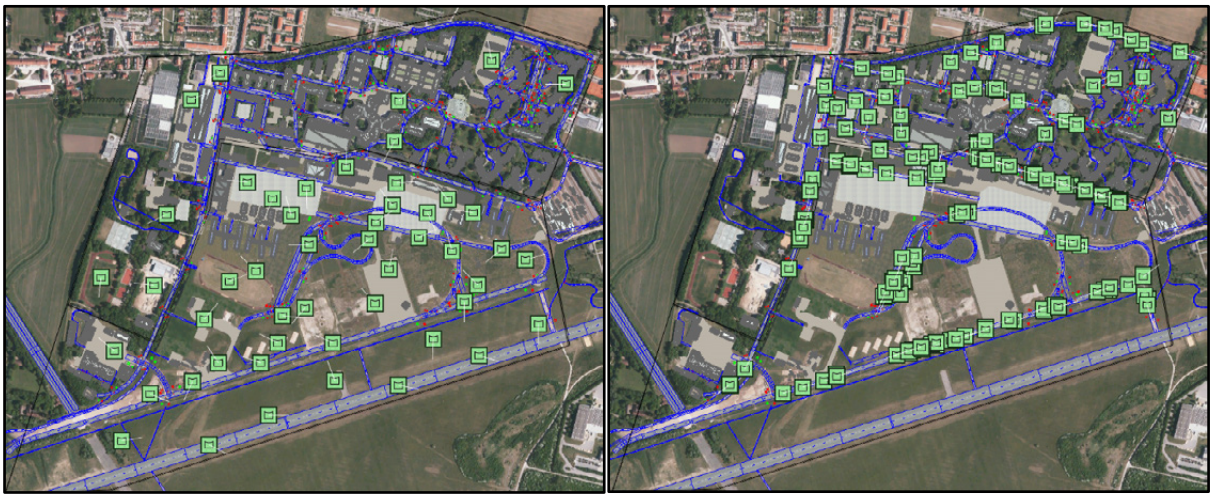


Abb. 57 Übersichtsdarstellung über die mit Hilfe des Szenariengenerators Stage zufällig generierten Fahrzeugpositionen für die Konfigurationen „Gelände“ links und „Straße“ rechts. In Blau ist das nachmodellierte und dafür verwendete Straßennetz zu sehen.

Die erste Variation betrachtet anstatt der 38 Fahrzeugmodelle (80 Modelle nach Umfärben) des Referenzdatensatzes nun 148 verschiedene Fahrzeugmodelle (466 Modelle nach Umfärben), wobei für jedes generierte Bild zufällig ein Modell aus diesem Set ausgewählt wird. Damit kann untersucht werden, ob eine Erhöhung der Variation in Bezug auf die vorkommenden Fahrzeugmodelle eine der Ursachen für den *Reality Gap* darstellt.

Die Variation „Position+“ beschreibt einen Datensatz, der statt der ursprünglich 6 speziell ausgewählten Positionen bei jedem Bild zufällig zwischen 200 verschiedenen Positionen wechselt. Diese wurden mit *Stage* erzeugt, das für die Szenariengenerierung in der *Presagis M&S Suite* zuständig ist. Die Auswahl der Positionen erfolgte dabei anhand des in der Simulationsumgebung nachmodellierten Straßennetzes und enthält 150 Positionen auf verschiedenen Straßentypen und 50 Positionen im Gelände, wobei Gebäudeumrisse und Ähnliches bei der automatischen Positionsgenerierung berücksichtigt werden (s. Abb.

57). Besonders zu erwähnen ist in diesem Zusammenhang die Orientierung der Fahrzeuge entlang des Straßenverlaufs, was in [113] als entscheidender Gestaltungsfaktor für die Generierung realistischer Szenen und für die Reduktion des *Reality Gaps* identifiziert wurde.

Im Datensatz „BB+“ werden anstatt einem Fahrzeug vier Fahrzeuge mit variablem Abstand im FOV der virtuellen Kamera platziert. Aufgrund der sich dadurch ändernden Größenverteilung der *Bounding Boxen* im Datensatz im Vergleich zum Referenzdatensatz, müssen die Ankerboxen für das Training des Detektormodells neu berechnet werden, was durch das grün hinterlegte Feld in Tab. 18 dargestellt wird.

Zur Erhöhung der Dichte von Störobjekten und zur Untersuchung des daraus resultierenden Einflusses auf den FP Anteil der Detektionsergebnisse, werden im Datensatz „Zusatz“ 10 aus 154 verfügbaren Zusatzmodellen (s. Abb. 36) in verschiedenen Abständen und Orientierungen um das Fahrzeugmodell platziert.

„Größe BB“ beschreibt einen ähnlichen Generierungsprozess wie beim Referenzdatensatz, jedoch mit deutlich größeren Flughöhen und Radien zum Objekt und dadurch fast um Faktor zwei kleineren Objektgrößen.

Im Datensatz „Zufall“ werden die geometrischen Parameter des Fahrzeugs nicht in diskreten Schrittweiten variiert, sondern sie werden ähnlich wie bei der Generierung der Testdaten zufällig ausgewählt und sind dadurch kontinuierlich verteilt.

Schließlich wird aus allen bisher vorgestellten Datensatzgenerierungsparametern durch zufällige Auswahl ein gemischter Datensatz erstellt („DG-Mix“), der dadurch nicht nur in Bezug auf einen einzelnen sondern in Bezug auf alle Parameter eine höhere Varianz enthält.

Tab. 18 Tabellarische Übersicht über die Parameterverteilung bei der Generierung der synthetischen Trainingsdatenvariationen anhand der Aufteilung in Datensatzgenerierungsparameter, Sensorparameter und Simulationsparameter. Blau hinterlegte Felder entsprechen den Konfigurationen des Referenzdatensatzes. Grün hinterlegte Variationen erforderten die Berechnung neuer Ankerboxen zum Training des Detektornetzwerks.
Orient.: Fahrzeugorientierung; p.B.: pro Bild; p.S.: pro Schritt; gl.vert.: gleichverteilte Parameter, BB: *Bounding Box*; DG: Datensatzgenerierung; DA: *Data Augmentation*; FXAA: *Fast Approximate* Anti-Aliasing

		Fahrzeugparameter			**Geometrische Parameter**			**Umgebungsparameter**			
		Modelle	Position	Orient.	Flughöhe	Radius	Orient.	Uhrzeit	Sicht-weite	Rau-schen	Zusatz-modelle
Parameter der Datensatzgenerierung	**Referenz**	*80(38)* *1 p.B.* diskret	*6* diskret	*0°* diskret	*15,30,50,90 m* diskret	*0,20,40,80 m* diskret	*30° p.S.* diskret	*6-18 Uhr* gl.vert.	*0,3-30 km* gl.vert.	*0-0,15* gl.vert.	*keine*
	Modell+	*466(148)* *1 p.B.* gl.vert.									
	Position+		*150 Straße* *50 Gelände* gl.vert.	*Stra-ßenver-lauf*							
	BB+	*80(38)* *4 p.B.* *2 - 35 m*		*0-359°* gl.vert.							
	Zusatz										*10 p. B.* *2 -15 m*
	Größe BB				*30,60,100,180 m* diskret	*0,40,80,160 m* diskret					
	Zufall			*0-359°* gl.vert.	*15-100 m* gl.vert.	*0-80 m* gl.vert.	*0-359°* gl.vert.				
	DG-Mix	**Mix** über **Datensatzgenerierungsparameter** (Referenz, Modell+, Zufall, Position+, BB+, Zusatz, Größe BB)									

Sensorparameter	**Farbe**	Farbstich + Farbtemperatur [232]; *4500-12 000 K* 33 % unveränderte Referenzbilder	gl.vert.
	Unschärfe	Gaußsche Unschärfe + Bokeh Unschärfe [233] *Kernel 1; 3; 5; 7; 9 px* 33 % unveränderte Referenzbilder	gl.vert.
	Auflösung	*1024 x 540* + *768 x 405* + *512 x 270* + *256 x 135* dadurch 25 % unveränderte Referenzbilder	gl.vert.

	Beleuchtung	Kontrast; *Faktor 0,4 - 1,9* DA Beleuchtung [234]; *Spot + diffus, zufällige Position, Orientierung, Stärke* Gamma-Wert; *Faktor 0,5; 0,7; 0,9; 1,5; 2,0; 2,5; 3,0* Weißabgleich 33 % unveränderte Referenzbilder	gl.vert.
	Impulsrauschen	Luminanzrauschen; *Anteil 0,0005 - 0,007* + Chrominanzrauschen; *Anteil 0,0005 - 0,004* 33 % unveränderte Referenzbilder	gl.vert.
	Patch Shuffle Regularization (PSR)	[61]; *2 x 2 Kernel* 95 % unveränderte Referenzbilder	
	Neural Style Transfer (NST)	[237]; $\alpha = 0{,}5$ 33 % unveränderte Referenzbilder	
Sim. Param.	**Anti-Aliasing (AA)**	6 Kombinationen: *Transparenz Ein/Aus; Anti-Aliasing: Aus, FXAA, Multisample* dadurch 1/6 unveränderte Referenzbilder	gl.vert.
	Hypertextur (HT)	*Ein/Aus; Texturauflösungen 0,001 - 0,05 m/Texel*	gl.vert.
	SensSimMix	**Mix** über **Sensorparameter** (Farbe, Unschärfe, Auflösung, Beleuchtung, Impulsrauschen, PSR, NST) und über **Simulationsparameter** (AA, HT)	
	AllMix	**Mix** über **Datensetgenerierungsparameter, Sensorparameter und Simulationsparameter**	

Sensorparameter

Die nächste Gruppe in Tab. 18 beschreibt die bei den Trainingsvariationen berücksichtigten Sensorparameter. Diese wurden teilweise bereits im vorigen Kapitel beschrieben und bilden die Datensätze „Farbe“, „Unschärfe“, „Auflösung“, „Beleuchtung“ und „Impulsrauschen“. In jeder dieser Gruppen werden entsprechende Bildeigenschaften zufällig verändert, wobei jeweils 25 % - 33 % unveränderte Referenzbilder enthalten sind, um eine Überanpassung auf bestimmte Bildeigenschaften zu verringern. Für die genauen Parameterwerte sei auf Tab. 18 verwiesen.

Des Weiteren ist die in [61] vorgestellte *Patch Shuffle Regularization* (PSR) enthalten, die zu den *Data Augmentation* Methoden gezählt wird. Sie vertauscht innerhalb eines definierten Kernels zufällig die vorkommenden Pixelwerte. Dies erhöht die lokale Variation, reduziert das Risiko einer Überanpassung, führt dadurch zu einer geringeren Anfälligkeit gegenüber Rauschen und zumindest in Bezug auf die Klassifikation im Allgemeinen zu einer höheren Generalisationsfähigkeit. Interessant ist in diesem Zusammenhang, ob dadurch auch die Variation der relevanten Merkmale in den synthetischen Daten erhöht und eine Überanpassung auf synthetische Merkmale verringert wird. Es werden die in [61] empfohlenen Hyperparameter (Kernelgröße 2×2 px, Modifizierungsanteil 5 %) verwendet.

Ein *Neural Style Transfer* (NST) [237] wird ebenfalls häufig als *Data Augmentation* Methode eingesetzt und versucht unter Verwendung eines tiefen neuronalen Netzes den Inhalt und den Stil eines Bildes zu extrahieren und zu vermischen. Verschiedene stilistische Eigenschaften wie beispielsweise Textur-, Beleuchtungs- und Farbvariationen können somit auf andere Bilddaten übertragen werden, während Formen und semantischer Bildinhalt weitestgehend unverändert bleiben [57]. Die hier implementierte NST Methode aus [237] führt einen zufallsbasierten Stil-Transfer für die Trainingsdaten durch und versucht somit die Robustheit von CNNs gegenüber Änderungen der Domäne (Realität ↔ Simulation) zu erhöhen. Dies ist in Übereinstimmung mit Erkenntnissen aus [49, 80, 97], die besagen, dass nicht unbedingt die photorealisitische Gestaltung synthetischer Daten im Vordergrund steht, sondern die Variation verschiedener Simulationsstile und Bildeigenschaften im Sinne der *Domain Randomization*, sodass die Realität lediglich als weitere Variation vom Detektor wahrgenommen wird.

Simulationsparameter

Die nächste Gruppe variiert verschiedene Simulationsparameter bei der Generierung der synthetischen Trainingsdaten. Dies umfasst Kombinationen mehrerer Anti-Aliasing Methoden und Transparenzeinstellungen sowie eine zufällige Auswahl verschiedener Darstellungsparameter von *Hypertexturen*.

Auch hier werden im „SensSimMix“ Datensatz wiederum zufällig Bilder aus den Gruppen der Sensor- und Simulationsparameter zusammengemischt, um den Einfluss der Erhöhung der Gesamtvarianz bzgl. aller betrachteten Parameter im Vergleich zur Erhöhung bzgl. eines einzelnen Parameters untersuchen zu können, wobei die Datensatzgröße nach wie vor unverändert bleibt.

Der „AllMix“ Datensatz dient dem gleichen Zweck unter Berücksichtigung aller Datensatzgenerierungs-, Sensor- und Simulationsparameter. Auf eine Neuberechnung der Ankerboxen wurde in diesem Fall bewusst verzichtet, da die Schwankungen bei der Clusteranalyse zur Berechnung neuer Ankerboxen größer sind als die Unterschiede zum Referenzdatensatz und zudem eine bessere Vergleichbarkeit gegeben ist.

5.3.3 Zusammenfassung

Tab. 19 liefert abschließend einen Gesamtüberblick über alle generierten Datensätze mit der zugehörigen Parametervariation für Trainings- und Testdatensätze und der jeweiligen Gruppenzugehörigkeit.

Tab. 19 Gesamtüberblick der generierten und untersuchten Parametervariationen für Trainings- und reale und synthetische Testdatensätze

Trainingsdatenvariationen		**Testdatenvariationen**			
Datensatzgenerierungsparameter	Modell+ Zufall Position+ BB+ Zusatz Größe BB	Sensorparameter	Farbe/Größe	Farbstich Farbtemperatur Monochrom Kompression Auflösung	Für synthetische Testdaten (S-UAV) Für reale Testdaten (R-UAV)
Mix	DG-Mix		Unschärfe	Bokeh Gauß	
Sensorparameter	Farbe Unschärfe Auflösung Beleuchtung Impulsrauschen Patch Shuffle Regularization Neural Style Transfer		Beleuchtung	Kontrast DA Beleuchtung Gamma Weißabgleich	
			Rauschen	Impuls (Chrom. + Lum.) Gauß (Chrom. + Lum. linear + Lum.) Speckle (Chrom.)	
Simulationsparameter	Anti-Aliasing Hypertexturen	Simulationsparameter		Anti-Aliasing: FXAA Anti-Aliasing: Aus Schatten: Geringe Qualität Schatten: Aus Hypertextur: Aus Vegetation: Aus Vegetation + Gebäude: Aus Künstliche Straßentexturen	
Mix	SensSimMix AllMix				

Die beiden rechten Spalten zeigen, welche Testdatenvariationen fürs das synthetische Datenset (S-UAV) und welche ausschließlich für das reale Datenmaterial (R-UAV) erstellt werden konnten. Insgesamt dienen die generierten und hier vorgestellten Parametervariationen ebenfalls zur Analyse der relevanten Einflussfaktoren und sollen helfen, die Prozesse und Anfälligkeiten *deep-learning* basierter Detektornetzwerke besser einschätzen und verstehen zu können. Sie sollen als Ergänzung des in Kapitel 4.5 beschriebenen Auswerteverfahrens auf Basis der Klassifikationskette gesehen werden und die damit identifizierten Einflussfaktoren in einer Art Rückkopplung bestätigen. Teil dieser Rückkopplung ist auch die Untersuchung der vorgestellten Trainingsvariationen, um den Einfluss der Trainingsdatengestaltung auf die Detektionsleistung ermitteln zu können und um daraus Anhaltspunkte für eine sinnvolle Zusammensetzung synthetischer Trainingsdaten für den hier betrachteten Anwendungsfall abzuleiten.

5.4 Implementierung des Detektortrainings

In Kapitel 4.3 wurde bereits begründet, warum das YOLOv3 Netzwerk für die Untersuchungen im Bereich der UAV-basierten Fahrzeugdetektion als Testalgorithmus ausgewählt wurde und welche grundlegende Funktionsweise diese Netzwerkarchitektur charakterisiert. Im Folgenden soll nun das Trainingsverhalten bei den verwendeten Datensätzen näher betrachtet und eine Hyperparameteroptimierung vorgenommen werden.

5.4.1 Ausgangsbedingungen und Trainingskonfiguration

Es gibt bereits frei verfügbare Gewichte für das YOLOv3 Netzwerk, die in einem langwierigen Trainingsprozess auf sehr umfangreichen Benchmark-Datensätzen zur Objektdetektion, wie z.B. Pascal VOC [153] oder ImageNet [152] trainiert wurden. Diese berücksichtigen eine große Bandbreite an Objektklassen und gewährleisten aufgrund der hohen Varianz in den Datensätzen eine gute Identifikation von allgemeinen und speziellen Merkmalen, die für ein Objekt im Bild charakteristisch sind. Die erwähnten Gewichte betrachten jedoch ausschließlich Objekte aus der Bodenperspektive im Gegensatz zu den in dieser Arbeit untersuchten Luftbildaufnahmen und erfassen dadurch komplett andere Blickwinkel, Größenverhältnisse und Hintergrundszenerien. Außerdem liegt der Fokus auf der Erkennung mehrerer Objektklassen, während im hier betrachteten Anwendungsfall ausschließlich die Klasse „Fahrzeug“ analysiert werden soll, um Verzerrungen bei der Identifikation relevanter Einflussfaktoren zu minimieren. Aus diesen Gründen ist ein erneutes Training mit den entsprechenden realen und synthetischen Trainingsdatensätzen nötig.

Als Ausgangspunkt für das erneute Training dienen die Gewichte des dem YOLOv3 Detektors zugrundeliegenden *Darknet-53* Netzwerks, das auf dem ImageNet Datensatz vortrainiert wurde. Dies ist sinnvoll, da die vortrainierten Gewichte durch die umfassenden Benchmark-Daten bereits sehr effizient einfache und dadurch vom Anwendungsfall unabhängige und robuste Merkmale für die Objektdetektion extrahieren und das Netzwerk somit nicht von Grund auf neu trainiert, sondern nur auf den neuen Anwendungsfall angepasst werden muss. Dies erfordert eine geringere Anzahl an Trainingsdaten, erhöht die Generalisationsfähigkeit trotz Verwendung weniger und speziellerer eigener Trainingsdaten und führt außerdem zu einer Steigerung des Lernfortschritts und einer geringeren Trainingsdauer.

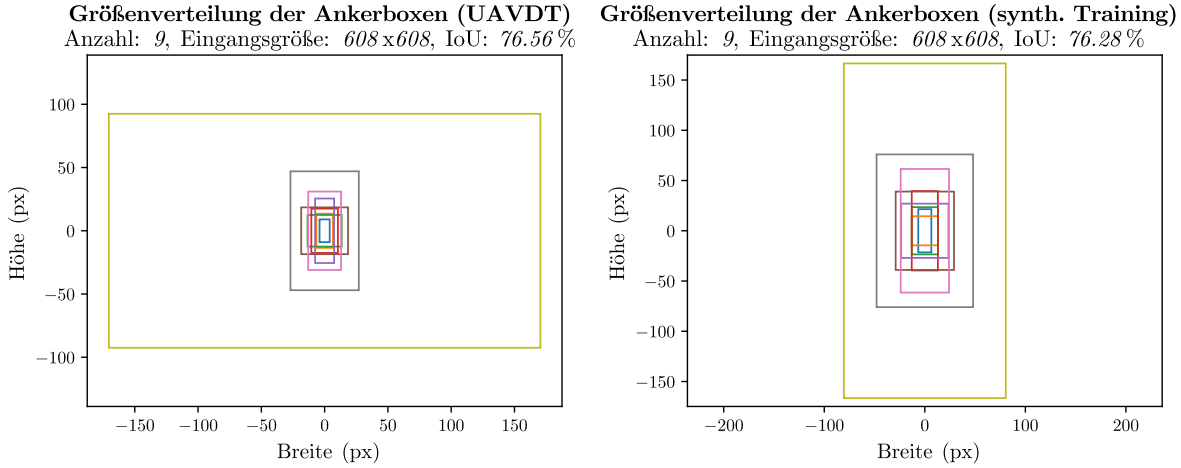


Abb. 58 Grafische Visualisierung der Dimensionen und Seitenverhältnisse der neun mit Hilfe des *K-Means* Clusteralgorithmus berechneten Ankerboxen für die UAVDT Trainingsdaten (links) im Vergleich zu den hier beschriebenen synthetisch generierten Trainingsdaten (rechts). Die IoU beschreibt jeweils die durchschnittliche Überlappung zu den im Datensatz vorkommenden *Bounding Boxen*. (vgl. [51])

Im Vorfeld des Trainings werden für jeden verwendeten Trainingsdatensatz durch eine *K-means* Clusteranalyse passende Ankerboxen berechnet. Dies führt zu einem stabileren Trainingsverhalten, da

dadurch nicht eine zufällig geschätzte *Bounding Box* auf ein zu detektierendes Objekt angepasst, sondern lediglich die Abweichung einer bereits gezielt ausgewählten Startgröße optimiert werden muss. Abb. 58 visualisiert die Abmessungen der berechneten Ankerboxen für die Trainingsdaten des UAVDT Datensatzes und des synthetischen Referenzdatensatzes. In beiden Fällen wird durch die Ankerboxen eine sehr gute Abdeckung der vorkommenden Objektgrößen erzielt, was eine IoU von ungefähr 76 % bestätigt. Bis auf die Orientierung der größten Box zeigen beiden Darstellungen eine ähnliche Verteilung in Bezug auf die Seitenverhältnisse, jedoch scheint die Fahrzeuggröße im UAVDT Datensatz tendenziell etwas kleiner zu sein als bei den synthetisch generierten Trainingsdaten.

Das erneute Training erfordert eine auf den Anwendungsfall abgestimmte Anpassung weiterer Konfigurationsparameter. Die Implementierung des YOLOv3 Detektors beinhaltet eine sogenannte *In-Place Data Augmentation*. Dies bedeutet, dass die Eingangsdaten während des Trainingsprozesses zufällig modifiziert werden und das Netzwerk somit in jedem Durchlauf neue unbekannte Daten sieht. Trotz Durchlaufens mehrerer Epochen wird so eine doppelte Verwendung von Trainingsdaten verhindert und dadurch die Robustheit der extrahierten Merkmale gesteigert. Variationen für Sättigung, Belichtung, Farbton, Bildzuschnitt und Änderung der Eingangsauflösung wurden verwendet. Die Drehung des Bildes um einen bestimmten Winkel wurde bewusst deaktiviert, da dies bei Fahrzeugen zu tendenziell eher unnatürlichen Orientierungen führt. Es wurde eine Stapelgröße von 64 mit einer Unterteilung von 16 gewählt, d.h. erst nach einer Iteration und 64 durchlaufenen Trainingsbildern werden die Gewichte des Netzwerkes aktualisiert, wobei 16 davon gleichzeitig auf der Grafikkarte verarbeitet werden. Große Gewichte bzw. große Gewichtsänderungen bei der Aktualisierung werden durch die Werte *Decay* = 0,0005 und *Momentum* = 0,9 beschränkt. Die Eingangsbildgröße betrug 608 × 608 px, wobei diese jeweils nach zehn durchlaufenen Stapeln durch die *Data Augmentation* zufällig verändert wird. Bei Bedarf müssen die neun neu berechneten Ankerboxen in der Konfigurationsdatei ersetzt werden, wobei sie je nach Größe einer der drei Skalierungsschichten des Netzwerks zugeordnet werden. Detektionen mit einem Zuverlässigkeitswert unter 0,005 werden für die Evaluierung nicht berücksichtigt. Dies begrenzt zwar die PR-Kurven hin zu hohen *Recall*-Werten und beeinflusst dadurch in geringem Maß die AP, ist jedoch nötig, um den Berechnungsaufwand in einem durchführbaren Rahmen zu halten. Bei der *Non-Maximum Suppression* kam die standardmäßig vorgegebene IoU-Schwelle von 0,45 zum Einsatz. Jeder Trainingsprozess startete mit einer sogenannten *Warm-up* Phase mit einer niedrigeren Lernrate über eine Dauer von 1000 Iterationen. Dies soll bewirken, dass die vortrainierten, einfachen und robusten Gewichte aus dem vorderen Teil des Netzwerks in den hinteren, für die Klassifikation neuer Objekte zuständigen Teil übertragen werden und das Trainings dadurch schneller konvergiert. Die Optimierung bisher noch nicht festgelegter Parameter wird im nachfolgenden Abschnitt beschrieben.

5.4.2 Hyperparameteroptimierung

Zur Optimierung des Trainingsverhaltens werden verschiedene Lernratenverläufe untersucht und anhand von Kriterien wie Detektionsleistung oder Trainingsdauer miteinander verglichen (vgl. [51]). Die Lernrate bestimmt, wie stark sich die Gewichte des Netzes pro Iteration ändern können. Sie weist zu Beginn des Trainings höhere Werte auf, fällt dann über die Zeit über einen definierten Verlauf ab und beeinflusst die Konvergenz des Trainings.

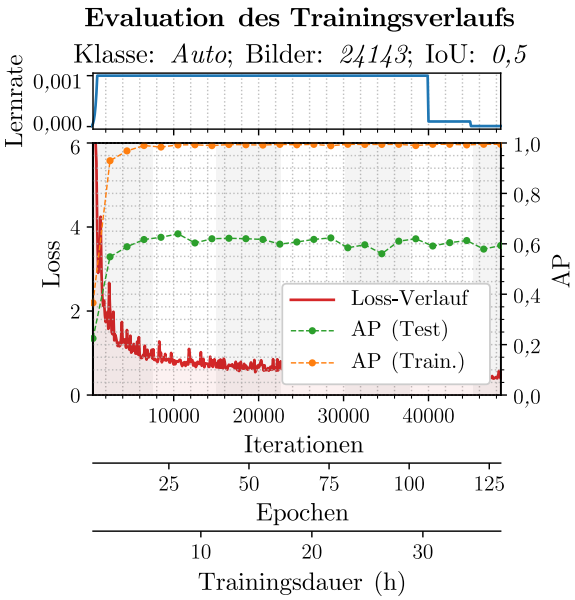


Abb. 59 Evaluation des Trainingsverhaltens anhand von Loss und AP für Trainings- und Testdaten über 50 000 Iterationen. Zur Anwendung kam ein stufenförmiger Lernratenverlauf aus der Konfigurationsdatei für das Training beim Pascal VOC Datensatz. (UAVDT Trainingssatz; IoU = 0,5) (vgl. [51])

Als Ausgangspunkt für die Untersuchungen dienen die Trainingsdaten des UAVDT Datensatzes, da angenommen wird, dass diese realen Daten die höchste Varianz und Komplexität bzgl. der vorkommenden Merkmale aufweisen und daher als Maßstab für das Trainingsverhalten angewendet werden können. Abb. 59 zeigt Loss und AP für einen stufenförmigen Lernratenverlauf, der in Anlehnung an die Trainingskonfiguration des Pascal VOC Datensatzes als Referenz gewählt wurde. Die Lernrate steigt dabei nach einer Warum-up Phase auf 0,001, bevor sie schließlich nach 40 000 und 45 000 Iterationen jeweils auf ein Zehntel ihres aktuellen Wertes reduziert wird. Insgesamt entspricht dies einem gemäßigten und eher langwierigen Training über 125 Epochen mit einer niedrigen initialen Lernrate. Eine Epoche entspricht dabei der Anzahl an Iterationen, die bei einer vorgegebenen Stapelgröße nötig sind, um den gesamten Trainingsdatensatz zu durchlaufen. Dabei bleibt zu bedenken, dass dem Netzwerk durch die integrierte *Data Augmentation* auch beim Durchlauf mehrerer Epochen stets geringfügig andere Eingangsbilddaten vorliegen. Die AP des zugehörigen UAVDT Testdatensatzes steigt dabei vermutlich aufgrund der vortrainierten Gewichte schnell an und schwankt nur geringfügig um AP-Werte von ca. 0,6. Ein Abfall der Detektionsleistung durch Überanpassung am Ende des Trainings kann nicht beobachtet werden.

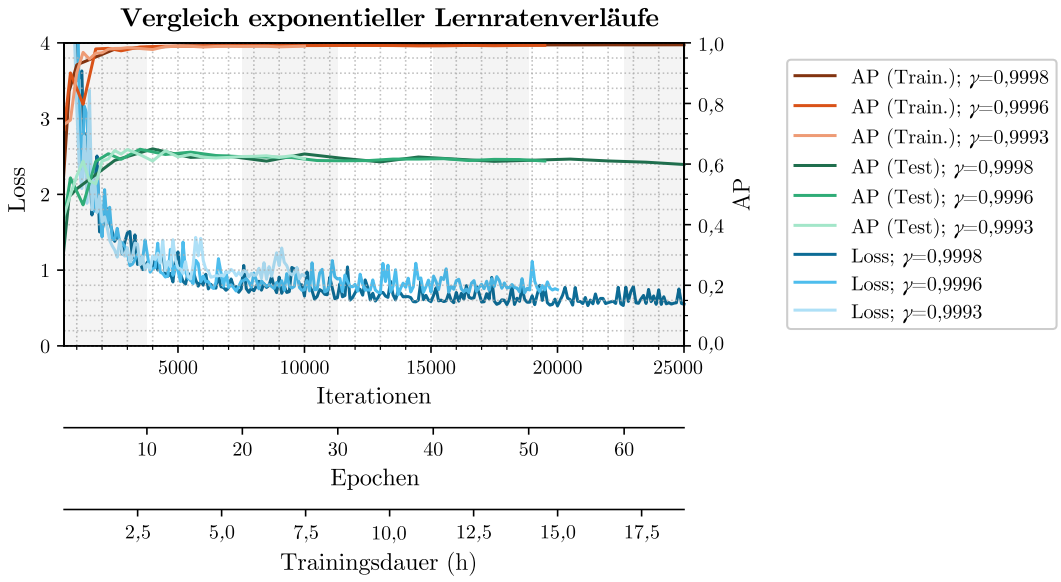


Abb. 60 Gegenüberstellung exponentieller Lernratenverläufe mit einer initialen Lernrate von 0,004 und verschiedenen Werten für die Basis γ. (UAVDT Trainingssatz; IoU = 0,5) (vgl. [51])

Aufgrund der schnellen Konvergenz von Loss und Detektionsrate wird im nächsten Schritt ein exponentieller Lernratenverlauf mit einer höheren initialen Lernrate lr_{init} untersucht. Die aktuelle Lernrate lr_{akt} wird dabei gemäß folgender Formel berechnet:

$$lr_{\text{akt}} = lr_{\text{init}} \cdot \gamma^{x} , \qquad x\text{: aktueller Iterationsschritt; } \gamma\text{: Basis} \tag{11}$$

Abb. 60 zeigt den resultierenden Verlauf von Loss und AP für verschiedene Werte der Basis γ, wobei die initiale Lernrate stets $lr_{\text{init}} = 0{,}004$ betrug. Niedrigere Werte der Basis führen zu einem schnelleren Abfall der Lernrate und einer kürzeren Trainingsdauer. Trotz der deutlich geringeren Anzahl an Iterationen erreicht die Detektionsleistung für Trainings- und Testdatensatz ähnliche Werte wie beim stufenförmigen Lernratenverlauf. Die AP des Testdatensatzes zeigt jedoch aufgrund des exponentiellen Abfalls einen deutlich stabileren Verlauf mit weniger Schwankungen, was die Auswahl eines finalen Modells vereinfacht. Die Loss sinkt auf Werte zwischen 0,5 und 1, was für ein kompaktes Modell spricht und zeigt ein deutliches und schnelles Konvergenzverhalten. Es wird vermutet, dass die geringe Anzahl nötiger Iterationen zum Teil auf die Verwendung von passenden vortrainierten Gewichten und zum Teil auf die Auswertung lediglich einer Objektklasse zurückzuführen ist, da in [238] eine Größenordnung von mindestens 2 000 Iterationen pro Klasse empfohlen wird.

Da die verschiedenen Basen gemäß Abb. 60 zu keinem signifikanten Leistungsunterschied führen, wird als Konfiguration eine initiale Lernrate $lr_{\text{init}} = 0{,}004$ mit der Basis $\gamma = 0{,}9993$ verwendet, da diese die geringste Trainingsdauer benötigt. Außerdem wird durch die Begrenzung auf 10 000 Iterationen vor allem bei der Verwendung einfacherer und synthetischer Datensätze das Risiko einer Überanpassung zusätzlich verringert.

Insgesamt kann festgehalten werden, dass diese Randbedingungen im vorliegenden Fall zu einem schnellen und gleichzeitig stabilen Lernprozess führen. Daher und aufgrund der Vergleichbarkeit werden diese Werte für alle nachfolgenden Untersuchungen angewandt.

6 Ergebnisse und Auswertung

Nach der Beschreibung der grundlegenden Methoden und der nötigen Implementierungsschritte sollen nun im Folgenden die Ergebnisse der durchgeführten Experimente vorgestellt werden. Die Auswertung gliedert sich gemäß den in Kapitel 3.1 aufgestellten Forschungsfragen in drei Abschnitte. Zu Beginn jedes Teilbereichs wird die jeweils behandelte Fragestellung in Erinnerung gerufen und die dafür vorgenommenen Untersuchungen erläutert und anhand einer Konzeptgrafik schematisch dargestellt. Abschließend werden alle Ergebnisse gesammelt und in ihrer Gesamtheit bewertet. Teilergebnisse in den einzelnen Unterkapiteln werden durch Umrandung hervorgehoben.

6.1 Wahl der Trainingsdatenzusammensetzung und Auswirkungen auf die Detektionsleistung

Im Folgenden wird der Einfluss der **Trainingsdatenzusammensetzung** auf das Detektionsergebnis in Bezug auf die reale und synthetische Domäne untersucht (zugehörige Forschungsfragen siehe Tab. 20).

Tab. 20 Wiederholung der Forschungsfragen zur Wahl der Trainingsdatenzusammensetzung

Trainingsdatenzusammensetzung - Forschungsfragen	
1.	Wie verhalten sich **real trainierte** Modelle auf synthetischen Testdaten im Vergleich zu realen Testdaten?
2.	Was sind die Auswirkungen der Verwendung rein **synthetischer Trainingsdaten**?
3.	Welche Leistung erreicht ein Modell, das mit **gemischten Trainingsdaten** aus beiden Domänen trainiert wurde?
4.	Wie unterscheidet sich die Leistung dieser drei Trainingskonfigurationen auf inhaltsgleichen **realen und synthetischen Bildduplikaten**?

Es kommt stets das in Kapitel 4.3 ausgewählte und beschriebene YOLOv3 Netzwerk zur Objektdetektion zum Einsatz, das jeweils mit den in Kapitel 5.4 beschriebenen Randbedingungen und Hyperparametern trainiert wird. Da in [154] festgestellt wurde, dass es für menschliche Betrachter schwierig ist, *Bounding Boxen* mit einer IoU-Schwelle von 0,3 von denen mit einer Schwelle von 0,5 zu unterscheiden, wird für die folgenden Auswertungen stets eine IoU-Schwelle von 0,3 verwendet.

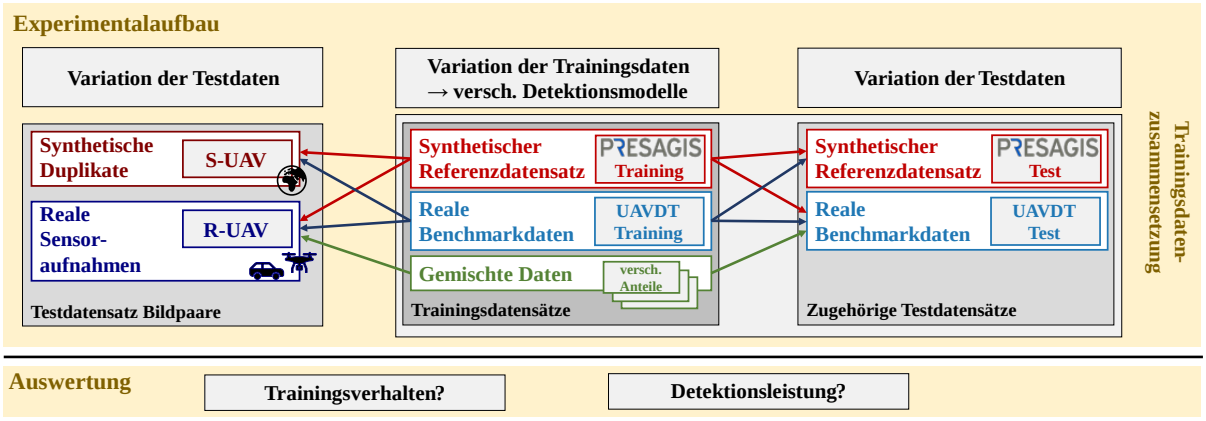


Abb. 61 Konzeptgrafik zur Untersuchung der Trainingsdatenzusammensetzungen: Es sind die verwendeten Trainings- und Testdatensatzkombinationen dargestellt, die im ersten Teil der Untersuchungen zur Auswertung der Detektionsleistung verwendet wurden.

Die Konzeptgrafik in Abb. 61 gibt einen Überblick über die für Trainings- und Testzwecke verwendeten Datensatzkombinationen, die im ersten Schritt der Untersuchungen verwendet werden, um die Auswirkungen verschiedener Trainingsdatenkombinationen und damit die in obiger Tabelle aufgelisteten

Forschungsfragen zu untersuchen. Dieser Block entspricht den ersten beiden Ebenen des Gesamtkonzepts aus Abb. 4 und ist entsprechend farbig gekennzeichnet.

Es werden drei Trainingskonfigurationen betrachtet und für das Modelltraining verwendet:

- Die eine beschreibt ein ausschließlich synthetisches Training mit dem in Kapitel 5.1.3 und 5.1.4 umschriebenen und eigens generierten synthetischen Referenzdatensatz.
- Eine andere verwendet das in Kapitel 4.1 ausgewählte und frei verfügbare reale UAVDT Datenset als Benchmark.
- Schließlich wird untersucht, welche Auswirkungen ein hybrider Ansatz, bei dem die realen Trainingsdaten mit verschiedenen Anteilen synthetischer Daten erweitert werden, auf die Detektionsleistung und die Generalisationsleistung hat.

Die Auswertung auf den zugehörigen Testdatensätzen dient der Analyse des *Reality Gaps* und soll zeigen, inwiefern dieser eine Richtungsabhängigkeit aufweist. Die Besonderheit im Vergleich zu den in Tab. 1 gesammelten Veröffentlichungen zum Einsatz von virtuellen Simulationsumgebungen bei CV-Anwendungen besteht darin, dass die beschriebenen Trainingsdatenkonfigurationen nicht nur auf den zugehörigen Testdaten evaluiert werden, sondern auch auf realen und synthetischen Bildpaaren (s. Kapitel 5.2). Somit werden für den hier behandelten Anwendungsfall alle möglichen Kombinationen betrachtet, was eine ganzheitliche Auswertung ermöglicht. Dies ist ebenfalls in Abb. 61 dargestellt.

Die dabei verwendeten Bildpaare weisen einen nahezu identischen Bildinhalt auf, unterscheiden sich lediglich in den visuellen Bildeigenschaften und ermöglichen somit detailliertere Analysen des *Reality Gaps*. Zudem besteht durch die örtlich ähnlichen Aufnahmepunkte (s. Abb. 46) in Bezug auf die Szenerie eine große Ähnlichkeit zu den synthetischen Trainingsdaten. Dies erlaubt genauere Rückschlüsse über die Gesamtzusammenhänge bei der Verwendung synthetischer Trainingsdaten und vor allem über die Auswirkungen einer selektiven Trainingsdatenerweiterung mit synthetischen Bildern aus dem späteren Einsatzbereich. Teilweise unterscheiden sich die im Folgenden vorgestellten absoluten Werte geringfügig von den in [51, 229] veröffentlichten Werten, da in der Zwischenzeit der real erflogene Datensatz erweitert und die synthetische Umgebung ausgebaut und aktualisiert wurde. Die daraus abgeleiteten Aussagen bleiben jedoch weitestgehend unverändert.

6.1.1 Training mit realen Benchmark Daten

Als Ausgangsbasis wird nun im ersten Schritt das Trainingsverhalten des real trainierten Detektormodells analysiert und die Detektionsleistung auf den zugehörigen Testdaten evaluiert. Tab. 21 wiederholt die Forschungsfrage und beschreibt das in diesem Kapitel betrachtete Experiment mit der zugehörigen Auswertung.

Tab. 21 Tabellarische Übersicht über die jeweils behandelte Forschungsfragestellung, das zugehörige Experiment und die einzelnen Bestandteile der Auswertung

1.	Wie verhalten sich real trainierte Modelle auf synthetischen Testdaten im Vergleich zu realen Testdaten?
	Experiment: Training des YOLOv3 Detektors auf dem realen UAVDT Benchmark Trainingsdatensatz
	Auswertung: Beurteilung des Trainingsverhaltens und der Detektionsleistung - auf den zugehörigen realen Testdaten - auf synthetisch generierten Testdaten

Die realen UAVDT Trainings- und Testdaten dienen dabei als Grundlage für die weiteren Untersuchungen. Abb. 62 zeigt Lernratenverlauf, Loss-Funktion und Detektionsleistungen über die Trainingsdauer.

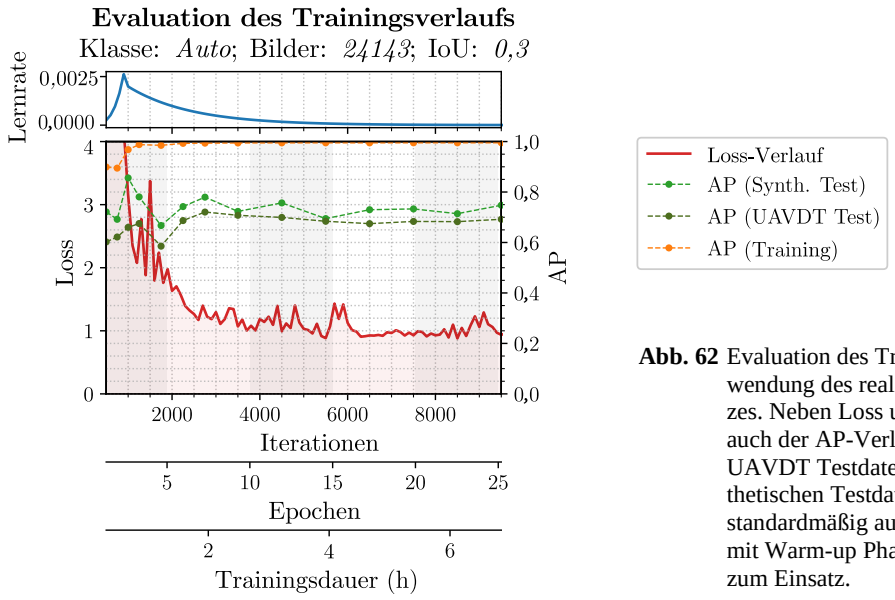


Abb. 62 Evaluation des Trainingsverhaltens unter Verwendung des realen UAVDT Trainingsdatensatzes. Neben Loss und AP der Trainingsdaten ist auch der AP-Verlauf der zugehörigen realen UAVDT Testdaten und der unabhängigen synthetischen Testdaten dargestellt. Es kommt der standardmäßig ausgewählte Lernratenverlauf mit Warm-up Phase und exponentiellem Abfall zum Einsatz.

Standardmäßig für alle Trainingskonfigurationen kommt der in Kapitel 5.4.2 beschriebene Lernratenverlauf mit *Warm-up* Phase und exponentiellem Abfall zum Einsatz. Der Loss-Verlauf konvergiert trotz einiger Ausreißer stetig zu einem Wert um 0,9-1,0, was für ein kompaktes Modell spricht. Durch die Verwendung von vortrainierten Gewichten stabilisiert sich auch die Detektionsleistung bereits nach ungefähr 2 500 Iterationen. Die ist in Übereinstimmung mit den Empfehlungen aus [238] wonach ungefähr 2 000 Trainingsiterationen pro Klasse benötigt werden. Abb. 62 zeigt den Verlauf der AP für die verwendeten realen Trainingsdaten, die zugehörigen, aber unbekannten realen UAVDT Testdaten und auch für die davon unabhängigen synthetischen Testdaten. Trotz des anspruchsvollen Datensatzes mit einer Vielzahl an Objekten erreicht die Detektionsleistung auf den Trainingsdaten sehr schnell eine nahezu ideale AP. Die AP der zugehörigen UAVDT Testdaten liegt relativ stabil bei ca. 70 % und die AP der synthetischen Testdaten sogar geringfügig darüber bei ca. 75 %, wobei diese dafür eine tendenziell größere Schwankung über die Trainingsdauer aufweist. Da der Verlauf der AP über beide Testdatensätze bei fortschreitender Trainingsdauer nicht nennenswert abfällt, kann davon ausgegangen werden, dass bei dieser Trainingskonfiguration keine Überanpassung stattfindet und Lernratenverlauf und Anzahl an Iterationen passend gewählt wurden. Insgesamt wird mit dieser Trainingskonfiguration eine sehr ähnliche Detektionsleistung auf realen und synthetischen Testdaten erzielt.

Zur Untersuchung der Lokalisationsgenauigkeit und der jeweiligen genaueren Zusammenhänge zwischen *Precision* und *Recall* werden zusätzlich die PR-Kurven betrachtet. Abb. 63 zeigt diese für die zugehörigen realen (links) und die unabhängigen synthetischen Testdaten (rechts). Die Spreizung der Kurven und die Veränderung der AP für verschiedene IoU-Schwellwerte ist ein Maß für die Lokalisationsgenauigkeit. Genaue Modelle zeichnen sich dadurch aus, dass die AP erst bei hohen IoU Werten stark abfällt. Mittelwert und Standardabweichung dienen dabei ebenfalls als Bewertungskriterium und repräsentieren in gewisser Weise die Überlappung detektierter *Bounding Boxen* mit der *Ground Truth*. Beim Vergleich der Daten aus Abb. 63 kann für die Testdaten aus den unterschiedlichen Domänen ein sehr ähnliches Verhalten beobachtet werden und zwar sowohl in Bezug auf die Form und die Verteilung der Kurven als auch in Bezug auf die absolute Detektionsleistung. Letztere liegt bei den synthetischen Testdaten sogar geringfügig höher, da synthetisch generierte Bilddaten im Allgemeinen weniger Störeinflüsse und dadurch markantere und deutlichere Merkmale aufweisen und zudem tendenziell einfachere und weniger detailreiche Szenarien enthalten.

