

Transfer Learning mit synthetischen Daten für Bildverarbeitungssysteme in der industriellen Kommissionierung

Zur Erlangung des akademischen Grades eines
DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)

von der KIT-Fakultät für Maschinenbau des
Karlsruher Instituts für Technologie (KIT)

angenommene

DISSERTATION

von

Moritz Weisenböhler, M.Sc.

Tag der mündlichen Prüfung:
Hauptreferent:
Korreferent:

27. März 2026
Prof. Dr.-Ing. Kai Furmans
Prof. Dr.-Ing. Christian Wurll



Dieses Werk ist lizenziert unter einer Creative Commons
Namensnennung 4.0 International Lizenz (CC BY 4.0):
<https://creativecommons.org/licenses/by/4.0/deed.de>

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Robotik und Autonome Systeme (IRAS) der Hochschule Karlsruhe (HKA) und als Doktorand am Institut für Fertigungstechnik und Logistiksysteme (IFL) des Karlsruher Instituts für Technologie (KIT).

Mein besonderer Dank gilt meinem Doktorvater Kai Furmans, unter dessen Betreuung und mit dessen Unterstützung ich am IFL forschen durfte. Seine Anleitung – insbesondere mit Blick auf wissenschaftliche Akribie sowie die Formulierung und Überprüfung statistischer Hypothesen – hat Erkenntnisse ermöglicht, die ansonsten nicht möglich gewesen wären, und meine wissenschaftliche Arbeit auch in Zukunft positiv prägen wird.

Ebenso möchte ich mich bei Christian Wurr und Björn Hein bedanken, die meine Promotion sowohl wissenschaftlich als auch weit darüber hinaus als Korreferent und Mentor begleitet und unterstützt haben. In den vergangenen Jahren durfte ich sie auf dem spannenden Weg der Gründung des IRAS begleiten und unterstützen.

Mein aufrichtiger Dank gilt meinen Kolleginnen und Kollegen an der HKA – sowohl im früheren K004b als auch am LTC – und am KIT mit denen ich in den vergangenen Jahren forschen und arbeiten durfte. Ich hoffe, dass wir noch so manchen Weg gemeinsam gehen werden.

Ich möchte meiner Familie und meinen Freunden außerhalb der Arbeit herzlich für ihre Unterstützung, Ermutigung und Geduld danken. Es bedeutet mir viel, euch in meinem Leben zu haben.

Mein tiefster Dank gilt meiner Frau Leonie und meinem Sohn Emil. Ihr seid die wichtigsten Menschen in meinem Leben und ohne euch hätte ich diesen Weg niemals gehen können und wollen.

Karlsruhe, November 2025

Moritz Weisenböhler

Kurzfassung

Mit der Automatisierung industrieller Produktions- und Logistikprozesse steigt auch der Bedarf an zugehörigen Bildverarbeitungssystemen, welche zunehmend auf Verfahren des maschinellen Lernens basieren. Aufgrund des hohen Datenbedarfs und des großen Aufwands der manuellen Datenerzeugung wird die Nutzung mittels Simulationssoftware erzeugter synthetischer Bilddaten immer relevanter. Jedoch führen Unterschiede zwischen der synthetischen und der realen Domäne, die sogenannte Domänenlücke (engl. domain gap), zu Performanzeinbußen der synthetisch trainierten neuronalen Netze (NNs) in der realen Anwendung.

Im Rahmen der vorliegenden Arbeit werden verschiedene Methoden zur Verbesserung von synthetischen Bilddaten mit dem Ziel untersucht, die Domänenlücke so weit wie möglich zu verkleinern – das Scene Engineering, die Domain Randomization und die Domain Adaptation sowie deren Kombination. Hierbei liegt der Fokus auf industriellen Anwendungsfällen, wobei für die Untersuchung primär eine exemplarische Item Picking-Aufgabe genutzt wird. Übergeordnet steht die Frage, ob es mit Blick auf die Performanz der neuronalen Netze möglich ist, mit synthetischen Trainingsdaten dasselbe Niveau wie beim Training mit Realdaten zu erreichen.

Für die untersuchten Anwendungsfälle wird die signifikante Überlegenheit des Scene Engineerings gegenüber kontextarmen Datengenerierungsmethoden belegt. Dabei werden insbesondere die realistischen Nachbildungen der Szene in Bezug auf den Ladungsträger und das Sichtfeld der Kamera als entscheidende Faktoren identifiziert.

Es wird die signifikante Verbesserung der synthetischen Bilddaten durch nachträgliche Domain Randomization über Augmentationstechniken nachgewiesen. Hierbei wird die Kombination aus starker Kontrast-, Saturations- und Hellwertvariation als beste Augmentationsmethode identifiziert.

Zwar wird auch die signifikante Steigerung der Performanz durch die Domain Adaptation mit bildstatistischen und generativen Verfahren belegt, jedoch kann bei der Kombination mit Domain Randomization keine generelle Verbesserung über diese hinaus nachgewiesen werden.

Durch die Übertragung auf weitere Datensätze, Metriken, NN-Architekturen und Anwendungsfälle wird die forschungsleitende Fragestellung abschließend bestätigt. Es wird belegt, dass mit synthetischen Trainingsdaten, in Abhängigkeit vom Evaluierungsdatensatz und den genutzten Metriken, Performanzkennwerte erreicht werden können, welche nicht signifikant niedriger und teilweise sogar höher sind als beim Training mit realen Bilddaten.

Abstract

With the automation of industrial production and logistics processes, the demand for image processing systems, which are increasingly based on machine learning methods, is also rising. Due to the large amount of data required and the substantial effort involved in manually generating data, the use of synthetic image data generated through simulation software is gaining relevance. However, discrepancies between the synthetic and real domains, known as the domain gap, lead to performance losses in synthetically trained neural networks (NNs) when applied to real-world scenarios.

This work examines various methods for enhancing synthetic image data, including scene engineering, domain randomization, and domain adaptation, along with the combinations of these methods, to minimize the domain gap. The focus is on industrial use cases, with a primary examination of an item picking task. The central research question is whether it is possible to achieve the same level of neural network performance with synthetic training data as with real data.

For the examined use cases, the significant superiority of scene engineering over low-context data generation methods is demonstrated. Realistic reproduction of the scene, particularly with respect to the bin and the camera's field of view, is identified as a crucial factor.

A significant improvement of the synthetic image data is achieved through subsequent domain randomization using augmentation techniques. Strong variations in contrast, saturation, and brightness are identified as the most effective augmentation methods.

Although domain adaptation using image statistical and generative methods significantly improves performance, combining these with domain randomization does not yield additional general improvements.

By applying the results to other datasets, metrics, NN architectures, and use cases, the research question is solved. It is shown that, depending on the evaluation dataset and metrics used, performance with synthetic training data can be comparable to, or in some cases even surpass, that achieved with real image data.

Inhaltsverzeichnis

1. Einleitung	1
1.1. Motivation	2
1.2. Problembeschreibung und Forschungsfragen	4
1.3. Aufbau der Arbeit	6
1.4. Konventionen	7
2. Hintergrund	9
2.1. Automatisierung in der Industrie am Beispiel von Item Picking	9
2.1.1. Item Picking Automobilindustrie (IP-A)	12
2.1.2. Item Picking Bekleidungsindustrie (IP-B)	14
2.1.