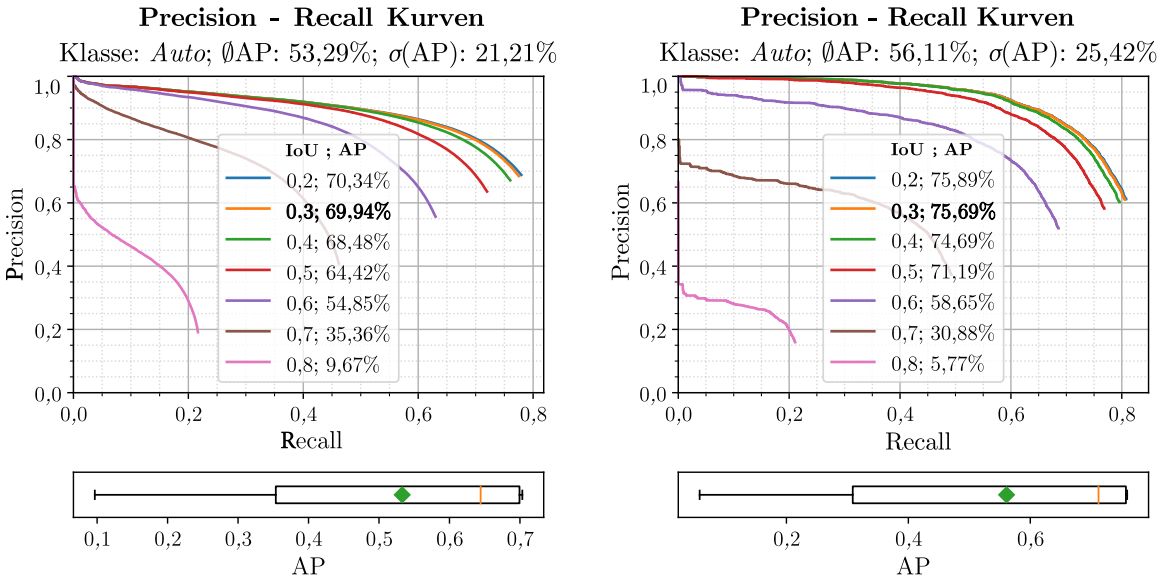


Abb. 63 PR-Kurven für verschiedene IoU Schwellwerte und zugehörige AP mit Mittelwert und Standardabweichung bei Verwendung des real trainierten Detektormodells nach 4500 Iterationen. Die linke Grafik zeigt die Anwendung auf das zugehörige UAVDT Testset, die rechte auf die synthetisch generierten Testdaten.

Zusammenfassung und Interpretation

Aus den Ergebnissen lässt sich schlussfolgern, dass sich im vorliegenden Fall das real trainierte Modell auf realen und synthetischen Testdaten ähnlich verhält und der *Reality Gap* in dieser Kombination gering ist.

Daher liegt die Annahme nahe, dass ähnliche Effekte in der Realität einen ähnlichen Einfluss in der Simulation bewirken. Es ist zu erwähnen, dass dies hier im ersten Schritt nur die Richtung von der Realität zur Simulation betrifft.

Dies ist in Übereinstimmung mit den Forschungsergebnissen aus der Literatur für den Anwendungsfall bodengestützter Objektdetektion. Carrillo et al. [105] wiesen unter anderem nach, dass bei real trainierten Modellen eine Hyperparameteroptimierung bzgl. Lernratenverlauf und Trainingsdauer in der Simulation möglich ist. Auch Gaidon et al. [101] zeigten in Übereinstimmung mit den hier vorgestellten Erkenntnissen, dass auf realen Daten vortrainierte Tracking-Algorithmen bei der Anwendung in beiden Domänen ein ähnliches Verhalten aufweisen und sich verschiedene Einflüsse in der realen und der synthetischen Welt ähnlich auf die Detektionsleistung auswirken. Sie nutzten daher eine virtuelle Umgebung zur Messung des Einflusses verschiedener Wetterbedingungen und Bildeigenschaften auf die Erkennungsleistung und stellten fest, dass diese bei Modellen, die mit realen Benchmark-Datensätzen trainiert wurden, zu einem signifikanten Leistungsabfall führen können.

Somit kann festgehalten werden, dass eine Evaluierung real trainierter Detektormodelle auf synthetischen Daten in vielen Fällen daher durchaus gerechtfertigt und eine Übertragbarkeit von der Realität zur Simulation möglich ist. Der *Reality Gap* in diese Richtung ist gering. Dies ermöglicht beispielsweise, dass der Einfluss bestimmter Umgebung- oder Testbedingungen auf die Detektionsleistung, die in der Realität nicht oder nur mit großem Aufwand nachgestellt werden könnten, in der Simulation mit synthetischen Daten analysiert werden kann. Inwiefern dies allgemein gültig ist und auch für einzelne spezielle Parameter gezeigt werden kann, wird in Kapitel 6.3.1.2 näher betrachtet.

6.1.2 Training mit synthetisch generierten Trainingsdaten

Im zweiten Schritt wird nun das Trainingsverhalten und die Detektionsleistung eines rein synthetisch trainierten Detektormodells analysiert. Tab. 22 wiederholt die Forschungsfrage und beschreibt das in diesem Kapitel betrachtete Experiment mit der zugehörigen Auswertung.

Tab. 22 Tabellarische Übersicht über die jeweils behandelte Forschungsfragestellung, das zugehörige Experiment und die einzelnen Bestandteile der Auswertung

2.	Was sind die Auswirkungen der Verwendung rein synthetischer Trainingsdaten?
	Experiment: Training des YOLOv3 Detektors auf dem synthetisch generierten Trainingsdatensatz
	Auswertung: Beurteilung des Trainingsverhaltens und der Detektionsleistung - auf den zugehörigen synthetischen Testdaten - auf den realen Benchmark Testdaten

Die in Abschnitt 5.1 beschriebene synthetische Datenbasis dient dabei als Trainings- und Testdatensatz. Abb. 64 zeigt Lernratenverlauf, Loss-Funktion und Detektionsleistungen über die Trainingsdauer.

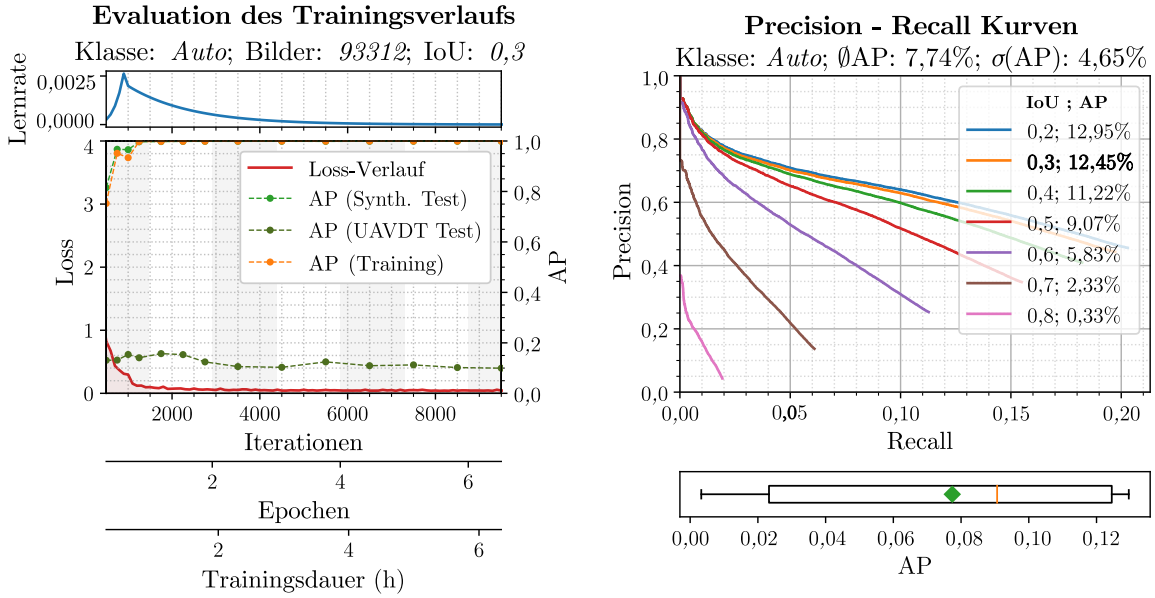


Abb. 64 *Links:* Evaluation des Trainingsprozesses bei Verwendung rein synthetischer Daten und AP-Werte des zugehörigen synthetischen Testdatensatzes und des unabhängigen realen UAVDT Testdatensatzes.
Rechts: PR-Kurven für verschiedene IoU Schwellwerte und zugehörige AP mit Mittelwert und Standardabweichung bei Anwendung des synthetisch trainierten Detektormodells nach 2750 Iterationen auf den realen UAVDT Testdaten.

Auffällig ist der schnelle und starke Abfall der Loss auf Werte im Bereich von 0,05. Im Vergleich zum realen Training aus Kapitel 6.1.1 mit Werten von 0,9 – 1,0 entspricht dies einem Abfall um mehr als eine Größenordnung und deutet darauf hin, dass ein vergleichsweise kleines Modell angelernt wurde, das auf wenige, spezielle Merkmale fokussiert ist. Dieses ist sehr gut auf die durch die Trainingsdaten vorgegebene Domäne angepasst, hat jedoch zu wenige allgemeine Merkmale angelernt, um eine gute Generalisationsfähigkeit zu erreichen. Das zeigt sich auch bei der Betrachtung der Detektionsleistung auf den synthetischen Trainings- und den zugehörigen Testdaten. In beiden Fällen werden nach einem schnellen und steilen Anstieg nahezu ideale AP-Werte von mehr als 99 % erreicht.

Dies hat mehrere Gründe. Zum einen wurden in dieser Konfiguration dieselben 3D-Modelle für die Generierung der Trainings- und Testdaten verwendet (vgl. Kapitel 5.1.4). Zum anderen weisen synthetisch generierte Daten im Allgemeinen auch weniger Störeffekte und eine geringe Komplexität auf und enthalten gleichzeitig ausgeprägtere Merkmale. Zudem sind die synthetischen Testdaten dem Detektor zwar unbekannt, sie stammen jedoch aus der gleichen Domäne und der gleichen geografischen

Umgebung, enthalten auf diese Weise sehr ähnliche Merkmale wie die bekannten Trainingsdaten und profitieren dadurch von einer möglichen Überanpassung.

Darüber hinaus zeigt dies aber auch, dass trotz der bei der Trainingsdatengenerierung diskret gewählten geometrischen Parameterstufen für Flughöhe, Radius und Orientierung (vgl. Kapitel 5.1.4) eine sehr gute Generalisationsfähigkeit in Bezug auf unbekannte, kontinuierlich verteilte und zufällig ausgewählte Abstufungen erzielt werden kann. Dies bedeutet im Umkehrschluss, dass die diskreten Schrittweiten sinnvoll gewählt wurden und auch für die reale Datengenerierung als Maßstab herangezogen werden können.

Die AP für die unabhängigen und aus der anderen Domäne stammenden realen UAVDT Testdaten ist ebenfalls in Abb. 64 Links dargestellt und erreicht nur relativ geringe Werte. Abb. 64 Rechts zeigt die zugehörigen PR-Kurven nach 2750 Trainingsiterationen. Die größere Spreizung der Kurven im Vergleich zum realen Training deutet auf eine geringere Lokalisationsfähigkeit des Detektors hin. Die allgemein niedrige Leistung zeigt, dass der *Reality Gap* bei synthetisch trainierten Modellen, die auf reale Daten angewandt werden, hier deutlich höher ist als im umgekehrten Fall. Dies bedeutet, dass in dieser Konfiguration nicht alle Merkmale, die für eine korrekte Detektion benötigt werden, vorhanden waren bzw. gelernt wurden, im umgekehrten Fall jedoch schon. Auffällig ist, dass die AP Werte nach einem kurzen Anstieg bei steigender Anzahl an Iterationen wieder abfallen, was insgesamt gesehen auf eine Überanpassung auf die synthetische Domäne hindeutet. Die maximal erreichte AP betrug 15,7 % nach 1750 Iterationen. Der in [22] mit den realen UAVDT Trainingsdaten trainierte Faster-RCNN Detektor erreichte ebenfalls lediglich eine AP von 22,3 %, jedoch für eine höhere IoU Schwelle und wahrscheinlich als Mittelwert über alle Fahrzeugklassen (Bus, Auto, LKW). Dennoch zeigt dies, dass das vorliegende Ergebnis als Ausgangspunkt für die weiteren nachfolgenden Untersuchungen zur Einflussanalyse genauer betrachtet werden sollte.

Ursachenanalyse

Die Übertragbarkeit zwischen den Domänen im Allgemeinen kann von mehreren Faktoren abhängen, wie z.B. der verwendeten Simulationsumgebung, der Qualität der Modellierung, der Datensatzzusammenstellung oder dem Algorithmus selbst. Im Folgenden werden daher im ersten Schritt einige zahlenmäßige Ursachen und Randbedingungen in Bezug auf die verwendeten Trainings- und Testdaten näher betrachtet (vgl. [51]).

Tab. 23 Vergleich zwischen dem UAVDT Datensatz und dem synthetisch generierten Datensatz in Bezug auf die vorkommenden Objekte und deren Anzahl (vgl. [51])

	Training		Test		Ø Anzahl Objekte pro Bild	Anzahl verschiedener Fahrzeuge
	Anzahl Bilder	Anzahl BB	Anzahl Bilder	Anzahl BB		
UAVDT	24 143	394 633	16 592	361 055	16,3	2700
Synth. Datensätze	93 312	86 480	9 720	9 520	0,93	38 (80 nach Umfärben)

Tab. 23 zeigt eine Gegenüberstellung der verwendeten realen und synthetischen Daten und offenbart einige relevante Unterschiede, die zu einem *Reality Gap* beitragen könnten. Zum einen weisen die realen Daten eine deutlich höhere Anzahl an *Bounding Boxen* auf, sowohl insgesamt betrachtet als auch pro Bild. Außerdem ist die Anzahl und damit die Varianz der vorkommenden Fahrzeugtypen bei den realen Daten mit 2700 deutlich höher als in den synthetischen Daten mit 80 verschiedenen Modellen. Bereits in Kapitel 5.4.1 wurde beim Vergleich der berechneten Ankerboxen in Abb. 58 die Vermutung geäußert, dass die Objekte des UAVDT Datensatzes durchschnittlich eher kleiner sind als bei den synthetischen Trainingsdaten. Die dadurch bedingte Verwendung falscher initialer *Bounding Boxen* bei der Detektion kann sich ebenfalls negativ auf die Leistung auswirken. Zudem erhalten die Daten im realen UAVDT

Datensatz hauptsächlich sehr detailreiche Aufnahmen innerstädtischer chinesischer Hauptstraßen, die vorwiegend Fahrzeuge auf mehrspurigen Fahrbahnen mit einer Vielzahl an Straßenmarkierungen zeigen. Wie Abb. 5 und Abb. 43 veranschaulichen, stellt dies in Bezug auf die Szenerie ebenfalls einen deutlichen Unterschied zu den synthetisch generierten Daten dar und kann sich unabhängig von der Domäne zusätzlich negativ auf die Detektionsleistung auswirken.

Zusammenfassung und Interpretation

In diesem Zusammenhang wird daher zwischen verschiedenen Bestandteilen des *Reality Gaps* unterschieden. Einer beschreibt den durch die unterschiedliche Domäne der Trainings- und Testdaten hervorgerufenen Leistungsunterschied. Ein anderer hingegen resultiert durch die Unterschiede in Datensatzzusammensetzung und Szenerie und kann auch bei der Verwendung verschiedener realer Datensätze zu signifikanten Leistungsunterschieden führen [106].

Auch in [98, 101, 108] und [105] führte ein ausschließlich synthetisches Training zu einer geringeren Leistungsfähigkeit als das Training mit realen Daten. Carrillo et al. [105] stellten außerdem fest, dass dabei durch eine zu geringe Variation der Merkmale im Verhältnis zur Anzahl der Bilder mit steigender Anzahl an Trainingsiterationen in Analogie zu Abb. 64 die Leistungsfähigkeit abnimmt und eine beginnende Überanpassung einsetzt. Prakash et al. [106] beobachteten dasselbe Phänomen bei Verwendung des VKITTI Datensatzes, leiteten daraus ab, dass die synthetische Bildgenerierung für die Objektdetektion zum Teil zu statisch in Bezug auf Geometrie, Straßen, Gebäude und Texturen ist und versuchten daher, den Generierungsprozess durch den Einsatz von *Structured Domain Randomization (SDR)* zu optimieren. Dies deckt sich mit den Aussagen aus [33], die schlussfolgern lassen, dass der Trainingswert eines einzelnen simulierten Bildes geringer ist als der eines realen Bildes, da der Detailgrad und die abgedeckte Variation in Bezug auf Beleuchtung, Farbe und Textur geringer ist. Jedoch konnten die selben Autoren [33] mit simulierten Trainingsdaten sogar bessere Ergebnisse erzielen als bei realem Training, unter der Voraussetzung, dass ausreichend synthetische Daten vorhanden waren. In [101] wurde dagegen ebenso wie in [108] gezeigt, dass die insgesamt beste Leistung durch ein synthetisches Vortrainieren und anschließendes Fine-Tuning auf realen Daten erreicht werden kann.

Insgesamt kann daher festgehalten werden, dass je nach Anwendungsfall, vorherrschenden Randbedingungen und verwendeten Datensätzen die Leistungsfähigkeit bei ausschließlicher Verwendung synthetischer Trainingsdaten in der Literatur unterschiedlich beurteilt und angegeben wird.

Für den komplexen Fall der luftgestützten Objekterkennung war ein rein synthetisches Training in der beschriebenen Form und mit der beschriebenen Datengenerierung nicht ausreichend, um auf unabhängigen beliebigen realen Testdaten eine vergleichbar gute Leistung zu erzielen wie mit realen Trainingsdaten. Dies bedeutet, dass der Reality Gap zwischen realem und synthetischem Bildmaterial eine deutliche Richtungsabhängigkeit aufweist, je nachdem, welche Domäne zum Trainieren des Netzes verwendet wurde.

Eine geringere Generalisationsfähigkeit der resultierenden Modelle und eine Überanpassung auf die synthetische Domäne sind neben der Zusammensetzung der Datensätze und deren zahlenmäßigen Zusammenhänge einige mögliche Ursachen. In Kapitel 6.2 soll auf Basis dessen mit dem in Kapitel 4.5 beschriebenen statistischen Auswerteverfahren analysiert werden, welche Bildeigenschaften in diesem Zusammenhang ausschlaggebend sind. Insgesamt dienen die daraus abgeleiteten Schlüsse auch als Anregungen für die Untersuchung verschiedener Optimierungen der synthetischen Trainingsdatengenerierung in Kapitel 6.3.2.

Es wurde gezeigt, dass trotz der diskreten Parameterabstufungen in den synthetischen Trainingsdaten (s. Kapitel 5.1.4) das Modell in der Lage ist, auf den zugehörigen synthetischen Testdaten zwischen allen vorkommenden kontinuierlich verteilten Zuständen zu generalisieren.

Im folgenden Kapitel wird nun der Einfluss gemischter Trainingsdaten für den hier betrachteten Anwendungsfall untersucht.

6.1.3 Training mit gemischten Trainingsdaten

Im letzten Schritt wird nun die Trainingskonfiguration mit gemischten Daten aus beiden Domänen betrachtet. Tab. 24 wiederholt die Forschungsfrage und beschreibt das in diesem Kapitel betrachtete Experiment mit der zugehörigen Auswertung.

Tab. 24 Tabellarische Übersicht über die jeweils behandelte Forschungsfragestellung, das zugehörige Experiment und die einzelnen Bestandteile der Auswertung

3.	Welche Leistung erreicht ein Modell, das mit gemischten Trainingsdaten aus beiden Domänen trainiert wurde?
	Experiment: Training des YOLOv3 Detektors auf einem gemischten Trainingsdatensatz, der neben den realen Bilddaten verschiedene Anteile synthetischer Trainingsdaten enthält
	Auswertung: Beurteilung des Trainingsverhaltens und der Detektionsleistung auf den realen Testdaten

Die gesamten realen UAVDT Trainingsdaten dienen dabei als Basis und werden mit einer bestimmten Menge an synthetischen Trainingsbildern erweitert. Bei der Evaluierung kommen die realen UAVDT Testdaten zum Einsatz, da dieser Fall der späteren tatsächlichen Anwendung am nächsten kommt.

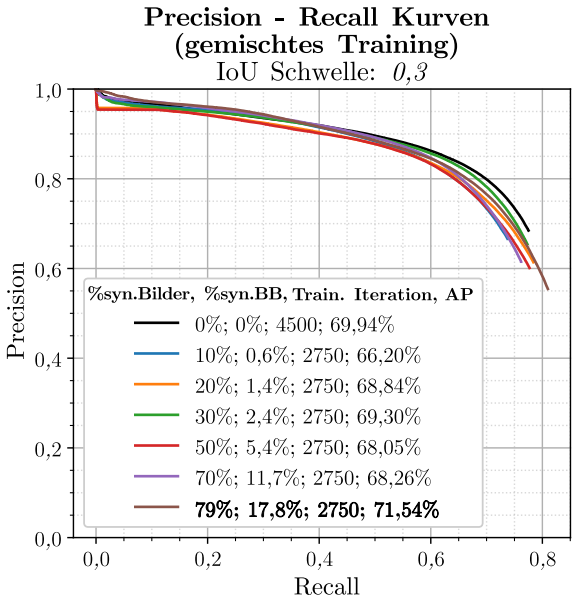


Abb. 65 PR-Kurven für verschiedene Anteile an synthetischem Bildmaterial in den verwendeten Trainingsdaten. Die realen UAVDT Trainingsdaten sind jeweils komplett enthalten. Darüber hinaus ist der Anteil synthetischer *Bounding Boxen*, der verwendete Iterationsschritt und die erreichte AP angegeben. Für die Evaluierung wurden stets die realen UAVDT Testdaten verwendet.

Abb. 65 zeigt einen Vergleich der resultierenden PR-Kurven für eine IoU-Schwelle von 0,3. Der Anteil der synthetischen Daten an den insgesamt verwendeten Trainingsdaten variiert dabei zwischen 10 % und 79 %. Zusätzlich ist der Anteil synthetischer *Bounding Boxen*, die jeweils betrachtete Trainingsiteration und die dabei erreichte AP aufgelistet. Die schwarze Kurve repräsentiert als Grundlage die Detektionsleistung bei ausschließlicher Verwendung der realen UAVDT Trainingsdaten. Es zeigt sich, dass bei einem Anteil von 79 % synthetischer Trainingsdaten, was einem Anteil von knapp 18 % der vorkommenden *Bounding Boxen* entspricht, die AP durch die synthetische Erweiterung um 1,6 Prozentpunkte gesteigert werden konnte.

Die realen UAVDT Trainings- und Testdaten stammen aus derselben geografischen Umgebung, enthalten ähnliche Szenarien und Fahrzeuge und weisen übereinstimmende Bildeigenschaften auf, wodurch sie sich sehr stark ähneln. Es ist erwähnenswert, dass durch Beimischung unabhängiger, sehr unterschiedlicher synthetischer Daten die Detektionsleistung dennoch, wenn auch nur geringfügig, gesteigert werden konnte. Diese Steigerung ist hauptsächlich darauf zurückzuführen, dass höhere *Recall*-Werte

erzielt werden, was im Umkehrschluss bedeutet, dass die Generalisationsfähigkeit des Detektormodells erhöht wurde und ein größerer Teil der vorkommenden Fahrzeuge erkannt wird.

Die *Precision* hingegen bleibt durch die Beimischung synthetischer Daten weitestgehend unverändert bzw. fällt sogar leicht ab. Ob und vor allem bei welchem synthetischen Bildanteil dieser Effekt eintritt, hängt von mehreren Faktoren ab, wie z.B. dem Anteil der *Bounding Boxen*, der Zusammensetzung der Szenerie, der Ähnlichkeit zu den realen Daten und nicht zuletzt dem Detailgrad und dem Trainingswert, den ein einzelnes synthetisches Bild liefert. Das Optimum bei einem Anteil von 79 % synthetischer Daten ist daher kein allgemeingültiger Wert, sondern lediglich ein Anhaltspunkt, der für jeden Anwendungsfall und für unterschiedliche Voraussetzungen und Datensätze neu überprüft werden muss (siehe [51]).

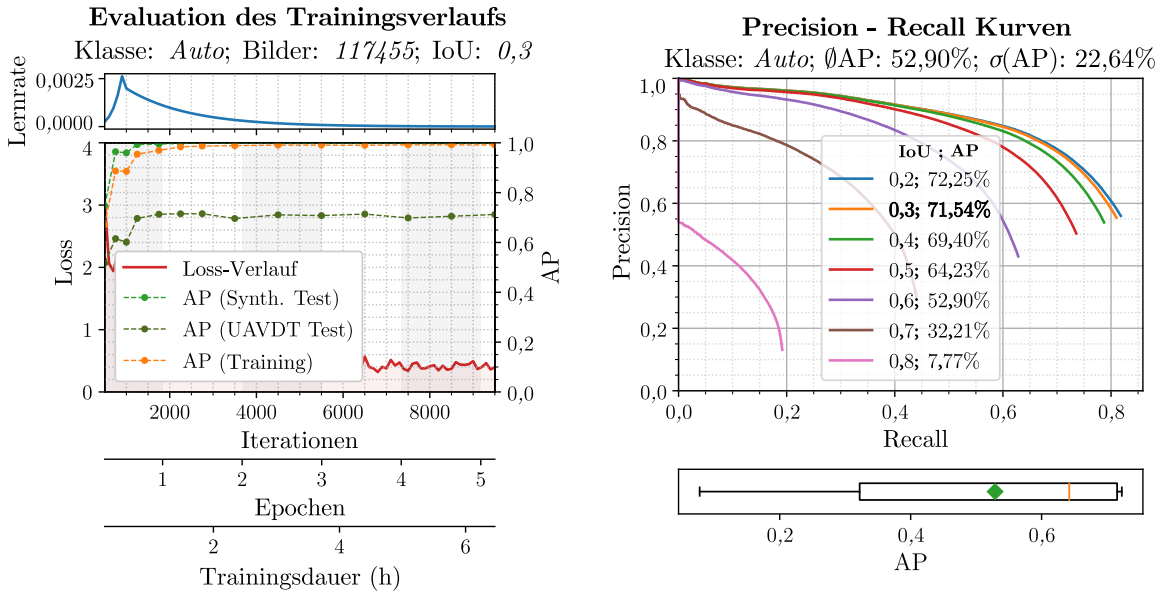


Abb. 66 *Links:* Evaluation des Trainingsprozesses bei Verwendung des gemischten Trainingsdatensatzes mit einem Anteil synthetischer Bilder von 79 %.
Rechts: PR-Kurven für verschiedene IoU Schwellwerte und zugehörige AP mit Mittelwert und Standardabweichung bei Anwendung des gemischt trainierten Detektormodells mit 79 % synthetischen Trainingsdaten nach 2750 Iterationen auf den realen UAVDT Testdatensatz.

Abb. 66 zeigt das Trainingsverhalten nach Beimischung von 79 % synthetischen Daten und die zugehörigen PR-Kurven für die verschiedenen IoU-Schwellen nach 2750 Iterationen. Es ist zu sehen, dass durch den synthetischen Anteil nicht nur bei den Trainingsdaten nach sehr kurzer Zeit nahezu ideale AP-Werte erzielt werden, sondern auch bei den synthetischen Testdaten. Das Modell hat demnach neben den relevanten realen Merkmalen auch die synthetischen Merkmale angelernt und betrachtet diese als eine Art Variation der Realität. Dies trägt höchstwahrscheinlich zu der erwähnten Erhöhung der Generalisationsfähigkeit bei und führt bei den realen UAVDT Testdaten zu leicht höheren AP-Werten. Interessant ist auch die Betrachtung des Verlaufs der AP auf den UAVDT Testdaten im Vergleich zu Abb. 62. Es fällt auf, dass die Beimischung synthetischer Daten neben der geringfügigen Steigerung der Detektionsleistung vor allem zu einem schnelleren und stabileren Anstieg der AP-Werte führt und dass diese insgesamt weniger Schwankungen unterworfen sind. Ein Abfall nach längerer Trainingsdauer als Zeichen für eine Überanpassung ist ebenfalls nicht erkennbar.

Im rechten Teil von Abb. 66 sind die zugehörigen PR-Kurven nach 2750 Iterationen dargestellt. Tab. 25 listet die daraus resultierende Veränderung der AP über die verschiedenen IoU Schwellwerte im Vergleich zur realen Trainingskonfiguration aus Abb. 63. Es zeigt sich, dass die Verbesserung durch den synthetischen Anteil bei steigenden IoU Schwellwerten stark abnimmt und sich sogar ins Gegenteil umkehrt. Dies deutet darauf hin, dass die im Detektionsnetzwerk verwendete *Bounding Box* Regression

anfällig ist für systematische oder zufällige Ungenauigkeiten der *Ground Truth* in den Trainingsdaten. Dies betrifft sowohl die händischen Annotationen, als auch die automatisch generierten Annotationen aus der Simulation. Letztere beruhen wie in Kapitel 5.1.2 beschrieben auf der 3D Hülle des Objekts und weisen dadurch vor allem bei schrägen Blickwinkeln je nach Fahrzeugform gewisse systematische Ungenauigkeiten auf. Durch die Wahl einer niedrigen IoU-Schwelle von 0,3 für die Auswertung wird verhindert, dass dies beim Vergleich zwischen realen und synthetischen Daten ebenfalls zum *Reality Gap* beiträgt.

Tab. 25 Gegenüberstellung der AP-Werte der rein realen und der gemischten Trainingskonfigurationen. Die Zahlenwerte und die zugehörigen PR-Kurven sind in Abb. 63 (reale UAVDT Trainingsdaten, 4500 Iterationen) und Abb. 66 (79 % synth. Trainingsdaten + reale UAVDT Trainingsdaten, 2750 Iterationen) zu finden. Evaluiert wurde jeweils auf den realen UAVDT Testdaten

IoU Schwellwert	0,2	**0,3**	0,4	0,5	0,6	0,7	0,8
AP (0 % synth. Trainingsbilder)	70,34 %	**69,94 %**	68,48 %	64,42 %	54,85 %	35,36 %	9,67 %
AP (79 % synth. Trainingsbilder)	72,25 %	**71,54 %**	69,40 %	64,23 %	52,90 %	32,21 %	7,77 %
AP Differenz	+ 1,91	**+ 1,60**	+ 0,92	- 0,19	- 1,95	- 3,15	- 1,90

Zusammenfassung und Interpretation

Insgesamt hat sich durch die Untersuchungen gezeigt, dass sich für die UAV-basierten Fahrzeugdetektion mit tiefen neuronalen Netzen der Einsatz gemischter Trainingsdaten positiv auf die Detektionsleistung auswirken kann und zu besseren Ergebnissen führt als die ausschließliche Verwendung realer Benchmark-Datensätze.

Dies ist zum Großteil auf die gesteigerte Generalisationsfähigkeit des angelernten Detektormodells zurückzuführen, da durch die synthetischen Bilder die Variation der Merkmale in den Trainingsdaten steigt. Der Anteil synthetischer Daten, der dabei verwendet werden sollte, ist stark von der Zusammensetzung und dem Trainingswert der einzelnen synthetischen Bilder abhängig und sollte je nach Datenbasis und Anwendungsfall gezielt optimiert werden. Um in diesem Zusammenhang einen Einfluss von Ungenauigkeiten oder Unterschieden in den Annotationen zu vermeiden, ist die Verwendung einer niedrigen IoU-Schwelle von z.B. 0,3 sinnvoll.

Auch de Souza et al. [47] haben festgestellt, dass für den Anwendungsfall „*Deep Action Recognition*“ die Kombination einer großen Menge synthetischer Daten mit wenigen realen Daten die Leistung erhöhen kann. Ähnliche Ansätze aus [101, 108] verwenden aus dem gleichem Grund synthetische Daten zum Vortrainieren der Netze und reale Daten für das Fine-Tuning.

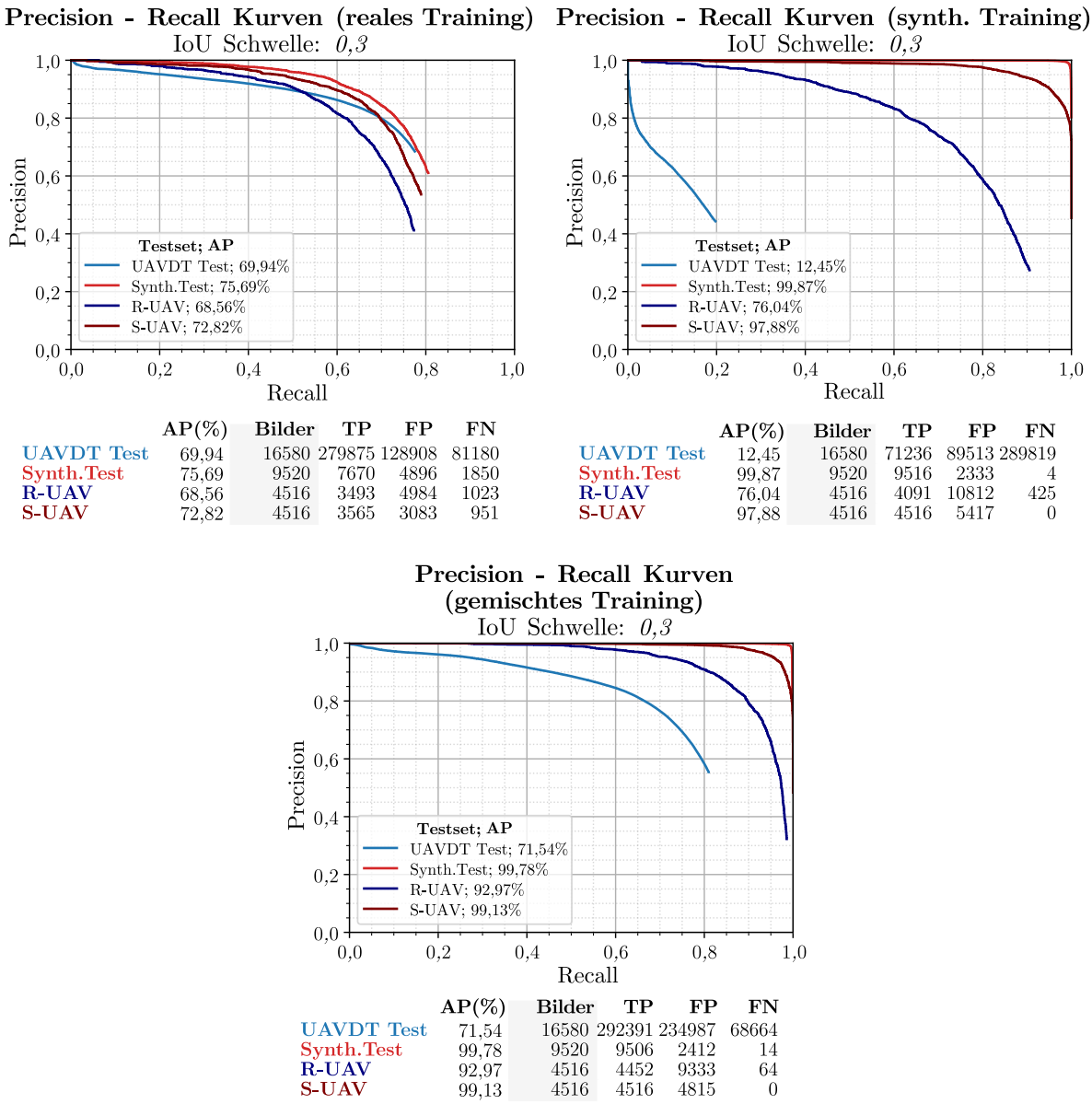
6.1.4 Auswertung der Konfigurationen auf realen und synthetischen Bildpaaren

Im Gegensatz zu den meisten bisher vorgestellten Veröffentlichungen werden im Folgenden die Leistungsunterschiede zusätzlich auf gekoppelten realen und synthetischen Bildpaaren untersucht (vgl. [229]). Tab. 26 wiederholt die Forschungsfrage und beschreibt das in diesem Kapitel betrachtete Experiment mit der zugehörigen Auswertung.

Tab. 26 Tabellarische Übersicht über die jeweils behandelte Forschungsfragestellung, das zugehörige Experiment und die einzelnen Bestandteile der Auswertung

4.	Wie unterscheidet sich die Leistung dieser drei Trainingskonfigurationen auf inhaltsgleichen realen und synthetischen Bildduplikaten?
	Experiment: Anwendung der verschiedenen Modelle (reales, synthetisches, gemischtes Training) auf die realen und synthetischen Bildpaare als Testdaten
	Auswertung: Beurteilung und Vergleich der Detektionsleistung - auf zugehörigen realen und synthetischen Testdaten - auf unabhängigen, inhaltsgleichen realen und synthetischen Bildpaaren

Die Bildpaare ermöglichen durch den identischen Bildinhalt eine detailliertere Analyse des *Reality Gaps* und kommen dem späteren Anwendungsfall deutlich näher als die Verwendung zugehöriger Testdatensätze. Dies erlaubt den Vergleich der Detektionsleistung in Bezug auf den Leistungsunterschied zwischen verschiedenen Domänen und gleichzeitig in Bezug auf den Leistungsunterschied zwischen verschiedenen Datensätzen der gleichen Domäne. Die in den vorigen Kapiteln beschriebenen realen, synthetischen und gemischten Trainingskonfigurationen dienen dafür als Ausgangspunkt.



	AP(%)	Bilder	TP	FP	FN
UAVDT Test	69,94	16580	279875	128908	81180
Synth.Test	75,69	9520	7670	4896	1850
R-UAV	68,56	4516	3493	4984	1023
S-UAV	72,82	4516	3565	3083	951

	AP(%)	Bilder	TP	FP	FN
UAVDT Test	12,45	16580	71236	89513	289819
Synth.Test	99,87	9520	9516	2333	4
R-UAV	76,04	4516	4091	10812	425
S-UAV	97,88	4516	4516	5417	0

	AP(%)	Bilder	TP	FP	FN
UAVDT Test	71,54	16580	292391	234987	68664
Synth.Test	99,78	9520	9506	2412	14
R-UAV	92,97	4516	4452	9333	64
S-UAV	99,13	4516	4516	4815	0

Abb. 67 Vergleichende Analyse der Detektionsleistungen für verschiedene Trainingskonfigurationen (real, synthetisch, gemischt) auf den zugehörigen realen und synthetischen Testdatensätzen (UAVDT, Synth. Test) und den gekoppelten realen und synthetischen Bildpaaren (R-/S-UAV). Die jeweils erzielte AP ist in der Legende und in der Tabelle zusammen mit der Anzahl an Testbildern und den TP, FP und FN Detektionen angegeben.

Abb. 67 visualisiert die Leistungsvergleiche für die betrachteten Konfigurationen. Die Evaluierung findet jeweils auf den realen UAVDT Testdaten, den synthetischen Testdaten und den realen und synthetischen Bildpaaren statt (R-/S-UAV).

6.1.4.1 Real trainiertes Modell

Die erste Grafik in Abb. 67 zeigt als Ausgangspunkt die Ergebnisse für rein reale Trainingsdaten. Der ähnliche Verlauf der PR-Kurven und die ähnlichen AP-Werte deuten darauf hin, dass diese Konfiguration zu einem allgemein anwendbaren Detektormodell führt, das unabhängig von der Domäne und dem Testdatensatz eine ähnliche Detektionsleistung erzielt. Mit den in den realen Benchmark-Daten enthaltenen und vom Netzwerk angelernten Merkmalen kann wie bereits in Kapitel 6.1.1 beschrieben auch auf den synthetischen Testdaten trotz der unterschiedlichen Domäne eine ähnliche bzw. aufgrund reduzierter Störeffekte sogar leicht bessere Leistung erzielt werden. Dies bedeutet, dass der *Reality Gap* in diese Richtung gering ist. Der Leistungsunterschied zwischen den beiden synthetischen Datensätzen (Synth. Test ↔ S-UAV) und den aus derselben Domäne stammenden realen Testdaten (UAVDT Test ↔ R-UAV) ist ebenfalls relativ gering.

Es ist dennoch erwähnenswert, dass auf den R-UAV Daten, die aus der gleichen Domäne stammen wie die Trainingsdaten, der geringste AP-Wert erzielt wird, da die *Precision* der Detektionen bei höheren *Recall*-Werten vergleichsweise stark abfällt. Dies deutet darauf hin, dass die Detektion unbekannter und von den Trainingsdaten unabhängiger realer Bilddaten mit teilweise leicht unterschiedlichen Szenerien und Objekten die komplexeste Form der Evaluierung darstellt. Dieser Fall kommt der späteren Anwendung am nächsten und sollte bei der Leistungsbeurteilung von Detektormodellen berücksichtigt werden.

6.1.4.2 Synthetisch trainiertes Modell

Die zweite Grafik in Abb. 67 visualisiert die Leistung des rein synthetisch trainierten Detektors. Trotz Verwendung der gleichen Testdatensätze sind dabei im Vergleich zum realen Training die Ergebnisse sehr unterschiedlich und die PR-Kurven über einen sehr weiten Bereich verteilt.

Während die Leistung auf den zugehörigen synthetischen Testdaten nahezu ideal ist (AP = 99,9 %), ist mit dem angelernten Modell eine Detektion auf den realen UAVDT Testdaten nicht oder nur sehr unzureichend möglich (AP = 12,5 %). In Kapitel 6.1.2 wurden bereits einige mögliche Gründe für diesen sehr deutlichen *Reality Gap* angesprochen, wie z.B. unterschiedliche Szenerien, unterschiedliche zahlenmäßige Datensatzzusammensetzungen oder eine Überanpassung des Modells auf die synthetischen Merkmale.

In diesem Kontext ist die Betrachtung der Detektionsleistung auf den realen und synthetischen Bildpaare nun besonders interessant. Auf den synthetischen Duplikaten wurde eine ebenfalls sehr hohe AP von 97,9 % erzielt, da diese mit derselben Simulationsumgebung erzeugt wurden und ähnliche Parameter und geografische Szenarien beinhalten. Der Leistungsabfall um zwei Prozentpunkte gegenüber den synthetischen Testdaten stammt sehr wahrscheinlich von der Tatsache, dass die Testfahrzeuge des S-UAV Datensatzes in Bezug auf Farbe und Modell in den synthetischen Trainingsdaten unterrepräsentiert waren. Insgesamt gesehen ist der Leistungsunterschied zwischen den beiden synthetischen Testdaten bei synthetischem Training aufgrund der dennoch hohen Ähnlichkeit gering. Betrachtet man nun die reale Domäne, sticht heraus, dass trotz des rein synthetischen Trainings auf den realen R-UAV Testdaten eine sehr gute AP von 76,0 % erreicht wird, während das Modell auf den ebenfalls realen UAVDT Testdaten nur sehr niedrige 12,5 % erreicht. Dies ist mit großer Wahrscheinlichkeit darauf zurückzuführen, dass die synthetischen Trainingsdaten sehr ähnliche Szenarien und damit sehr ähnliche Objekt-, Kontext- und Umgebungsparameter enthalten wie die R-UAV Daten, da sie zu einem Teil auf dem nachmodellierten Testfluggelände generiert wurden (s. Verteilung der Aufnahmepositionen in Abb. 46).

Durch diese gezielte Betrachtung der verschiedenen Testdaten und Bildpaare wird deutlich, dass der sehr große Leistungsunterschied bzw. *Reality Gap* zwischen den realen UAVDT Testdaten und den synthetischen Testdaten (AP: 12,5 % ↔ 99,9 %) aus mehreren Bestandteilen besteht, die den einzelnen Flächen zwischen den PR-Kurven zugeordnet werden können. Der erste davon ist unabhängig von der

Domäne und dem synthetischen Rendering-Prozess und wird lediglich durch den Bildinhalt beeinflusst. Er wird daher auch als *Content Gap* bezeichnet und entspricht der Leistungsdifferenz zwischen dem UAVDT Testsatz und den R-UAV Bildern (AP: 12,5 % ↔ 76,0 %). Diese stammen zwar beide aus der realen Domäne, aber nur letztere sind in Bezug auf den Bildinhalt ähnlich zu den hier verwendeten synthetischen Trainingsdaten.

Der zweite Bestandteil beschreibt den Leistungsunterschied zwischen den realen und den synthetischen Bildpaaren (AP: 76,0 % ↔ 97,9 %). Diese weisen durch die möglichst exakte Nachmodellierung einen nahezu identischen Bildinhalt auf, was wiederum bedeutet, dass der Leistungsunterschied ausschließlich durch die synthetische Bildgenerierung und das Rendering zustande kommt. Aus diesem Grund wird dieser Anteil auch als *Appearance Gap* bezeichnet. Der letzte Bestandteil beschreibt wiederum einen *Content Gap* und drückt sich durch den Abstand der PR-Kurven von S-UAV und synthetischem Testdatensatz aus (AP: 97,9 % ↔ 99,9 %). Aufgrund des sehr ähnlichen Bildinhalts fällt dieser jedoch sehr gering aus.

6.1.4.3 Gemischt trainiertes Modell

Die letzte Grafik in Abb. 67 zeigt die im vorigen Kapitel beschriebene Trainingskonfiguration mit gemischten Trainingsdaten. Von besonderem Interesse ist erneut die Betrachtung der realen und synthetischen Bildpaare. Zwar ist der Unterschied in Bezug auf die Detektionsleistung bei den verschiedenen Testdatensätzen geringer, die Rangfolge der PR-Kurven bleibt jedoch identisch wie bereits beim rein synthetischen Training. Dies bestätigt die Annahme, dass sich der *Reality Gap* in einen *Content Gap* und einen *Appearance Gap* aufspaltet und beide Bestandteile nach wie vor vorhanden sind, jedoch in einer stark verminderten Form.

In diesem Zusammenhang ist auch die jeweilige Veränderung der Detektionsleistung im Vergleich zum realen Training von Interesse. Tab. 27 liefert einen Überblick über die erreichten Werte und die Veränderungen. Das bereits erwähnte stabilere Trainingsverhalten und die höhere Generalisationsfähigkeit führen bei allen Testdaten zu einem teilweise sogar sehr deutlichen Anstieg der AP-Werte (s. auch Tab. 27). In Abb. 67 wird diese gesteigerte Generalisationsfähigkeit vor allem durch den Vergleich der erreichten *Recall*-Werte deutlich. Diese sind beim gemischten Training deutlich höher, was wiederum bedeutet, dass ein größerer Anteil der in den Testdaten vorkommenden Fahrzeuge vom Detektor korrekt erkannt werden kann. Wie bereits im vorigen Kapitel beschrieben, kann durch die Zumischung synthetischer Daten die Detektionsleistung auf den UAVDT Testdaten nur um ca. 1,6 Prozentpunkte gegenüber der ausschließlichen Verwendung der rein realen UAVDT Trainingsdaten gesteigert werden (s. Tab. 27), da sich UAVDT Trainings- und Testdaten in Bezug auf Szenerie und Objekt-, Kontext- und Umgebungsparameter bereits sehr stark ähneln.

Im Gegensatz dazu steigt bei den R-UAV Testdaten der AP-Wert sehr deutlich um 24,4 Prozentpunkte an, obwohl bei realem Training auf UAVDT und R-UAV Testdaten eine ähnliche Leistung erzielt wird. Dieses Ergebnis ist in mehrerlei Hinsicht hervorzuheben. Es zeigt zum einen, dass durchaus deutliche Leistungsunterschiede zwischen Testdatensätzen aus der gleichen Domäne möglich sind und zum anderen, dass durch passende Beimischung synthetischer Daten die Detektionsleistung für eine spätere Anwendung deutlich erhöht werden kann. Dabei ist zu beachten, dass die beigefügten synthetischen Trainingsdaten unter anderem am nachmodellierten Testfluggelände erstellt wurden, das auch für die zur Erstellung des R-UAV Testdatensatzes durchgeführten realen Flüge genutzt wurde. Dadurch enthalten die synthetischen Trainingsdaten ähnliche Umgebungen und Szenarien wie die R-UAV Testdaten. Insgesamt zeigt diese Konstellation, dass eine selektive Erweiterung allgemeiner realer Benchmark Trainingsdaten mit synthetischen Daten aus dem späteren Einsatzgebiet zu einer signifikant besseren Anpassung des Detektormodells auf die vorkommenden Einsatzbedingungen führt.

Tab. 27 Vergleich der Detektionsleistung für verschiedene Trainingskonfigurationen auf verschiedenen Testdatensätzen auf Basis der AP. Zusätzlich ist die jeweilige Veränderung der AP im Vergleich zum realen Training angegeben, das als Ausgangspunkt dient. Die höchste erreichte Detektionsleistung für den jeweilige Testdatensatz ist fett gedruckt.

	UAVDT Test		Synth. Test		R-UAV		S-UAV	
Reales Training	69,9 %		75,7 %		68,6 %		72,8 %	
Synth. Training	12,5 %	- 57,4	**99,9 %**	+ 24,2	76,0 %	+ 7,4	97,9 %	+ 25,1
Gemischtes Training	**71,5 %**	+ 1,6	99,8 %	+ 24,1	**93,0 %**	+ 24,4	**99,1 %**	+ 26,3

6.1.4.4 Zusammenfassung und Interpretation

Diese Erkenntnisse decken sich mit der Literatur, in der ebenfalls synthetische Daten für unterschiedlichste Anwendungen verwendet werden. Auch in [106] hat sich gezeigt, dass Differenzen zwischen verschiedenen Datensätzen derselben Domäne eine ähnliche Größenordnung haben als die Differenzen zwischen den Domänen. In [34, 46, 96, 101] wurde übereinkommend festgestellt, dass sowohl für den Fall der Objektdetektion als auch bei der semantischen Segmentierung eine Erweiterung mit synthetischen Trainingsdaten zu einer Leistungssteigerung führt. Ob lediglich synthetische Daten hinzugemischt werden, Methoden der *Domain Adaptation* genutzt werden oder die Modelle auf synthetischen Daten vortrainiert werden, scheint im ersten Schritt für den Effekt keine Rolle zu spielen. Kar et al [113] beobachteten ebenfalls eine Aufteilung in *Appearance Gap* und *Content Gap* und versuchten daher die Wahl der Attribute des Szenengraphen und dessen Zusammensetzung bei der synthetischen Datengenerierung zu optimieren. Im Gegensatz zu der hier vorgestellten Arbeit wird dabei jedoch nur der *Content Gap* berücksichtigt und es wird kein systematisches automatisiertes Auswerteverfahren zur Einflussanalyse angewandt, sondern lediglich die subjektive menschliche Beurteilung anhand von Beispielbildern.

Insgesamt hat sich gezeigt, dass nur durch die Analyse gekoppelter realer und synthetischer Bildpaare eine umfassende Beurteilung der grundlegenden Zusammenhänge möglich ist:

Das rein reale Training lieferte dabei ein allgemein anwendbares Detektormodell mit geringen Leistungsunterschieden zwischen den Testdaten. Ein willkürliches synthetisches Training ist für die Anwendung auf komplexen unabhängigen realen Daten nicht ausreichend. Stammen die synthetischen Trainingsdaten jedoch aus demselben Anwendungsgebiet wie die realen Testdaten, ist durchaus eine sehr gute Detektionsleistung mit rein synthetischem Training möglich. Durch die Verwendung gekoppelter Bildpaare konnte gezeigt werden, dass sich der *Reality Gap* aus einem *Content Gap* und einem *Appearance Gap* zusammensetzt.

Das bedeutet auch, dass bei der Evaluierung berücksichtigt werden muss, dass sowohl die Domäne als auch die Szenerie einen Einfluss auf die Detektionsleistung haben, wodurch nur mit unabhängigen realen Bilddaten des späteren Anwendungsfalles eine sehr zuverlässige Evaluierung eines Modells möglich ist. Ein mit gemischten Trainingsdaten trainiertes Modell liefert aufgrund der höheren Generalisationsfähigkeit auf allen Testdaten die höchste Detektionsleistung.

Durch gezielte synthetische Erweiterung vorhandener realer Trainingsdaten kann ein Modell sehr effektiv auf den späteren Einsatzzweck hin optimiert und angepasst werden und erzielt dann in real unbekannten Umgebungen und Situationen eine deutlich bessere Detektionsgüte als bei ausschließlich realem Training mit allgemeinen Benchmark-Daten.

6.2 Statistische Auswertung der Einflussfaktoren auf die Detektion

Nun folgt eine gezielte statistische Auswertung der Ursachen der im vorigen Kapitel beobachteten Detektionsunterschiede. Es wird die Fragestellung behandelt, wie eine **statistische Auswertung** zur

Identifikation relevanter Bildunterschiede zwischen realen und synthetischen Sensordaten und zur **Identifikation der Einflussfaktoren** der durch die Bildunterschiede hervorgerufenen **Leistungsunterschiede** vorgenommen werden kann (zugehörige Forschungsfragen siehe Tab. 28).

Tab. 28 Wiederholung der Forschungsfragen zur statistischen Auswertung der Einflussfaktoren

Statistische Auswertung - Forschungsfragen	
1.	Mit welcher Zuverlässigkeit kann mit Hilfe von Bildbeschreibern zwischen **realen und synthetischen Bildpaaren** unterschieden werden und welche Einflussfaktoren sind dabei entscheidend?
2.	Welche Zuverlässigkeit erreicht der Ansatz bei der Unterscheidung zwischen **voneinander unabhängigen realen und synthetischen Trainings- und Testdaten** und welche Einflussfaktoren sind dabei entscheidend**?**
3.	Wie hoch ist die Güte der Klassifikation in **korrekte und inkorrekte Detektionsergebnisse** und welche Einflussfaktoren sind dabei entscheidend**?**

Grundlage dafür bildet die Extraktion geeigneter Merkmale als unabhängige Variablen zur Beschreibung der für die Bild- und Leistungsunterschiede verantwortlichen Faktoren. Dies wird durch ein ausgewähltes Set an Bildschreibern bewerkstelligt, das in Kapitel 4.4 näher beschrieben wurde und die jeweilige Bildinformation charakterisiert. Dieses Set dient als Eingangsgröße für die statistische Auswertekette. Als Methoden für die Auswertung kommen dabei sowohl Regressionsverfahren als auch Klassifikationsalgorithmen in Frage, wobei der Fokus auf letzteren liegt, da damit bessere Ergebnisse erzielt werden konnten und diese universeller einsetzbar sind (s. Kapitel 4.5). In beiden Fällen wird im ersten Schritt anhand von Trainingsdaten das Modell bestimmt. Anschließend wird mit Hilfe von Testdaten validiert, wie gut das Modell die Zusammenhänge beschreibt, bevor schließlich im letzten Schritt der Kette die für die Modellbildung relevanten Merkmale identifiziert werden.

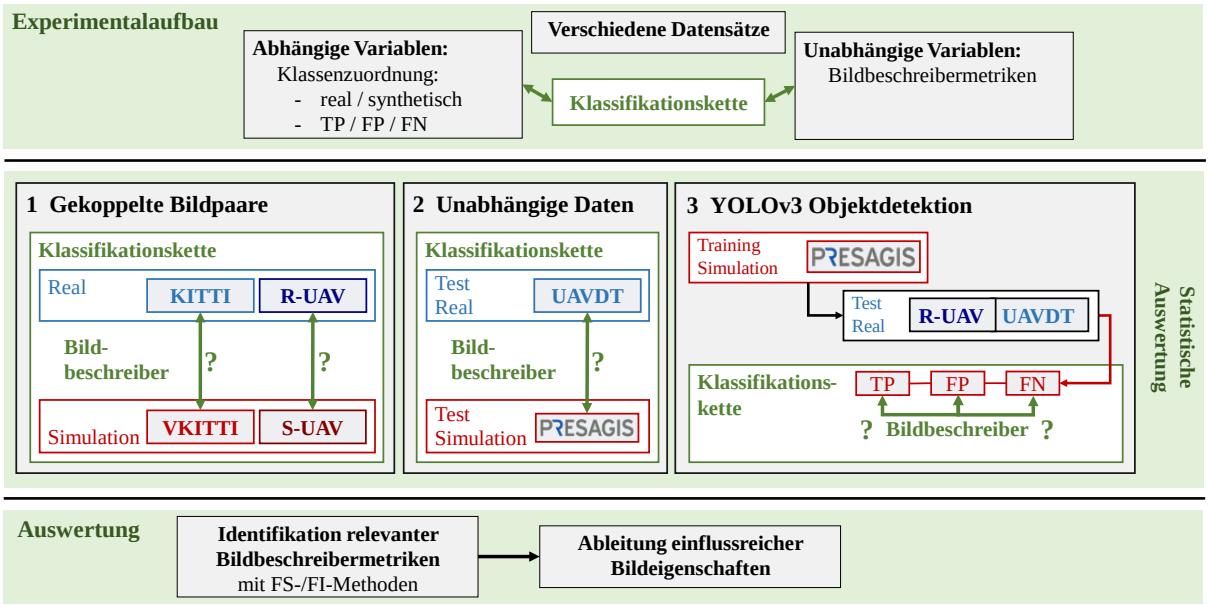


Abb. 68 Konzeptgrafik zur statistischen Auswertung: Übersicht über die mit Hilfe der Bildbeschreiber und der Klassifikationskette analysierten Zusammenhänge, die im zweiten Teil der Untersuchung für die Identifikation der relevanten Einflussparameter verwendet werden.

Die Konzeptgrafik in Abb. 68 gibt einen Überblick über die genauen Zusammenhänge, die auf diese Weise analysiert werden sollen. Dieser Block entspricht den Ebenen 3 und 4 des Gesamtkonzepts aus Abb. 4 und ist entsprechend farbig gekennzeichnet.

Insgesamt können drei Teilbereiche unterschieden werden:

- Im ersten wird untersucht, inwieweit es möglich ist, mit dem beschriebenen Verfahren und den ausgewählten Bildbeschreibern zwischen gekoppelten realen und synthetischen Bildpaaren zu

unterscheiden und so hauptsächlich die für den *Appearance Gap* relevanten Bildeigenschaften zu identifizieren.

- Im nächsten Teil wird dieser Ansatz auf voneinander unabhängige reale und synthetische Datensätze erweitert, wodurch der *Reality Gap* in seiner Gesamtheit untersucht werden kann. Diese beiden Teile dienen der Analyse der Bildunterschiede.
- Im letzten Abschnitt wird schließlich versucht, darauf aufbauend die Leistungsunterschiede genauer zu betrachten. Dazu werden mit Hilfe der Bildbeschreiber die Bildeigenschaften der vom synthetisch trainierten Detektor gelieferten *Bounding Boxen* beschrieben, bevor diese anschließend auf Basis der dadurch extrahierten Bildinformation in korrekte (TP: *True Positive*) oder inkorrekte (FP: *False Positive*, FN: *False Negative*) *Bounding Boxen* klassifiziert werden sollen.

Insgesamt soll dadurch eine Möglichkeit geschaffen werden, die im ersten Teil der Auswertung beschriebenen *Black-Box* Detektormodelle und die zugehörigen Leistungsunterschiede zu analysieren und dabei Bildeigenschaften zu identifizieren, die inkorrekte Detektionen begünstigen.

Dieses Vorgehen ermöglicht eine umfassende, unabhängige und detaillierte Analyse der Einflussfaktoren, die auch auf andere Anwendungsfälle übertragen werden kann und den *Reality Gap* in seiner Gesamtheit analysiert, d.h. sowohl in Bezug auf die Bildunterschiede als auch in Bezug auf die Leistungsunterschiede. Aufgrund erweiterter und aktualisierter Datensätze unterscheiden sich die im Folgenden vorgestellten Daten geringfügig von den in [188] veröffentlichten Werten. Die daraus abgeleiteten Aussagen bleiben jedoch weitestgehend unverändert.

6.2.1 Voruntersuchungen auf Basis einer Regressionsanalyse

Im ersten Teil der Untersuchungen soll nun analysiert werden, inwiefern eine Regressionsanalyse, wie in Kapitel 4.5.1 beschrieben, zur Identifikation relevanter Einflussfaktoren eingesetzt werden kann. Diese Art der Auswertung basiert auf der Verwendung gekoppelter Bildpaare (R-UAV und S-UAV Datensatz) und dem rein synthetisch trainierten Detektormodell.

Die davon paarweise berechneten Differenzen der Bildbeschreibermetriken bilden als unabhängige Variablen die Datenmatrix $\boldsymbol{X}$, welche in 4516 Zeilen für jedes Bildpaar des Datensatzes die 107 Merkmalsdifferenzen enthält. Diese Differenzen sind gleichmäßig über das Datenset verteilt, d.h. es sind keine systematischen Gruppierungen in der verwendeten Reihenfolge der Bildpaare zu erkennen. Die Vorschrift zur Berechnung der Distanz zwischen den Vektoren der MPEG7 Bildbeschreiber ist im zugehörigen Standard festgelegt und liefert auf Basis der Manhattan-Distanz bzw. der Euklidischen Distanz je einen Wert pro Bildbeschreiber.

In Abb. 69 ist die Verteilung dieser sieben Werte visualisiert, welche bei der Berechnung automatisch normiert werden und somit im Bereich zwischen Null und Eins liegen. Es wird deutlich, dass sich die Bildpaare hauptsächlich in Bezug auf CLD, CSD unterscheiden, geringfügig in Bezug auf EHD und HTD und sich sehr ähnlich sind in Bezug auf DCD und SCD. Die Differenzen der weiteren zusätzlichen Bildbeschreiber (s. Tab. 10) bilden die restlichen 100 Werte der Matrix. Bei einzelnen Eigenschaften, wie z.B. der Helligkeit, können die jeweiligen Werte direkt subtrahiert werden, bei mehrdimensionalen Eigenschaften wird wiederum die Manhattan-Distanz berechnet. Zusätzlich wurden bei jeder Gruppe zusammengehöriger Bildeigenschaften zur Beschreibung der Gesamtdifferenz durch einen Wert weitere Differenzmetriken wie eine standardisierte Summe der Einzeldifferenzen, die *Kosinusähnlichkeit* mit und ohne Standardisierung und die *Pearson Korrelation* zwischen den Bildbeschreibervektoren mit in die Datenmatrix aufgenommen. Die *Kosinusähnlichkeit* beschreibt die Ähnlichkeit zweier Merkmalsvektoren anhand ihrer Ausrichtung und wird zum Beispiel beim Vergleich von Dokumenten verwendet.

Abb. 69 zeigt wiederum deren Verteilung für die jeweiligen Gruppierungen an Bildbeschreibern anhand eines Violinplots. Die deutlichsten Unterschiede sind im Bereich BLC und ET zu beobachten. Die kategorischen Variablen aus der Bildbeschreibergruppe für Umweltbedingungen (Env-Vektor) führen zu einer Verzerrung der Berechnung, weshalb die Darstellung in diesem Bereich nur eine geringe Aussagekraft besitzt. Interessant ist jedoch, dass sich die Bildpaare in Bezug auf die Bildbeschreibermetriken aus der Gruppe „Sha“ (s. Kapitel 4.4.2) sehr stark ähneln. Dies ist durchaus plausibel, da die in der Gruppe „Sha“ zusammengefassten Metriken durch Segmentierungsansätze die Zusammensetzung der Szenerie beschreiben, welche per Definition bei den gekoppelten realen und synthetischen Bildpaaren möglichst identisch sein sollte.

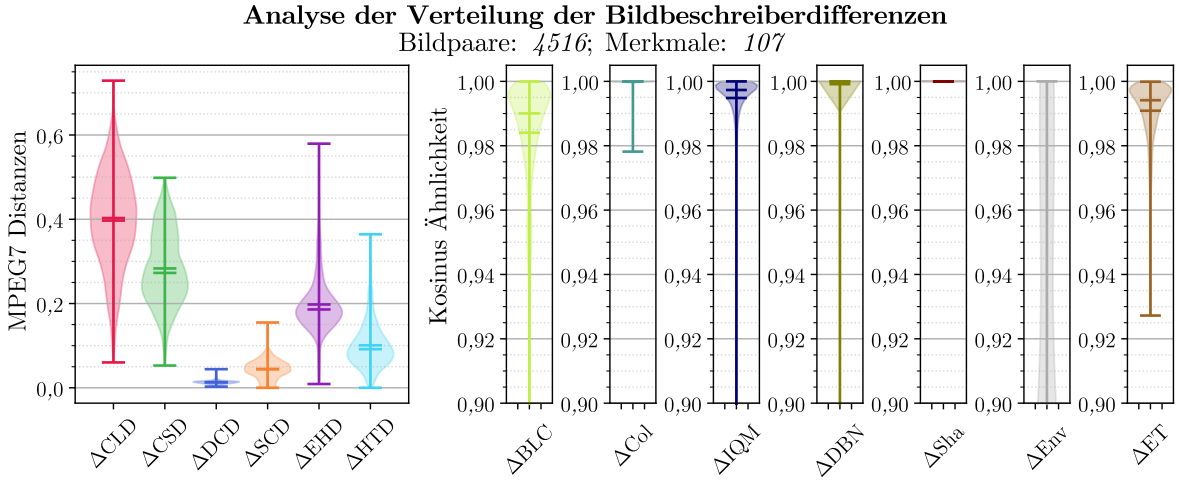


Abb. 69 Visualisierung der Verteilung der paarweisen Differenzen der Bildbeschreiber für die jeweilige Gruppe. Differenzen innerhalb jeder Gruppe an Bildbeschreibern sind jeweils in Form eines Histogramms horizontal aufgetragen, wobei zusätzlich Median und Mittelwert eingezeichnet sind.
Datensatz: R-UAV/S-UAV.

Den zweiten Bestandteil der Regressionsanalyse bildet der Zielgrößenvektor $\boldsymbol{y}$, der als abhängige Variablen die vorhandenen Leistungsunterschiede zwischen den Bildpaaren enthält. Diese können durch mehrere Metriken ausgedrückt werden (s. Kapitel 4.3.3). Anders als bei der Evaluierung ganzer Datensätze steht für die Berechnung nur jeweils ein Bild zur Verfügung, was sich negativ auf die Zuverlässigkeit der resultierenden Werte auswirken kann. Die sonst üblicherweise verwendete AP ist daher nicht geeignet, da sie auf der PR-Kurve beruht, zu deren Berechnung eine Vielzahl von Detektionen nötig sind. Um dennoch die Genauigkeit und die Sensitivität des Detektors zu berücksichtigen, wurde der F1-Score für die Auswertung herangezogen.

Die resultierenden Leistungsunterschiede und deren Verteilung ist in Abb. 70 dargestellt. Im linken Teil der Grafik zeigt sich, dass die erreichte Detektionsleistung auf realen und synthetischen Daten relativ gleichmäßig über den Datensatz verteilt ist. Die Gesamtverteilung offenbart bei synthetischen Daten eine deutliche Anhäufung des F1-Scores bei sehr hohen Detektionsleistungen, während diese bei den realen Daten mehr im mittleren Bereich angesiedelt ist. Die resultierende Differenz ist daher meist positiv und relativ gleichmäßig über den mittleren positiven Bereich verteilt, wobei auch ein deutlicher Anteil an Bildpaaren vorhanden ist, der nur einen geringen bzw. überhaupt keinen Leistungsunterschied aufweist.

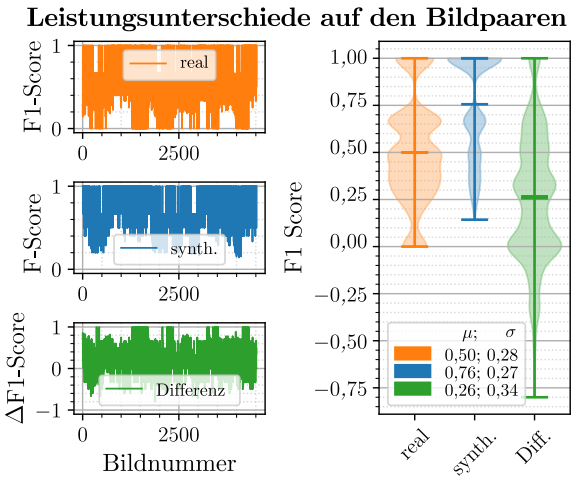


Abb. 70 Übersicht zum Verlauf der Detektionsleistung und deren Differenz über den Datensatz der Bildpaare. Im rechten Teil ist ergänzend die Gesamtverteilung der jeweiligen F1-Scores dargestellt.
Datensatz: R-UAV/S-UAV; Detektor: synth. trainiert.

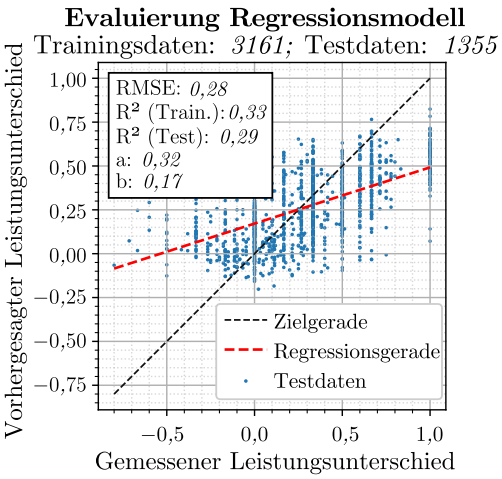


Abb. 71 *Predicted-vs.-Measured-Plot* zur Evaluierung des multiplen linearen Regressionsmodells zusammen mit den berechneten Gütekriterien.
Datensatz: R-UAV/S-UAV;
Detektor: synth. trainiert.

Ziel der multiplen linearen Regression (MLR) ist nun die Bestimmung eines passenden Regressionsvektors $\boldsymbol{b}$, der versucht, die resultierenden Leistungsunterschiede durch die Bildbeschreiberdifferenzen zu erklären und einen Zusammenhang zwischen beiden herzustellen. Abb. 71 zeigt die grafische Evaluierung des berechneten Regressionsmodells. Sowohl bei den Trainings- als auch bei den Testdaten wird lediglich ein sehr niedriges Bestimmtheitsmaß $R^2 = 0{,}33$ bzw. $0{,}29$ erreicht.

Zusammenfassung und Interpretation

Da nur ein niedriges Bestimmtheitsmaß erreicht wird, ist das Modell nicht in der Lage, aus den Bildbeschreiberdifferenzen die Differenzen des F1-Scores zuverlässig abzuleiten. Auf eine Auswertung des Regressionsvektors und der zugrundeliegenden Einflussfaktoren wurde daher aufgrund der zu geringen Aussagekraft an dieser Stelle verzichtet. Eine Regressionsanalyse ist somit für den hier betrachteten Anwendungsfall nicht als Auswertemethode geeignet.

Es kommen mehrere Gründe als Erklärung für dieses Ergebnis in Frage. Einer der Hauptgründe ist sicherlich die Tatsache, dass vor allem bei rein synthetischem Training der allgemeine Leistungsunterschied zwischen den Bildpaaren nicht auf einzelne globale Bildeigenschaften zurückzuführen ist, sondern auch in einem gewissen Maß durch die Trainingskonfiguration verursacht wird, wodurch es zu einer Verzerrung der Eingangsdaten kommt. Darüber hinaus betrachtet der Detektionsalgorithmus einzelne lokale Regionen im Bild und ermittelt dafür mögliche ROIs für die Objektklassifikation. Die globale Berechnung der Bildbeschreiberdifferenzen über das gesamte Bild erfasst und beschreibt daher nicht die benötigten lokalen Merkmale, die für eine gute Regression auf die Detekorleistung nötig wären. Wie bereits erwähnt führt die sehr geringe Anzahl an *Bounding Boxen* pro Bild zusätzlich zu einer tendenziell instabilen Berechnung der Leistungsmetrik, was ebenfalls zu Fehlertermen führen kann und die Aussagekraft der Methode einschränkt.

Aus diesen Gründen wird in der vorliegenden Arbeit ein alternativer Ansatz auf Grundlage einer Klassifikationskette bevorzugt (s. Kapitel 4.5.2). Dieser hat zudem den Vorteil, dass er nicht nur auf Bildpaare beschränkt ist und daher sowohl die Auswertung von Bildunterschieden zwischen unabhängigen Testdatensätzen ermöglicht als auch die Auswertung von Leistungsunterschieden durch eine Klassifikation in korrekte und inkorrekte *Bounding Boxen*. Als Datenbasis können im Gegensatz zur Regression

die absoluten Werte der Bildbeschreiber verwendet werden, was wiederum eine Verzerrung des Modells durch die Differenzbildung ausschließt.

6.2.2 Klassifikation realer und synthetischer Bildpaare

Im Folgenden wird nun die in Kapitel 4.5.2 beschriebene Klassifikationskette zur Identifikation relevanter Einflussfaktoren verwendet. Als Grundlage für die weiteren Untersuchungen wird im ersten Schritt analysiert, inwiefern mit dieser Methode auf Basis der berechneten Bildbeschreiber zwischen realen und synthetischen Bildpaaren unterschieden werden kann und welche Bildeigenschaften für die Entscheidung relevant sind. Tab. 29 wiederholt die Forschungsfrage und beschreibt das in diesem Kapitel betrachtete Experiment mit der zugehörigen Auswertung.

Tab. 29 Tabellarische Übersicht über die jeweils behandelte Forschungsfragestellung, das zugehörige Experiment und die einzelnen Bestandteile der Auswertung

1.	Mit welcher Zuverlässigkeit kann mit Hilfe von Bildbeschreibern zwischen realen und synthetischen Bildpaaren unterschieden werden und welche Einflussfaktoren sind dabei entscheidend?
Experiment:	Klassifikationsanalyse mit dem Ziel der Unterscheidung der Domänen (real / synthetisch) auf Basis von Bildbeschreiberwerten
Auswertung:	1. Beurteilung und Analyse der Klassifikationsgüte (F1-Score) 2. Identifikation der einflussreichen Bildbeschreiber (FS- / FI-Methoden) - für den KITTI / VKITTI Datensatz - für den R-UAV / S-UAV Datensatz → Ableitung relevanter Bildeigenschaften und Gestaltungsmerkmale

Die Auswertung soll in erster Linie Aufschluss darüber geben, wie sich inhaltlich identische Sensordaten beider Domänen in Bezug auf ihre visuelle Erscheinung voneinander unterscheiden. Als Datensätze wurden dabei sowohl die real erflogenen und synthetisch nachgebildeten R-UAV und S-UAV Bildpaare betrachtet als auch die KITTI und VKITTI Daten, die als Benchmark Bildpaare aus dem Umfeld des autonomen Fahrens als Vergleich dienen. Die absoluten Werte der Bildbeschreiber bilden als unabhängige Variablen die Datenmatrix. Ziel der Methode ist es, ein beliebiges Bild aufgrund dieser Eingangsdaten der entsprechenden Domäne zuzuordnen, die als abhängige Variable den Zielgrößenvektor bildet, und anschließend mit *Feature Selection* und *Feature Importance* Methoden (s. Kapitel 4.5.2.4 und 4.5.2.5) die für diese Klassifikation relevanten Merkmale zu identifizieren.

Grundlegende Analysen

Obwohl diese Klassifikationsanalyse die absoluten Werte der Bildbeschreiber verwendet, sind in Abb. 72 zum Vergleich mit Abb. 69 die Differenzen zwischen den KITTI und VKITTI Bildpaaren dargestellt. Es zeigt sich, dass in beiden Fällen sowohl in Bezug auf die MPEG7 Distanzen als auch in Bezug auf die übrigen Gruppierungen trotz unterschiedlichem Anwendungsfall und anderer zugrundeliegender *Render-Engine* eine sehr hohe Übereinstimmung vorliegt und ähnliche Effekte beobachtet werden können, die bereits im vorigen Kapitel beschrieben wurden.

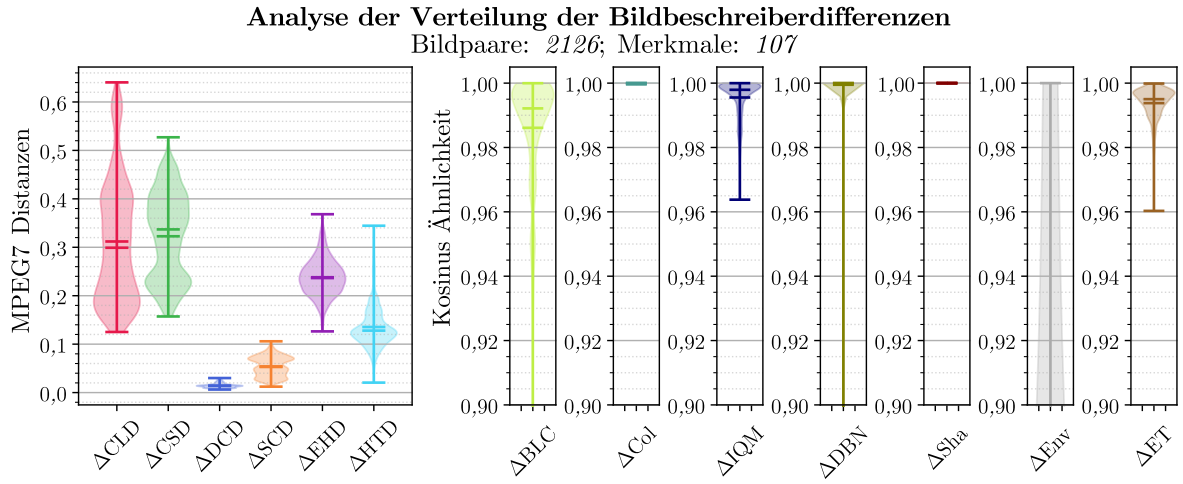


Abb. 72 Visualisierung der Verteilung der paarweisen Differenzen der Bildbeschreiber für die jeweilige Gruppe. Datensatz: KITTI/VKITTI.

Bei der Vorverarbeitung wird jeweils die Multikollinearität in den Eingangsdaten untersucht und im Zuge einer separaten FS durch die Entfernung konstanter, quasi-konstanter und doppelt vorhandener Merkmalsvektoren reduziert. Abb. 73 zeigt als Maß für die Multikollinearität die Korrelationsmatrix und die VIF Werte für den KITTI/VKITTI Datensatz vor und nach der FS. Es zeigt sich, dass in diesem Fall trotz der Vorverarbeitung eine teilweise sehr hohe Multikollinearität in den Daten vorhanden ist, d.h. einzelne Merkmalsvektoren sind stark miteinander korreliert und können durch die Linearkombination anderer Merkmale vorhergesagt werden. Dennoch ist eine Reduktion durch die Anwendung weitreichenderer FS-Methoden nicht sinnvoll, da diese dazu führen würden, dass die spätere Interpretation von der Reihenfolge der Merkmale in der Datenmatrix abhängig ist. Zudem ist die verwendete nichtlineare Klassifikation auf Basis eines *Decision Trees* unsensibel gegenüber derartigen Abhängigkeiten in den Daten.

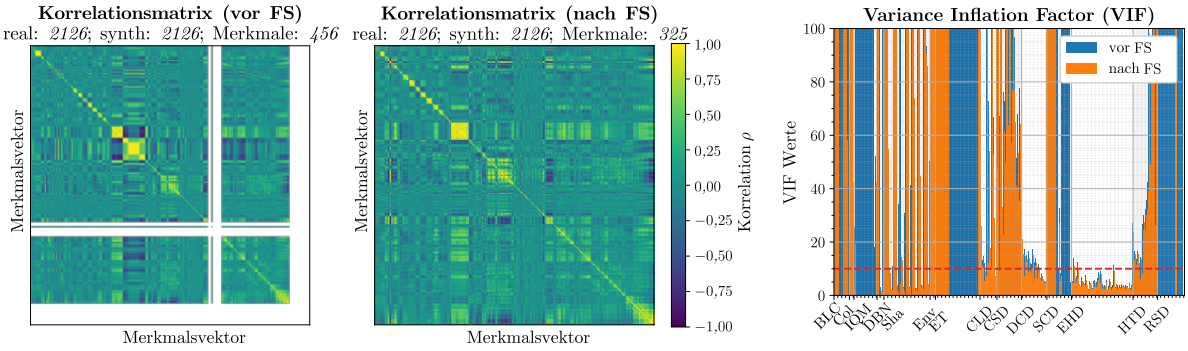


Abb. 73 Untersuchung der Multikollinearität in den Eingangsdaten anhand der Korrelationsmatrix und der VIF Werte vor und nach der *Feature Selection*. Weiße Zeilen und Spalten in der Korrelationsmatrix beschreiben Merkmale, die keine Varianz aufweisen und werden im Zuge der FS als quasikonstante Merkmale entfernt. Datensatz: KITTI/VKITTI.

Klassifikation zwischen KITTI und VKITTI Bildpaaren

Im ersten Schritt wird nun ein Modell zur Klassifikation der Bildpaare des KITTI/VKITTI Datensatzes berechnet, da diese als Benchmark dienen. Die Datenmatrix besteht nach der FS aus 325 Merkmalen und weist wie eben beschrieben eine deutliche Multikollinearität auf. Es zeigt sich, dass das Modell sowohl bei den Trainings- als auch bei den Testdaten eine ideale Klassifikation der Domäne ermöglicht und keinerlei Fehlklassifikationen auftreten. Im linken Teil von Abb. 74 ist der zugehörige Entscheidungsbaum visualisiert, der lediglich aus einer Ebene besteht und sich am Merkmal DBN8 orientiert.

Aus Tab. 10 ist ersichtlich, dass es sich dabei um einen Bildbeschreiber für Rauschen handelt (Merkmal an Position 8 in der Gruppe DBN).

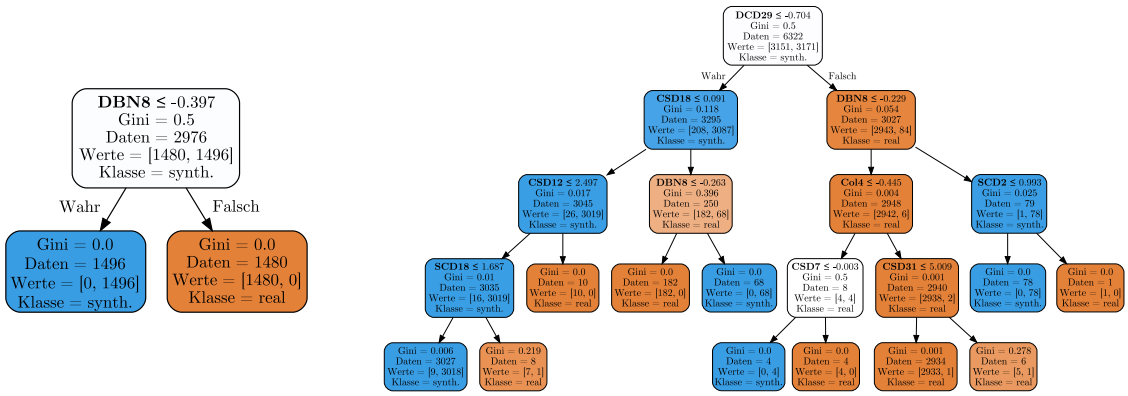


Abb. 74 Grafische Veranschaulichung der Baumstruktur des *Decision Tree* Klassifikators für die verschiedenen Datensätze mit Bildpaaren. An den Knotenpunkten können die Kriterien für die Aufteilung abgelesen werden. Links: KITTI/VKITTI; Rechts: R-UAV/S-UAV

In Abb. 75 ist links das Ergebnis des Auswahlprozesses der einflussreichsten Merkmale anhand der in Kapitel 4.5.2.4 und 4.5.2.5 beschriebenen FI- und FS-Methoden für diese Klassifikation dargestellt. Die y-Achse beschreibt die verwendeten FI- und FS-Methoden. Die Bildbeschreiber auf der x-Achse sind entsprechend ihrer Auswahlhäufigkeit angeordnet und repräsentieren dadurch die einflussreichsten Merkmale. Eine Zuweisung zu den jeweils beschriebenen Bildeigenschaften kann anhand von Tab. 10 vorgenommen werden. Um möglichst zuverlässige Ergebnisse zu erhalten, werden nur Merkmale aufgeführt, die von mindestens drei Methoden gleichzeitig als einflussreich bewertet wurden.

In diesem Fall deutet die hohe Multikollinearität und die dadurch mehrfach verfügbare Information, die ideale Klassifikationsleistung und die sehr geringe Baumtiefe darauf hin, dass sogar einzelne Merkmalsvektoren ausreichend sind, um zwischen den Domänen unterscheiden zu können. Da die FI-Methoden lediglich das aktuelle Klassifikationsmodell betrachten und dieses lediglich auf einem Merkmal beruht, kann auch nur dieses ausgewählt werden (s. Abb. 75 links, DBN8).

Die FS-Methoden betrachten im Gegensatz dazu unabhängig vom Modell die gesamten Eingangsdaten und zeigen, dass in mehreren Merkmalen relevante Information vorhanden ist. Es zeigt sich, dass eine Metrik für Rauschen (DBN8) überproportional großen Einfluss hat, was darauf hindeutet, dass die idealen gerenderten synthetischen Sensordaten zu wenig natürliches Rauschen enthalten. Darüber hinaus enthält die Liste mehrere Merkmale der auf der Berechnung eines Histogramms basierenden MPEG7 Farbdeskriptoren SCD und CSD. In [156] wird dazu erwähnt, dass der SCD Deskriptor stark diskriminativ ist für synthetisch generiertes Datenmaterial. CSD hingegen ist ein Maß für die lokale Farbstruktur. Daher liegt es nahe, dass das Fehlen feiner Strukturen auf den homogenen synthetischen Materialien (s. zum Beispiel Abb. 6 rechts im Bereich der Straße oder des Himmels) und der höhere Detailgrad der realen Aufnahmen für diesen Einfluss verantwortlich ist. Des Weiteren werden die Helligkeit im Bild (BLC17) und die Anzahl der vorkommenden Farben (Col5) aufgelistet. Im rechten Bildpaar aus Abb. 6 kann dieser Einfluss im Bereich des Himmels wiederum visuell nachvollzogen werden.

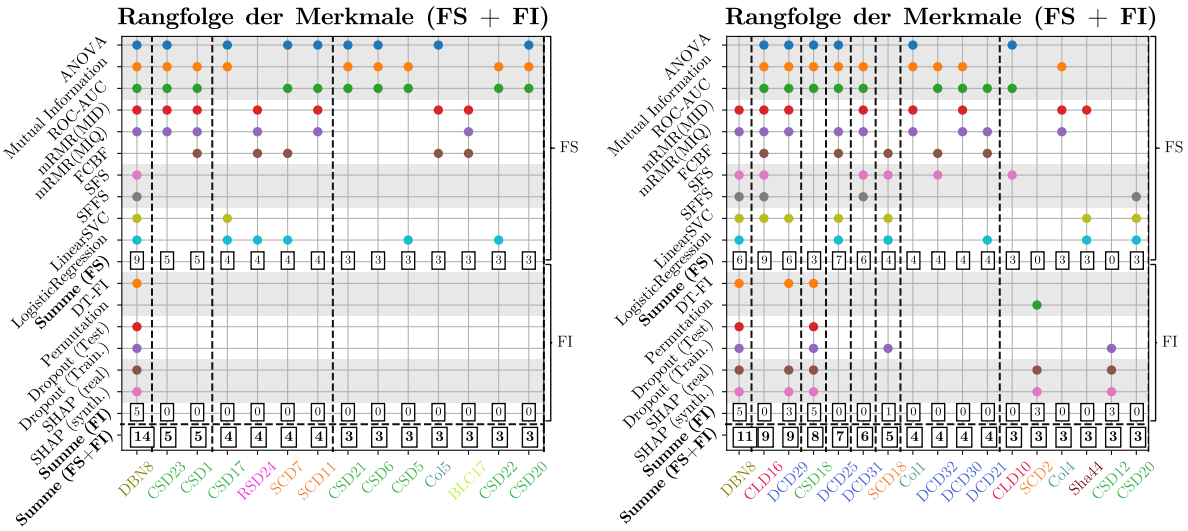


Abb. 75 Überblick und globale Rangfolge der als einflussreich identifizierten Bildbeschreiber. Auf der y-Achse sind die FS- und FI-Methoden aufgelistet, die für die Auswahl verwendet wurden. Die Punkte markieren die von der jeweiligen Methode ausgewählten Bildbeschreibermetriken, welche anschließend auf der x-Achse gemäß der Häufigkeit der Auswahl angeordnet sind. Die jeweils korrespondierende Bildeigenschaft kann aus Tab. 10 entnommen werden. Dabei enthält der erste Teil der Abkürzung die Gruppe (z.B. DBN), während die nachfolgende Nummer die Position des ausgewählten Merkmals in dieser Gruppe beschreibt (z.B. DBN8: Merkmal Nummer 8 in Gruppe DBN = Rauschen).
Links: KITTI/VKITTI; Rechts: R-UAV/S-UAV

Klassifikation zwischen R-UAV und S-UAV Bildpaaren

Im nächsten Schritt werden nun auf die gleiche Weise die nach dem in Kapitel 5.2 beschriebenen Schema generierten Bildpaare des R-UAV und S-UAV Datensatzes analysiert. Die Ausgangsbedingungen sind sehr ähnlich. Die Eingangsdaten enthalten wiederum eine deutlich erkennbare Multikollinearität und bestehen aus 331 Merkmalen. Abb. 74 zeigt im rechten Teil die zugrundeliegende Baumstruktur, die im Vergleich zu den vorherigen Benchmark Bildpaaren deutlich komplexer ist und aus vier Ebenen besteht. Dies spricht für eine gute und sorgfältige Nachbildung der synthetischen Duplikate. Das berechnete Modell erreicht dennoch wiederum eine nahezu ideale Klassifikationsgüte mit einem F1-Score von 0,996 und nur einzelnen Fehlklassifikationen. Abb. 76 zeigt einen Überblick über den Verlauf von Mittelwert und Standardabweichung der einzelnen Metriken in den jeweiligen Bildbeschreibergruppen. Es ist sichtbar, dass sich die Domänen in mehreren Merkmalen deutlich voneinander unterscheiden.

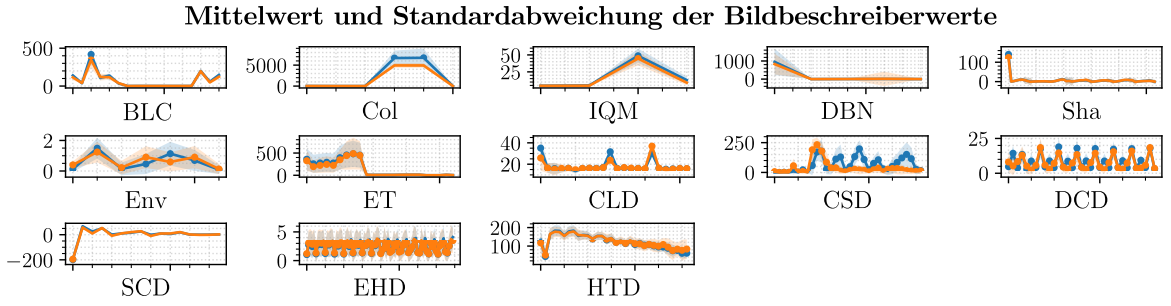


Abb. 76 Vergleich des Verlaufs des Mittelwerts der Merkmale in den einzelnen Bildbeschreibergruppen für die realen (blau) und synthetisch nachgebildeten Bildpaare (orange) und der zugehörigen Standardabweichung.
Datensatz: R-UAV/S-UAV.

Der rechte Teil von Abb. 75 listet erneut die Ergebnisse der Einflussanalyse. Trotz unterschiedlicher *Render-Engine* wird aus den bereits erwähnten Gründen wiederum Rauschen (DBN8) als insgesamt sehr ausschlaggebendes Merkmal identifiziert. Bei der detaillierteren Betrachtung der FS-Methoden LinearSVC und Logistic Regression spielt dieses Merkmal sowohl in Bezug auf das Vorzeichen als auch in Bezug auf den zugewiesenen Wert eine herausragende Rolle, ebenso wie bei der Berechnung der

SHAP-Werte, die zu den FI-Methoden zählt. Auch bei der zur Klassifikation verwendeten Baumstruktur wird die Metrik in beiden Zweigen des Baumes zur Entscheidungsfindung verwendet, was deren Einfluss unterstreicht.

Die Evaluierung zeigt außerdem, dass alle vier MPEG7 Farbdeskriptoren als relevant eingestuft wurden, wobei der die dominante Farbe im Bild beschreibende DCD Deskriptor zahlenmäßig am häufigsten auftritt. Dies bedeutet, dass Bildunterschiede sowohl in der lokalen und globalen Histogramm-basierten Farbverteilung vorliegen als auch in den lokalen und globalen dominanten Farben. In diesem Zusammenhang ist interessant, dass auch die beiden Metriken für Farbstich (Col1) und für Farbtemperatur (Col4) in der Liste aufgeführt sind. Dies deckt sich mit dem Einfluss der MPEG7 Farbdeskriptoren und auch anhand der Beispielbilder aus Abb. 54 sind die lokalen und globalen Farbunterschiede deutlich sichtbar, die durchaus als Farbstich oder Änderung der Farbtemperatur wahrgenommen werden können.

Trotz der möglichst exakten synthetischen Nachbildung ist schließlich noch ein Wert aus der Sha-Gruppe enthalten, die die allgemeine Szenerie beschreibt. Sha44 repräsentiert die Summe der Vegetation in der semantischen Segmentierung des Bildes. Da bei der Modellierung großer Wert auf die exakte Platzierung der Vegetation gelegt wurde, wird vermutet, dass dieser Einfluss durch die unterschiedliche Einstufung der Bodenvegetation durch die semantische Segmentierung zustande kommt, die unter Umständen Probleme bei der Klassifizierung der Grasbereiche in den synthetischen Daten hat, die durch das Luftbild nachgebildet und durch *Hypertexturen* aufgewertet werden. Auch in [239] wurde gezeigt, dass eine qualitativ hochwertige Bodentextur eine große Rolle spielt, wenn reale und synthetische Bilddaten auf Basis der Leistungsfähigkeit von Merkmalsdetektoren miteinander verglichen werden.

Zusammenfassung und Interpretation

Die Untersuchungen haben gezeigt, dass mit der vorgestellten Methode trotz einer großen Varianz und verschiedensten Szenarien in den Datensätzen eine nahezu ideale Zuordnung der Bildpaare zur jeweiligen Domäne möglich ist.

Dies dient als Grundlage für die nachfolgenden Untersuchungen und unterstreicht, dass die Bildbeschreibermetriken passend ausgewählt wurden und dass diese einige Bildeigenschaften beschreiben, die für den *Reality Gap* in Bezug auf eine unterschiedliche Erscheinungsform zwischen realen und synthetischen Daten verantwortlich sind. Der Vergleich mit dem KITTI/VKITTI Datensatz zeigt, dass die Methode unabhängig von der *Render-Engine* eingesetzt werden kann, obwohl in beiden Fällen leicht unterschiedliche Bildeigenschaften für den *Reality Gap* verantwortlich sind.

Insgesamt spielt jedoch bei beiden Datensätzen das fehlende bzw. nicht realistisch modellierte Rauschen die Hauptrolle bei der Unterscheidung. Weitere relevante Bildeigenschaften sind je nach Datensatz zum Beispiel das Fehlen feiner Strukturen auf den homogenen synthetischen Materialien, die Helligkeit und die Anzahl der vorkommenden Farben oder im anderen Fall die allgemeine Farbgebung mit Fokus auf Farbtemperatur und Farbstich.

Diese Ergebnisse können anhand von Bildpaaren aus dem jeweiligen Datensatz visuell nachvollzogen werden und bestätigen damit die Auswertemethode.

6.2.3 Klassifikation unabhängiger realer und synthetischer Trainings- und Testdaten

Auf Basis der bisher vorgestellten Ergebnisse soll nun untersucht werden, inwiefern auch eine Klassifikation voneinander unabhängiger realer und synthetischer Datensätze möglich ist. Dies spielt zum Beispiel beim Einsatz synthetischer Trainings- und Testdatensätze eine Rolle und soll helfen, Bildeigenschaften zu identifizieren, anhand derer sich diese synthetischen Datensätze von realen Datensätzen unterscheiden. Im Gegensatz zur Verwendung von Bildpaaren sind die dabei auftretenden Effekte nicht mehr einfach visuell anhand einzelner Beispielbilder nachvollziehbar, sondern beziehen sich auf die

Datensätze in ihrer Gesamtheit und begründen damit den eigentlichen Nutzen des Auswerteschemas. Tab. 30 wiederholt die Forschungsfrage und beschreibt das in diesem Kapitel betrachtete Experiment mit der zugehörigen Auswertung.

Tab. 30 Tabellarische Übersicht über die jeweils behandelte Forschungsfragestellung, das zugehörige Experiment und die einzelnen Bestandteile der Auswertung

2.	Welche Zuverlässigkeit erreicht der Ansatz bei der Unterscheidung zwischen voneinander unabhängigen realen und synthetischen Trainings- und Testdaten und welche Einflussfaktoren sind dabei entscheidend?
Experiment:	Klassifikationsanalyse mit dem Ziel der Unterscheidung der Domänen (real / synthetisch) auf Basis von Bildbeschreiberwerten
Auswertung:	1. Beurteilung und Analyse der Klassifikationsgüte (F1-Score) 2. Identifikation der einflussreichen Bildbeschreiber (FS- / FI-Methoden) - für den UAVDT und den synthetisch generierten Testdatensatz → Ableitung relevanter Bildeigenschaften und Gestaltungsmerkmale

Folgender Vergleich bezieht sich auf den realen UAVDT Testdatensatz und die in Kapitel 5.1.4 beschriebenen synthetisch generierten Testdaten. Auch auf diesen unabhängigen Bilddaten konnte eine nahezu ideale Klassifikation in die jeweilige Domäne mit einem F1-Score von 0,999 erzielt werden. Der zugehörige Entscheidungsbaum enthält 4 Ebenen. Abb. 77 zeigt, dass bis auf die Permutation Methoden alle FS- und FI-Verfahren in der Lage waren, einflussreiche Merkmale zu identifizieren. Insgesamt ist daher eine zuverlässige Evaluierung der Einflussfaktoren möglich.

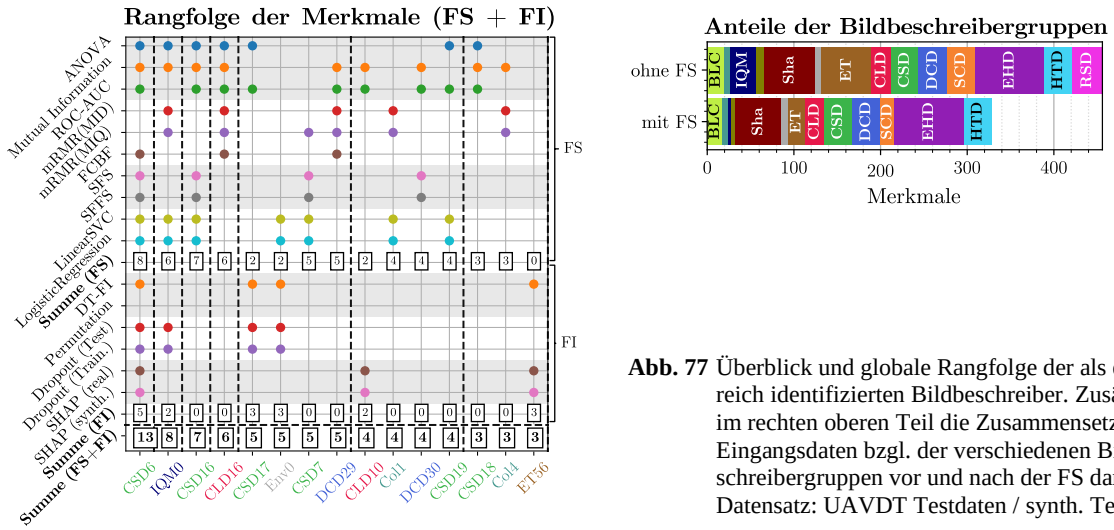


Abb. 77 Überblick und globale Rangfolge der als einflussreich identifizierten Bildbeschreiber. Zusätzlich ist im rechten oberen Teil die Zusammensetzung der Eingangsdaten bzgl. der verschiedenen Bildbeschreibergruppen vor und nach der FS dargestellt. Datensatz: UAVDT Testdaten / synth. Testdaten

Der CSD Deskriptor, der die lokale Farbverteilung im Bild beschreibt, ist sowohl hinsichtlich der Gewichtung als auch hinsichtlich der Häufigkeit der einflussreichste Bildbeschreiber in der Auswertung aus Abb. 77. Zum Teil ist dies, ähnlich wie bereits in Kapitel 6.2.2 erläutert, wiederum auf das Fehlen feiner Strukturen und Details auf den homogenen synthetischen Materialien zurückzuführen. In diesem speziellen Fall jedoch trägt auch der Bildaufbau entscheidend dazu bei, da die synthetisch generierten Daten deutlich häufiger großflächige homogene Szenerien mit weniger Struktur und Vielfalt enthalten als die sehr detailreichen und kleinstrukturierten realen UAVDT Bilder. Ein Vergleich von Abb. 5 und Abb. 43 macht diesen Unterschied deutlich.

In IQM0 wird eine ästhetische Bewertung der Fotoqualität vorgenommen. Dies kann nicht direkt einer speziellen Bildeigenschaft zugeordnet werden, entspricht aber unter Berücksichtigung der in [169] gegebenen Beispielbilder durchaus dem allgemeinen Eindruck.

Darüber hinaus sind mehrere Metriken aus der CLD und DCD Gruppe gelistet, welche die lokale und globale Verteilung der dominanten Farben im Bild beschreiben. Auch dafür gibt es zwei mögliche

Ursachen, die unter Umständen auch beide einen Beitrag zum Einfluss dieser Merkmale liefern. Zum einen enthalten die realen UAVDT Daten innerstädtische chinesische Hauptstraßen, während die synthetisch generierten Bilder hauptsächlich ländliche Gebiete mit Vegetation oder Industriebereichen zeigen, was zu einer unterschiedlichen Farbzusammensetzung führen kann. Zum anderen wurden auch bereits bei der Unterscheidung zwischen realen R-UAV und synthetischen S-UAV Bildpaaren die MPEG7 Farbdeskriptoren als Einflussfaktoren identifiziert. Da dabei sowohl die gleiche virtuelle Welt als auch die gleiche zugrunde liegende *Render-Engine* zur Anwendung kam, liegt es nahe, dass auch im hier betrachteten Fall wiederum die synthetische Farbgebung durch die Gruppen CLD, DCD und auch CSD beschrieben wird. Diese Annahme wird vor allem dadurch bestärkt, dass ebenfalls wieder die Metriken Col1 und Col4 zur Beschreibung von Farbstich und Farbtemperatur als Einflussfaktoren ausgewählt wurden.

Schließlich sind noch ET56 (Glattheit) und Env0 (Schattenanteil) in der Liste der relevanten Merkmale aufgeführt. Da gemäß der Korrelationsmatrix die Metriken der Gruppe ET stark mit denjenigen der Gruppe CSD korreliert sind, beschreiben beide sehr wahrscheinlich dieselbe zugrunde liegende Ursache. Die Ursache für einen Einfluss des Schattenanteils im Bild kann nicht genau abgeleitet werden. Es wird vermutet, dass eine Falschklassifizierung dunkler Bereich oder Teile von Gebäuden als Schatten infolge spezieller Beleuchtungseffekte eine Rolle spielen könnte. Im Vergleich zu den anderen identifizierten Bildunterschieden ist dieser Aspekt bei der synthetischen Datengestaltung jedoch wahrscheinlich eher vernachlässigbar.

Abb. 77 visualisiert zusätzlich im rechten oberen Teil die Zusammensetzung der Datenmatrix vor und nach der FS. Diese ist für alle betrachteten Klassifikationsaufgaben ähnlich und zeigt, dass die Anzahl an Merkmalen und damit die Größe einer Gruppe von Bildbeschreibern in keinem Zusammenhang mit deren Häufigkeit bei der Evaluierung steht und somit keinen Einfluss auf die Auswahl der FS- und FI-Methoden ausübt.

Zusammenfassung und Interpretation

Letztendlich konnte nachgewiesen werden, dass der verwendete Ansatz auf Basis einer Klassifikationskette auch bei voneinander unabhängigen Datensätzen zur Unterscheidung der realen und synthetischen Domäne verwendet werden kann und eine insgesamt sehr hohe Zuverlässigkeit erreicht wird.

Dies ermöglicht wiederum die Ableitung von Einflussfaktoren, die zu Unterschieden zwischen realen und synthetischen Trainings- und Testdatensätzen führen können. Die Auswertung zeigte, dass diese Einflussfaktoren in zwei Gruppen aufgeteilt werden können:

Die erste Gruppe beschreibt die synthetische Farbrepräsentation in Bezug auf die dominante Farbverteilung (CLD, DCD) und Metriken für Farbstich und Farbtemperatur.

Diese Bildeigenschaften wurden bereits bei der Analyse der Bildpaare aus dem R-UAV/S-UAV Datensätzen aufgelistet, da für deren Generierung dieselbe virtuelle Modellierung und *Render-Engine* verwendet wurde. Diese Gruppe beinhaltet somit denjenigen Anteil des *Reality Gaps*, der durch die visuelle Erscheinung der Bilddaten herrührt.

Die zweite Gruppe beschreibt durch fehlende feine Strukturen auf den synthetischen Materialien und großflächige homogene Bereiche und Szenarien eher den synthetischen Bildinhalt und daher den inhaltsbasierten Anteil des Reality Gaps.

Dies zeigt, dass parallel zu den Beobachtungen in 6.1.4 auch auf Basis der Einflussanalyse eine Aufteilung des *Reality Gaps* in *Appearance Gap* und *Content Gap* nachgewiesen werden konnte. Die ausgewählten Bildbeschreibermetriken decken wiederum alle Bildunterschiede sehr gut ab und sind universell anwendbar.

6.2.4 Klassifikation korrekter und inkorrekter Detektionsergebnisse

Im letzten Schritt soll nun untersucht werden, inwiefern es mit dem vorgestellten Klassifikationsansatz möglich ist, zwischen korrekten (TP) und inkorrekten (FP, FN) Detektionen zu unterscheiden. Ziel dabei ist die Identifikation von Bildeigenschaften, die einen Einfluss auf die Leistungsfähigkeit von *deep-learning* basierten Fahrzeugdetektoren besitzen. Tab. 31 wiederholt die Forschungsfrage und beschreibt das in diesem Kapitel betrachtete Experiment mit der zugehörigen Auswertung.

Tab. 31 Tabellarische Übersicht über die jeweils behandelte Forschungsfragestellung, das zugehörige Experiment und die einzelnen Bestandteile der Auswertung

3.	Wie hoch ist die Güte der Klassifikation in korrekte und inkorrekte Detektionsergebnisse und welche Einflussfaktoren sind dabei entscheidend?
Experiment:	Klassifikationsanalyse mit dem Ziel der Unterscheidung der Detektionsergebnisse (TP / FP / FN) auf Basis von Bildbeschreiberwerten der *Bounding Boxen*
Auswertung:	1. Beurteilung und Analyse der Klassifikationsgüte (F1-Score) 2. Identifikation der einflussreichen Bildbeschreiber (FS- / FI-Methoden) - für das synthetisch trainierte Modell auf den realen UAVDT Testdaten - für das synthetisch trainierte Modell auf den realen R-UAV Daten → Ableitung relevanter Bildeigenschaften und Gestaltungsmerkmale

Als Ausgangsbasis für die Evaluierung dient das rein synthetisch trainierte Detektormodell, das auf zwei verschiedenen Testdatensätzen evaluiert wird. Bei der Anwendung auf die realen UAVDT Testdaten ist die Detektionsleistung relativ gering. Daher stellt sich die Frage, ob ausschließlich auf Basis der durch die Bildbeschreiber ausgewerteten Bildinformation in den *Bounding Boxen* eine Unterscheidung zwischen TP, FP und FN Detektionen möglich ist und ob somit Bildeigenschaften identifiziert werden können, die zur geringen Detektionsleistung beitragen und Fehldetektionen verursachen.

Bei der Anwendung des Modells auf die zu den synthetischen Trainingsdaten inhaltlich sehr ähnlichen R-UAV Testdaten wird eine sehr gute Detektionsleistung erzielt. Der Vergleich soll zeigen, welche Unterschiede bezüglich der einflussreichen Bildeigenschaften in diesem Fall vorliegen und wie dadurch eine zukünftige synthetische Trainingsdatengenerierung verbessert werden könnte. In beiden Fällen bilden die für alle *Bounding Boxen* berechneten Bildbeschreiberwerte die Datenmatrix auf Basis derer die Art der Detektionen (TP, FP, FN) abgeleitet werden soll.

6.2.4.1 Geometrische Analyse der *Bounding Boxen*

Zu Beginn wurden einige grundlegende geometrische Eigenschaften der *Bounding Boxen* analysiert, um vorab mögliche Einflussfaktoren zu identifizieren, die nicht auf bestimmte Bildeigenschaften zurückzuführen sind. Dabei wurden jeweils die *Ground Truth Bounding Boxen* der Trainings- und Testdatensätze und auch die Gruppen TP, FP und FN miteinander verglichen, um zu analysieren, ob inkorrekte Detektionen in einem Zusammenhang zu den geometrischen Bedingungen stehen. Ausgangsbasis war wie immer in diesem Abschnitt das synthetisch trainierte Detektormodell.

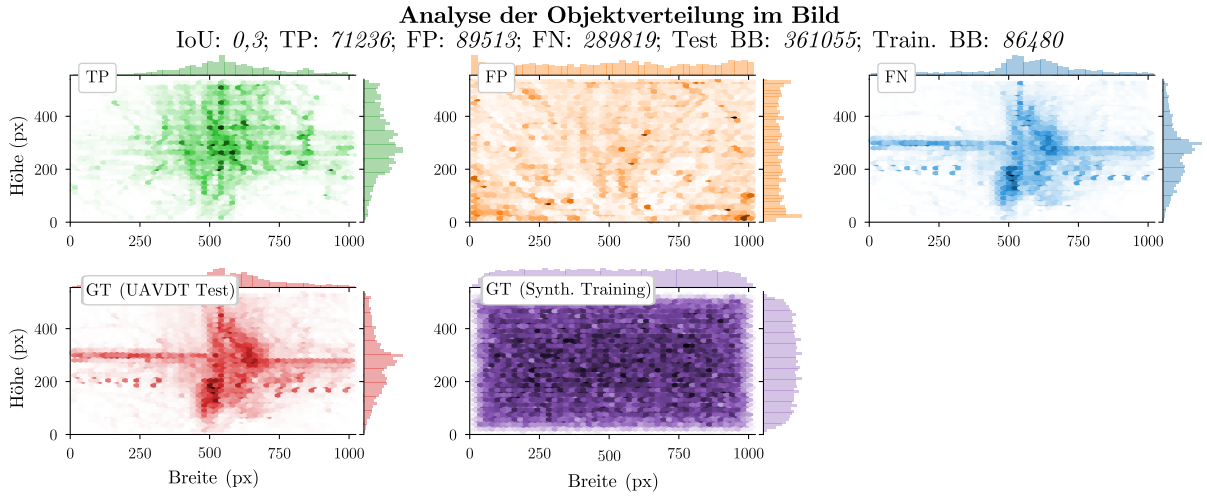


Abb. 78 Grafische Visualisierung der örtlichen Verteilung der verschiedenen Detektionen und *Ground Truth Bounding Boxen* im Bildausschnitt.
Datensatz: UAVDT Testdaten; synth. trainiertes Detektormodell

In Abb. 78 ist die örtliche Verteilung der Detektionen für die jeweiligen Gruppen aufgetragen. Während die Fahrzeuge im synthetischen Trainingssatz wie durch die Parameterwahl erwartet gleichförmig über das Bild verteilt sind, kann man bei den UAVDT Testdaten eine Häufung im mittleren x- und y-Bereich beobachten, da dies dem typischen Straßenverlauf entspricht. Die FP-Detektionen sind ebenfalls gleichmäßig verteilt, während die TP- und FN-Detektionen wiederum die dem Straßenverlauf geschuldete Verteilung aufweisen. Die Häufung im mittleren y-Bereich ist vor allem bei den FN-Detektionen ausgeprägt, während sie bei den TP-Detektionen nicht auftritt. Bei der Betrachtung von Beispielbildern (s. Abb. 5) lässt sich daraus vermuten, dass vor allem kleine, dicht gedrängte Fahrzeuge auf quer zum Bild verlaufenden Straßen, die aus größeren Flughöhen aufgenommen wurden, tendenziell schlechter erkannt werden.

In Abb. 79 ist daher die Größenverteilung der verschiedenen Gruppen anhand der Parameter Höhe, Breite, Fläche und Seitenverhältnis gegeneinander aufgetragen. Die ebenfalls eingezeichneten Ankerboxen dienen als eine Art Vorlage für die Anpassung der vom YOLOv3 Netzwerk detektierten *Bounding Boxen* und sind auf die Größenverteilung im Trainingsdatensatz angepasst. Die Verteilung zeigt, dass die Objektgrößen in den synthetischen Trainingsdaten durchschnittlich größer sind als in den davon unabhängigen realen UAVDT Testdaten. Im Allgemeinen bestehen zwischen den Verteilungen der Gruppen TP, FP und FN nur sehr geringe Unterschiede, da das Modell in der Lage ist, bis zu einem bestimmten Grad zwischen den vorkommenden Größen der *Bounding Boxen* zu generalisieren. Bei Betrachtung des Parameters „Fläche“ ist jedoch eine leichte Anhäufung falscher Detektionen bei geringen Objektgrößen erkennbar. Dieser Effekt hat wahrscheinlich nur einen sehr geringen Einfluss auf die Detektionsleistung, sollte jedoch bei der allgemeinen Auswertung der Einflussfaktoren mitberücksichtigt werden.

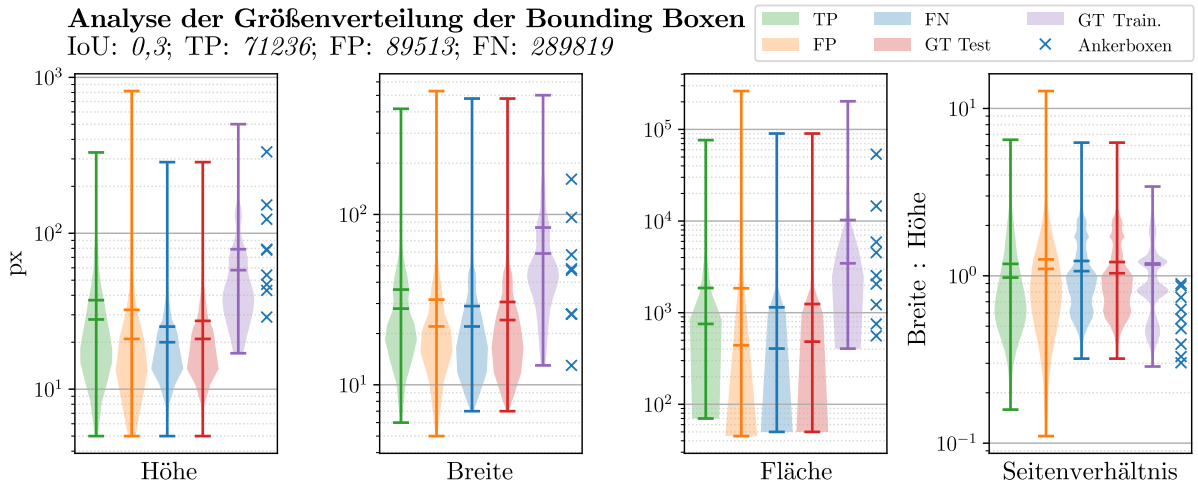


Abb. 79 Violinplot mit Median und Mittelwert der Verteilung einiger geometrischer Eigenschaften der *Bounding Boxen* zum Vergleich zwischen Trainings- und Testdatensatz und TP-, FP- und FN-Detektionen.
Datensatz: UAVDT Testdaten; synth. trainiertes Detektormodell

In Abb. 80 ist derselbe Sachverhalt für eine Anwendung des synthetisch trainierten Modells auf den realen R-UAV Daten dargestellt. Es zeigt sich, dass diese als Testdaten eine zu den Trainingsdaten deutlich ähnlichere Größenverteilung besitzen. Zusätzlich fällt auf, dass sich in diesem Fall die Gruppe der FP-Detektionen bei allen Parametern signifikant weiter in den Bereich kleiner Objektgrößen ausdehnt. Dies bedeutet, dass die fälschlicherweise als Fahrzeug detektierten Störobjekte häufig kleinere Objektgrößen aufweisen als die eigentlich vorkommenden Fahrzeuge. Auch dieser Umstand sollte bei der weiteren Evaluierung der Modelle beachtet werden.

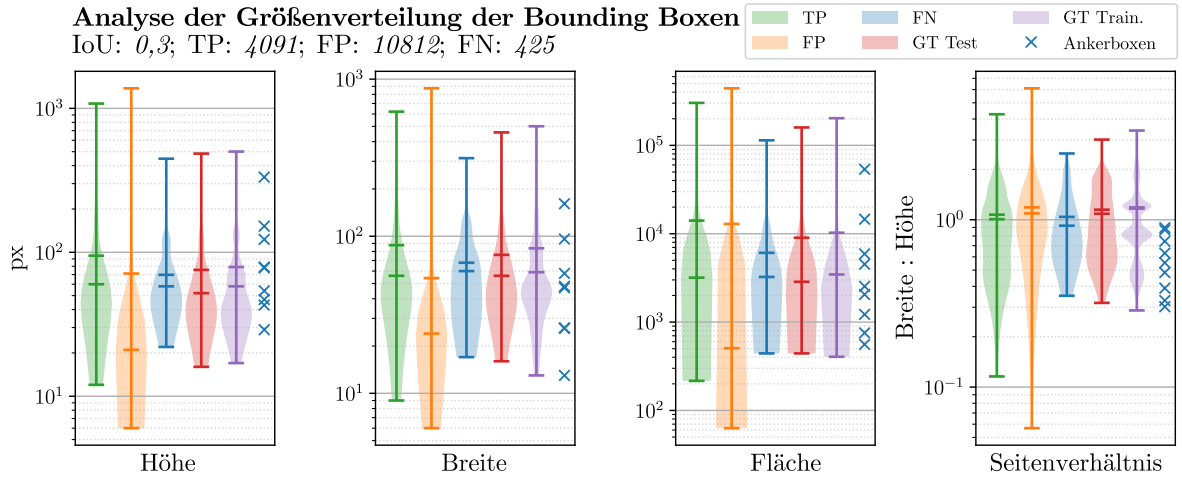


Abb. 80 Violinplot mit Median und Mittelwert der Verteilung einiger geometrischer Eigenschaften der *Bounding Boxen* zum Vergleich zwischen Trainings- und Testdatensatz und TP-, FP- und FN-Detektionen.
Datensatz: R-UAV Testdaten; synth. trainiertes Detektormodell

6.2.4.2 Synthetisch trainiertes Modell auf realen UAVDT Benchmark Daten

Im Folgenden soll nun untersucht werden, inwiefern mit den auf Basis der *Bounding Boxen* berechneten Bildbeschreiberwerten eine Klassifikation der Detektionen in die Gruppen TP, FP und FN möglich ist. Die dazu betrachteten Detektionen stammen aus der Anwendung des rein synthetisch trainierten Modells auf die realen UAVDT Testdaten. Das Modell erreichte dabei nur eine sehr geringe AP von 12,45 %, weshalb nun versucht wird, diejenigen Bildeigenschaften zu identifizieren, die für eine Unterscheidung zwischen korrekten und inkorrekten Detektionen herangezogen werden können und somit bei der zukünftigen Trainingsdatengenerierung verstärkt berücksichtigt werden müssen.

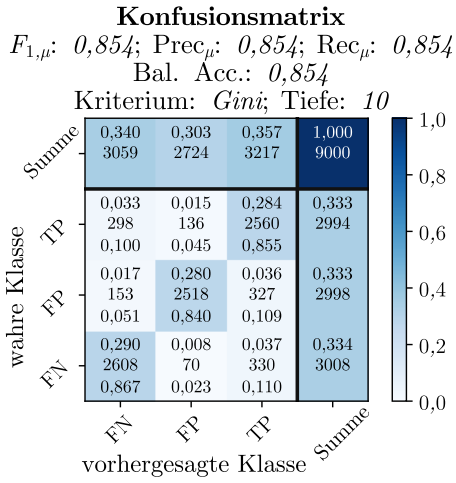


Abb. 81 Konfusionsmatrix zur detaillierteren Evaluierung des Klassifikationsmodells in Bezug auf die Leistungsfähigkeit pro Klasse.
Obere Reihe: Relative Werte
Mittlere Reihe: Absolute Werte
Untere Reihe: Relative Werte auf die Klasse normiert
Prec.: *Precision*; Rec.: *Recall*; Bal. Acc.: *Balanced Accuracy*

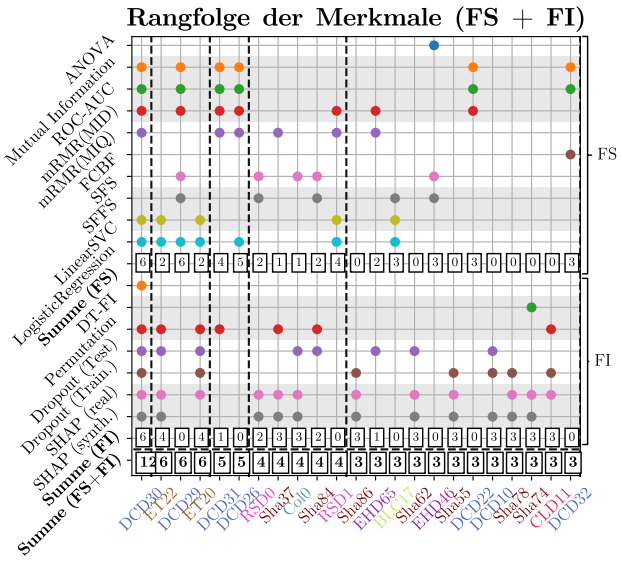


Abb. 82 Überblick und globale Rangfolge der als einflussreich identifizierten Bildbeschreiber.
Datensatz: UAVDT Testdaten; synth. trainiertes Detektormodell

Die Güte der Klassifikation ist in Abb. 81 anhand der Konfusionsmatrix dargestellt. Im Vergleich zu den vorherigen Untersuchungen ist diese Art der Klassifikation deutlich komplexer, weshalb der dafür verwendete Entscheidungsbaum 10 Ebenen besitzt. Dennoch wird beim Training ein F1-Score von ungefähr 0,92 erzielt, der auf den unbekannten Testdaten auf 0,854 abfällt und damit immer noch eine sehr zuverlässige Zuordnung der Detektionen ermöglicht. Durch die Konfusionsmatrix lässt sich zusätzlich nachweisen, dass keine signifikanten Leistungsunterschiede bei der Klassifikation der einzelnen Klassen (TP, FP, FN) auftreten. Insgesamt bestätigt dies die Verwendung der beschriebenen Klassifikationskette und ermöglicht die in Abb. 82 dargestellte Auswertung der Einflussfaktoren.

Anhand der Rangfolge aber auch anhand der Häufigkeit kommt dem DCD Deskriptor eine große Bedeutung zu. Ob dieser Einfluss infolge der dominanten Farben des Hintergrunds in der *Bounding Box* entsteht oder infolge der dominanten Farben des Fahrzeugobjekts kann nicht eindeutig unterschieden werden. Da jedoch auch bereits bei der Unterscheidung der realen und synthetischen Bildpaare (R-UAV/S-UAV, s. Kapitel 6.2.2) und bei der Unterscheidung der zugehörigen realen und synthetischen Testdaten (s. Kapitel 6.2.3) der DCD Deskriptor als einflussreich aufgeführt wurde und in all diesen Fällen dieselbe virtuelle Modellierung und Simulationsumgebung verwendete wurde, kommt sehr wahrscheinlich ein deutlicher Anteil durch den Hintergrund zustande.

Als weitere Faktoren sind zwei Werte des MPEG7 Texturdeskriptors EHD und die Merkmale ET20 und ET22 (GLCM, Homogenität) aufgelistet. Diese beschreiben die Struktur der Texturen und die lokale räumliche Verteilung der Kanten im Ausschnitt der *Bounding Box*. Da Fahrzeuge eine sehr charakteristische Kontur aufweisen, wird vermutet, dass diese Merkmale verwendet werden, um zwischen *Bounding Boxen* mit Fahrzeug (TP, FN) und solchen ohne Fahrzeug (FP) zu unterscheiden. Da die *Bounding Boxen* teilweise einen sehr kleinen Bildausschnitt betrachten, sind vor allem an den Übergängen zwischen Kanten und homogenen Bereichen deutliche Blockartefakte der JPEG Komprimierung zu sehen, welche ebenfalls einen Einfluss auf diese Merkmale haben können. Eine dritte Erklärung wären schließlich charakteristische Straßenmarkierungen im Bereich der *Bounding Boxen*, welche bei den realen UAVDT Testdaten sehr häufig vorkommen, bei den synthetischen Trainingsdaten jedoch eher

unterrepräsentiert sind. Eine diesbezügliche Variation in den Trainingsdaten in Kapitel 6.3.2 soll zeigen, ob dies einen Einfluss auf die Detektionsleistung hat.

Da bei der Berechnung des MPEG7 Formdeskriptors RSD keine Segmentierungsmaske verwendet wurde, beschreiben die Merkmale RSD0 und RSD1 die Form und Größe der *Bounding Boxen* und stehen damit vermutlich in Zusammenhang mit den in Abb. 79 beschriebenen Größenverteilungen.

Die zweite große Gruppe neben den DCD Merkmalen mit einem vor allem wegen ihrer Häufigkeit sehr großen Einfluss bilden die Sha-Vektoren. Diese Formdeskriptoren sind von der semantischen Segmentierung der *Bounding Box* und des gesamten dazugehörenden Bildes abgeleitet und beschreiben dadurch in gewisser Weise die Szenerie des Bildes und das Umfeld der Detektionen. Während lediglich zwei Deskriptoren von der semantischen Segmentierung der *Bounding Box* stammen (Sha37, Sha55), sind die restlichen fünf Deskriptoren aus der semantischen Segmentierung des zugehörigen Gesamtbildes abgeleitet. Dies lässt darauf schließen, dass die gesamte Szenerie des Testbildes einen Einfluss auf die Detektionsleistung hat. Die Deskriptoren Sha86, Sha62, Sha78, Sha74 repräsentieren dabei bestimmte Hu-Momente zur Beschreibung der Segmentierungsmasken der Klassen Vegetation, Straße, Gebäude und Objekten, während der Deskriptor Sha84 die Summe der Vegetation im Bild beschreibt. Dies ist interessant, da diese Eigenschaft bereits bei der Unterscheidung der realen und synthetischen Bildpaare aufgeführt wurde (entspricht Sha44 in Abb. 75, da in diesem Fall keine *Bounding Boxen* vorkamen) und hier wiederum Parallelitäten aufgrund der verwendeten *Render-Engine* erkennbar sind.

Dies ist wahrscheinlich auch die Ursache für die Relevanz der Merkmale Col0 (Farbigkeit) und CLD11, die in Zusammenhang mit den bereits bei der Unterscheidung der zugehörigen realen und synthetischen Testdaten (s. Kapitel 6.2.3) erwähnten Farbunterschieden stehen.

Schließlich ist noch der BLC17 Deskriptor aufgeführt, der die Helligkeit im Bild beschreibt. Dies kann durch die Datenzusammensetzung erklärt werden, da der reale UAVDT Testdatensatz Aufnahmen bei Nacht enthält, der synthetische Trainingsdatensatz jedoch nicht.

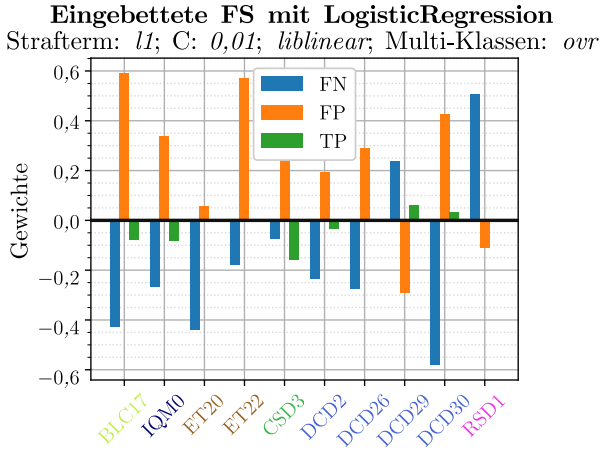


Abb. 83 Visualisierung der Gewichte derjenigen Merkmale, die durch die zu den FS-Methoden gehörende *LogisticRegression* als relevant eingestuft wurden. Dabei wird zwischen den Klassen TP, FP und FN unterschieden.
ovr: *One-vs.-Rest* Mulitklassen Strategie
Datensatz: UAVDT Testdaten; synth. trainiertes Detektormodell

Zusammenfassung und Interpretation

Es ist festzuhalten, dass das Klassifikationsmodell in der Lage ist, im Vorfeld ausschließlich auf Basis der in den *Bounding Boxen* enthaltenen Bildinformation mit hoher Zuverlässigkeit (F1-Score = 0,854) das Detektionsergebnis vorherzusagen.

Dies bestätigt, dass die Bildbeschreiber auch für diese Aufgabe passend gewählt wurden und die für die Detektion relevanten Bildeigenschaften charakterisieren. Die verschiedenen Vorzeichen der Gewichte der *LogisticRegression* (FS-Methode) für *Bounding Boxen* mit Fahrzeugen (TP, FN) und solchen ohne Fahrzeuge (FP) in Abb. 83 zeigen, dass es sogar möglich ist, im Vorfeld der Klassifikation

ausschließlich auf Basis der Daten der Bildbeschreiber zwischen *Bounding Boxen* mit und ohne Fahrzeugen zu unterscheiden. Einige Bildeigenschaften, wie z.B. die allgemeine Farbgebung und dabei vor allem die Zusammensetzung der dominanten Farben (DCD) waren bereits bei der Unterscheidung der realen und synthetischen Bildpaare auffällig und sind daher wiederum der Erscheinungsform und dem Rendering der simulierten Daten zuzuordnen. Die andere Gruppe der Bildbeschreiber, wie z.B. die aus der semantischen Segmentierung stammenden Sha-Werte, sind im Gegensatz dazu eher dem Bildinhalt zuzuordnen und belegen, dass auch die Szenerie des Bildes und das Umfeld der Detektion einen Einfluss auf die Detektionsleistung nimmt.

Es zeigt sich, dass in diesem Fall wiederum Einflussfaktoren aus dem Bereich *Appearance Gap* und aus dem Bereich *Content Gap* enthalten sind. Die Auswertung hat ebenfalls ergeben, dass zumindest bei zufällig ausgewählten und nicht speziell modifizierten Testdaten die Detektionen relativ unempfindlich sind gegenüber ebenfalls durch die Bildbeschreiber erfasste Störgrößen wie z.B. Rauschen, Kontrast oder Unschärfe.

6.2.4.3 Synthetisch trainiertes Modell auf real erflogenen R-UAV Testdaten

Zum Vergleich wird nun im letzten Schritt eine Klassifikation derjenigen Detektionen betrachtet, die das rein synthetisch trainierte Modell auf den real erflogenen R-UAV Testdaten liefert. Da diese aus derselben geografischen Umgebung stammen, sind sie den synthetischen Trainingsdaten inhaltlich sehr ähnlich und das Modell erzielt dabei eine sehr gute AP von 76 %. Neben den mehrfach erwähnten Bildbeschreibermetriken können in diesem Fall auch die in Tab. 17 aufgelisteten Parameter des Realfluges als Merkmale für die Klassifikation verwendet werden.

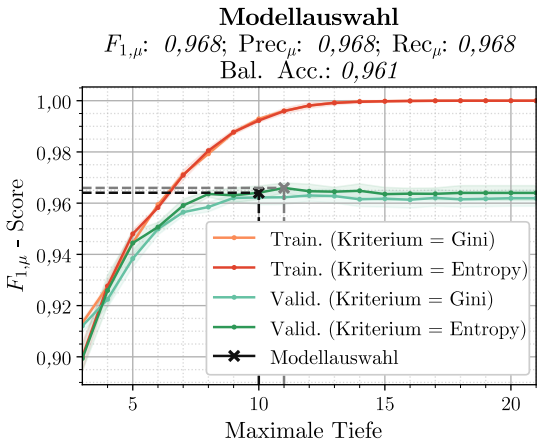


Abb. 84 Prozess zur Modellauswahl anhand des F1-Scores und der Standardabweichung innerhalb der Testdaten. In Rot ist zum Vergleich die Leistung auf den Trainingsdaten aufgetragen.
Datensatz: R-UAV Testdaten; synth. trainiertes Detektormodell

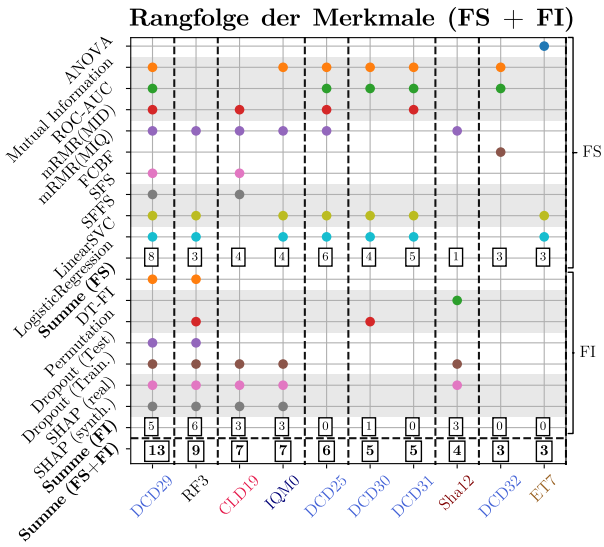


Abb. 85 Überblick und globale Rangfolge der als einflussreich identifizierten Bildbeschreiber.
Datensatz: R-UAV Testdaten; synth. trainiertes Detektormodell

Auch in diesem Fall erzielt die vorgestellte Klassifikationskette eine sehr hohe Güte und erreicht einen F1-Score von 0,968. In Abb. 84 ist der zugehörige Prozess zur Modellauswahl abgebildet. Mit Hilfe einer Kreuzvalidierung wird für jede Baumtiefe die Leistung und die zugehörige Standardabweichung berechnet und anschließend das Modell mit der niedrigsten Baumtiefe ausgewählt, dessen Leistung sich dennoch innerhalb der Standardabweichung der maximal möglichen Leistung befindet. Dieses Verfahren soll dazu beitragen, die Modellkomplexität und die Überanpassung zu reduzieren und ist fester Bestandteil der in allen Fällen verwendeten Klassifikationskette. Insgesamt resultiert daraus wie bereits bei

der Klassifikation der Detektionen auf den UAVDT Testdaten im vorigen Kapitel ein Entscheidungsbaum mit 10 Ebenen. Trotz der guten Detektionsleistung kann auch in diesem Fall eine sehr hohe Klassifikationsgüte erzielt werden, was wiederum eine Analyse der Einflussfaktoren ermöglicht, welche in Abb. 85 dargestellt ist. Obwohl jede Methode in der Lage ist, mehrere relevante Bildbeschreiber auszuwählen, fällt auf, dass es im Vergleich zu den bisherigen Analysen weniger Überschneidungen gibt und daher insgesamt weniger Merkmale als einflussreich eingestuft werden.

Der DCD Deskriptor ist in Bezug auf Rangfolge und Häufigkeit der wichtigste Bildbeschreiber. Dies ist in Übereinstimmung mit den Ergebnissen des vorigen Kapitels, wobei zu erwähnen ist, dass größtenteils sogar die gleichen Merkmale des DCD-Vektors aufgeführt sind. Dieser Einfluss kommt wie bereits erläutert durch die Verwendung der virtuellen Modellierung und Simulationsumgebung zustande und beschreibt in gewisser Weise den *Appearance Gap*.

RF3 (Aufnahmeposition bei Gebäude) stammt aus dem in diesem Fall zusätzlich mit aufgenommenem Vektor, der die zum Bild gehörigen Parameter der realen Aufnahme enthält, und deutet darauf hin, dass die Positionierung des Fahrzeugs und damit das Umfeld einen Einfluss auf die Detektionsleistung hat. Weitere geometrische Aufnahmeparameter, Objekt- oder Umgebungsparameter sind ebenfalls im RF-Vektor enthalten, wurden in dieser Analyse aber tendenziell als weniger einflussreich eingestuft.

Der CLD Deskriptor beschreibt die lokale Verteilung dominanter Farben und IQM0 ist ein Maß für die ästhetische Bewertung von Fotografien. Beide Merkmale gehen daher ebenfalls in die Richtung des *Appearance Gaps* und kamen bereits bei der Klassifikation der Domänen vor.

ET7 (GLCM, Kontrast) beschreibt auf Basis der Grauwertematrix die Textureigenschaft Kontrast und Sha12 repräsentiert den Bildanteil, den die Segmentierungsmaske aus der Vordergrundsegmentierung einnimmt. Die genauen Ursachen können nicht eindeutig erkannt werden, es wird jedoch vermutet, dass in Verbindung zum Parameter RF3 der Fahrzeuguntergrund bzw. die Umgebung eine Rolle spielen könnte.

Hervorzuheben ist, dass im Gegensatz zur Klassifikation der Detektionen auf den UAVDT Testdaten im hier betrachteten Fall keinerlei Sha-Merkmale aus der semantischen Segmentierung aufgelistet werden. Dies betrifft sowohl die semantische Segmentierung der *Bounding Box* als auch diejenige des gesamten zugehörigen Sensorbildes. Da die R-UAV Testdaten aus derselben geografischen Umgebung stammen wie die verwendeten synthetischen Trainingsdaten, enthalten sie sehr ähnliche Szenerien und Umgebungen. Daher ist es durchaus plausibel, dass die Analyse der Einflussfaktoren in diesem Fall keine Metriken enthält, die vorwiegend den Bildinhalt beschreiben, da der *Content Gap* hier vernachlässigbar ist. Dies ist auch in Übereinstimmung mit der höheren Detektionsleistung trotz des rein synthetischen Trainings und bestätigt die in Kapitel 6.1.4 gemachten Erkenntnisse zur Aufteilung des *Reality Gaps*.

Zusammenfassung und Interpretation

Abb. 86 zeigt die Gewichtungen der von der zu den FS-Methoden zählenden *LogisticRegression* ausgewählten Merkmale. Ebenso wie in Abb. 83 ist auch hier anhand des Vorzeichens eine Unterscheidung zwischen *Bounding Boxen* mit Fahrzeugen (TP, FN) und solchen ohne Fahrzeugen (FP) ausschließlich auf Basis der Daten möglich. Die zu den FI-Methoden zählende SHAP Analyse bewertet den durchschnittlichen Einfluss der Merkmale auf das Modellergebnis der Klassifikation. Anhand von Abb. 87 wird deutlich, dass dabei der Einfluss für korrekte (TP) und inkorrekte (FP, FN) Detektionen mit unterschiedlichem Vorzeichen gewertet wird.

Insgesamt unterstreichen die Analysen, dass eine Klassifikation der Detektionen ausschließlich auf Basis der Bildinformation in den *Bounding Boxen* mit einer allgemein sehr hohen Güte möglich ist. Dies

ist zudem unabhängig von der Detektionsleistung auf den Testdaten und ist sowohl bei Fällen mit geringer als auch bei solchen mit mittlerer bis hoher Detektionsleistung möglich.

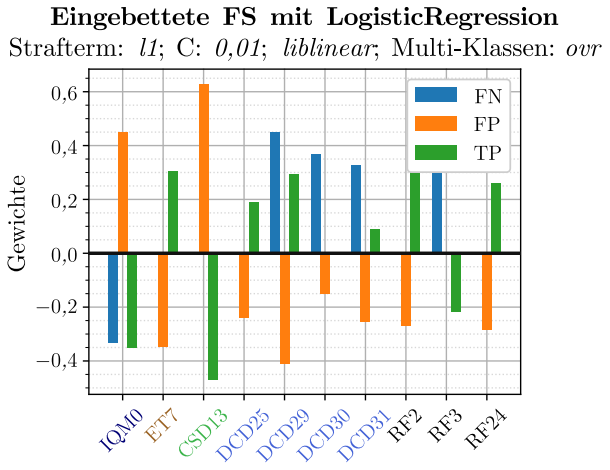


Abb. 86 Visualisierung der Gewichte derjenigen Merkmale, die durch die zu den FS-Methoden gehörende LogisticRegression als relevant eingestuft wurden. Dabei wird zwischen den Klassen TP, FP und FN unterschieden.
ovr: One-vs.-Rest Mulitklassen Strategie
Datensatz: R-UAV Testdaten; synth. trainiertes Detektormodell

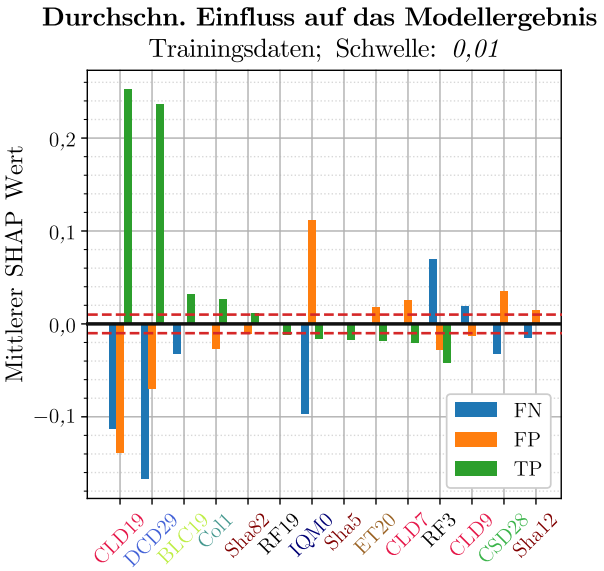


Abb. 87 Überblick über die SHAP Werte (FI-Methode) für jede Klasse als Maß für den Einfluss auf das Klassifikationsergebnis des Modells.
Datensatz: R-UAV Testdaten; synth. trainiertes Detektormodell

Im Allgemeinen hat sich herausgestellt, dass die globale dominante Farbverteilung in der *Bounding Box* (DCD Deskriptor) sehr großen Einfluss auf das Detektionsergebnis nimmt.

Im Gegensatz zu den UAVDT Testdaten sind jedoch in diesem Fall durch die szenarische Ähnlichkeit zu den Trainingsdaten keine Bildbeschreibermetriken aufgeführt, die den Content Gap beschreiben, sondern lediglich solche, die eher dem Appearance Gap zugeordnet werden können.

Die Auswertung der in diesem Fall speziell als Merkmale hinzugenommenen Realflugparameter zeigte, dass zwar das Fahrzeugumfeld die Detektionsleistung beeinflusst, aber weitere geometrische Aufnahmeparameter, Objekt- oder Umgebungsparameter nicht vorkommen und daher durch die Generalisationsfähigkeit des Netzes und die Berücksichtigung bei der Variation in den Trainingsdaten eher geringen Einfluss haben. Ähnlich wie bei den UAVDT Testdaten trifft dies in gewisser Weise auch für Störgrößen wie z.B. Rauschen, Kontrast oder Unschärfe zu.

6.3 Analyse von Parametereinflüssen auf die Detektionsleistung

Im dritten Teil werden die Auswirkungen einer gezielten **Variation einzelner entkoppelter Parameter** auf das Detektormodell und die Detektionsleistung betrachtet (zugehörige Forschungsfragen siehe Tab. 32).

Tab. 32 Wiederholung der Forschungsfragen zur Analyse von Parametereinflüssen auf die Detektionsleistung

Parametereinflüsse - Forschungsfragen	
1.	Welche Einflüsse haben verschiedene **Objekt-, Umgebungs-, Sensor- und Simulationsparameter** auf die Detektionsleistung?
2.	Wie müssen synthetische Trainingsdatensätze in Bezug auf **Datensatz-, Sensor- und Simulationsparameter** gestaltet werden?

Dies dient in einer Art rückgekoppelten Analyse zur Bestätigung der mit Hilfe der statistischen Auswertung identifizierten Einflussfaktoren und soll Verbesserungsansätze für die zukünftige synthetische Trainingsdatengenerierung liefern. Im Konzept ist dabei sowohl die Evaluierung der bereits beschriebenen Detektormodelle auf modifizierten Testdaten als auch das Trainieren neuer Modelle auf Basis modifizierter Trainingsdaten vorgesehen.

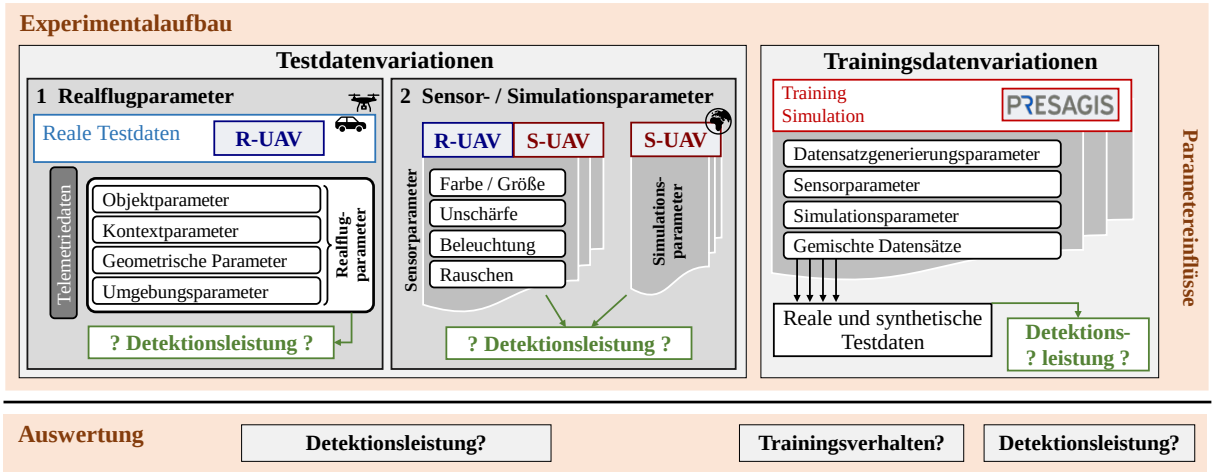


Abb. 88 Konzeptgrafik zur Analyse der Parametereinflüsse: Übersicht über die verschiedenen Gruppierungen und Bestandteile der entkoppelten Parametervariationen bei Trainings- und Testdatensätzen, die im dritten Teil der Untersuchungen zur direkten Einflussanalyse herangezogen wurden.

Die Konzeptgrafik in Abb. 88 gibt einen Überblick über die verschiedenen Bestandteile dieser Analysen. Dieser Block entspricht unter anderem der obersten Ebene und dem Rückkopplungszweig des Gesamtkonzepts aus Abb. 4 und ist entsprechend farbig gekennzeichnet.

Es werden zwei Herangehensweisen betrachtet:

Im ersten Teil wird die Änderung der Detektionsleistung durch bestimmte Parameterzustände in den Testdaten untersucht. Dies umfasst zum einen die in Kapitel 5.2.3 beschriebenen und während des Realflugs erfassten Parameter im R-UAV Testdatensatz, wobei bei der Datengenerierung speziell drauf geachtet wurde, diese möglichst entkoppelt voneinander zu erfassen. Zum anderen werden darüber hinaus die realen R-UAV Bilder und die synthetisch nachmodellierten Duplikate in Nachhinein mit verschiedenen Sensoreffekten wie z.B. Rauschen oder Unschärfe beaufschlagt, um die diesbezügliche Stabilität des Detektormodells und die Anfälligkeit bestehender Modelle auf sich ändernde Bedingungen zu untersuchen. Bei den synthetischen S-UAV Bildern ist zusätzlich die Variation einiger Simulationsparameter möglich, die ebenfalls in die Auswertung miteinfließt.

Im zweiten Teil des Kapitels liegt der Fokus auf der Evaluierung neuer Detektormodelle, die auf Basis modifizierter synthetischer Trainingsdaten angelernt wurden. Dadurch werden ausgewählte Faktoren der Datensatzgenerierung und deren Einfluss auf das Trainingsverhalten analysiert, um deren Anpassung auf die späteren realen Einsatzbedingungen zu verbessern. Dies schließt den Kreis und bezieht wieder das Training als elementaren Bestandteil des Detektionsalgorithmus in die Auswertung mit ein. Aufgrund erweiterter und aktualisierter Datensätze unterscheiden sich die im Folgenden vorgestellten Daten geringfügig von den in [229] veröffentlichten Werten. Die daraus abgeleiteten Aussagen bleiben jedoch weitestgehend unverändert.

6.3.1 Parametervariationen in den Testdatensätzen

Im ersten Teil werden nun die verschiedenen Parametervariationen in den Testdatensätzen und deren Einfluss auf die Detektionsleistung untersucht. Zum Einsatz kommen dabei die bereits in Kapitel 6.1

beschriebenen Detektionsmodelle auf Basis realer, rein synthetischer oder gemischter Trainingsdaten. Die realen und synthetischen Bildpaare (R-UAV / S-UAV) dienen als Testdatensätze.

6.3.1.1 Realflugparameter

Im Gegensatz zu den gängigen verfügbaren Datensätzen enthalten die im Rahmen dieser Arbeit real erflogenen R-UAV Testdaten neben den Annotationen der *Bounding Boxen* auch weiterführende Annotationen zu den im Bild vorkommenden Objekt-, Kontext- und Umgebungsparametern, die in Tab. 17 zusammengefasst sind. Auf diese Weise können Anfälligkeiten der trainierten *Black-Box* Detektormodelle gegenüber bestimmten Parametervariation erkannt werden, was in gewisser Weise einer indirekten Analyse des *Content Gap* entspricht und Ansätze für eine Verbesserung der Trainingsdaten liefern kann. Tab. 33 wiederholt die Forschungsfrage und beschreibt das in diesem Kapitel betrachtete Experiment mit der zugehörigen Auswertung.

Tab. 33 Tabellarische Übersicht über die jeweils behandelte Forschungsfragestellung, das zugehörige Experiment und die einzelnen Bestandteile der Auswertung

1.	Welche Einflüsse haben verschiedene Objekt-, Umgebungs-, Sensor- und Simulationsparameter auf die Detektionsleistung?
	Experiment: Filterung und Kategorisierung der Testdaten nach bestimmten Objekt- und Umgebungsparametern
	Auswertung: Analyse und Vergleich der Detektionsleistung auf den jeweiligen Untergruppen des Testdatensatzes - für das synthetisch trainierte Modell - für das real trainierte Modell - für das gemischt trainierte Modell

6.3.1.1.1 Synthetisch trainiertes Modell

Abb. 89 gibt einen Überblick über die Änderungen in Bezug auf die Detektionsleistung für bestimmte Untergruppen, die spezielle Parametereigenschaften aufweisen und vergleicht diese mit der in schwarz eingetragenen Detektionsleistung für den gesamten Datensatz, die als Referenz dient.

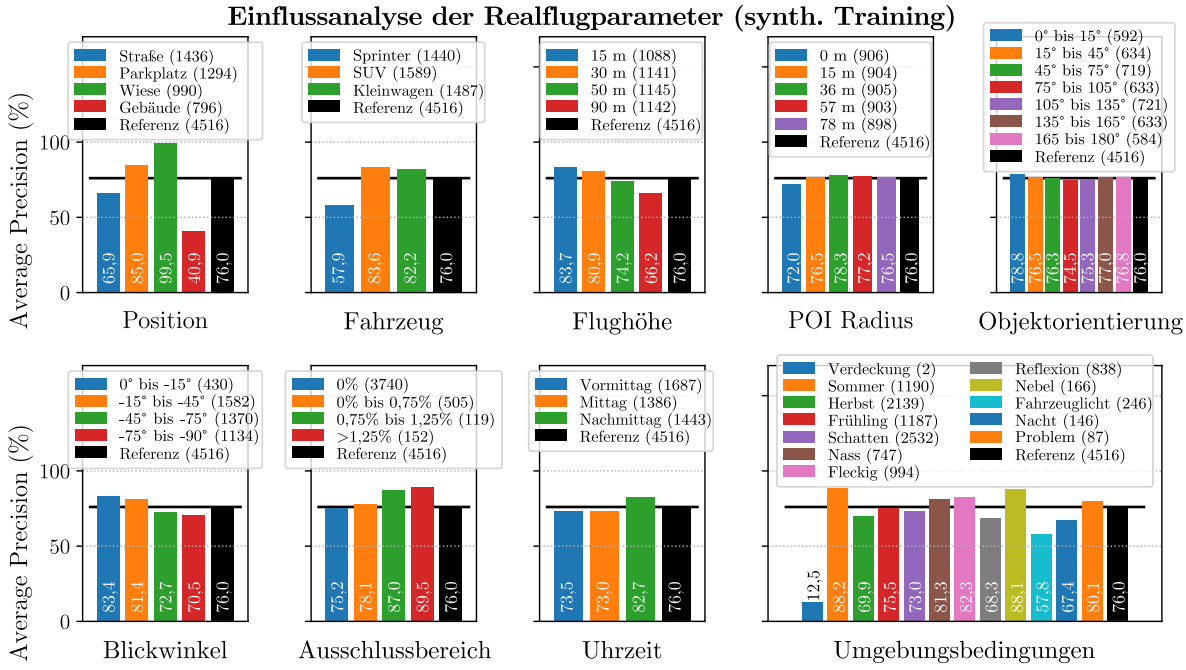


Abb. 89 Überblick über den Einfluss auf die Detektionsleistung bei selektiver Auswahl von Untergruppen an Testdaten mit verschiedenen annotierten Parametern. Zum Vergleich ist jeweils die Referenzleistung auf dem gesamten Datensatz dargestellt. In Klammern ist die Größe der Untergruppe angegeben.
Datensatz: R-UAV; synth. trainiertes Detektormodell

In Abb. 89 wird dazu im ersten Schritt das rein synthetisch trainierte Modell betrachtet, das auf den R-UAV Testdaten eine AP von 76 % erreicht. Der erste Parameter gruppiert die Bilddaten anhand ihrer Aufnahmeposition und zeigt, dass zwischen den verschiedenen Positionen sehr deutliche Leistungsunterschiede vorliegen. Während die Detektionsleistung an den Standorten „Wiese“ und „Parkplatz“ überdurchschnittlich hoch ist, fällt sie am Standort „Straße“ deutlich niedriger aus und erreicht am Standort „Gebäude“ lediglich eine AP von 40,9 %. Diese Differenzen können zwar mehrere Ursachen haben, der ausschlaggebende Faktor ist aber mit großer Wahrscheinlichkeit das Fahrzeugumfeld und der Fahrzeuguntergrund. An den Positionen mit hoher Detektionsleistung befindet sich das Fahrzeug auf einer großen homogenen Fläche (Asphalt, Graß, vgl. Abb. 46). Dadurch heben sich die Testfahrzeuge bei allen Blickwinkeln deutlich vom Hintergrund ab und die Konturen werden nur sehr selten durch Störobjekte im Hintergrund verändert, was in Summe eine einfachere und zuverlässigere Detektion ermöglicht.

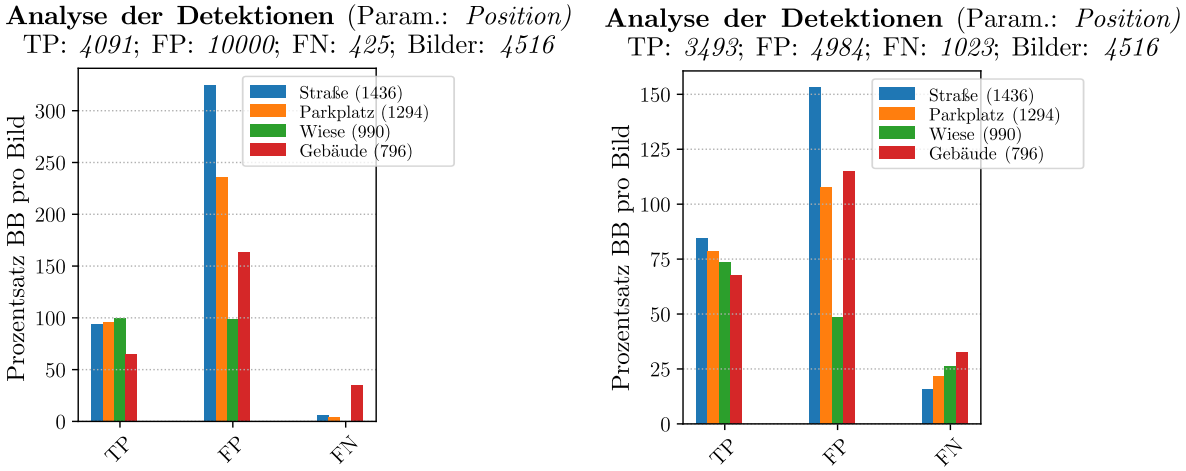


Abb. 90 Analyse der TP/FP/FN Detektionen in Beziehung zu der Anzahl an Bildern für den Parameter „Position“, der das Fahrzeugumfeld repräsentiert. Jedes Bild enthält jeweils nur ein Testfahrzeug.
links: Datensatz: R-UAV; synth. trainiertes Detektormodell
rechts: Datensatz: R-UAV; real trainiertes Detektormodell

Abb. 90 zeigt eine detailliertere Analyse der TP/FP/FN-Detektionen in Bezug zur Anzahl an Bildern für die jeweiligen Untergruppen des betrachteten Parameters „Position“. Es ist zu beachten, dass bei den R-UAV Daten jeweils nur ein Fahrzeug im Bild vorkommt, um eine möglichst entkoppelte Erfassung der verschiedenen Parameter zu gewährleisten. Dies bedeutet zum einen, dass bei einem TP-Wert von 100 % alle Fahrzeuge der Untergruppe korrekt erkannt wurden und zum anderen, dass die FN-Werte direkt von den TP-Werten abhängen und daher keine weitere Information beinhalten. In der linken Grafik von Abb. 90 ist die Aufteilung für das synthetisch trainierte Detektormodell dargestellt. Es zeigt sich, dass neben den Positionen mit einer überdurchschnittlichen AP („Parkplatz“, „Wiese“) auch an der Position „Straße“ ein sehr hoher TP-Anteil erreicht wird. Die niedrigere AP an der Position „Straße“ wird daher zum größten Teil durch die hohe Anzahl an FP hervorgerufen, was wiederum am Fahrzeugumfeld und der dargestellten Szenerie liegt, da die häufig im Bild enthaltenen Betonblöcke neben der Straße sehr leicht fälschlicherweise als Fahrzeug klassifiziert werden können (vgl. Abb. 46). Bei der Position „Gebäude“ hingegen ist zwar die FP-Rate im mittleren Bereich, jedoch werden deutlich weniger der vorkommenden Fahrzeuge korrekt erkannt. Der Abfall in der Detektionsleistung ist daher in diesem Fall auf den niedrigeren TP-Anteil zurückzuführen.

Abb. 89 zeigt außerdem, dass die Detektionsleistung auch sehr stark vom vorkommenden Testfahrzeug abhängig ist. Trotz seiner Größe und seiner markanten Kontur wird der Transporter („Sprinter“) deutlich unzuverlässiger erkannt als die beiden anderen Fahrzeugtypen („SUV“, „Kleinwagen“). Dies ist plausibel, da der Anteil dazu ähnlicher Fahrzeugtypen und damit auch die Variation in Hinblick auf diesen Fahrzeugtypen bei den zur Erstellung der Trainingsdaten verwendeten 38 verschiedenen 3D-Modellen

vergleichsweise gering ist. Dies bedeutet, dass bei der Datengenerierung darauf zu achten ist, dass auch spezielle Objektkonfigurationen mit einer ausreichenden Variation berücksichtigt werden müssen und es daher in vielen Fällen nicht sinnvoll ist, die reale Verteilung beim Training nachzubilden. Diese Verteilung der Variation, die bei einer speziellen Unterkategorie von Objekten zum Teil überproportional hoch ist im Vergleich zu deren tatsächlichen Vorkommen, ist nötig, um den beobachteten Abfall der Detektionsleistung zu verhindern und eine vom Detektionsobjekt unabhängige und stabile Leistung zu erzielen.

Wie bereits in Kapitel 6.1.2 bei der Analyse der synthetischen Testdaten vermutet, ist das Modell in der Lage trotz der verwendeten diskreten Parameterabstufungen bei der Trainingsdatengenerierung zwischen allen kontinuierlichen Abstufungen in den Testdaten zu generalisieren. Dies betrifft die Parameter „Flughöhe“, „POI Radius“, „Objektorientierung“ und „Blickwinkel“ und kann auch im hier betrachteten Fall für die Evaluierung auf den realen R-UAV Testdaten bestätigt werden, da die Auswertung zeigt, dass die Detektionsleistung für die verschiedenen Untergruppen in diesen Parametern sehr ähnlich ist. Lediglich bei höheren Flughöhen und schrägeren Blickwinkeln sinkt die Leistung leicht ab. Die genauere Analyse zeigt, dass dies hauptsächlich auf geringere TP-Werte zurückzuführen ist, die aufgrund der geringen Objektgrößen und dem dadurch verbundenen höheren Schwierigkeitsgrad durchaus zu erwarten sind. Dennoch ist es in Übereinstimmung mit der in Kapitel 6.2.4.1 besprochenen Analyse zur Größenverteilung der *Bounding Boxen* durchaus sinnvoll, trotz der ähnlichen Größenverteilung der Objekte in Trainings- und Testdaten auch durchaus kleinere Objekte beim Training zu berücksichtigen, da der Großteil der FP-Detektionen eine unterdurchschnittliche Größe der *Bounding Box* aufweist (s. Abb. 80) und das Modell daher in diesem Bereich weniger zuverlässige und stabile Ergebnisse liefert, was sich auch hier in dieser Parameteranalyse wiederspiegelt. Daher ist es wiederum durchaus sinnvoll, Randwerte oder sogar Werte, die zwar in der realen Verteilung nicht oder nur indirekt vorkommen, zu berücksichtigen, da diese dennoch die Gesamtstabilität des Modells erhöhen und die Störanfälligkeit von diesem in bestimmten Randbereichen verringern.

Eine Auswertung des Einflusses der Ausschlussbereiche ist nicht zuverlässig möglich, da die Untergruppen zu wenige Daten enthalten. Diese Bereiche kommen jedoch nur bei sehr wenigen Bildern vor und nehmen auch nur einen sehr geringen Teil der Bildfläche ein, weshalb die Effekte vernachlässigbar sind.

Der Einfluss der verschiedenen Umgebungsbedingungen kann ebenfalls nur schwer statistisch ausgewertet werden, da Nebeneffekte durch die einflussreichen Parameter „Position“ und „Fahrzeug“ die Ergebnisse verzerren. Dies ist vor allem dann ein Problem, wenn nur wenige Daten für die einzelnen Untergruppen verfügbar sind. Im Allgemeinen kann jedoch durchaus davon ausgegangen werden, dass das rein synthetisch trainierte Modell gewisse Anfälligkeiten gegenüber unbekannten Umgebungszuständen aufweist, vor allem wenn diese die Erscheinungsform der Sensordaten und Objekte stark beeinflussen, wie das zum Beispiel bei Aufnahmen in der Dämmerung oder bei Reflexionen im Bereich des Fahrzeugs der Fall ist.

6.3.1.1.2 Real trainiertes Modell

Abb. 91 zeigt nun die gleiche Auswertung für das real trainierte Detektormodell, das aufgrund der identischen Domäne zwar keinen *Appearance Gap,* aber durchaus einen nicht zu vernachlässigenden *Content Gap* aufweist.

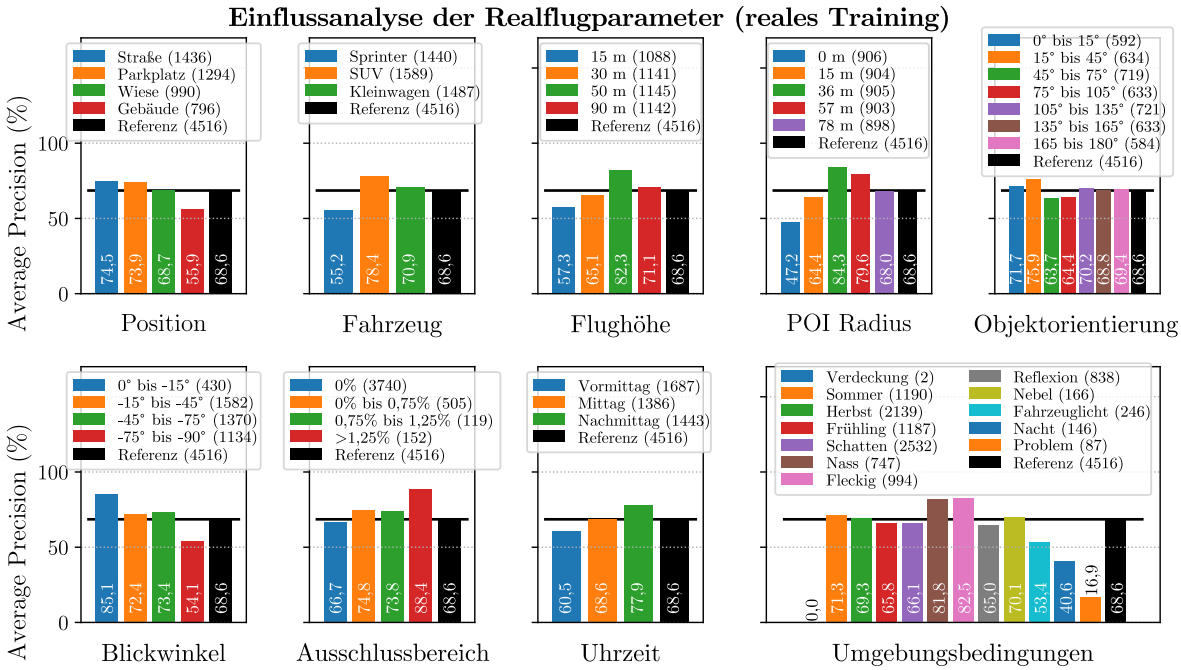


Abb. 91 Überblick über den Einfluss auf die Detektionsleistung bei selektiver Auswahl von Untergruppen an Testdaten mit verschiedenen annotierten Parametern.
Datensatz: R-UAV; real trainiertes Detektormodell

Der Parameter „Position“ beeinflusst wiederum die Detektionsleistung, jedoch in einem geringeren Maß als beim rein synthetischen Training. Die Untergruppe „Straße“ erreicht in diesem Fall die durchschnittliche Detektionsleistung, was auf die gesteigerte TP-Rate zurückzuführen ist. Die FP-Detektionen sind ähnlich verteilt wie bei rein synthetischem Training, ihr absoluter Anteil ist jedoch bei allen Untergruppen um ungefähr die Hälfte gesunken (s. Abb. 90 rechts). Beides deutet auf eine besser Generalisationsfähigkeit des Detektormodells in diesem Bereich hin.

Dennoch weist dieses im Gegensatz zum rein synthetisch trainierten Modell deutliche Abhängigkeiten gegenüber den Parametern „Flughöhe“, „POI Radius“ und „Blickwinkel“ auf. Dies bestätigt den Nutzen der systematischen und diskreten Parametervariation bei der synthetischen Datengenerierung für das Modelltraining. Bei der willkürlichen Erfassung realer Daten sind einige Parameterzustände unter Umständen über- oder unterrepräsentiert oder kommen überhaupt nicht vor, was in Summe im hier betrachteten Fall negative Auswirkungen auf das Trainingsverhalten und die Stabilität gegenüber diesen Parametern mit sich bringt. Dieser Effekt tritt sehr wahrscheinlich auch in beliebigen anderen realen Testdaten auf, ist aufgrund der meist fehlenden systematischen Aufteilung jedoch nur schwer nachzuweisen und wird bei der Berechnung der Gesamtleistung herausgemittelt. Eine systematische Berücksichtigung dieser Parameter auch bei der realen Trainingsdatengenerierung könnte daher auch im Allgemeinen zu einer Steigerung der Detektionsleistung führen. Eine Auswertung des Einflusses der Umgebungsbedingungen ist aufgrund der vielen Nebeneffekte auch hier nicht zielführend.

6.3.1.1.3 Gemischt trainiertes Modell

Im letzten Schritt werden nun die Einflüsse für das mit gemischten Daten trainierte Modell untersucht (s. Abb. 92).

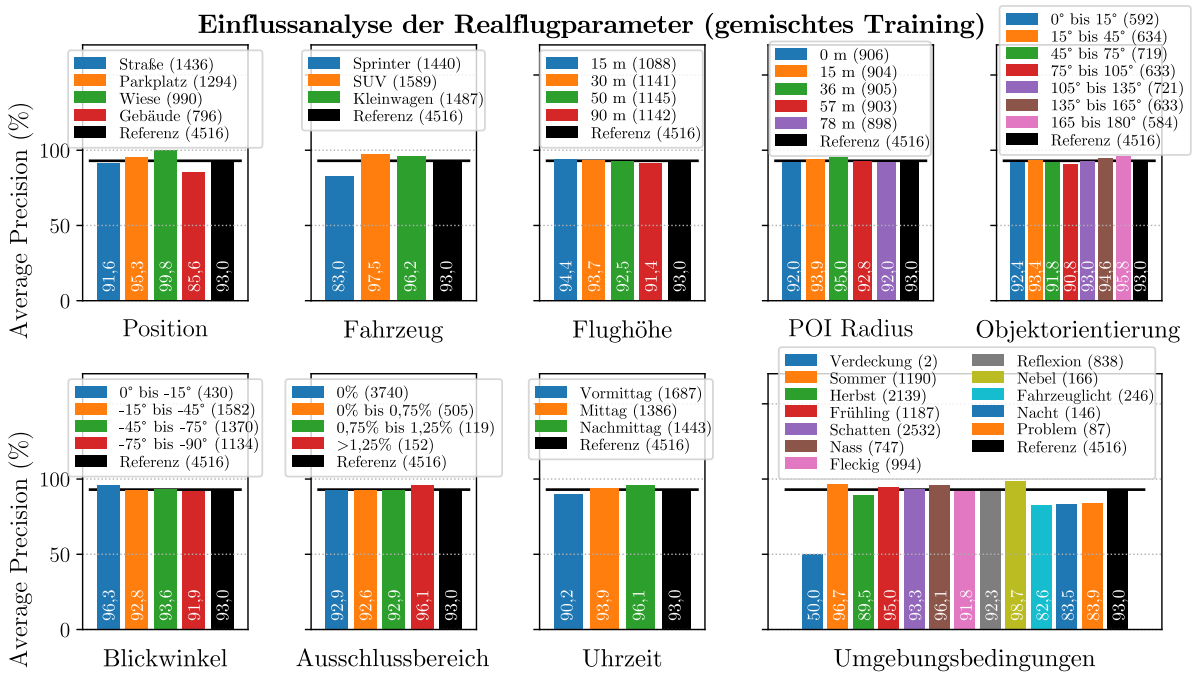


Abb. 92 Überblick über den Einfluss auf die Detektionsleistung bei selektiver Auswahl von Untergruppen an Testdaten mit verschiedenen annotierten Parametern.
Datensatz: R-UAV; gemischt trainiertes Detektormodell

Zusätzlich zur deutlich höheren mittleren Detektionsleistung (AP = 93 %) fällt auf, dass die Schwankungen in den einzelnen Parametervariationen durch die höhere Generalisationsfähigkeit des Modells stark reduziert wurden. Dies bedeutet, dass durch die Beimischung passend gewählter synthetischer Daten nicht nur die Detektionsleistung gesteigert werden kann, sondern insbesondere auch die Stabilität der Detektionen erhöht wird und die Anfälligkeit des Modells gegenüber Störgrößen oder Parametervariationen deutlich reduziert wird.

Dies ermöglicht nun auch die Analyse der Umgebungsbedingungen. Es stellt sich heraus, dass die verschiedenen Jahreszeiten und die damit verbundene Änderung der Hintergrundvegetation keinen nennenswerten Einfluss auf die Detektionsleistung hat, ebenso wie die Uhrzeit oder der Schattenwurf. Es ist erwähnenswert, dass auch eine durch Regen nasse oder fleckige Fahrbahn, Reflexionen oder Nebel keine negativen Auswirkungen haben, da diese Effekte wahrscheinlich bereits durch die Variationen in den Trainingsdaten abgedeckt werden. Bilder bei Dämmerung und mit eingeschaltetem Fahrzeuglicht und Bilder, auf denen auch für den menschlichen Betrachter das Testfahrzeug schwer zu erkennen ist, weisen jedoch eine geringfügig niedrigere Detektionsleistung auf.

6.3.1.1.4 Zusammenfassung und Interpretation

Der beschriebene Versuchsaufbau mit den annotierten Parametervariationen ermöglichte eine umfassende Analyse des Content Gaps und der Anfälligkeit der verschiedenen Modelle gegenüber bestimmten Parametereinflüssen. Es hat sich gezeigt, dass beides sehr deutlich von der Trainingskonfiguration abhängt.

In Übereinstimmung mit Kapitel 6.1.4 wiederum wurde dabei nachgewiesen, dass eine gezielte Erweiterung vorhandener realer Trainingsdaten mit synthetischem Bildmaterial aus dem späteren Einsatzszenario im Vergleich das beste und stabilste Detektionsmodell liefert, das nur sehr geringe Anfälligkeiten gegenüber Parametervariationen aufweist.

Sowohl die Position bzw. das Fahrzeugumfeld als auch der Fahrzeugtyp wurden als einflussreiche Parameter identifiziert, was sowohl bei der Anwendung als auch bei der Trainingsdatengenerierung zu berücksichtigen ist.

Dies ist in Übereinstimmung mit den zugehörigen Ergebnissen der Klassifikationsanalyse aus Kapitel 6.2.4.3, bei der ebenfalls die Fahrzeugposition und die dominante Farbe in der *Bounding Box* als relevante Merkmale hervorgehoben wurden.

Der Einfluss der Umgebungsbedingungen hängt von der Zusammensetzung der Trainingsdaten ab und ist bei einem robusten Modell vergleichsweise gering. Es wurde bestätigt, dass die systematische Trainingsdatengenerierung mit diskreten Parameterabstufungen wie sie bei der synthetischen Datenerzeugung verwendet wurde, für ein umfassendes Anlernen der verschiedenen Grundparameter wie zum Beispiel Objektorientierung und -größe sinnvoll ist und auch bei der Zusammenstellung realer Trainingsdaten vermehrt berücksichtigt werden sollte.

Neuere *deep-learning* basierte Detektormodelle weisen aufgrund der tiefen Netzwerkstrukturen eine sehr großes Lernpotential auf und neigen zur Überanpassung. Es hat sich daher herausgestellt, dass es nicht immer vorteilhaft ist, die reale Verteilung der Bildeigenschaften direkt nachzubilden, sondern durchaus darüber hinauszugehen und Randwerte oder spezielle Objektkonfigurationen mit einer ebenso großen Variation beim Training zu berücksichtigen wie die häufiger vorkommenden Zustände, da dies die Gesamtstabilität des Modells erhöht, die Störanfälligkeit verringert und auch in komplexeren Situationen zu einer verlässlicheren Detektionsleistung führt.

6.3.1.2 Sensor- und Simulationsparameter

Tab. 34 Tabellarische Übersicht über die jeweils behandelte Forschungsfragestellung, das zugehörige Experiment und die einzelnen Bestandteile der Auswertung

1.	Welche Einflüsse haben verschiedene Objekt-, Umgebungs-, Sensor- und Simulationsparameter auf die Detektionsleistung?
Experiment:	1. Überlagerung der Testdaten mit verschiedenen Sensoreffekten 2. Generierung der Testdaten mit anderen Simulationsparametern
Auswertung:	Analyse und Vergleich der Detektionsleistung auf den jeweiligen modifizierten Testdatensatzes - für das synthetisch trainierte Modell - für das real trainierte Modell - für das gemischt trainierte Modell

Dieser Teil beschäftigt sich wiederum mit der Untersuchung des Parametereinflusses auf die Detektionsleistung bei der Evaluierung der bestehenden Modelle. Allerdings liegt diesmal der Fokus auf den Auswirkungen einer Überlagerung der R-UAV und S-UAV Testdaten mit verschiedenen Sensor- und Störeffekten und den Auswirkungen der Veränderung verschiedener Simulationsparameter. Dies soll wiederum diesbezügliche Anfälligkeiten der Modelle aufdecken und ermöglicht nun durch die Betrachtung der gekoppelten realen und synthetischen Bildpaare als Testdaten auch detailliertere Aussagen über den *Reality Gap* in diesem Bereich. Die Forschungsfrage in Tab. 34 bleibt unverändert, jedoch wird in diesem Kapitel eine andere experimentelle Herangehensweise untersucht.

Die dabei betrachteten Parametervariationen sind in Kapitel 5.3.1 genauer erläutert. Es wird vermutet, dass die auftretenden Effekte und Einflüsse stark von den verwendeten Daten und dem verwendeten Modell abhängig sind und somit keine allgemein gültigen Aussagen abgeleitet werden können. Die Auswertung liefert aber dennoch wertvolle Hinweise, welche Parameter bei der Trainingsdatengenerierung in Kapitel 6.3.2 zur Erhöhung der allgemeinen Varianz im Datensatz verstärkt betrachtet werden sollten.

6.3.1.2.1 Synthetisch trainiertes Modell

Abb. 93 gibt einen Überblick über die Ergebnisse bei Verwendung des synthetisch trainierten Detektormodells.

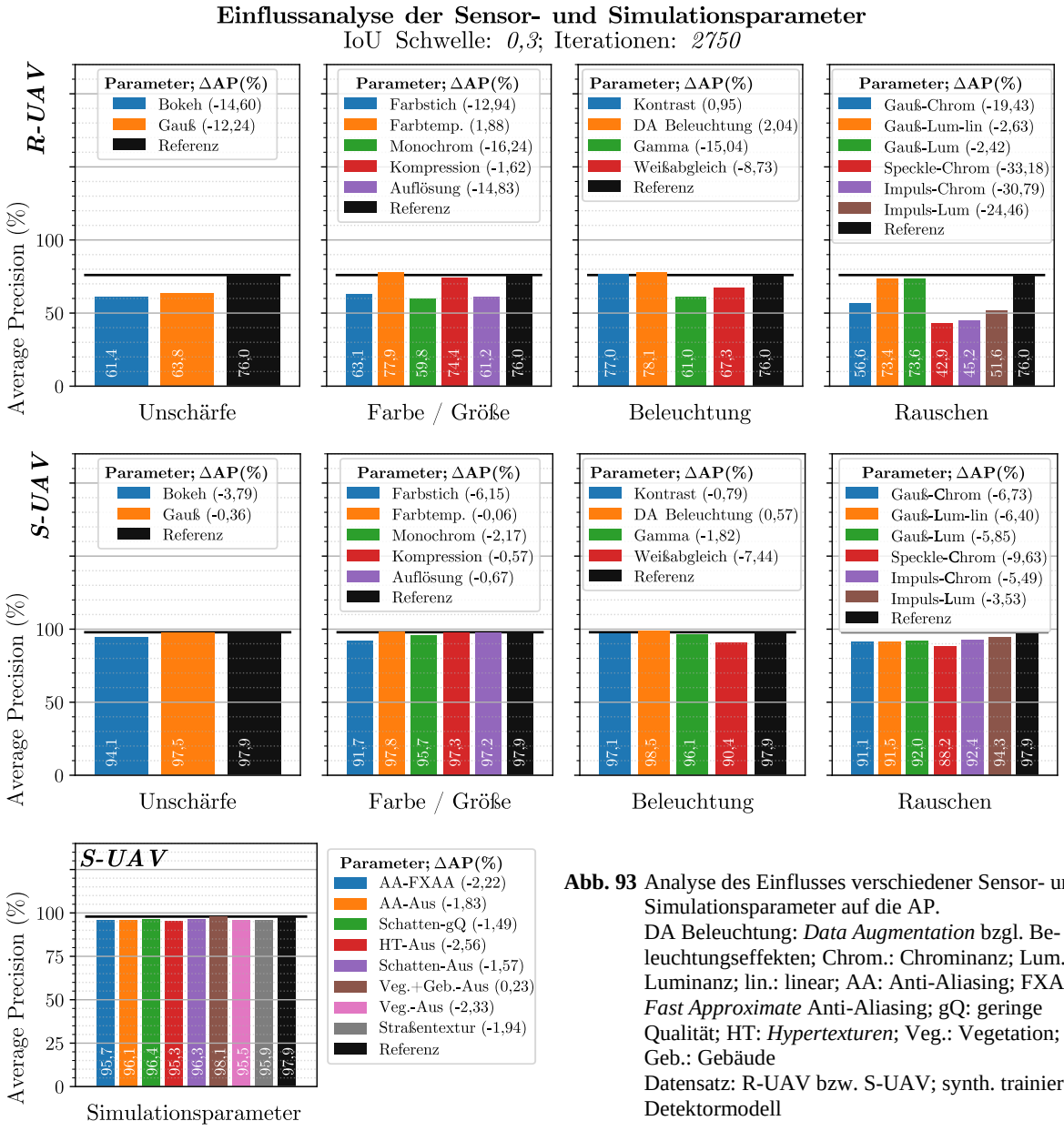


Abb. 93 Analyse des Einflusses verschiedener Sensor- und Simulationsparameter auf die AP.
DA Beleuchtung: *Data Augmentation* bzgl. Beleuchtungseffekten; Chrom.: Chrominanz; Lum.; Luminanz; lin.: linear; AA: Anti-Aliasing; FXAA: *Fast Approximate* Anti-Aliasing; gQ: geringe Qualität; HT: *Hypertexturen*; Veg.: Vegetation; Geb.: Gebäude
Datensatz: R-UAV bzw. S-UAV; synth. trainiertes Detektormodell

Die erste Zeile zeigt die Änderungen für die realen R-UAV Testdaten mit einer AP von 76 % als Ausgangsbasis und die beiden nachfolgenden für die synthetischen S-UAV Duplikate mit einer AP von 97,9 % als Basis. In diesem Fall liegt der Fokus auf der Auswertung der Einflüsse auf den realen Testdaten, da dies eher dem späteren Anwendungsfall entspricht und die Ergebnisse auf den synthetischen S-UAV Daten durch die sehr ähnlichen synthetischen Trainingsdaten und die resultierende sehr hohe Detektionsleistung weniger aussagekräftig sind. Zudem ist das synthetisch trainierte Modell hier vergleichsweise unempfindlich in Bezug auf eine Überlagerung der synthetischen S-UAV Testdaten mit Sensor- oder Störeffekten.

Es stellt sich heraus, dass beide Arten von Unschärfe bei den realen R-UAV Testdaten einen negativen Einfluss auf die Detektionsleistung haben, da sie auf den komplexen realen Sensordaten zu einer größeren Unbestimmtheit bei der Extraktion der relevanten Merkmale führen. Bei den synthetischen Testdaten hat dieser Effekt aufgrund der ausgeprägteren Merkmale keinen Einfluss. Genauere Analyse haben

jedoch gezeigt, dass in beiden Fällen durch die unklareren Merkmale im Bild auch weniger falsche Detektionen (FP) zu verzeichnen sind. Verschiedene Formen und Intensitäten von Unschärfe sollten daher bei der künftigen Trainingsdatengenerierung verstärkt berücksichtigt werden.

Farbänderungen können je nach Ausprägung durchaus einen Einfluss auf die Detektionsleistung haben. Während das Modell gegenüber einer Änderung der Farbtemperatur stabil ist, wirken sich z.B. Farbstiche im Bild negativ auf die Detektionsleistung aus. Auch ein Fehlen der Farbinformation hat einen negativen Einfluss, wobei dieser fast ausschließlich auf deutlich höhere FP-Werte zurückzuführen ist. Der Einfluss einer Kompression der realen Testdaten ist zu vernachlässigen, eine Reduktion der Auflösung jedoch nicht.

Bei der nächsten Gruppe an Parametern ist das Modell in der Lage zwischen Kontrast- und Beleuchtungsänderungen zu generalisieren, eine Erhöhung des Gamma-Wertes oder die Nachbildung von lokalen Überbelichtungen durch den Weißabgleich führt jedoch zu Abstrichen in der Detektionsgüte durch einen früheren Abfall der *Recall*-Werte. Dies spricht dafür, dass auch hier eine Erhöhung der Varianz in den Trainingsdaten sinnvoll sein kann.

Zusätzlich wurden verschiedene Formen von Luminanz- und Chrominanzrauschen untersucht. Es stellt sich heraus, dass Rauschen unter allen betrachteten Parametern den größten Einfluss auf die Detektionsleistung hat, Gaußsches Luminanzrauschen jedoch keinen nennenswerten Abfall der Leistung bewirkt. Der Grund dafür liegt sehr wahrscheinlich darin, dass beim Generierungsprozess die verwendeten synthetischen Trainingsdaten bereits mit zufälligen Intensitäten von weißem Gaußschem Luminanzrauschen überlagert wurden (s. Kapitel 5.1.4) und das Modell daher mit diesem bereits bekannten Störanteil besser umgehen kann. Dieses Ergebnis ist vor allem im Kontext der *Domain Randomization* interessant. Diese besagt, dass bei ausreichend Variation in den synthetisch generierten Trainingsdaten die Realität vom Netzwerk lediglich als weitere Variation betrachtet wird und dass eine Erhöhung der Diversität in den synthetischen Daten effektiver ist als eine Optimierung der Simulation in Hinblick auf die Erzeugung möglichst real wirkender Daten [49, 81]. Im Gegensatz dazu führen alle anderen untersuchten Arten von Rauschen zu einem sehr deutlichen Abfall der Detektionsleistung. Erwähnenswert ist in diesem Zusammenhang vor allem das Impulsrauschen, das auch als *Salt-and-Pepper* Rauschen bezeichnet wird und einzelne Pixel im Bild verändert. Einige Veröffentlichungen des separaten Forschungsbereichs *„One Pixel Attack“* und *„Adversarial Attack“* zeigen in Übereinstimmung damit, dass eine Täuschung des Detektors und ein Abfall der Detektionsleistung durch gezielte Pixelmanipulation oder Rauschen zu erwarten ist. Insgesamt ist daher auch bei dieser Gruppe eine Erhöhung der Varianz und vor allem eine Berücksichtigung aller Arten von Rauschen sinnvoll.

Abschließend wird in der letzten Zeile von Abb. 93 untersucht, inwiefern eine Änderung verschiedener Simulationsparameter bei der Generierung der synthetischen Testdaten die Detektionsleistung beeinflusst. Aufgrund der mehrfach erwähnten Gründe sind auch in diesem Fall bei den synthetischen Testdaten die Einflüsse vergleichsweise gering. Da jedoch bei nahezu allen Parametern dennoch ein geringfügiger Leistungsabfall vorhanden ist, wird in Kapitel 6.3.2 testweise eine Variation der Anti-Aliasing Methoden und der Generierungsparameter *Hypertexturen* untersucht.

6.3.1.2.2 Real trainiertes Modell

Abb. 94 zeigt dieselbe Analyse für das real trainierte Modell. Es wird sehr schnell deutlich, dass größtenteils andere Effekte die Detektionsleistung beeinflussen.

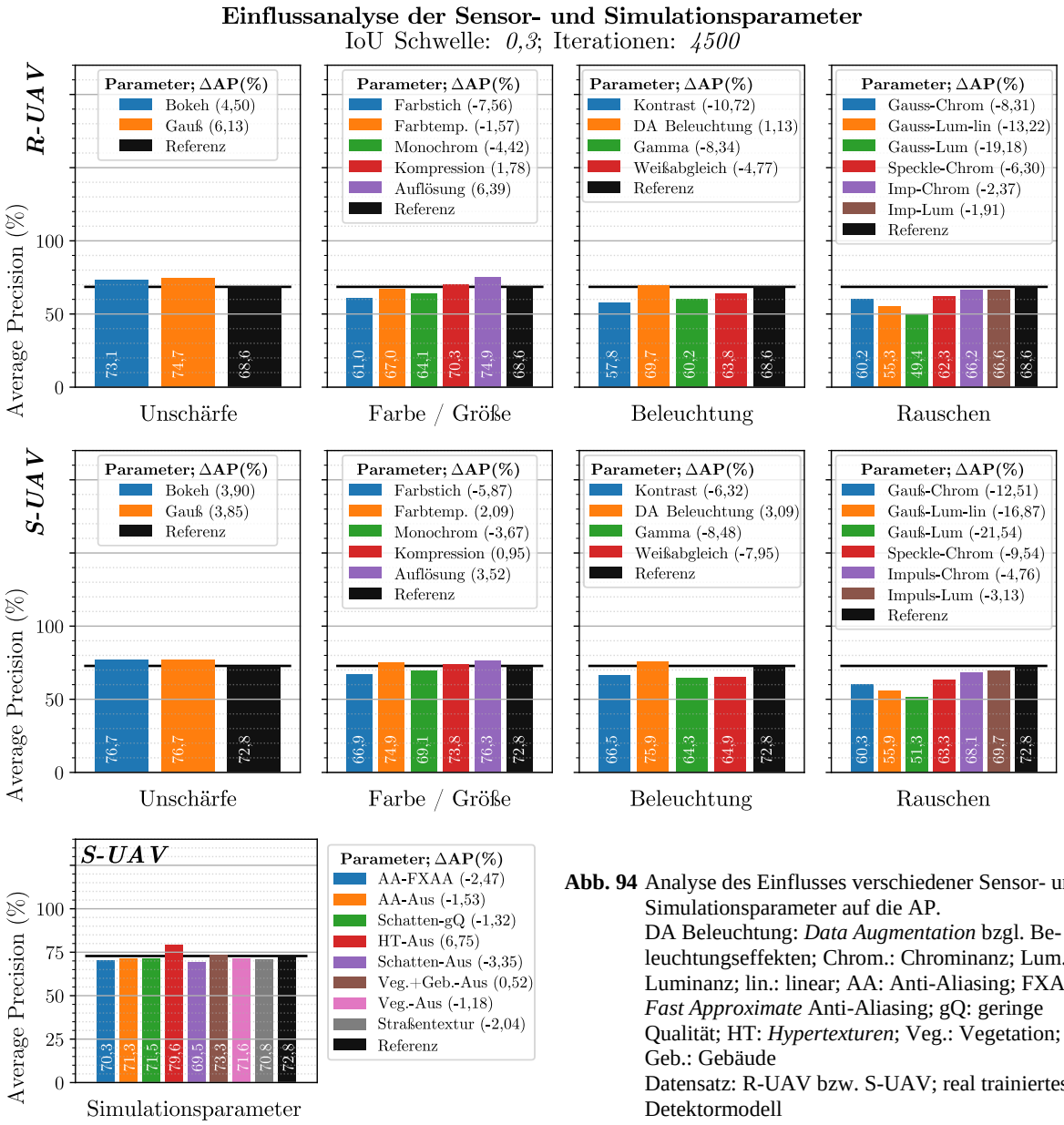


Abb. 94 Analyse des Einflusses verschiedener Sensor- und Simulationsparameter auf die AP.
DA Beleuchtung: *Data Augmentation* bzgl. Beleuchtungseffekten; Chrom.: Chrominanz; Lum.; Luminanz; lin.: linear; AA: Anti-Aliasing; FXAA: *Fast Approximate* Anti-Aliasing; gQ: geringe Qualität; HT: *Hypertexturen*; Veg.: Vegetation; Geb.: Gebäude
Datensatz: R-UAV bzw. S-UAV; real trainiertes Detektormodell

Im Gegensatz zum synthetischen Training haben hier Unschärfe oder eine geringere Auflösung der Testdaten keinen oder sogar einen positiven Einfluss. Der negative Einfluss durch die Überlagerung mit Impulsrauschen ist vermutlich infolge der höheren Stabilität und der geringeren Überanpassung des Modells geringer. Dafür führt eine Überlagerung der Testdaten mit Gauß´schem Luminanzrauschen zu deutlich höheren Leistungseinbußen. Dies ist wahrscheinlich darauf zurückzuführen, dass dieser Effekt bei der realen Trainingsdatengenerierung nicht variiert wird. Das bedeutet, dass in Übereinstimmung mit den Ergebnissen aus dem vorigen Kapitel die Trainingsdatenzusammensetzung und -gestaltung großen Einfluss auf die Parameterabhängigkeit und die Stabilität des Modells hat.

Interessant ist diesem Fall auch die Betrachtung der synthetischen Testdaten. Diese sind deutlich anfälliger gegenüber Parametervariationen als bei synthetischem Training. Dies kann zum Teil an der allgemein niedrigeren AP liegen, wird aber auch dadurch beeinflusst, dass beim Training keine Daten aus der gleichen Domäne enthalten waren.

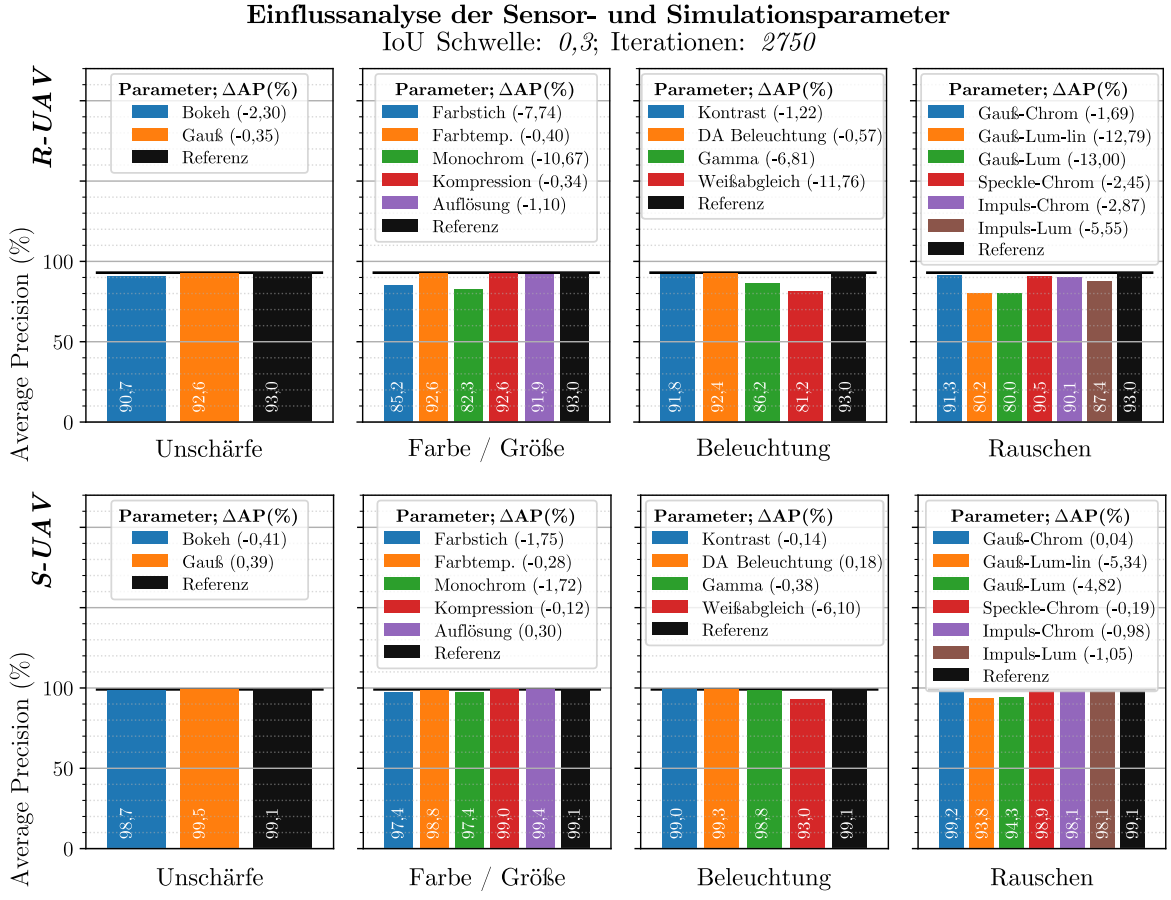
Zudem wird überaus deutlich, dass sich die realen und die synthetischen Testdaten hier sehr ähnlich verhalten. Dabei spielen in beiden Domänen nicht nur die gleichen Parameter eine Rolle, sondern deren Einfluss ist auch in Bezug auf die Größenordnung sehr ähnlich und vergleichbar. Dies ist in

Übereinstimmung mit der in Kapitel 6.1.1 aufgestellten Behauptung, dass bei realem Training der *Reality Gap* bei der Detektionsleistung sehr gering ist. Diese weiterführende Untersuchung zeigt nun, dass diese Übertragbarkeit von der Realität in die Simulation nicht nur pauschal für die Detektionsleistung gilt, sondern auch im Speziellen für eine Vielzahl von Parametereinflüssen und Sensoreffekten. Aufgrund der sehr ähnlichen Einflüsse ist damit eine erste Evaluation bestimmter Effekte und spezieller Szenarien, die in der Realität nur schwer und mit großem Aufwand erfasst werden können, in der Simulation möglich.

Der letzte Teil von Abb. 94 visualisiert wiederum die Auswirkungen verschiedener Simulationsparameter. Im Vergleich zu den Sensoreffekten sind die Einflüsse hier gering und die Detektionsleistung bleibt sehr stabil. Lediglich in Hinblick auf eine Änderung der *Hypertexturen* ist ein Einfluss zu verzeichnen, was für die Berücksichtigung einer erhöhten Variation in diesem Punkt bei der Trainingsdatengenerierung spricht (s. Kapitel 6.3.2).

6.3.1.2.3 Gemischt trainiertes Modell

Abschließend werden in Abb. 95 die Einflüsse auf ein mit gemischten Daten trainiertes Detektormodell dargestellt.



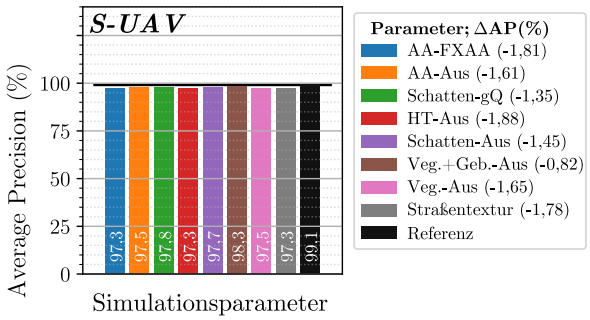


Abb. 95 Analyse des Einflusses verschiedener Sensor- und Simulationsparameter auf die AP.
DA Beleuchtung: *Data Augmentation* bzgl. Beleuchtungseffekten; Chrom.: Chrominanz; Lum.; Luminanz; lin.: linear; AA: Anti-Aliasing; FXAA: *Fast Approximate* Anti-Aliasing; gQ: geringe Qualität; HT: *Hypertexturen*; Veg.: Vegetation; Geb.: Gebäude
Datensatz: R-UAV bzw. S-UAV; mit gemischten Daten trainiertes Detektormodell

Grundsätzlich sind diese ähnlich zu den Einflüssen, die beim rein real trainierten Modell beobachtet wurden, jedoch in deutlich abgeschwächter Form. Durch die Kombination der Trainingsdaten aus beiden Domänen wird nicht nur die Detektionsleistung deutlich gesteigert, sondern auch die Stabilität gegenüber Störeinflüssen erhöht. Dies betrifft beide Domänen der Testdaten und bestätigt die Analysen des vorigen Kapitels, die bei der Betrachtung verschiedener Parameteranalysen zum selben Ergebnis kamen. Es ist zu beachten, dass durch die Beimischung der synthetischen Trainingsdaten die Effekte auf den synthetischen Testdaten geringere Schwankungen verursachen als auf den realen Testdaten und somit ein gewisser *Reality Gap* bei der Auswertung entsteht. Dieser muss beispielsweise dann berücksichtigt werden, wenn spezielle Szenarien und Einflussfaktoren in der Simulation evaluiert werden sollen und dabei ein gemischt trainiertes Modell zur Anwendung kommt. Der Einfluss verschiedener Simulationsparameter ist in der letzten Zeile von Abb. 95 dargestellt und wie bereits bei den vorherigen Modellen sehr gering.

6.3.1.2.4 Zusammenfassung und Interpretation

Insgesamt lässt sich festhalten, dass auch diese Form der Parameterevaluierung dazu beigetragen hat, in einer Art rückgekoppelten Analyse die Ergebnisse der vorherigen Untersuchungen zum einen zu bestätigen und zum anderen detaillierter zu untersuchen.

Im Allgemeinen hat sich herausgestellt, dass die Einflüsse durch die Überlagerung der Testdaten mit Sensor- und Störeffekten wiederum vom Modell und den Trainingsdaten abhängig sind. Die Auswirkungen auf den synthetischen Testdaten sind teilweise geringer als die auf den realen Testdaten, vor allem wenn beim Training synthetisches Datenmaterial berücksichtigt wurde. Jedoch konnte gezeigt werden, dass sich ein real trainiertes Modell auf realen und synthetischen Testdaten sehr ähnlich verhält und somit der *Reality Gap* in diesem speziellen Fall sehr gering ist. Dies trifft sowohl auf die Detektionsleistung als auch auf die Einflüsse einzelner Parameter zu und ermöglicht daher eine gezielte Parameteranalyse in der Simulation bei der Evaluierung derartiger Modelle.

Rauschen mit seinen verschiedenen Ausprägungen und Formen hatte bei allen Trainingskonfigurationen im Vergleich der untersuchten Parameter die größten negativen Auswirkungen und sollte daher bei der zukünftigen Trainingsdatengenerierung durch eine höhere Variation verstärkt berücksichtigt werden. Der Einfluss der untersuchten Simulationsparameter bei der synthetischen Testdatengenerierung war hingegen durchweg gering. In Übereinstimmung mit den Ergebnissen der vorigen Kapitel wurde auch deutlich, dass das mit gemischten Daten aus beiden Domänen trainierte Modell nicht nur die höchste Detektionsleistung aufweist, sondern vor allem auch die höchste Stabilität und Generalisationsfähigkeit gegenüber Sensoreffekten und Störeinflüssen.

6.3.2 Parametervariationen bei der Trainingsdatengenerierung

In diesem Kapitel liegt der Fokus der Untersuchungen auf dem Trainieren neuer Modelle auf Basis modifizierter Trainingsdaten und der Evaluation der dadurch hervorgerufenen Unterschiede in der

Detektionsleistung. Tab. 35 wiederholt die Forschungsfrage und beschreibt das in diesem Kapitel betrachtete Experiment mit der zugehörigen Auswertung.

Tab. 35 Tabellarische Übersicht über die jeweils behandelte Forschungsfragestellung, das zugehörige Experiment und die einzelnen Bestandteile der Auswertung

2.	Wie müssen synthetische Trainingsdatensätze in Bezug auf Datensatz-, Sensor- und Simulationsparameter gestaltet werden?
Experiment:	1. Generierung synthetischer Trainingsdatensätze mit unterschiedlichen Parametervariationen 2. Modelltraining und -evaluierung
Auswertung:	1. Analyse der Detektionsleistung aller Modelle und Vergleich mit dem Referenzmodell - auf den realen UAVDT Testdaten - auf den realen R-UAV Daten 2. Analyse der Detektionsleistung für gemischt trainierte Modelle mit optimierten synthetischen Trainingsdaten und Vergleich mit dem Referenzmodell 3. Evaluierung des Trainingsverlaufs ausgewählter Modelle 4. Vergleich der Detektionsleistung ausgewählter Modelle anhand von *Precision-Recall* Kurven für alle vorhandenen Testdatensätze (synth. Testdaten, UAVDT Testdaten, R-UAV, S-UAV)

Es kommen stets rein synthetische Trainingsdaten zum Einsatz, da dadurch ausgewählte Faktoren der Datensatzgenerierung gezielt modifiziert werden können. Dies soll dazu dienen, das Training als elementaren Bestandteil des Detektionsalgorithmus bei der vorliegenden allgemeinen Auswertung der Einflussfaktoren mit zu berücksichtigen und Ansätze für eine möglichst optimale Parameterverteilung bei der Datensatzgenerierung zu liefern. In Kapitel 5.3.2 sind die dabei betrachteten Trainingsdatensätze mit ihren jeweiligen Parameterverteilungen beschrieben. Die entsprechenden Modelle werden anschließend auf dem synthetischen Testdatensatz (s. Kapitel 5.1.4), den realen und synthetischen Bildpaaren (S-UAV und R-UAV) und den realen UAVDT Testdaten evaluiert. Im Folgenden werden die einzelnen Bestandteile der Auswertung dargestellt und Rückschlüsse für eine verbesserte Trainingsdatengenerierung abgeleitet.

Vergleich der Detektionsleistung auf dem realen UAVDT Testdatensatz

Abb. 96 zeigt den Einfluss verschiedener synthetischer Trainingsdatenkonfigurationen auf die Detektionsleistung auf dem UAVDT Testdatensatz. Dabei werden sowohl Datensätze betrachtet, die sich nur in einzelnen Parametern der Datensatzgenerierung oder einzelnen Sensor- und Simulationsparametern von der Referenz unterscheiden, als auch gemischte Datensätze, bei denen Bilddaten aus mehreren Parametervariationen verwendet werden und die dadurch eine breite Verteilung der Varianz aufweisen. Die Größe der Datensätze ist dabei stets identisch. Als Referenz dient das mit den ursprünglichen in Kapitel 5.1.4 beschriebenen synthetischen Trainingsdaten trainierte Modell, das hier auf den UAVDT Testdaten lediglich eine sehr niedrige AP von 12,5 % erzielt.

Die Untersuchung der verschiedenen darauf aufbauenden Trainingsdatenkonfigurationen soll zeigen, welche Parameter bei der synthetischen Datengenerierung Einfluss auf die Leistungsfähigkeit des Detektors haben und wie durch eine optimierte Parameterauswahl der Nutzen synthetischer Daten gesteigert werden kann. Die bisherigen Untersuchungen haben gezeigt, dass in diesem Fall bei Verwendung synthetischer Trainingsdaten und realer UAVDT Testdaten der *Reality Gap* sowohl aus einem *Appearance Gap* als auch einem *Content Gap* besteht, wobei letzterer im Allgemeinen sogar die größeren Leistungsunterschiede verursacht. Dies ist auch ein Grund dafür, dass in Abb. 96 die Parameter der Datensatzgenerierung durchweg einen großen Effekt auf die Leistungsfähigkeit des Modells haben.

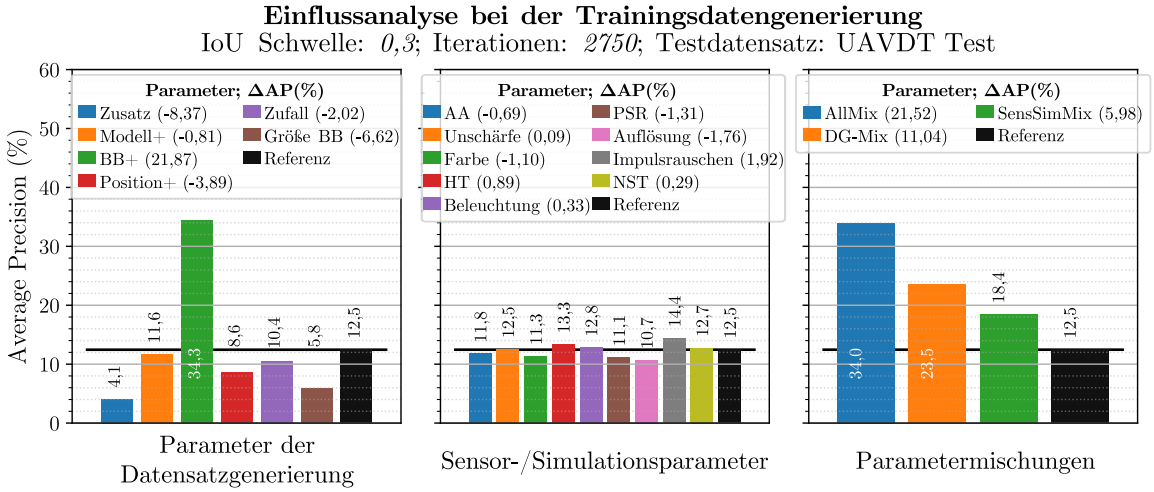


Abb. 96 Vergleich des Einflusses verschiedener Parametervariationen bei der rein synthetischen Trainingsdatengenerierung auf die Leistungsfähigkeit des Detektormodells. Die Referenz zeigt die Detektionsleistung, die mit der ursprünglichen in Kapitel 5.1.4 beschriebenen Parameterverteilung erzielt wurde.
Testdatensatz: reale UAVDT Testdaten

Bei der Analyse ist zu beachten, dass in den synthetischen Referenzdaten stets nur ein Fahrzeug pro Bild enthalten ist. Die Platzierung weiterer Zusatzmodelle („Zusatz“) um dieses Fahrzeug, die selbst jedoch keine Fahrzeuge darstellen, sollte eigentlich die Robustheit des Modells gegenüber Störobjekten erhöhen, führte jedoch zu einer geringeren Detektionsleistung. Die Ursache dafür liegt wahrscheinlich im Aufbau der realen UAVDT Testdaten, da diese stark befahrene innerstädtische Hauptstraßen zeigen und dadurch im Gegensatz zu den Trainingsdaten die um ein Fahrzeug verteilten Objekte in den meisten Fällen wiederum Fahrzeuge darstellen.

Demgegenüber konnte durch die Platzierung weiterer Fahrzeugobjekte im Bild („BB+“) die Detektionsleistung um fast 22 Prozentpunkte gesteigert und eine AP von 34,3 % erzielt werden. Die genauere Analyse zeigt, dass in dieser Konfiguration eine sehr viel höhere Anzahl an TP-Detektionen erzielt wird als mit dem Referenzmodell, jedoch auch eine sehr viel höhere FP-Rate. Es lässt sich schlussfolgern, dass auch hier der *Content Gap* eine wichtige Rolle spielt und die Trainingsdatenzusammensetzung die inhaltlichen Gegebenheiten der späteren Anwendung berücksichtigen sollte.

Die Verwendung einer größeren Anzahl an 3D-Fahrzeugmodellen („Modell+“) führte zu keinen signifikanten Veränderungen. Die Verwendung von 200 zufällig ausgewählten Positionen anstatt 6 speziell ausgewählten diskreten Positionen zur Fahrzeugplatzierung („Position+“), die zufällige Auswahl der geometrischen Platzierungsparameter des Fahrzeugs („Zufall“) und die Verwendung kleinerer Objektgrößen verursachten alle einen deutlichen Leistungsabfall. Dies bestätigt wiederum die bisherigen Ergebnisse, wonach eine diskrete Parameterverteilung in den Trainingsdaten sinnvoller ist als eine kontinuierliche bzw. zufallsbasierte Auswahl. Zudem zeigt dies, dass eine zu hohe Varianz in einem einzelnen Parameter durchaus einen negativen Effekt auf das gesamte Trainingsverhalten haben kann. Die häufig aufgeführte Aussage, dass eine Erhöhung der Varianz in den Trainingsdaten anzustreben ist, ist daher nicht im Allgemeinen gültig, sondern nur unter der Voraussetzung, dass die Varianz breit genug gestreut ist und nicht einzelne Parameter überproportional berücksichtigt werden.

Die Variationen der Sensor- und Simulationsparameter in den Trainingsdaten und die Anwendung verschiedener Methoden zur *Data Augmentation* zeigten nur geringfügige Einflüsse auf die Detektionsleistung, wie im mittleren Teil von Abb. 96 zu sehen ist. Lediglich die Berücksichtigung von Impulsrauschen in den synthetischen Trainingsdaten brachte in Übereinstimmung mit den Ergebnissen aus Kapitel 6.3.1 einen nennenswerten positiven Beitrag von 2 Prozentpunkten.

Der rechte Teil von Abb. 96 zeigt schließlich die Ergebnisse unter Verwendung von Datensätzen, die vermischte Bilddaten aus verschiedenen Parameterkonfigurationen einer Gruppe enthalten. Diese führen alle zu einer deutlichen Leistungssteigerung. Daraus lassen sich mehrere interessante Schlussfolgerungen ableiten. Es zeigt sich bei Betrachtung der Sensor- und Simulationsparameter („SensSimMix“), dass durch Kombination mehrerer Parametervariationen die Leistungsfähigkeit stärker gesteigert werden kann als durch die Summe der einzelnen Beiträge. Dies trifft ebenso für die Mischung der Parameter der Datensatzgenerierung zu, wobei hier der negative Einfluss einzelner Variationen durch die insgesamt breiter gefächerte Varianz vollständig verschwindet und zu einer signifikanten Verbesserung des Modells um 11 Prozentpunkte führt. Durch eine Mischung aller Konfigurationen („AllMix“) kann diese Verbesserung auf fast 22 Prozentpunkte gesteigert werden. Da lediglich durch Platzierung einer größeren Anzahl an Fahrzeugen („BB+“) bereits eine ebenso hohe Leistungssteigerung erzielt werden konnte, sollte an dieser Stelle erneut die Bedeutung des *Content Gaps* und der szenarischen Ähnlichkeit zwischen Trainings- und Testdaten unterstrichen werden. Sind jedoch die späteren Einsatzbedingungen und inhaltlichen Gegebenheiten unbekannt, so zeigt dies, dass auch durch eine möglichst hohe und vor allem breit gestreute Varianz ebenfalls sehr großes Verbesserungspotential besteht.

Auswirkungen auf gemischt trainierte Modelle

In diesem Zusammenhang soll nun auch betrachtet werden, wie sich die Anpassungen bei der Generierung der synthetischen Daten bei der Verwendung gemischter Trainingsdaten aus beiden Domänen auf die Detektionsleistung des Modells auswirken. In Kapitel 6.1.3 wurde bereits gezeigt, dass trotz der sehr ähnlichen realen UAVDT Trainings- und Testdaten durch die Hinzunahme der synthetisch generierten Referenzdaten in den Trainingsdatensatz eine Steigerung der Detektionsleistung möglich ist. Abb. 97 zeigt nun auch den Vergleich für den Fall, dass die beiden Variationen „AllMix“ und „BB+“ als synthetischer Anteil beim Training verwendet werden. Es wird deutlich, dass auch hier die Detektionsleistung im Vergleich zum rein realen Training hauptsächlich durch höhere *Recall*-Werte gesteigert wird. Durch Verwendung der inhaltlich eher auf die UAVDT Testdaten angepassten synthetischen Bilder der „BB+“-Konfiguration wird dabei die größte Leistungssteigerung um 4,76 Prozentpunkte erzielt.

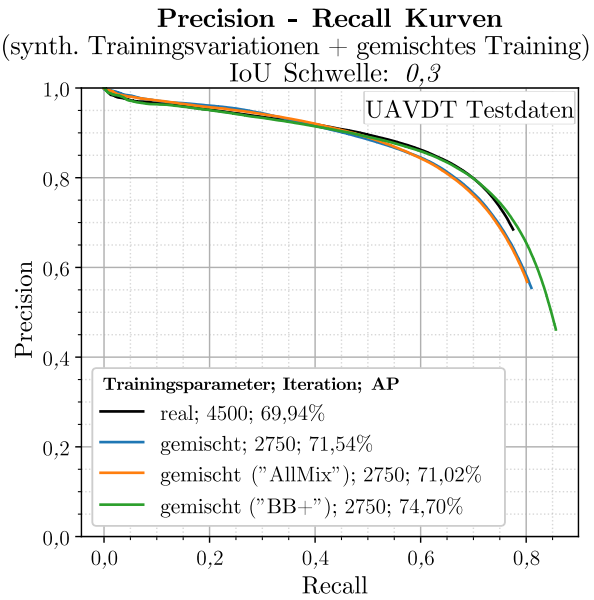


Abb. 97 Vergleich der PR-Kurven für gemischte Trainingsdaten und verschiedene Generierungsparameter der dabei verwendeten synthetischen Bilddaten. Der synthetische Anteil an den Gesamtdaten beträgt dabei stets 79 %. Als Referenz ist die Detektionsleistung mit dem real trainierten Modell dargestellt.
Testdatensatz: UAVDT Testdaten

Vergleich der Detektionsleistung auf dem realen R-UAV Datensatz

Abb. 98 zeigt nun wiederum die Auswertung der rein synthetischen Trainingsdatenkonfigurationen für die realen R-UAV Daten als Testdaten, die von der geografischen Umgebung und der Szenerie den

synthetischen Referenz-Trainingsdaten sehr ähnlich sind, wodurch bereits eine AP von 76,0 % als Referenz erreicht wird und der *Content Gap* vergleichsweise gering ist.

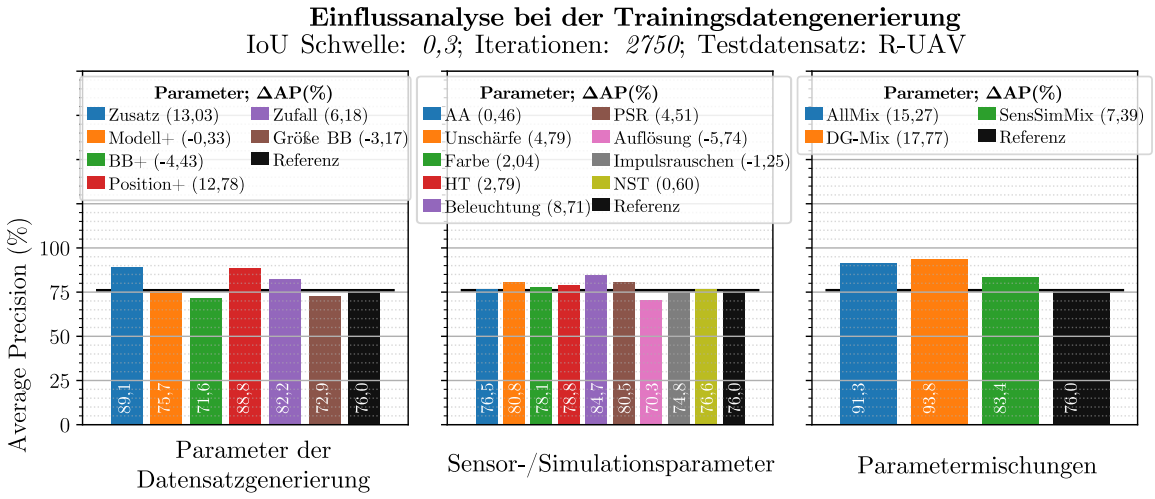


Abb. 98 Vergleich des Einflusses verschiedener Parametervariationen bei der rein synthetischen Trainingsdatengenerierung auf die Leistungsfähigkeit des Detektormodells. Die Referenz zeigt die Detektionsleistung, die mit der ursprünglichen in Kapitel 5.1.4 beschriebenen Parameterverteilung erzielt wurde.
Testdatensatz: reale R-UAV Daten als Testdatensatz

Es zeigt sich wiederum, dass die Parameter der Datensatzgenerierung größere Leistungsunterschiede verursachen als die Sensor- und Simulationsparameter. Die R-UAV Daten enthalten ebenso wie die Trainingsdaten lediglich ein Testfahrzeug pro Bild. Dies erklärt auch, warum in diesem Fall die Platzierung weiterer Zusatzmodelle mit Kleinteilen zu einer Leistungssteigerung führt, während die Platzierung weiterer Fahrzeuge im Bild eher einen leichten Leistungsabfall bewirkt, da dies den *Content Gap* vergrößert und ebenso wie im vorigen Fall zu einem starken Anstieg der FP-Detektionen führt.

Wie bereits in der Klassifikationsanalyse und bei den Parametervariationen der Testdatensätze ermittelt, nimmt hier die Fahrzeugplatzierung („Position+“) starken Einfluss auf die Modellbildung, weshalb eine Erhöhung der Variation in diesem Punkt einen positiven Effekt hat, ebenso wie die zufällige Auswahl der geometrischen Fahrzeugparameter, die zu einer Erhöhung der Variation in Bezug auf verschiedene Blickwinkel über das Testfluggelände führt.

Bis auf einen positiven Effekt bei der Variation der Beleuchtungsbedingungen haben die untersuchten Sensor- und Simulationsparameter nur geringen Einfluss auf die Modellleistung. Auch die betrachteten Methoden zur *Data Augmentation* (PSR, NST) führen einzeln betrachtet nur zu einer geringen Aufwertung der synthetischen Trainingsdaten. Durch die Kombination mehrerer Parametervariationen ist jedoch ähnlich wie bei den UAVDT Testdaten eine deutliche Leistungssteigerung möglich, die sowohl durch höhere *Precision*- als auch *Recall*-Werte hervorgerufen wird.

Abschließend wird nun der Vollständigkeit halber noch kurz auf die Einflüsse der verschiedenen Trainingskonfigurationen bei Anwendung auf die synthetischen Testdaten und die S-UAV Daten eingegangen. Wie bereits die Untersuchungen des Kapitels 6.3.1 gezeigt haben, sind auf dem synthetischen Datenmaterial deutlich geringere Einflüsse auf die Detektionsleistung zu verzeichnen. Dies liegt unter anderem daran, dass durch eine Überanpassung auf die synthetischen Merkmale bereits eine nahezu ideale Detektionsleistung mit dem synthetischen Referenz-Trainingsdatensatz erzielt wurde. Auf den zugehörigen synthetischen Testdaten betrug die AP beispielsweise 99,9 %. Die anderen hier betrachteten Trainingskonfigurationen führten dabei zu keiner Veränderung, bewirkten aber auch keine nennenswerte Verschlechterung. Da auch die synthetischen S-UAV Testdaten bereits sehr ähnliche Szenerien und Bedingungen enthalten, wird als Referenz ebenfalls eine sehr gute AP von 97,9 % erzielt. Trotz teilweise auftretender sehr geringer Verschlechterungen durch einzelne Parametervariationen in der

Trainingsdatenerzeugung kann mit der Mischung aller betrachteter Parametervariationen im Datensatz „AllMix“ oder durch Mischung der betrachteten Parameter der Datensatzgenerierung „DG-Mix“ dennoch eine Verbesserung um rund 2 Prozentpunkte auf eine nahezu ideale AP von 99,8 – 99,9 % erzielt werden. Dies zeigt, dass auch bei einer bereits sehr hohen Modellgüte durch die betrachteten Parameterveränderungen bei der Trainingsdatengenerierung eine Optimierung der Detektionsleistung möglich ist.

Evaluierung und Vergleich des Trainingsverlaufs ausgewählter Modelle

Im Folgenden wird nun das Trainingsverhalten für die einflussreichsten Parametervariationen genauer analysiert. Abb. 99 zeigt den Loss-Verlauf über die Trainingsiterationen. Dieser ist unabhängig von der bisher betrachteten Leistungsanalyse auf den Testdatensätzen und beschreibt das Lernverhalten auf den synthetischen Trainingsdaten. Während der Loss-Verlauf bei den Datensätzen mit gemischten Parametervariationen nur geringfügig höher liegt als bei den Referenzdaten, hebt sich die Konfiguration mit den zusätzlichen Fahrzeugen im Bild („BB+“) deutlich nach oben ab. Dies bedeutet, dass das Modell komplexer wird und weniger auf die Trainingsdaten spezialisiert ist, was für eine verringerte Überanpassung spricht. Zur Stabilitätsanalyse sind im unteren Teil von Abb. 99 die Detektionsleistungen auf dem UAVDT Testdatensatz über die Trainingsdauer dargestellt. Die bisherigen Analysen betrachteten und verglichen die Leistung nach 2750 Iterationen. Der Verlauf macht deutlich, dass die Konfiguration „SensSimMix“ im Mittel eine nur geringfügig höhere Leistung aufweist als das Referenzmodell, wobei beide Kurvenverläufe vergleichsweise stabil sind. Die Konfiguration „DG-Mix“ zeigt über den gesamten Bereich sehr starke Schwankungen, während die Detektionsleistung der Modelle „BB+“ und vor allem „AllMix“ über die Trainingsdauer stabil bleibt und zudem die höchsten absoluten Werte aufweist.

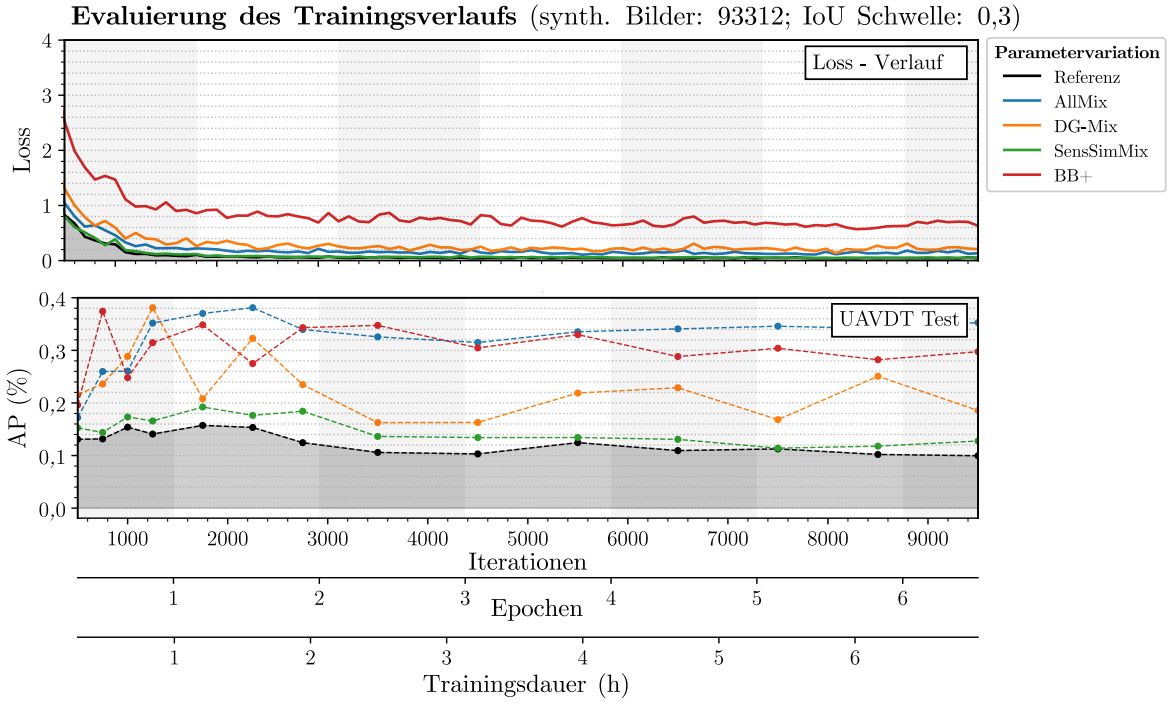


Abb. 99 Evaluierung des Trainingsverlaufs anhand der Loss-Funktion und der resultierenden Detektionsleistung auf den realen UAVDT Testdaten für verschiedene Parametervariationen bei der synthetischen Trainingsdatengenerierung.

Vergleich der Detektionsleistung ausgewählter Modelle anhand von Precision-Recall Kurven

Im Folgenden werden nun anhand von Abb. 100 die *Precision-Recall* Kurven für diejenigen Trainingsdatenkonfigurationen genauer analysiert, die bei den bisher beschriebenen Analysen die größten Leistungssteigerungen gegenüber der Verwendung der synthetischen Referenzdaten verursachten. Es kann

im Allgemeinen festgehalten werden, dass mit jeder der betrachteten Konfigurationen vor allem bei den realen Testdatensätzen höhere *Recall*-Werte erreicht werden.

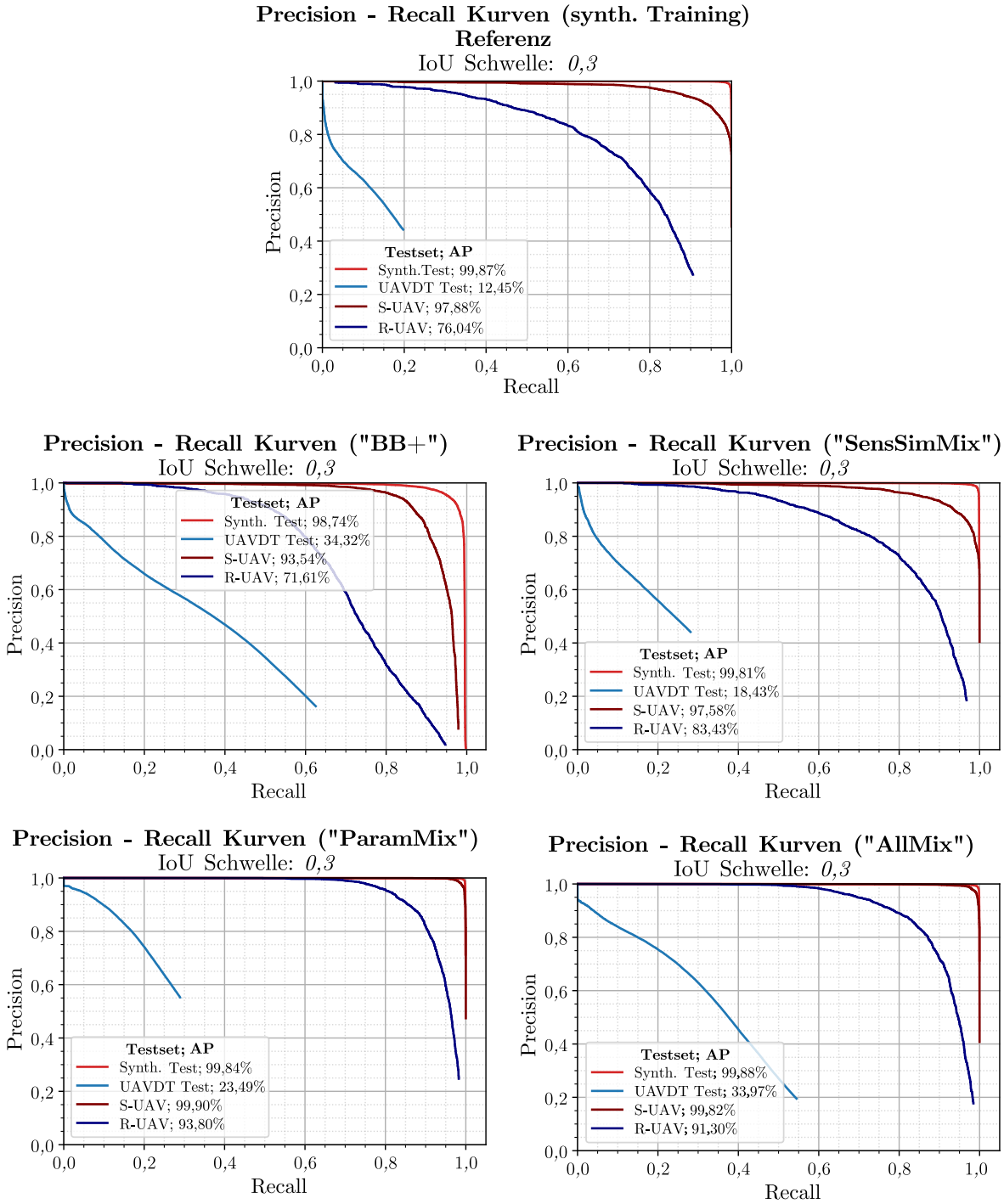


Abb. 100 Vergleichende Analyse der Detektionsleistungen für verschiedene Konfigurationen synthetischer Trainingsdaten auf den zugehörigen realen und synthetischen Testdatensätzen (UAVDT, Synth. Test) und den gekoppelten realen und synthetischen Bildpaaren (R-/S-UAV). Die oberste Abbildung dient als Referenz für die Detektionsleistung mit den bereits analysierten synthetischen Referenz-Trainingsdaten. Die jeweils erzielte AP ist in der Legende angegeben.

Die „BB+“ Trainingsdaten haben je nach Testdatensatz einen sehr spezifischen Einfluss auf den Kurvenverlauf. Bei den UAVDT Testdaten steigt der Recall durch die besser angepasste Szenerie und Objektkonfiguration sehr deutlich an, da mehr der vorkommenden Fahrzeuge erkannt werden. Dafür sinkt die Precision bei den R-UAV Testdaten, da wie bereits erwähnt auch die Anzahl der FP-Detektionen steigt. Die Konfiguration „SensSimMix“ hat auch hier eher geringe Einflüsse. Im Vergleich fällt auf,

dass durch die Mischung der Parameter der Datensatzgenerierung („ParamMix“) nun vor allem die Precision gesteigert werden kann und somit die Zuverlässigkeit der Detektionen steigt. Im Fall der realen UAVDT Testdaten wird dadurch aufgrund der weiterhin niedrigen Recall-Werte dennoch eine relativ geringe Leistungssteigerung erzielt. Durch die Mischung der Trainingsdaten aus allen untersuchten Konfigurationen („AllMix“), ist das gelernte Modell nun in der Lage, die Vorteile beider Untergruppen zu kombinieren und sowohl höhere Precision als auch höhere Recall-Werte zu erzielen.

Tab. 36 bietet im linken Teil einen Überblick über die erzielten Detektionsleistungen auf den beiden realen Testdaten und zeigt, dass damit zwar nicht absolut gesehen für einen speziellen Testdatensatz die beste Leistung erzielt wird, aber das Modell allgemein gesehen für reale Anwendungsfälle als genereller Detektor am besten geeignet ist. Die Evaluierung der Modelle auf den realen R-UAV Testdaten belegt, dass ein rein synthetisch trainiertes Modell sogar deutlich höhere Leistungen erzielen kann, als ein mit realen Daten trainiertes Modell, wenn auf die Einsatzbedingungen angepasstes synthetisches Datenmaterial verwendet wird.

Im rechten Teil von Tab. 36 wird abschließend der Einfluss der Modelle auf die Leistungsunterschiede zwischen den verschiedenen Testdaten evaluiert, die in gewisser Weise den *Content* und *Appearance Gap* repräsentieren. Die Konfiguration „BB+“ halbiert den *Content Gap* durch eine bessere Berücksichtigung der charakteristischen Fahrzeugverteilung im Bild, hat aber keinen positiven Einfluss auf den *Appearance Gap*. Durch eine Variation der Sensor- und Simulationsparameter („SensSimMix“), die hauptsächlich die Erscheinungsform der synthetischen Daten betreffen, wird dagegen lediglich der *Appearance Gap* geringfügig verbessert. Bei der Mischung der Datensatzgenerierungsparameter („ParamMix“) spielen mehrere Effekte eine Rolle. Durch die sehr starke Leistungssteigerung auf den R-UAV Daten steigt der *Content Gap* leicht an, während der *Appearance Gap* sehr deutlich absinkt, da die hohe Leistung auf den R-UAV Daten näher an die ideale Leistung auf den synthetischen S-UAV Duplikaten herankommt. Dies zeigt, dass beide Bestandteile des *Reality Gaps* immer in gewisser Weise verbunden sind und sich gegenseitig beeinflussen. Es wird wiederum deutlich, dass die Kombination aller Parametervariationen „AllMix“ die höchste Generalisationsfähigkeit erzielt, da die Abstände und Unterschiede zwischen den PR-Kurven der Testdatensätze verringert werden und das Modell somit für eine breite Menge an Bilddaten anwendbar ist. Auch in Bezug auf die Verringerung des *Content Gaps* und des Appearance Gaps erreicht diese Konfiguration zwar einzeln betrachtet nicht das Optimum, ist aber insgesamt als einzige Konfiguration in der Lage, beide Bestandteile des *Reality Gaps* zu verringern und ist somit durch die erhöhte und breit gefächerte Varianz für allgemeine Anwendungen die beste synthetische Trainingskonfiguration.

Tab. 36 Übersicht und Vergleich der Detektionsleistungen und deren Differenzen zur Beurteilung der verschiedenen Konfigurationen an Trainingsdaten. Im linken Teil ist die AP und die Differenz zum mit den synthetischen Referenzdaten trainierten Modell für die beiden realen Testdatensätze angegeben. Im rechten Teil sind die Differenzen in Bezug auf die AP und deren Veränderungen für die verschiedenen Bestandteile des *Reality Gaps* aufgelistet.

	UAVDT Test	**R-UAV**	**UAVDT Test ↔ R-UAV** *“Content Gap”*	**R-UAV ↔ S-UAV** *„Appearance Gap“*
Synth. Training Referenz	12,5 %	76,0 %	+ 63,6	+ 21,8
BB+	**34,3 %** + 21,8	71,6 % - 4,4	**+ 37,3** - 26,3	+ 21,9 + 0,1
SensSimMix	18,4 % + 5,9	83,4 % + 7,4	+ 65,0 + 1,4	+ 14,2 - 7,6
ParamMix	23,5 % + 11,0	**93,8 %** + 17,8	+ 70,3 + 6,7	**+ 6,1** - 15,7
AllMix	34,0 % + 21,5	91,3 % + 15,3	+ 57,3 - 6,3	+ 8,5 - 13,3

Zusammenfassung und Interpretation

Insgesamt lässt sich in Bezug auf die Untersuchungen zur Trainingsdatengenerierung festhalten, dass der Fokus darauf gelegt werden sollte, die inhaltlichen und szenarischen Gegebenheiten der späteren realen Anwendung in den synthetischen Trainingsdaten möglichst exakt nachzubilden, da sich

herausgestellt hat, dass die Parameter der Datensatzgenerierung generell einen höheren Einfluss auf die spätere Modellleistung haben als verschiedene Sensor- und Simulationsparameter.

Auch bei der Verwendung gemischter Trainingsdaten kann durch den angepassten synthetischen Datenanteil eine deutliche Leistungssteigerung erreicht werden. Sind das Einsatzgebiet und die dortigen Gegebenheiten im Vorfeld unbekannt, so ist es sinnvoll durch eine sehr hohe und vor allem sehr breit gefächerte Varianz in den Trainingsdaten ein generelles und allgemein anwendbares Modell zu trainieren. Die Aussage, dass eine Erhöhung der Varianz in den Trainingsdaten anzustreben ist, ist daher nicht im Allgemeinen gültig, sondern nur unter der Voraussetzung, dass die Varianz breit genug gestreut ist und nicht einzelne Parameter überproportional berücksichtigt werden. In diesem Zusammenhang wurde erneut gezeigt, dass eine diskrete Parameterverteilung bei der Trainingsdatengenerierung meist bessere Ergebnisse erzielt als eine kontinuierliche bzw. zufallsbasierte Parameterverteilung. Die Verwendung von Methoden der *Data Augmentation* zeigte in Kombination mit den synthetischen Trainingsdaten einzeln betrachtet keinen oder nur geringfügigen Einfluss auf die Detektionsleistung.

Eine Kombination mehrerer Parametervariationen bei der Trainingsdatengenerierung steigert jedoch die Generalisationsfähigkeit und die Detektionsleistung des damit trainierten Modells mehr als der Beitrag einzelner Parameter. Dies begründet auch, dass die Berücksichtigung aller betrachteter Parametervariationen in den Trainingsdaten ("AllMix“) zu einem Datensatz mit sehr hoher und vor allem sehr breit gefächerter Varianz führt und somit insgesamt die zu empfehlende Konstellation darstellt, wenn nicht für einen speziellen Anwendungsfall spezialisierte synthetische Daten erzeugt werden können. Dies ermöglicht das Training eines generellen Detektormodells und reduziert sowohl den *Content* als auch den *Appearance Gap*.

Die Evaluierung zeigte, dass mit synthetischen Trainingsdaten, die auf die späteren Einsatzbedingungen angepasst sind, sogar deutlich höhere Leistungen erzielt werden können als mit allgemeinen realen Benchmark-Daten. Für Fälle, in denen dies nicht möglich ist, und um einen allgemein anwendbaren Detektor zu erhalten, ist jedoch nach wie vor der Einsatz gemischter realer und synthetischer Trainingsdaten empfohlen.

6.4 Zusammenfassung und Schlussfolgerungen

Im Folgenden sollen nun für ein besseres Verständnis die verschiedenen Einzelergebnisse der vorigen Kapitel gesammelt und miteinander in Zusammenhang gebracht werden.

6.4.1 Wahl der Trainingsdatenzusammensetzung und Auswirkungen auf die Detektionsleistung

Der erste Teil der Untersuchungen befasste sich mit der Analyse und dem Vergleich verschiedener Trainingsdatenzusammensetzungen in Bezug auf die Domäne. Dabei kamen ein eigens generierter synthetischer Trainingsdatensatz, ein realer Benchmark-Datensatz (UAVDT) und ein gemischter Trainingsdatensatz zum Einsatz. Die damit erzeugten Detektormodelle wurden nicht nur auf den zugehörigen Testdatensätzen evaluiert, sondern zusätzlich auch auf realen und synthetischen Bildpaaren, die durch den nahezu identischen Bildinhalt eine detailliertere Analyse des *Reality Gaps* ermöglichten. Die grundlegende Fragestellung in diesem Untersuchungsabschnitt war, wie sich die unterschiedlich trainierten Konfigurationen in Bezug auf die Detektionsleistung auf den Testdaten verhalten und welche Leistungsunterschiede zwischen den Domänen festgestellt werden können. Tab. 37 zeigt eine Zusammenfassung der Ergebnisse und der daraus abgeleiteten Schlussfolgerungen.

Tab. 37 Zusammenfassung der Ergebnisse zum Vergleich verschiedener Trainingskonfigurationen und der Auswertung der damit generierten Modelle auf den zugehörigen Testdatensätzen und zusätzlichen gekoppelten realen und synthetischen Bildpaaren

<table>
<tr><th colspan="2" rowspan="2"></th><th colspan="4">Testdaten</th></tr>
<tr><th>Reale Testdaten (UAVDT)</th><th>Synth. Testdaten</th><th>Reale Aufnahmen (R-UAV)</th><th>Synth. Duplikate (S-UAV)</th></tr>
<tr><td rowspan="6">Trainingsdaten</td><td rowspan="2">Reale Benchmarkdaten (UAVDT)</td><td colspan="2">Sehr ähnliche Leistung in beiden Domänen
Geringer Reality Gap in dieser Konfiguration
→ Evaluierung real trainierter Modelle auf synthetischen Daten ist möglich ✓</td><td>Niedrigste Leistung unter den Testdaten
→ Unabhängige reale Bilddaten mit unterschiedlichen Szenarien sind komplexeste Form der Evaluierung</td><td></td></tr>
<tr><td colspan="4">Allgemein und unabhängig von Domäne und Testdatensatz anwendbares Detektormodell mit guter Leistungsfähigkeit und hoher Generalisationsfähigkeit</td></tr>
<tr><td rowspan="2">Synthetischer Referenzdatensatz</td><td>Geringe Leistung ✗, da
- Unterschiedliche Anzahl an Objekten im Bild
- Andere Szenerie
- Überanpassung auf synthetische Daten und geringe Generalisationsfähigkeit
→ Richtungsabhängigkeit des Reality Gaps
→ Rein synthetisches Training zu wenig ✗</td><td>Ideale Leistung ✓
→ Diskrete Schrittweite der synth. Datengenerierung wurde passend gewählt und ist ausreichend für Detektion kontinuierlicher Abstufungen ✓</td><td>Sehr gute Leistung ✓
trotz anderer Domäne, da gleiche Szenerien
→ Reality Gap besteht aus Content Gap und Appearance Gap</td><td>Ideale Leistung ✓, da selbe Simulationsumgebung, Parameter und Szenerie
→ beginnende Überanpassung</td></tr>
<tr><td colspan="4">Leichte Überanpassung auf synthetische Domäne und vor allem Szenerie und daher nicht Allgemein anwendbar, aber sehr gute Leistung, wenn synthetische Daten das spätere Anwendungsgebiet abdecken.</td></tr>
<tr><td rowspan="2">Gemischte Daten</td><td colspan="2">Geringer Leistungsanstieg gegenüber realem Modell durch höhere Recall-Werte ✓</td><td>Sehr deutlicher Leistungsanstieg gegenüber realem Modell ✓
→ Selektive Erweiterung allgemeiner realer Benchmark Daten mit synth. Daten aus dem späteren Einsatzgebiet führt zu signifikant besserer Anpassung des Detektormodells auf die späteren realen Einsatzbedingungen
→ Verringerter Reality Gap
→ weiterhin Aufteilung Content Gap/Appearance Gap</td><td></td></tr>
<tr><td colspan="4">Allgemein anwendbares, sehr stabiles Modell mit gesteigerter Detektions- und Generalisationsleistung gegenüber rein realem Training und hohem Optimierungspotential durch selektive synthetische Datenbeimischung.</td></tr>
</table>

Als Schlussfolgerung lässt sich festhalten, dass die realen Benchmark-Trainingsdaten ein allgemein anwendbares Modell mit guter Leistungsfähigkeit liefern, das sich auf realen und synthetischen Testdaten sehr ähnlich verhält. Die Verwendung des synthetischen Referenzdatensatzes führt zu einem Modell mit leichter Überanpassung auf die synthetische Domäne, das nur dann auch auf realen Daten eine sehr hohe Detektionsleistung erreicht, wenn diese ähnliche Szenarien, Umgebungen und Bedingungen aufweisen wie die synthetischen Trainingsdaten. Durch die gezielte Erweiterung der realen Benchmark-Daten mit synthetischen Daten aus dem späteren Einsatzgebiet wird insgesamt gesehen die höchste Detektions- und Generalisationsleistung erzielt und ein allgemein anwendbares und sehr stabiles Modell angelernt. Die Betrachtung der gekoppelten realen und synthetischen Bildpaare hat dabei bestätigt, dass sich der *Reality Gap* in einen *Content Gap* und einen *Appearance Gap* aufteilt.

6.4.2 Statistische Auswertung der Einflussfaktoren auf die Detektion

Im zweiten Teil wurde ein statistisches Auswerteverfahren entwickelt und angewandt, welches eine gezielte Analyse der einflussreichen Bildeigenschaften erlaubt. Ziel war zum einen die Identifikation relevanter Bildunterschiede zwischen realen und synthetischen Sensordaten und zum anderen die Identifikation der Einflussfaktoren der durch die Bildunterschiede hervorgerufenen Leistungsunterschiede.

Grundlage dafür bildete ein Set an ausgewählten Bildbeschreibern, die als Merkmale dienten. Es stellte sich heraus, dass eine Auswertung auf Basis einer Regressionsanalyse dafür nur bedingt geeignet ist, da sie einerseits ausschließlich für den Vergleich von Bildpaaren eingesetzt werden kann und andererseits eine Vorhersage der Leistungsunterschiede auf den Bildpaaren damit aus mehreren Gründen nur mit sehr geringer Güte möglich war. Die implementierte Klassifikationskette lieferte hingegen durchweg Modelle mit einer hohen Güte, die die Identifikation der für die Modellbildung relevanten Merkmale bzw. Bildeigenschaften erlaubte.

Es wurden dabei drei Teilbereiche betrachtet: Erstens die Unterscheidung zwischen gekoppelten realen und synthetischen Bildpaaren zur Analyse des *Appearance Gaps*, zweitens die Unterscheidung voneinander unabhängiger realer und synthetischer Datensätze zur Analyse des *Reality Gaps* in seiner Gesamtheit und drittens die Klassifikation der Detektionen (TP, FP, FN) auf Basis der Bildinformation in den *Bounding Boxen* zur Analyse der Einflussfaktoren auf die eigentlichen Leistungsunterschiede. Das Vorgehen erlaubte somit eine umfassende Analyse des *Reality Gaps*, d.h. sowohl in Bezug auf die Bildunterschiede als auch in Bezug auf die Leistungsunterschiede.

Tab. 38 Zusammenfassung der Ergebnisse des statistischen Auswerteverfahren zur Identifikation einflussreicher Bildeigenschaften in Bezug auf den *Reality Gap* und seiner Bestandteile für die drei Teilaspekte der Untersuchung. Zusammengehörende bzw. vermehrt auftretende Einflüsse sind durch farbige Blöcke am Rand gekennzeichnet.

		R-UAV / S-UAV	**KITTI / VKITTI**
Gekoppelte Bildpaare real ↔ synthetisch		Ideale Klassifikationsgüte ✓ Baumtiefe = 4 → hohe Ähnlichkeit der Bildpaare	Ideale Klassifikationsgüte ✓ Baumtiefe = 2 → sehr einfaches Modell
		- Rauschen (DBN8) **→ Die idealen gerenderten synthetischen Sensordaten enthalten zu wenig natürliches Rauschen**	- Rauschen (DBN8): überproportionaler Einfluss
		- CLD, CSD, DCD, SCD: Alle 4 Farbdeskriptoren → lokale/globale Histogramm-basierten Farbverteilung und lokale/globale dominante Farben **→ Fehlende feine Strukturen und Details auf den homogenen synthetischen Materialien** **→ Höherer Detailgrad der realen Aufnahmen**	- SCD, CSD → lokale/globale Histogramm-basierte Farbverteilung
		- Farbstich (Col1), Farbtemperatur (Col4) **→ Unterschiedliche Farbgebung durch synth. Modellierung / *Render-Engine***	- Anzahl der vorkommenden Farben (Col5) - Helligkeit (BLC17)
		→ Ideale Unterscheidung der Domäne ist möglich → Auswerteverfahren / Bildbeschreiber wurden passend gewählt → Auswerteverfahren ist **unabhängig von der *Render-Engine*** einsetzbar → Einflüsse sind visuell **anhand von Beispielbildern nachvollziehbar** ✓	

		reale UAVDT Testdaten / synthetisch generierte Testdaten
Unabhängige Datensätze real ↔ synthetisch		Ideale Klassifikationsgüte ✓, Baumtiefe = 4
		- CSD: überproportionaler Einfluss der lokalen Farbverteilung **→ Fehlende feine Strukturen und Details auf den homogenen synthetischen Materialien** **→ Bildaufbau:** großflächige homogene Szenerien mit weniger Struktur und Vielfalt ↔ sehr detailreiche und kleinstrukturierte reale Bilder
		- ästhetische Bewertung der Fotoqualität (IQM0)
		- CLD, DCD: lokale/globale Verteilung der dominanten Farben - Farbstich (Col1), Farbtemperatur (Col4) **→ Unterschiedliche Farbzusammensetzung durch unterschiedliche Szenerie** **→ Unterschiedliche Farbgebung durch synth. Modellierung / *Render-Engine***

	→ Ideale Unterscheidung der Domäne ist möglich → ausgewählte Bildbeschreiber sind universell anwendbar → **Aufteilung des *Reality Gaps* in zwei Gruppen:** **visuelle Erscheinungsform (*Appearance Gap*)** und **synthetischer Bildinhalt (*Content Gap*)**

	Synth. trainiertes Modell → reale UAVDT Testdaten AP = 12,5 %	**Synth. Trainiertes Modell → reale R-UAV Testdaten AP = 76,0 %**
Detektionsergebnisse TP ↔ FP ↔ FN	Gute Klassifikation (F1 = 0,85) ✓ Baumtiefe = 10 → Komplexe Aufgabenstellung	Sehr gute Klassifikation (F1 = 0,97) ✓ Baumtiefe = 10 → Komplexe Aufgabenstellung
	- DCD: überproportionaler Einfluss - Farbigkeit (Col0) und CLD → **Dominante Farbe von Fahrzeugobjekt oder Hintergrund** → ***Appearance Gap*:** Farbgebung durch synth. Modellierung / *Render-Engine*	- DCD: überproportionaler Einfluss - CLD
	- Sha: **semantische Segmentierung, da andere Szenerie / Umfeld der Detektion → hoher *Content Gap*** - Größe der BB (RSD0, RSD1) - Helligkeit (BLC17): → UAVDT Testdaten enthalten Aufnahmen bei Nacht - EHD und Homogenität (GLCM, ET20, ET22) → Struktur der Texturen und lokale räumliche Verteilung der Kanten im Ausschnitt der *Bounding Box* → Blockingartefakte (JPEG) oder **Unterscheidung BB mit und ohne Fahrzeug durch charakteristische Fahrzeugkontur**	**- keine Sha-Merkmale,** da selbe geografische Umgebung und Szenerie → **geringer *Content Gap*** - ästhetische Bewertung der Fotoqualität (IQM0) - Aufnahmeposition (RF3): → Einfluss **Fahrzeugumfeld / -positionierung** **- geometrische Aufnahmeparameter, Objekt- oder Umgebungsparameter ohne Einfluss**
	→ **Unterscheidung korrekter / inkorrekter Detektionen ist im Vorfeld mit großer Zuverlässigkeit möglich** → Auswerteverfahren ist bei niedriger und mittlerer bis hoher Detektionsleistung anwendbar → Bildbeschreiber erfassen die für die Detektion relevanten Bildeigenschaften → Je nach Testdatensatz: *Appearance Gap*, *Content Gap* oder Merkmale aus beiden enthalten → Im Allgemeinen **einflussreich**: - Globale dominante Farbverteilung (DCD) in der BB → Im Allgemeinen **durch Training / Generalisationsfähigkeit abgedeckt**: - Geometrische Aufnahmeparameter, Objekt- und Umgebungsparameter - Störgrößen wie Rauschen, Kontrast, Unschärfe	

Es wurde gezeigt, dass mit dem beschriebenen Verfahren in allen drei Teilbereichen eine Auswertung der Einflussfaktoren möglich ist. Anhand der Farbgebung in Tab. 38 wird ersichtlich, dass dabei im ersten Teil beim Vergleich der Bildpaare wie zu erwarten durch den identischen Bildinhalt weitestgehend Merkmale aufgelistet sind, die den *Appearance Gap* (rötliche Farbgebung) beschreiben. Bei der Unterscheidung realer und synthetischer Datensätze hingegen spielt durch unterschiedliche Szenerien und Umgebungen nicht nur der *Appearance Gap* sondern auch der *Content Gap* (bläuliche Farbgebung) und diesbezügliche Merkmale eine Rolle. Noch deutlicher wird dies bei der Analyse der Detektionsergebnisse. Während bei den inhaltlich zu den synthetischen Trainingsdaten sehr unterschiedlichen UAVDT Testdaten eine ganze Reihe von Merkmalen aus der semantischen Segmentierung der Szenerie hervorstechen, sind diese bei den inhaltlich zu den Trainingsdaten sehr ähnlichen R-UAV Testdaten überhaupt nicht aufgeführt.

In Bezug auf Gestaltungsempfehlungen zur Erhöhung der Ähnlichkeit zwischen realen und synthetischen Daten lässt sich daraus und unter der Einbeziehung von Beispielbildern schlussfolgern, dass der Fokus in erster Linie auf der inhaltlichen Gestaltung und der passenden virtuellen Modellierung im Kontext des späteren Anwendungsfalles liegt, um Einflüsse des Content Gaps gering zu halten. Im Hinblick auf die Erscheinungsform und den Appearance Gap hat sich herausgestellt, dass gerenderte synthetische Sensordaten im Allgemeinen zu wenig natürliches Rauschen und zu wenige feine Strukturen und Details auf den homogenen synthetischen Materialien aufweisen. Zudem werden durch die synthetische Modellierung oder die virtuelle Simulation der Umgebungsbedingungen durch die Render-Engine Unterschiede in der Farbgebung und dem allgemeinen Farbeindruck verursacht, die ebenfalls sehr häufig als einflussreich eingestuft wurden. Auch bei der Unterscheidung korrekter und inkorrekter Detektionen ist eine Minimierung des Content Gaps die Grundvoraussetzung für weitere Optimierungsschritte.

Zudem wurde für beide Testdatensätze ein überproportionaler Einfluss der dominanten Farbverteilung in der Bounding Box festgestellt, was dafürspricht, dass das verwendete deep-learning basierte Detektormodell diesbezüglich sensibel reagiert. Verschiedenste geometrische Aufnahmeparameter, Objekt- und Umgebungsparameter oder Störgrößen wie Rauschen, Kontrast oder Unschärfe wurden hingegen durch das Training bzw. die Generalisationsfähigkeit des Detektors vergleichsweise gut kompensiert und nahmen keinen oder nur geringen Einfluss auf das Detektionsergebnis.

6.4.3 Analyse von Parametereinflüssen auf die Detektionsleistung bei Evaluierung und Trainingsdatengenerierung

Der dritte Teil behandelte schließlich die Auswirkungen einer gezielten Variation einzelner entkoppelter Parameter auf das Detektormodell und die Detektionsleistung. Dabei wurde sowohl die Evaluierung der bereits beschriebenen Detektormodelle auf modifizierten Testdaten als auch das Trainieren neuer Modelle auf Basis modifizierter Trainingsdaten untersucht. Als Parametervariationen der Testdaten wurden zum einen die Auswirkungen verschiedener geometrischer Aufnahmeparameter, Objekt- und Umgebungsparameter, die bei der Erfassung der realen UAV-Sensorbilder als Metadaten mit aufgezeichnet wurden, betrachtet und zum anderen die Anfälligkeit der Modelle gegenüber einer Überlagerung bzw. Modifikation verschiedener Sensor-, Stör- und Simulationsparameter evaluiert. Die Modifikation der Trainingsdaten erfolgte ausschließlich auf Basis der synthetischen Daten, da dadurch ausgewählte Faktoren der Datensatzgenerierung gezielt modifiziert werden können. Dies soll dazu dienen die Trainingsdatengenerierung bzw. das Training an sich zu analysieren und zu optimieren.

Tab. 39 gibt einen Überblick über die Einflüsse der bei den realen Aufnahmen erfassten Metadaten auf das Detektionsergebnis bei Verwendung der mit unterschiedlichen Datenzusammensetzungen trainierten Modelle.

Tab. 39 Überblick über die grobe Abhängigkeit der Detektionsleistung von verschiedenen Parametern für die unterschiedlichen Trainingsdatenzusammensetzungen. Die gemittelte Detektionsleistung (AP) auf dem gesamten Datensatz ist ebenfalls angegeben.
Rot: starke Abhängigkeit; Gelb: mittlere Abhängigkeit; Grün: keine Abhängigkeit
✗: Auswertung aufgrund von Nebeneffekten oder zu geringer Datenbasis nicht aussagekräftig

Training (AP)	**Fahrzeug-position**	**Fahr-zeugtyp**	**Flug-höhe**	**POI Radius**	**Objektori-entierung**	**Blick-winkel**	**Ausschluss-bereich**	**Uhr-zeit**	**Umgebungs-bedingungen**
Real (68,6 %)							✗		✗
Synth. (76,0 %)							✗		✗
Gemischt (93,0 %)							✗		

Der Versuchsaufbau ermöglichte eine umfassende Analyse des *Content Gaps* in Abhängigkeit der verschiedenen Trainingsdatenzusammensetzungen. Das Fahrzeugumfeld und der Fahrzeugtyp konnten im Allgemeinen als einflussreich identifiziert werden, was bei der Trainingsdatengenerierung und der Anwendung zu berücksichtigen ist. Dies deckt sich mit den Ergebnissen des statistischen Auswerteverfahrens, das ebenfalls das Fahrzeugumfeld und die dominante Farbe von Fahrzeugobjekt oder Hintergrund als ausschlaggebend einstufte. Es hat sich außerdem gezeigt, dass eine systematische Trainingsdatengenerierung mit diskreten Parameterabstufungen wie sie bei der synthetischen Datenerzeugung verwendet wurde, für ein umfassendes Anlernen der verschiedenen Grundparameter sinnvoll ist und auch bei der Zusammenstellung realer Trainingsdaten vermehrt berücksichtigt werden sollte. Zudem ist es nicht immer vorteilhaft, nur die reale Verteilung der Bildeigenschaften direkt nachzubilden. Vielmehr sollten die Trainingsdaten durch das hohe Lernpotential der *deep-learning* Detektoren darüber hinausgehen und auch Randwerte oder spezielle Objektkonfigurationen mit einer ebenso großen Variation beim Training berücksichtigen, um eine höhere Zuverlässigkeit und eine geringere Störanfälligkeit in komplexeren Situationen zu erzielen. Tab. 39 zeigt sehr deutlich, dass die Verwendung von gemischten

Trainingsdaten nicht nur in Bezug auf die höhere Detektionsleistung zu empfehlen ist, sondern ebenso in Bezug auf die stark reduzierten Parameterabhängigkeiten und die daraus resultierende höhere Gesamtstabilität.

Im nächsten Schritt wurde nun auf ähnliche Weise die Anfälligkeit der Modelle gegenüber einer Überlagerung bzw. Modifikation verschiedener Sensor-, Stör- und Simulationsparameter evaluiert. Es wird jedoch vermutet, dass diese Einflüsse stark von den verwendeten Daten und dem verwendeten Modell abhängig sind.

Tab. 40 Überblick über die grobe Abhängigkeit der Detektionsleistung von verschiedenen Sensor-, Stör- und Simulationsparametern für die unterschiedlichen Trainingsdatenzusammensetzungen und Testdaten. Die Haupteinflussfaktoren sind jeweils mit angegeben, ebenso wie die Detektionsleistung (AP) auf den Referenzdaten.
Rot: starke negative Abhängigkeit; Gelb: mittlere negative Abhängigkeit; Grün: keine Abhängigkeit

Training	*Test (AP)*	**Unschärfe**	**Farbe / Größe**	**Beleuchtung**	**Rauschen**	**Simparameter**
Real	R-UAV (68,6 %)				Gauß	
	S-UAV (72,8 %)				Gauß	
Synth.	R-UAV (76,0 %)	Blur; Bokeh	Farbstich; Graustufen; Auflösung	Gamma; Weißabgleich	Gauß-Chrom, Speckle, Impuls	
	S-UAV (97,9 %)					
Gemischt	R-UAV (93,0 %)					
	S-UAV (99,1 %)					

Es hat sich herausgestellt, dass die Auswirkungen auf den synthetischen Testdaten zum Teil geringer sind als die auf den realen Testdaten, vor allem wenn beim Training synthetisches Datenmaterial berücksichtigt wurde. Dabei ist hervorzuheben, dass sich ein real trainiertes Modell auf realen und synthetischen Testdaten sehr ähnlich verhält und somit der *Reality Gap* in diesem speziellen Fall sehr gering ist. Dies bestätigt die Ergebnisse aus dem ersten Teil der Untersuchungen und zeigt, dass dies nicht nur auf die Detektionsleistung zutrifft, sondern auch auf die Einflüsse einzelner Parameter und deren Größenordnung, was bei der Evaluierung eine gezielte Parameteranalyse in der Simulation möglich macht. Die untersuchten Arten und Formen von Rauschen verursachten bei allen Trainingskonfigurationen vergleichsweise die größten negativen Auswirkungen, während unterschiedliche Simulationsparameter nur sehr geringen Einfluss auf die Detektionsleistung ausüben. Die Parametereinflüsse auf das gemischt trainierte Detektormodell sind ähnlich zu den Einflüssen auf das real trainierte, jedoch deutlich abgeschwächt. Dies zeigt, dass auch hier eine gemischte Trainingsdatenzusammensetzung positiv zu bewerten ist und zu einer höheren Stabilität gegenüber Störeffekten führt. Insgesamt hat auch dieser Teil dazu beigetragen, in einer Art rückgekoppelten Analyse die Ergebnisse der vorherigen Untersuchungen zum einen zu bestätigen und zum anderen detaillierter zu untersuchen.

Tab. 41 Überblick über die Auswirkungen unterschiedlicher Modifikationen der Trainingsdaten bei Anwendung der neu trainierten Modelle auf die verschiedenen Testdatensätze. Die zugehörige Detektionsleistung (AP) des Referenzmodells ist in der ganz linken Spalte aufgeführt. Trainingsdatenkonfigurationen mit besonders positivem oder negativen Einfluss sind ebenfalls mit der zugehörigen Änderung der AP angegeben.
Rot: negativer Einfluss; Gelb: geringer Einfluss; Grün: positiver Einfluss

Testdaten (AP)	**Datensatzgenerierung**		**Sensor-/Simparameter**	**Parametermischungen**
UAVDT (12,5 %)	Zusatz (-8,4)	BB+ (+21,9)		AllMix (+21,5); DG-Mix (+11,0); SensSimMix (+6,0)
R-UAV (76,0 %)	Zusatz (+13,9); Position (+12,8)			AllMix (+15,3); DG-Mix (+17,8); SensSimMix (+7,4)
S-UAV (97,9 %)				AllMix (+1,9); DG-Mix (+2,0)
Synth. Testdaten (99,9 %)				

Abschließend folgen nun Ergebnisse zum Trainieren neuer Modelle auf Basis modifizierter synthetischer Trainingsdaten und der Evaluation der dadurch hervorgerufenen Unterschiede in der Detektionsleistung. Die entsprechenden Modelle werden auf dem synthetischen Testdatensatz, den realen und

synthetischen Bildpaaren (S-UAV und R-UAV) und den realen UAVDT Testdaten evaluiert. Tab. 41 visualisiert die Einflüsse und die resultierenden Auswirkungen auf die Detektionsleistung.

Durch eine Überanpassung auf die synthetische Domäne und eine bereits sehr hohe Leistung des Referenzmodells sind die Auswirkungen auf den synthetischen Testdaten eher gering. Da dies auch nicht dem typischen Anwendungsfall entspricht, liegt der Fokus auf den realen Testdaten. Die Parameter der Datensatzgenerierung beeinflussen dabei in erster Linie den *Content Gap* und haben dadurch im Allgemeinen deutlich stärkere Auswirkungen auf die vom Modell erreichte Detektionsleistung als verschiedene Sensor- und Simulationsparameter. Wie im Fall der UAVDT Testdaten zu sehen ist, sind dabei durch Parametervariationen bei der Trainingsdatengenerierung sowohl positive als auch negative Einflüsse zu verzeichnen. Das Ziel ist daher eine möglichst exakte Nachbildung der inhaltlichen und szenarischen Gegebenheiten der späteren realen Anwendung in den synthetischen Trainingsdaten.

Zudem konnten auch in diesem Kontext die bisherigen Ergebnisse bestätig werden, wonach eine diskrete Parameterverteilung in den Trainingsdaten sinnvoller ist als eine kontinuierliche bzw. zufallsbasierte Auswahl. Dies betrifft auch die Aussage, dass eine Erhöhung der Varianz in den Trainingsdaten anzustreben ist, die daher nicht im Allgemeinen gültig ist, sondern nur unter der Voraussetzung, dass die Varianz breit genug gestreut ist und nicht einzelne Parameter überproportional berücksichtigt werden. Dies unterstreicht auch die Auswertung der Trainingskonfigurationen aus der Gruppe „Parametermischungen“, bei denen Bilddaten aus mehreren Parametervariationen verwendet werden und die durch die großflächig verteilte Varianz im Allgemeinen zu einer hohen bzw. sehr hohen Leistungssteigerung des damit trainierten Modells führten. Zudem konnten durch die Mischung die negativen Effekte einzelner Parametervariationen komplett vermieden werden (s. DG-Mix bei der Evaluierung auf den UAVDT Testdaten).

Die Verwendung von Methoden der *Data Augmentation* zeigte in Kombination mit den synthetischen Trainingsdaten einzeln betrachtet keinen oder nur geringfügigen Einfluss auf die Detektionsleistung. Durch die Mischung mehrerer Sensor- und Simulationsparameter konnte jedoch eine Leistungssteigerung erreicht werden, was wiederum zeigt, dass eine Kombination von Variationen positivere Auswirkungen hat als die Beiträge einzelner Parameter. Dies begründet auch, dass die Berücksichtigung aller betrachteter Parametervariationen in den Trainingsdaten ("AllMix“) die insgesamt zu empfehlende Konstellation darstellt, wenn nicht für einen speziellen Anwendungsfall spezialisierte synthetische Daten erzeugt werden können. Durch die Mischung aller Parametervariationen wird sowohl der Content als auch der *Appearance Gap* reduziert und es werden höhere *Precision* und höhere *Recall*-Werte erreicht, was für eine allgemein hohe Generalisationsfähigkeit spricht. Die Evaluierung zeigte somit, dass mit synthetischen Trainingsdaten, die auf die späteren Einsatzbedingungen angepasst sind und vielfältige Parametervariationen enthalten, sogar deutlich höhere Leistungen erzielt werden können als mit allgemeinen realen Benchmark-Daten. Für Fälle, in denen dies nicht möglich ist und um einen allgemein anwendbaren Detektor zu erhalten, wird jedoch nach wie vor der Einsatz gemischter realer und synthetischer Trainingsdaten empfohlen.

7 Bewertung und Ausblick

Die stetige Weiterentwicklung unbemannter fliegender Systeme erfordert zunehmend die maschinelle Auswertung und Interpretation von Missionssensordaten. Neuere *deep-learning* basierte Verfahren weisen diesbezüglich eine sehr hohe Leistungsfähigkeit auf. Diese ist jedoch abhängig von der Menge, Verfügbarkeit und Varianz passender Trainingsdaten. Ein Lösungsansatz ist daher die Verwendung virtueller Simulationsumgebungen zur Generierung synthetischer Sensordaten. Die vorliegende Arbeit beschäftigte sich in diesem Kontext mit der grundlegenden Fragestellung, wie das synthetische Datenmaterial für diesen Zweck beschaffen sein muss und welche Randbedingungen beim Einsatz von synthetischen Sensordaten eine Rolle spielen, um den sogenannten *Reality Gap* zu verringern. Dieser beschreibt die Leistungsdifferenz, die durch die Verwendung synthetischer anstatt realer Daten entsteht. Als Anwendungsfall diente die Fahrzeugdetektion auf UAV-basierten Luftbildern. Die Untersuchungen gliederten sich dabei in drei Forschungsschwerpunkte: Die Wahl der Trainingsdatenzusammensetzung, die statistische Auswertung der Einflussfaktoren und die Analyse von Parametereinflüssen. Die einzelnen Forschungsgebiete ergänzen sich untereinander und ermöglichen nur in ihrer Gesamtheit eine zuverlässige Ableitung von Rückschlüssen und Zusammenhängen.

Wissenschaftliche Beiträge

Die wissenschaftlichen Beiträge werden in diesem Abschnitt noch einmal gesammelt dargestellt. Ziel war die Untersuchung und Optimierung des Einsatzes synthetischer Sensordaten in Verbindung mit *deep-learning* basierten Bildverarbeitungsalgorithmen und die Ableitung von Gestaltungsempfehlungen für virtuelle Umgebungen und von Richtlinien für die synthetische Trainingsdatengenerierung. Insgesamt behandelte die vorliegende Arbeit dabei eine Vielzahl relevanter Aspekte bei der Verwendung synthetischer Daten und lieferte eine umfassende Auswertung der zu beachtenden Zusammenhänge und Randbedingungen. Folgende wissenschaftliche Beiträge sind dabei hervorzuheben und wurden teilweise bereits in Vorveröffentlichungen dem Fachpublikum vorgestellt:

- Ein *deep-learning* basierter Detektor wurde mit verschiedenen realen und synthetischen Trainingsdatenzusammensetzungen für den Anwendungsfall der UAV basierten Fahrzeugdetektion angelernt und anschließend evaluiert [51]. Im Gegensatz zu bisherigen Veröffentlichungen wurden dabei bei der Auswertung nicht nur zugehörige, sondern auch unabhängige Testdaten in Form von inhaltsgleichen realen und synthetischen Bildpaaren verwendet und nicht nur Leistungsunterschiede betrachtet, sondern auch das Trainingsverhalten. Somit konnte nach den vorliegenden Recherchen erstmals systematisch und wissenschaftlich belegt werden, dass der *Reality Gap* eine Richtungsabhängigkeit aufweist und sich aus *Content Gap* und *Appearance Gap* zusammensetzt. Zudem wurden Empfehlungen für die Trainingsdatenzusammensetzung abgeleitet. Es wurde nachgewiesen, dass die Verwendung gekoppelter realer und synthetischer Bildpaare dabei im Allgemeinen zu empfehlen ist, da erst dadurch eine detaillierte Beurteilung des *Reality Gaps* ermöglicht wird.
- Es wurde erstmals ein allgemeines statistisches Auswerteverfahren auf Basis einer Klassifikationskette entwickelt, das bei der Anwendung *deep-learning* basierter Algorithmen zur Einflussanalyse verwendet werden kann [188]. Die Validierung zeigte, dass es eine zuverlässige Identifikation der relevanten Einflussfaktoren erlaubt und eine Möglichkeit darstellt, das Verhalten von *deep-learning* basierten Detektormodellen besser verstehen und analysieren zu können. Dabei wurde der *Reality Gap* im Gegensatz zu bisherigen Veröffentlichungen erstmals sowohl in Bezug auf Bildunterschiede als auch in Bezug auf Leistungsunterschiede hin analysiert und es wurden mit einer statistischen Methode Bildeigenschaften identifiziert, die derartige Unterschiede zwischen den Domänen hervorrufen. Auf Basis derer wurden schließlich umfassende

Gestaltungshinweise für virtuelle Umgebungen und eine synthetische Datengenerierung abgeleitet.

- Durch die Untersuchung systematischer Parametervariationen in den Testdaten wurde zudem die Modellanfälligkeit gegenüber äußeren Parametereinflüssen miteinbezogen [229]. Auf diese Weise konnten die Ergebnisse der statistischen Auswertung unabhängig bestätigt und erweitert werden. Die Recherchen haben ergeben, dass eine derartige rückgekoppelte Analyse zur Bestätigung in der bisherigen Literatur nicht durchgeführt wurde. Insgesamt wurden dadurch umfassende Gestaltungsrichtlinien für eine optimierte Anwendung synthetischer Daten bei ähnlichen Anwendungsfällen hergeleitet.
- Zudem wurde eine Optimierung der synthetischen Trainingsdatengenerierung vorgenommen und Empfehlungen für einen möglichst gewinnbringenden Einsatz von synthetischem Datenmaterial erarbeitet. Derartige Empfehlungen sind in bisherigen Veröffentlichungen zwar vereinzelt zu finden, werden aber meist nicht systematisch hergeleitet und bestätigt.

Da sämtliche Untersuchungen unter gleichen Randbedingungen an einem bestimmten Anwendungsfall evaluiert wurden, ist eine gute Vergleichbarkeit der einzelnen Teilergebnisse gegeben. Bestimmte untersuchungsübergreifende Rückschlüsse und Analysen waren zudem durch diesen Aufbau überhaupt erst möglich. Das zugrundeliegende Konzept ist für beliebige Anwendungsfälle und Testalgorithmen adaptierbar und kann als Ausgangspunkt für ähnliche bzw. weiterführende Untersuchungen dienen. Es liefert daher in seiner Gesamtheit einen wichtigen wissenschaftlichen Beitrag zur Untersuchung der Einflussfaktoren und des Trainingsverhaltens bei der Verwendung von synthetischem Datenmaterial für *deep-learning* basierte Algorithmen.

Bewertung und Ausblick

Das vorgestellte Konzept wurde auf Basis des Anwendungsfalls der UAV basierten Fahrzeugdetektion evaluiert. Es bleibt zu überprüfen, inwiefern die dabei erhaltenen Ergebnisse allgemein gültig sind und eine Übertragbarkeit auf andere Anwendungsbereiche gegeben ist. Dies betrifft zunächst die Verwendung anderer Datensätze, anschließend die Einbindung anderer Detektionsalgorithmen und schließlich die Betrachtung anderer CV-Aufgaben neben der Objektdetektion. Auch wenn die Vermutung nahe liegt, dass jede Aufgabe andere Randbedingungen und Gestaltungsrichtlinien erfordert und somit erneute Analysen nötig sind, so ist anzunehmen, dass das Auswertekonzept allgemein eingesetzt werden kann. Es kann an eine Vielzahl von CV-Aufgaben, die auf dem Einsatz deep-learning basierter Algorithmen beruhen, angepasst werden und als Ausgangspunkt für ähnliche derartige Analysen dienen.

In diesem Zusammenhang wäre bei zukünftigen Untersuchungen auch die Verwendung von *GANs* zur Erzeugung künstlicher, synthetischer Bildinstanzen auf Basis eines vorgegebenen Datensatzes interessant. Hier stellt sich ebenfalls die Frage, inwiefern mit derartigem Datenmaterial das Trainingsverhalten verbessert werden kann und welche Auswirkungen bei dieser Art von Daten im Gegensatz zu Daten aus virtuellen Simulationsumgebungen auf den *Reality Gap* und die relevanten Einflussfaktoren zu verzeichnen sind. Neue Methoden aus dem Bereich *„Domain Adaptation“* versuchen hingegen eine Angleichung von Daten aus unterschiedlichen Domänen zu erreichen, wobei der Bildinhalt identisch bleibt und lediglich der Stil angepasst wird. Auch hier ist offen, ob dies eine Möglichkeit zur Reduktion des *Appearance Gaps* darstellt und wenn ja, welche positiven oder negativen Auswirkungen der Einsatz angepasster synthetischer Daten auf die Gesamtvariation in den Trainingsdaten hat. Auch die Verwendung moderner *Game-Engines* oder photorealistischer *Render-Engines* zur Datengenerierung wurde in dieser Arbeit nicht im Detail untersucht, kann aber ebenfalls mit dem vorgestellten Auswerteverfahren analysiert und optimiert werden. Zuletzt bleibt noch zu erwähnen, dass moderne *deep-learning* basierte Verfahren häufig sogenanntes Fine-Tuning unterstützen, bei dem lediglich die letzten spezialisierten Schichten des Netzwerks für eine bestimmte Aufgabe neu angelernt wurden. Da die vorgestellten

Untersuchungen gezeigt haben, dass gemischte Trainingsdaten mit speziell auf den Einsatzzweck angepasstem synthetischen Datenanteil im Allgemeinen die beste Leistungsfähigkeit erzielen, wäre in diesem Zusammenhang ein iterativer Lernprozess durchaus denkbar. Eine Untersuchung, inwiefern dabei zum Beispiel auf Basis falscher Detektionen bzw. auf Basis einer Vorhersage falsche Detektionen mit der vorgestellten Klassifikationskette eine automatisierte synthetische Datengenerierung und damit ein automatisierter iterativer Lernprozess möglich ist, bietet durchaus weiteres Verbesserungspotential und kann den Ausgangspunkt für zukünftige Arbeiten bilden.

Vorveröffentlichungen

Im Rahmen der Erstellung dieser Dissertation wurde auf folgende Vorveröffentlichungen zurückgegriffen:

- Krump, M., Hellert, C., Hummel, G., Stütz, P.: Übersicht zur Übertragbarkeit der Leistungscharakteristik von Computer Vision Algorithmen bei synthetischen und realen Luftbildaufnahmen. In: Längle, T., Puente León, F., and Heizmann, M. (eds.) Forum Bildverarbeitung 2018. KIT Scientific Publishing, Karlsruhe (2018).
- Krump, M., Ruß, M., Stütz, P.: Deep Learning Algorithms for Vehicle Detection on UAV Platforms: First Investigations on the Effects of Synthetic Training. In: Modelling and Simulation for Autonomous Systems. MESAS 2019. Lecture Notes in Computer Science. pp. 50–70 (2020).
- Krump, M., Stütz, P.: UAV Based Vehicle Detection with Synthetic Training: Identification of Performance Factors Using Image Descriptors and Machine Learning. In: Modelling and Simulation for Autonomous Systems. MESAS 2020. Lecture Notes in Computer Science. pp. 62–85 (2021).
- Krump, M., Stütz, P.: UAV Based Vehicle Detection on Real and Synthetic Image Pairs: Performance Differences and Influence Analysis of Context and Simulation Parameters. In: Modelling and Simulation for Autonomous Systems. MESAS 2021. Lecture Notes in Computer Science. pp. 3–25 (2022).
- Krump, M.; Stütz, P.: Deep Learning Based Vehicle Detection on Real and Synthetic Aerial Images: Training Data Composition and Statistical Influence Analysis. Sensors 2023, 23, 3769. (2023).

Literaturverzeichnis

1. Verband Unbemannte Luftfahrt: Analyse des deutschen Drohnenmarktes. (2019).
2. Geng, L., Zhang, Y.F., Wang, J.J., Fuh, J.Y.H., Teo, S.H.: Mission planning of autonomous UAVs for urban surveillance with evolutionary algorithms. In: 2013 10th IEEE International Conference on Control and Automation (ICCA). pp. 828–833. IEEE (2013).
3. Kanistras, K., Martins, G., Rutherford, M.J., Valavanis, K.P.: A survey of unmanned aerial vehicles (UAVs) for traffic monitoring. In: 2013 International Conference on Unmanned Aircraft Systems (ICUAS). pp. 221–234. IEEE (2013).
4. Metni, N., Hamel, T.: A UAV for bridge inspection: Visual servoing control law with orientation limits. Autom. Constr. 17, 3–10 (2007).
5. Máthé, K., Buşoniu, L.: Vision and Control for UAVs: A Survey of General Methods and of Inexpensive Platforms for Infrastructure Inspection. Sensors. 15, 14887–14916 (2015).
6. Herwitz, S.., Johnson, L.., Dunagan, S.., Higgins, R.., Sullivan, D.., Zheng, J., Lobitz, B.., Leung, J.., Gallmeyer, B.., Aoyagi, M., Slye, R.., Brass, J..: Imaging from an unmanned aerial vehicle: agricultural surveillance and decision support. Comput. Electron. Agric. 44, 49–61 (2004).
7. Groos, A.R., Bertschinger, T.J., Kummer, C.M., Erlwein, S., Munz, L., Philipp, A.: The Potential of Low-Cost UAVs and Open-Source Photogrammetry Software for High-Resolution Monitoring of Alpine Glaciers: A Case Study from the Kanderfirn (Swiss Alps). Geosciences. 9, 356 (2019).
8. Drauschke, M., Bartelsen, J., Reidelstürz, P.: Towards UAV-based Forest Monitoring. Proc. Work. UAV-basaed Remote Sens. Methods Monit. Veg. - Kölner Geogr. Arbeiten, 94. (2014).
9. Nagai, M., Tianen Chen, Shibasaki, R., Kumagai, H., Ahmed, A.: UAV-Borne 3-D Mapping System by Multisensor Integration. IEEE Trans. Geosci. Remote Sens. 47, 701–708 (2009).
10. Remondino, F., Barazzetti, L., Nex, F., Scaioni, M., Sarazzi, D.: UAV Photogrammetry for Mapping and 3D Modeling - Current Status and Future Perspectives. ISPRS - Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XXXVIII-1/, 25–31 (2012).
11. Waharte, S., Trigoni, N.: Supporting Search and Rescue Operations with UAVs. In: 2010 International Conference on Emerging Security Technologies. pp. 142–147. IEEE (2010).
12. Erdelj, M., Natalizio, E., Chowdhury, K.R., Akyildiz, I.F.: Help from the Sky: Leveraging UAVs for Disaster Management. IEEE Pervasive Comput. 16, 24–32 (2017).
13. Benjdira, B., Khursheed, T., Koubaa, A., Ammar, A., Ouni, K.: Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3. 1st Int. Conf. Unmanned Veh. Syst. UVS 2019. 1–6 (2019).
14. Pérez, A., Chamoso, P., Parra, V., Sánchez, A.J.: Ground vehicle detection through aerial images taken by a UAV. FUSION 2014 - 17th Int. Conf. Inf. Fusion. (2014).
15. Schulte, A.: Kognitive und kooperative Automation zur Führung unbemannter Luftfahrzeuge. 2. Interdiszip. Work. Kognitive Syst. (2012).
16. Tvaryanas, A.P.: Human systems integration in remotely piloted aircraft operations. Aviat. Space. Environ. Med. 77, 1278–82 (2006).
17. Schmitt, M., Stuetz, P.: Multi-UAV based Helicopter Landing Zone Reconnaissance: Information Level Fusion and Decision Support. In: Engineering Psychology and Cognitive Ergonomics - 14th International Conference, EPCE 2017, Held as Part of HCI International 2017. Springer International Publishing, Vancouver, BC, Canada (2017).
18. Ruf, C., Stütz, P.: Model-Driven Sensor Operation Assistance for a Transport Helicopter Crew in Manned-Unmanned Teaming Missions: Selecting the Automation Level by Machine Decision-Making. In: Savage-Knepshield, P. and Chen, J. (eds.) Advances in Human Factors in

Robots and Unmanned Systems: Proceedings of the AHFE 2016 International Conference on Human Factors in Robots and Unmanned Systems. pp. 253–265. Springer International Publishing, Cham (2017).

19. Ruf, C., Stütz, P.: Model-driven payload sensor operation assistance for a transport helicopter crew in manned–unmanned teaming missions: Assistance realization, modelling experimental evaluation of mental workload. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 10275 LNAI, 51–63 (2017).
20. Hellert, C.: Algorithmenauswahl für den adaptiven Sensoreinsatz an Bord unbemannter Luftfahrzeuge, (2019).
21. Radovic, M., Adarkwa, O., Wang, Q.: Object Recognition in Aerial Images Using Convolutional Neural Networks. J. Imaging. 3, (2017).
22. Du, D., Qi, Y., Yu, H., Yang, Y., Duan, K., Li, G., Zhang, W., Huang, Q., Tian, Q.: The unmanned aerial vehicle benchmark: Object detection and tracking. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 11214 LNCS, 375–391 (2018).
23. Li, Q., Mou, L., Xu, Q., Zhang, Y., Zhu, X.X.: R^3-Net: A Deep Network for Multi-oriented Vehicle Detection in Aerial Images and Videos. IEEE Geosci. Remote Sens. Soc. 57, 5028–5042 (2019).
24. Tayara, H., Soo, K.G., Chong, K.T.: Vehicle Detection and Counting in High-Resolution Aerial Images Using Convolutional Regression Neural Network. IEEE Access. 6, 2220–2230 (2017).
25. Cheng, P., Zhou, G., Zheng, Z.: Detecting and counting vehicles from small low-cost UAV images. Am. Soc. Photogramm. Remote Sens. Annu. Conf. 2009, ASPRS 2009. 1, 138–144 (2009).
26. Azevedo, C.L., Cardoso, J.L., Ben-Akiva, M., Costeira, J.P., Marques, M.: Automatic Vehicle Trajectory Extraction by Aerial Remote Sensing. Procedia - Soc. Behav. Sci. 111, 849–858 (2014).
27. Zheng, Z., Wang, X., Zhou, G., Jiang, L.: Vehicle detection based on morphology from highway aerial images. Int. Geosci. Remote Sens. Symp. 5997–6000 (2012).
28. Girshick, R.: Fast R-CNN. (2015).
29. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. (2015).
30. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. (2015).
31. Redmon, J., Farhadi, A.: YOLO9000: Better, Faster, Stronger. Cvpr2017. 7263–7271 (2016).
32. Redmon, J., Farhadi, A.: YOLOv3: An Incremental Improvement. (2018).
33. Johnson-Roberson, M., Barto, C., Mehta, R., Sridhar, S.N., Vasudevan, R.: Driving in the Matrix: Can Virtual Worlds Replace Human-Generated Annotations for Real World Tasks? In: IEEE International Conference on Robotics and Automation (ICRA). pp. 746–753 (2017).
34. Richter, S.R., Vineet, V., Roth, S., Koltun, V.: Playing for Data: Ground Truth from Computer Games. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 102–118 (2016).
35. Thacker, N.A., Clark, A.F., Barron, J.L., Ross Beveridge, J., Courtney, P., Crum, W.R., Ramesh, V., Clark, C.: Performance characterization in computer vision: A guide to best practices. Comput. Vis. Image Underst. 109, 305–334 (2008).
36. Hummel, G.: On synthetic datasets for development of computer vision algorithms in airborne reconnaissance applications. Dissertation. Universität der Bundeswehr München (2017).
37. Carrillo, J., Gates, B., Monroe, G., Newell, B., Durst, P.: Using Physics-Based M&S for Training

and Testing Machine Learning Algorithms. In: Modelling and Simulation for Autonomous Systems. MESAS 2018. Lecture Notes in Computer Science. pp. 445–455 (2019).

38. Redmill, K.A., Martin, J.I.: Virtual Environment Simulation for Image Processing Sensor Evaluation. Electr. Eng. (2000).

39. Evans, D.C.: Computer generated images for aircraft use. Aeronaut. J. 82, 342–345 (1978).

40. Bossu, J., Gruyer, D., Smal, J.C., Blosseville, J.M.: Validation and Benchmarking for Pedestrian Video Detection based on a Sensors Simulation Platform. IEEE Intell. Veh. Symp. Proc. 115–122 (2010).

41. Nentwig, M., Stamminger, M.: Hardware-in-the-Loop Testing of Computer Vision Based Driver Assistance Systems. 2011 IEEE Intell. Veh. Symp. 339–344 (2011).

42. Nentwig, M., Stamminger, M.: A method for the reproduction of vehicle test drives for the simulation based evaluation of image processing algorithms. IEEE Conf. Intell. Transp. Syst. Proceedings, ITSC. 1307–1312 (2010).

43. Sadeghi, F., Levine, S.: CAD2RL: Real Single-Image Flight without a Single Real Image. In: Robotics: Science and Systems 2017 (2017).

44. Call, B., Beard, R., Taylor, C., Barber, D.: Obstacle Avoidance for Unmanned Air Vehicles Using Image Feature Tracking. AIAA Guid. Navig. Control Conf. Exhib. 1–9 (2006).

45. Hrabar, S.: 3D path planning and stereo-based obstacle avoidance for rotorcraft UAVs. 2008 IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS. 807–814 (2008).

46. Shafaei, A., Little, J.J., Schmidt, M.: Play and Learn: Using Video Games to Train Computer Vision Models. Br. Mach. Vis. Conf. (2016).

47. de Souza, C.R., Gaidon, A., Cabon, Y., Peña, A.M.L.: Procedural Generation of Videos to Train Deep Action Recognition Networks. (2016).

48. Vaudrey, T., Rabe, C., Klette, R., Milburn, J.: Differences between stereo and motion behaviour on synthetic and real-world stereo sequences. In: 2008 23rd International Conference Image and Vision Computing New Zealand. pp. 1–6. IEEE (2008).

49. Tobin, J., Fong, R., Ray, A., Schneider, J., Zaremba, W., Abbeel, P.: Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (2017).

50. Hummel, G., Kovács, L., Stütz, P., Szirányi, T.: Data simulation and testing of visual algorithms in synthetic environments for security sensor networks. Commun. Comput. Inf. Sci. 318 CCIS, 212–215 (2012).

51. Krump, M., Ruß, M., Stütz, P.: Deep Learning Algorithms for Vehicle Detection on UAV Platforms: First Investigations on the Effects of Synthetic Training. In: Modelling and Simulation for Autonomous Systems. MESAS 2019. Lecture Notes in Computer Science. pp. 50–70 (2020).

52. Ellis, T.: Performance metrics and methods for tracking in surveillance. In: Ferryman, J.M. (ed.) 3rd IEEE Workshop on Performance Evaluation of Tracking and Surveillance. pp. 26–31. , Copenhagen, Denmark: IEEE Computer Society (2002).

53. Orjales, F., Lopez Peña, F., Paz-Lopez, A., Deibe, A., Duro, R.J.: On the use of mixed reality for setting up control and coordination strategies for teams of autonomous UAV. Adv. Intell. Syst. Comput. 693, 529–540 (2018).

54. Ferwerda, J.A.: Three Varieties of Realism in Computer Graphics. In: Rogowitz, B.E. and Pappas, T.N. (eds.) Proceedings SPIE Human Vision and Electronic (2003).

55. Myszkowski, K., Tawara, T., Akamine, H., Seidel, H.-P.: Perception-guided global illumination solution for animation rendering. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH ’01. pp. 221–230. ACM Press, New York,

New York, USA (2001).

56. Yee, H., Pattanaik, S., Greenberg, D.P.: Spatiotemporal sensitivity and visual attention for efficient rendering of dynamic environments. ACM Trans. Graph. 20, 39–65 (2001).
57. Shorten, C., Khoshgoftaar, T.M.: A survey on Image Data Augmentation for Deep Learning. J. Big Data. 6, 60 (2019).
58. Mikolajczyk, A., Grochowski, M.: Data augmentation for improving deep learning in image classification problem. In: 2018 International Interdisciplinary PhD Workshop (IIPhDW). pp. 117–122. IEEE (2018).
59. Wu, R., Yan, S., Shan, Y., Dang, Q., Sun, G.: Deep Image: Scaling up Image Recognition. (2015).
60. Moreno-Barea, F.J., Strazzera, F., Jerez, J.M., Urda, D., Franco, L.: Forward Noise Adjustment Scheme for Data Augmentation. In: 2018 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 728–734. IEEE (2018).
61. Kang, G., Dong, X., Zheng, L., Yang, Y.: PatchShuffle Regularization. (2017).
62. Inoue, H.: Data Augmentation by Pairing Samples for Images Classification. (2018).
63. Takahashi, R., Matsubara, T., Uehara, K.: Data Augmentation using Random Image Cropping and Patching for Deep CNNs. (2018).
64. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random Erasing Data Augmentation. (2017).
65. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. (2017).
66. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: SSD: Single Shot MultiBox Detector. (2015).
67. Moosavi-Dezfooli, S.-M., Fawzi, A., Frossard, P.: DeepFool: a simple and accurate method to fool deep neural networks. (2015).
68. Su, J., Vargas, D.V., Sakurai, K.: One Pixel Attack for Fooling Deep Neural Networks. IEEE Trans. Evol. Comput. 23, 828–841 (2019).
69. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and Harnessing Adversarial Examples. (2014).
70. Goodfellow, I.J., Warde-Farley, D., Mirza, M., Courville, A., Bengio, Y.: Maxout Networks. (2013).
71. Li, S., Chen, Y., Peng, Y., Bai, L.: Learning More Robust Features with Adversarial Training. (2018).
72. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative Adversarial Networks. Adv. Neural Inf. Process. Syst. 3, (2014) . http://arxiv.org/abs/1406.2661.
73. Radford, A., Metz, L., Chintala, S.: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. (2015).
74. Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., Greenspan, H.: GAN-based Synthetic Medical Image Augmentation for increased CNN Performance in Liver Lesion Classification. (2018).
75. Zhu, J.-Y., Park, T., Isola, P., Efros, A.A.: Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. (2017).
76. Zhu, X., Liu, Y., Qin, Z., Li, J.: Data Augmentation in Emotion Classification Using Generative Adversarial Networks. (2017).
77. Gatys, L.A., Ecker, A.S., Bethge, M.: A Neural Algorithm of Artistic Style. (2015).
78. Jo, H., Na, Y.-H., Song, J.-B.: Data augmentation using synthesized images for object detection.

In: 2017 17th International Conference on Control, Automation and Systems (ICCAS). pp. 1035–1038. IEEE (2017).

79. Montserrat, D.M., Lin, Q., Allebach, J., Delp, E.J.: Training object detection and recognition CNN models using data augmentation. IS T Int. Symp. Electron. Imaging Sci. Technol. 27–36 (2017).

80. Rozantsev, A., Lepetit, V., Fua, P.: On Rendering Synthetic Images for Training an Object Detector. Comput. Vis. Image Underst. 137, 24–37 (2015).

81. Tremblay, J., Prakash, A., Acuna, D., Brophy, M., Jampani, V., Anil, C., To, T., Cameracci, E., Boochoon, S., Birchfield, S.: Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization. (2018).

82. Battaglia, P.W., Hamrick, J.B., Tenenbaum, J.B.: Simulation as an engine of physical scene understanding. Proc. Natl. Acad. Sci. 110, 18327–18332 (2013).

83. Papon, J., Schoeler, M.: Semantic Pose Using Deep Networks Trained on Synthetic RGB-D. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 774–782. IEEE (2015).

84. Handa, A., Patraucean, V., Badrinarayanan, V., Stent, S., Cipolla, R.: SceneNet: Understanding Real World Indoor Scenes With Synthetic Data. (2015).

85. Kamilaris, A., Brink, C. van den, Karatsiolis, S.: Training Deep Learning Models via Synthetic Data: Application in Unmanned Aerial Vehicles. arXiv. (2019).

86. Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Gordon, D., Zhu, Y., Gupta, A., Farhadi, A.: AI2-THOR: An Interactive 3D Environment for Visual AI. (2017).

87. Marin, J., Vazquez, D., Geronimo, D., Lopez, A.M.: Learning appearance in virtual scenarios for pedestrian detection. In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 137–144. IEEE (2010).

88. Broggi, A., Fascioli, A., Grisleri, P., Graf, T., Meinecke, M.: Model-based validation approaches and matching techniques for automotive vision based pedestrian detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops. pp. 1–1. IEEE.

89. Vazquez, D., Lopez, A.M., Marin, J., Ponsa, D., Geronimo, D.: Virtual and Real World Adaptation for Pedestrian Detection. IEEE Trans. Pattern Anal. Mach. Intell. 36, 797–809 (2014).

90. Xu, J., Vazquez, D., Lopez, A.M., Marin, J., Ponsa, D.: Learning a Part-Based Pedestrian Detector in a Virtual World. IEEE Trans. Intell. Transp. Syst. 15, 2121–2131 (2014).

91. Nentwig, M., Miegler, M., Stamminger, M.: Concerning the Applicability of Computer Graphics for the Evaluation of Image Processing Algorithms. 2012 IEEE Int. Conf. Veh. Electron. Safety, ICVES 2012. 205–210 (2012).

92. Yang, S., Deng, W., Liu, Z., Wang, Y.: Analysis of Illumination Condition Effect on Vehicle Detection in Photo-Realistic Virtual World. SAE Tech. Pap. Part F1298, (2017).

93. Hummel, G., Smirnov, D., Kronenberg, A., Stütz, P.: Prototyping and training of computer vision algorithms in a synthetic UAV mission test bed. AIAA SciTech 2014. 1–10 (2014).

94. Hejrati, M., Ramanan, D.: Analysis by Synthesis: 3D Object Recognition by Object Reconstruction. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 2449–2456. IEEE (2014).

95. Movshovitz-Attias, Y., Sheikh, Y., Naresh Boddeti, V., Wei, Z.: 3D Pose-by-Detection of Vehicles via Discriminatively Reduced Ensembles of Correlation Filters. In: Proceedings of the British Machine Vision Conference 2014. pp. 53.1-53.12. British Machine Vision Association (2014).

96. Pepik, B., Stark, M., Gehler, P., Schiele, B.: Teaching 3D geometry to deformable part models.

In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. pp. 3362–3369. IEEE (2012).

97. Sun, B., Saenko, K.: From Virtual to Reality: Fast Adaptation of Virtual Object Detectors to Real Domains. In: Proceedings of the British Machine Vision Conference 2014. pp. 82.1-82.12. British Machine Vision Association (2014).

98. Peng, X., Sun, B., Ali, K., Saenko, K.: Learning Deep Object Detectors from 3D Models. (2014).

99. Müller, M., Smith, N., Gahem, B.: A Benchmark and Simulator for UAV Tracking. In: Computer Vision - ECCV 2016. pp. 445–461 (2016).

100. Müller, M., Casser, V., Lahoud, J., Smith, N., Ghanem, B.: Sim4CV: A Photo-Realistic Simulator for Computer Vision Applications. Int. J. Comput. Vis. 126, 902–919 (2018).

101. Gaidon, A., Wang, Q., Cabon, Y., Vig, E.: Virtual Worlds as Proxy for Multi-object Tracking Analysis. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 4340–4349 (2016).

102. Hönig, W., Milanes, C., Scaria, L., Phan, T., Bolas, M., Ayanian, N.: Mixed reality for robotics. IEEE Int. Conf. Intell. Robot. Syst. 5382–5387 (2015).

103. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A Large Dataset to Train Convolutional Networks for Disparity, Optical Flow, and Scene Flow Estimation. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4040–4048. IEEE (2016).

104. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A Naturalistic Open Source Movie for Optical Flow Evaluation. Presented at the (2012).

105. Carrillo, J., Davis, J., Osorio, J., Goodin, C., Durst, J.: High-fidelity physics-based modeling and simulation for training and testing convolutional neural networks for UGV systems. Model. Simul. Auton. Syst. MESAS 2019.

106. Prakash, A., Boochoon, S., Brophy, M., Acuna, D., Cameracci, E., State, G., Shapira, O., Birchfield, S.: Structured Domain Randomization: Bridging the Reality Gap by Context-Aware Synthetic Data. (2018).

107. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the Inception Architecture for Computer Vision. (2015).

108. Ros, G., Sellart, L., Materzynska, J., Vazquez, D., Lopez, A.M.: The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3234–3243. IEEE (2016).

109. Shah, S., Dey, D., Lovett, C., Kapoor, A.: AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles. (2017).

110. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An Open Urban Driving Simulator. (2017).

111. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets Robotics : The KITTI Dataset. Int. J. Robot. Res. (2013).

112. Ros, G., Stent, S., Alcantarilla, P.F., Watanabe, T.: Training Constrained Deconvolutional Networks for Road Scene Semantic Segmentation. (2016).

113. Kar, A., Prakash, A., Liu, M.Y., Cameracci, E., Yuan, J., Rusiniak, M., Acuna, D., Torralba, A., Fidler, S.: Meta-sim: Learning to generate synthetic datasets. Proc. IEEE Int. Conf. Comput. Vis. 4550–4559 (2019).

114. Xu, Y., Yu, G., Wang, Y., Wu, X., Ma, Y.: Car Detection from Low-Altitude UAV Imagery with the Faster R-CNN. J. Adv. Transp. 2017, 1–10 (2017).

115. Tang, T., Deng, Z., Zhou, S., Lei, L., Zou, H.: Fast vehicle detection in UAV images. RSIP 2017 - Int. Work. Remote Sens. with Intell. Process. Proc. 1–5 (2017).

116. Li, W., Li, H., Wu, Q., Chen, X., Ngan, K.N.: Simultaneously Detecting and Counting Dense Vehicles from Drone Images. IEEE Trans. Ind. Electron. (2019).

117. Lu, J., Ma, C., Li, L., Xing, X., Zhang, Y., Wang, Z., Xu, J.: A Vehicle Detection Method for Aerial Image Based on YOLO. J. Comput. Commun. 06, 98–107 (2018).

118. Lechgar, H., Bekkar, H., Rhinane, H.: Detection of cities vehicle fleet using YOLO V2 and aerial images. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch. 42, 121–126 (2019).

119. Razakarivony, S., Jurie, F.: Vehicle Detection in Aerial Imagery : A small target detection benchmark. J. Vis. Commun. Image Represent. (2015).

120. Tanner, F., Colder, B., Pullen, C., Heagy, D., Eppolito, M., Carlan, V., Oertel, C., Sallee, P.: Overhead imagery research data set - An annotated data library & tools to aid in the development of computer vision algorithms. Proc. - Appl. Imag. Pattern Recognit. Work. 1–8 (2009).

121. Mundhenk, N.T., Konjevod, G., Sakla, W.A., Boakye, K.: A Large Contextual Dataset for Classification, Detection and Counting of Cars with Deep Learning. Comput. Vis. – ECCV 2016 14th Eur. Conf. Amsterdam. 9905, 785–800 (2016).

122. Xia, G.S., Bai, X., Ding, J., Zhu, Z., Belongie, S., Luo, J., Datcu, M., Pelillo, M., Zhang, L.: DOTA: A Large-Scale Dataset for Object Detection in Aerial Images. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 3974–3983 (2018).

123. Liu, K., Mattyus, G.: Fast Multiclass Vehicle Detection on Aerial Images. IEEE Geosci. Remote Sens. Lett. 12, 1938–1942 (2015).

124. Azimi, S.M., Bahmanyar, R., Henry, C., Kurz, F.: EAGLE: Large-scale Vehicle Detection Dataset in Real-World Scenarios using Aerial Imagery. (2020).

125. Robicquet, A., Sadeghian, A., Alahi, A., Savarese, S.: Learning Social Etiquette : Human Trajectory. Eur. Conf. Comput. Vis. 549–565 (2016).

126. Bozcan, I., Kayacan, E.: AU-AIR: A Multi-modal Unmanned Aerial Vehicle Dataset for Low Altitude Traffic Surveillance. (2020).

127. Lyu, S., Chang, M.C., Du, D., Li, W., Wei, Y., Coco, M. Del, Carcagni, P., Schumann, A., Munjal, B., Dang, D.Q.T., Choi, D.H., Bochinski, E., Galasso, F., Bunyak, F., Seetharaman, G., Baek, J.W., Lee, J.T., Palaniappan, K., Lim, K.T., Moon, K., Kim, K.J., Sommer, L., Brandlmaier, M., Kang, M.S., Jeon, M., Al-Shakarji, N.M., Acatay, O., Kim, P.K., Amin, S., Sikora, T., Dinh, T., Senst, T., Che, V.G.H., Lim, Y.C., Song, Y.M., Chung, Y.S.: UA-DETRAC 2018: Report of AVSS2018 IWT4S Challenge on Advanced Traffic Monitoring. Proc. AVSS 2018 - 2018 15th IEEE Int. Conf. Adv. Video Signal-Based Surveill. (2019).

128. Hsieh, M.R., Lin, Y.L., Hsu, W.H.: Drone-Based Object Counting by Spatially Regularized Regional Proposal Network. Proc. IEEE Int. Conf. Comput. Vis. 4165–4173 (2017).

129. Zhu, P., Wen, L., Du, D., Bian, X., Ling, H.: VisDrone-VDT2018: The Vision Meets Drone Video Detection and Tracking Challenge Results. 11206, 1–23 (2018).

130. Wu, Z., Suresh, K., Narayanan, P., Xu, H., Kwon, H., Wang, Z.: Delving into Robust Object Detection from Unmanned Aerial Vehicles: A Deep Nuisance Disentanglement Approach. (2019).

131. Presagis - Modelling and Simulation Software, https://www.presagis.com/en/, https://www.presagis.com/en/page/academic-programs/.

132. Unreal Engine, https://www.unrealengine.com/en-US/.

133. Unity Echtzeit-Entwicklungsplattform, https://unity.com/de.

134. CryEngine, https://www.cryengine.com/.

135. Bohemia Interactive Simulations: Virtual Battelspace 3, https://bisimulations.com/products/vbs3.

136. Cornette, W.M.: MOSART: Modeling the Radiative Environment of Earth's Atmosphere, Terrain, Oceans, and Space. J. Washingt. Acad. Sci. 98, 27–46 (2012).

137. SpeedTree - 3D Vegetation Modelling and Middleware, https://store.speedtree.com/.

138. Choi, J., Yang, Y.: Vehicle detection from aerial images using local shape information. Adv. Image Video Technol. 5414, 227–236 (2009).

139. Hinz, S., Stilla, U.: Car detection in aerial thermal images by local and global evidence accumulation. Pattern Recognit. Lett. 27, 308–315 (2006).

140. Niu, X.: A semi-automatic framework for highway extraction and vehicle detection based on a geometric deformable model. ISPRS J. Photogramm. Remote Sens. 61, 170–186 (2006).

141. Xu, Y., Yu, G., Wang, Y., Wu, X., Ma, Y.: A hybrid vehicle detection method based on viola-jones and HOG + SVM from UAV images. Sensors (Switzerland). 16, (2016).

142. Moranduzzo, T., Melgani, F.: Automatic car counting method for unmanned aerial vehicle images. IEEE Trans. Geosci. Remote Sens. 52, 1635–1647 (2014).

143. Moranduzzo, T., Melgani, F.: Detecting cars in UAV images with a catalog-based approach. IEEE Trans. Geosci. Remote Sens. 52, 6356–6367 (2014).

144. Sommer, L.W., Schuchert, T., Beyerer, J.: Fast deep vehicle detection in aerial images. Proc. - 2017 IEEE Winter Conf. Appl. Comput. Vision, WACV 2017. 311–319 (2017).

145. Youssef, Y., Elshenawy, M.: Automatic vehicle counting and tracking in aerial video feeds using cascade region-based convolutional neural networks and feature pyramid networks. Transp. Res. Rec. 2675, 304–317 (2021).

146. Schweitzer, D., Agrawal, R.: Multi-Class Object Detection from Aerial Images Using Mask R-CNN. Proc. - 2018 IEEE Int. Conf. Big Data, Big Data 2018. 3470–3477 (2019).

147. Wu, Y., Abdel-Aty, M., Zheng, O., Cai, Q., Zhang, S.: Automated Safety Diagnosis Based on Unmanned Aerial Vehicle Video and Deep Learning Algorithm. Transp. Res. Rec. 2674, 350–359 (2020).

148. Cai, Z., Vasconcelos, N.: Cascade R-CNN: High quality object detection and instance segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 43, 1483–1498 (2021).

149. Westlake, N., Cai, H., Hall, P.: Detecting People in Artwork with CNNs. Comput. Vis. – ECCV 2016 Work. 9913, 825–841 (2016).

150. Zafar, I. and Tzanidou, G. and Burton, R. and Patel, N. and Araujo, L.: Hands-On Convolutional Neural Networks with TensorFlow: Solve computer vision problems with modeling in TensorFlow and Python. Packt Publishing (2018).

151. Kathuria, A.: What´s new in YOLOv3? TowardsDataScience. (2018) . https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b.

152. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. Int. J. Comput. Vis. 115, 211–252 (2015).

153. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. 88, 303–338 (2010).

154. Russakovsky, O., Li, L.-J., Fei-Fei, L.: Best of both worlds: Human-machine collaboration for object annotation. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2121–2131. IEEE (2015).

155. Zitnick, C.L., Dollár, P.: Edge Boxes: Locating Object Proposals from Edges. Presented at the (2014).

156. Eidenberger, H.: Statistical analysis of content-based MPEG-7 descriptors for image retrieval. Multimed. Syst. 10, 84–97 (2004).

157. Messing, D.S.., van Beek, P., Errico, J.H.: The MPEG-7 Color Structure Descriptor: Image Description Using Color and Local Spatial Information. IEEE Int. Conf. Image Process. (2001).

158. Oelbaum, T.: Design and Verification of Video Quality Metrics, (2008).

159. Manjunath, B.S., Salembier, P., Sikora, T.: Introduction to MPEG-7: Multimedia Content Description Interface. Wiley (2002).

160. Sikora, T.: The MPEG-7 visual standard for content description - An overview. IEEE Trans. Circuits Syst. Video Technol. 11, 696–702 (2001).

161. ISO/IEC JTC1/SC29/WG11N6828: MPEG-7 Overview (version 10), (2004).

162. Cao, G., Huang, L., Tian, H., Huang, X., Wang, Y., Zhi, R.: Contrast enhancement of brightness-distorted images by improved adaptive gamma correction. Comput. Electr. Eng. 66, 569–582 (2018) . https://github.com/leowang7/iagcwd.

163. Ke, Y., Tang, X., Jing, F.: The design of high-level features for photo quality assessment. Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 1, 419–426 (2006) . https://github.com/szakrewsky/quality-feature-extraction.

164. Wang, J., Allebach, J.: Automatic Assessment of Online Fashion Shopping Photo Aesthetic Quality. Int. Conf. Image Process. 2915–2919 (2015) . https://github.com/szakrewsky/quality-feature-extraction.

165. Hasler, D., Sabine, S.: Measuring colourfulness in natural images. Proc. SPIE - Int. Soc. Opt. Eng. 87–95 (2003).

166. Li, F., Wu, J., Wang, Y., Zhao, Y., Zhang, X.: A color cast detection algorithm of robust performance. 2012 IEEE 5th Int. Conf. Adv. Comput. Intell. ICACI 2012. 662–664 (2012) . https://github.com/hwp9527/color_cast.

167. Robertson, A.R.: Computation of Correlated Color Temperature and Distribution Temperature. J. Opt. Soc. Am. 58, 1528 (1968) . https://www.colour-science.org/.

168. Hernández-Andrés, J., Lee, R.L., Romero, J.: Calculating correlated color temperatures across the entire gamut of daylight and skylight chromaticities. Appl. Opt. 38, 5703 (1999) . https://www.colour-science.org/.

169. Talebi, H., Milanfar, P.: NIMA: Neural Image Assessment. (2017) . https://github.com/idealo/image-quality-assessment.

170. Mittal, A., Moorthy, A.K., Bovik, A.C.: No-Reference Image Quality Assessment in the Spatial Domain. IEEE Trans. Image Process. 21, 4695–4708 (2012) . https://github.com/bukalapak/pybrisque.

171. Mai, L., Le, H., Niu, Y., Liu, F.: Rule of thirds detection from photograph. Proc. - 2011 IEEE Int. Multimedia, ISM 2011. 91–96 (2011) . https://github.com/szakrewsky/quality-feature-extraction.

172. Kumar, J., Chen, F., Doermann, D.: Sharpness estimation for document and scene images. Proc. - Int. Conf. Pattern Recognit. 3292–3295 (2012) . https://github.com/umang-singhal/pydom.

173. Narvekar, N.D., Karam, L.J.: A No-Reference Image Blur Metric Based on the Cumulative Probability of Blur Detection (CPBD). IEEE Trans. Image Process. 20, 2678–2683 (2011) . https://github.com/0x64746b/python-cpbd.

174. Su, B., Lu, S., Tan, C.L.: Blurred image region detection and classification. In: Proceedings of the 19th ACM international conference on Multimedia - MM ’11. p. 1397. ACM Press, New York, New York, USA (2011).

175. Hanghang, T., Mingjing, L., Hongjiang, Z., Changshui, Z.: Blur detection for digital images using wavelet transform. 2004 IEEE Int. Conf. Multimed. Expo. . https://github.com/szakrewsky/quality-feature-extraction.

176. Rakhshanfar, M., Amer, M.A.: Estimation of Gaussian, Poissonian-Gaussian, and Processed

Visual Noise and its Level Function. IEEE Trans. Image Process. 1–1 (2016) . https://github.com/meisamrf/ivhc-estimator.

177. Chen, G., Zhu, F., Heng, P.A.: An Efficient Statistical Method for Image Noise Level Estimation. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 477–485. IEEE (2015).

178. Ming-Kuei Hu: Visual pattern recognition by moment invariants. IEEE Trans. Inf. Theory. 8, 179–187 (1962).

179. DeepLabv3 ResNet101, https://pytorch.org/hub/pytorch_vision_deeplabv3_resnet101/, last accessed 2020/07/30.

180. Zhu, L., Deng, Z., Hu, X., Fu, C.W., Xu, X., Qin, J., Heng, P.A.: Bidirectional feature pyramid network with recurrent attention residual modules for shadow detection. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 122–137 (2018) . https://github.com/zijundeng/BDRAR.

181. CNN Weather Classification Models, https://github.com/666-zhf/weather-predicition, https://github.com/imaaditya-stack/Weather-image-classification, https://github.com/NgoJunHaoJason/weather-classification, https://github.com/berkgulay/WeatherPredictionFromImage, last accessed 2020/07/30.

182. Haralick, R.M., Shanmugam, K., Dinstein, I.: Textural Features for Image Classification. IEEE Trans. Syst. Man. Cybern. SMC-3, 610–621 (1973).

183. Royo, C.V.: Image-Based Query by Example Using MPEG-7 Visual Descriptors, (2010).

184. Spyrou, E., Tolias, G., Mylonas, P., Avrithis, Y.: Concept detection and keyframe extraction using a visual thesaurus. Multimed. Tools Appl. 41, 337–373 (2009).

185. Rayar, F.: ImageNet MPEG-7 Visual Descriptors - Technical Report. 21–23 (2017).

186. Manjunath, B.S., Ohm, J.R., Vasudevan, V. V., Yamada, A.: Color and texture descriptors. IEEE Trans. Circuits Syst. Video Technol. 11, 703–715 (2001).

187. Won, C.S., Park, D.K., Park, S.J.: Efficient use of MPEG-7 edge histogram descriptor. ETRI J. 24, 23–30 (2002).

188. Krump, M., Stütz, P.: UAV Based Vehicle Detection with Synthetic Training: Identification of Performance Factors Using Image Descriptors and Machine Learning. In: Modelling and Simulation for Autonomous Systems. MESAS 2020. Lecture Notes in Computer Science. pp. 62–85 (2021).

189. Spiessl, W.: Farb- und Textur-Extraktion und -Deskription nach dem MPEG-7-Standard, (2006).

190. Group, M.P.E.: Information Technology - Multimedia Content Description Interface - Part 6: Reference Software. In: ISO/IEC JTC 1 SC29 (Vol. 15938–6:20) (2003).

191. Bastan, M., Cam, H., Gudukbay, U., Ulusoy, O.: Bilvideo-7: an MPEG-7- compatible video indexing and retrieval system. IEEE Multimed. 17, 62–73 (2010) . https://github.com/mubastan/mpeg7fex.

192. Kasutani, E., Yamada, A.: The MPEG7 Color Layout Descriptor: A Compact Image Feature Description for High-Speed Image/Video Segment Retrieval. IEEE Int. Conf. Image Process. (2001).

193. Tamura, H., Mori, S., Yamawaki, T.: Textural Features Corresponding to Visual Perception. IEEE Trans. Syst. Man. Cybern. 8, 460–473 (1978).

194. Ro, Y.M., Kim, M., Kang, H.K., Manjunath, B.S., Kim, J.: MPEG-7 homogeneous texture descriptor. ETRI J. 23, 41–51 (2001).

195. Bober, M.: MPEG-7 visual shape descriptors. IEEE Trans. Circuits Syst. Video Technol. 11, 716–719 (2001).

196. Salembier, P.: Introduction to MPEG 7: Multimedia Content Description Language. Wiley

(2002).

197. Tang, X., Luo, W., Wang, X.: Content-based photo quality assessment. IEEE Trans. Multimed. 15, 1930–1943 (2013).

198. Finley, D.R.: HSP Color Model — Alternative to HSV (HSB) and HSL, http://alienryderflex.com/hsp.html, last accessed 2020/07/29.

199. Otsu, N.: A Threshold Selection Method from Gray-Level Histograms. IEEE Trans. Syst. Man. Cybern. 9, 62–66 (1979).

200. Chen, L.-C., Papandreou, G., Schroff, F., Adam, H.: Rethinking Atrous Convolution for Semantic Image Segmentation. (2017).

201. Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P.: Microsoft COCO: Common Objects in Context. Comput. Vis. Pattern Recognit. (2014).

202. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal Visual Object Classes Challenge: A Retrospective. Int. J. Comput. Vis. 111, 98–136 (2015).

203. MIT Driving Scene Segmentation. . https://colab.research.google.com/github/lexfridman/mit-deep-learning/blob/master/tutorial_driving_scene_segmentation/tutorial_driving_scene_segmentation.ipynb.

204. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes Dataset for Semantic Urban Scene Understanding. (2016).

205. Basha, S.M., Rajput, D.S.: Survey on Evaluating the Performance of Machine Learning Algorithms: Past Contributions and Future Roadmap. In: Deep Learning and Parallel Computing Environment for Bioengineering Systems. pp. 153–164. Elsevier (2019).

206. Backhaus, K., Erichson, B., Plinke, W., Weiber, R.: Multivariate Analysemethoden. Springer Berlin Heidelberg, Berlin, Heidelberg (2016).

207. Rönz, B., Förster, E.: Regressions- und Korrelationsanalyse. Gabler Verlag (1992).

208. Scikit-Learn Dokumentation - 1.10 Decision Trees, https://scikit-learn.org/stable/modules/tree.html.

209. Brodersen, K.H., Ong, C.S., Stephan, K.E., Buhmann, J.M.: The Balanced Accuracy and Its Posterior Distribution. In: 2010 20th International Conference on Pattern Recognition. pp. 3121–3124. IEEE (2010).

210. Chawla, N. V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic Minority Over-sampling Technique. J. Artif. Intell. Res. 16, 321–357 (2002).

211. Wooldridge, J.M.: Introduction Econometrics: A Modern Approach. (2005).

212. Kessler, W.: Multivariate Datenanalyse für die Pharma-, Bio- und Prozessanalytik. (206)AD.

213. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification And Regression Trees. Routledge (2017).

214. Jovic, A., Brkic, K., Bogunovic, N.: A review of feature selection methods with applications. In: 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). pp. 1200–1205. IEEE (2015).

215. Chandrashekar, G., Sahin, F.: A survey on feature selection methods. Comput. Electr. Eng. 40, 16–28 (2014).

216. Shroff, K.P., Maheta, H.H.: A comparative study of various feature selection techniques in high-dimensional data set to improve classification accuracy. 2015 Int. Conf. Comput. Commun. Informatics, ICCCI 2015. (2015).

217. Hanchuan Peng, Fuhui Long, Ding, C.: Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans. Pattern Anal. Mach. Intell. 27, 1226–1238 (2005).

218. Yu, L., Liu, H.: Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution. Proceedings, Twent. Int. Conf. Mach. Learn. 2, 856–863 (2003).

219. Saarela, M., Jauhiainen, S.: Comparison of feature importance measures as explanations for classification models. SN Appl. Sci. 3, 272 (2021).

220. Ribeiro, M.T., Singh, S., Guestrin, C.: "Why Should I Trust You?" In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1135–1144. ACM, New York, NY, USA (2016).

221. Lundberg, S., Lee, S.-I.: A Unified Approach to Interpreting Model Predictions. (2017) . https://github.com/slundberg/shap.

222. Lundberg, S.M., Erion, G.G., Lee, S.-I.: Consistent Individualized Feature Attribution for Tree Ensembles. (2018).

223. Strobl, C., Boulesteix, A.-L., Zeileis, A., Hothorn, T.: Bias in random forest variable importance measures: Illustrations, sources and a solution. BMC Bioinformatics. 8, 25 (2007).

224. Shapley, L.S.: A Value for n-Person Games. In: Contributions to the Theory of Games (AM-28), Volume II. pp. 307–318. Princeton University Press (1953).

225. Geodatenbasis: Bayerische Vermessungsverwaltung (Landesamt für Digitalisierung, Breitband und Vermessung). . http://vermessung.bayern.de/file/pdf/7203/Nutzungsbedingungen_%0AViewing.pdf; https://www.ldbv.bayern.de/.

226. Turbosquid: Zusatzmodelle und Kleinteile, https://www.turbosquid.com/3d-models/construction-2-3d-3ds/810700, https://www.turbosquid.com/3d-models/urban-street-props-pack-max/852225.

227. Turbosquid: Fahrzeugmodelle, https://www.turbosquid.com/3d-models/3d-60-urban-cars-vehicles-model-1192514, https://www.turbosquid.com/3d-models/city-vehicles-3d-model-1500161, https://www.turbosquid.com/3d-models/city-vehicles-3d-model-1500292, https://www.turbosquid.com/3d-models/ci.

228. DJI M210 RTK V2, Zenmuse XT2, Multikoptersetup, https://droneparts.de/dji-matrice-210-v2-rtk-d-rtk-2-mobile-station-combo-drohne-fuer-vermessung, https://www.dji.com/de/matrice-200-series-v2.

229. Krump, M., Stütz, P.: UAV Based Vehicle Detection on Real and Synthetic Image Pairs: Performance Differences and Influence Analysis of Context and Simulation Parameters. In: Modelling and Simulation for Autonomous Systems. MESAS 2021. Lecture Notes in Computer Science. pp. 3–25 (2022).

230. Skalski, P.: MAKE SENSE Labeling Tool, https://www.makesense.ai/, https://github.com/SkalskiP/make-sense.

231. Sundog: Silver Lining Sky: Dynamische Wolkensimulation, https://sundog-soft.com/.

232. Siess, A.: RGB to color temperature, https://andi-siess.de/rgb-to-color-temperature/.

233. Ravikumar, R.: Bokehlicious Selfies, https://rahulrav.com/blog/bokehlicious.html.

234. Fan, Z.: Adjust Local Brightness for Image Augmentation. Medium. . https://medium.com/@fanzongshaoxing/adjust-local-brightness-for-image-augmentation-8111c001059b.

235. Qiu, S., Liu, Q., Zhou, S., Wu, C.: Review of Artificial Intelligence Adversarial Attack and Defense Technologies. Appl. Sci. 9, 909 (2019).

236. Vargas, D.V., Su, J.: Understanding the One-Pixel Attack: Propagation Maps and Locality

Analysis. (2019).

237. Jackson, P.T., Atapour-Abarghouei, A., Bonner, S., Breckon, T., Obara, B.: Style Augmentation: Data Augmentation via Style Randomization. (2018) . https://github.com/philipjackson/style-augmentation.

238. Bochkovskiy, A.: Yolo v4, v3 and v2 for Windows and Linux, https://github.com/AlexeyAB/darknet.

239. Hummel, G., Stütz, P.: Evaluation of Synthetically Generated Airborne Image Datasets using Feature Detectors as Performance Metric. IPCV 2015. 231–237 (2015).

Danksagung

Im Folgenden möchte ich die Gelegenheit nutzen, mich bei allen zu bedanken, die mich während meiner Tätigkeit als wissenschaftlicher Mitarbeiter unterstützt haben und die in direkter oder indirekter Weise zum Gelingen dieser Arbeit beigetragen haben.

Mein Dank gilt dabei besonders Herrn Prof. Dr. Peter Stütz, der es mir ermöglicht hat, diese Arbeit am Institut für Flugsysteme zu schreiben. Als Doktorvater hat er die Arbeit stets unterstützt und durch zahlreiche fachliche und organisatorische Ratschläge begleitet. Ich möchte mich dabei auch recht herzlich für das zu jedem Zeitpunkt entgegengebrachte Vertrauen bedanken, das vor allem während der Corona Pandemie für mich sehr wichtig war.

Bedanken möchte ich mich außerdem bei dem Vorsitzenden des Prüfungsausschusses Herrn Prof. Dr. Michael Schmitt und vor allem auch bei dem zweiten Berichterstatter Herrn Prof. Dr. Dr. Wolfram Hardt für das Interesse an dieser Arbeit und die Übernahme des Koreferates.

Ein besonderer Dank gilt auch meinem Kollegen Martin Ruß, der mir zu Beginn des Projektes sehr geholfen hat. Mit zahlreichen Ideen und vor allem mit seinem tatkräftigen Einsatz bei der Inbetriebnahme der Simulationsumgebung und der Beschaffung des Kartenmaterials hat er mich sehr unterstützt.

Zudem habe ich mich über das allzeit gute Verhältnis und Miteinander unter den Kollegen sehr gefreut. Darüber hinaus möchte ich mich insbesondere auch bei Armin Tichacek bedanken, der mir bei der Modellierung geholfen hat. Mein Dank gilt auch der Werkstatt unter Melanie Finze und Jessica Lach für die Weiterführung der Modellierung und vor allem das teils langwierige Labeln der Bilddaten. Ich danke Alexander Schelle für die sehr strukturierte Einführung in den Flugbetrieb am Institut und insbesondere danke ich auch Jessica Lach für die tatkräftige Mithilfe bei meinen Flügen und die Bereitstellung ihrer beiden Autos als Testfahrzeuge.

Mein herzlicher Dank gilt auch meinen Eltern und meiner Schwester Christina, die das Ganze all die Jahre hinweg unterstützt haben und mir in den stressigen Phasen zur Seite standen.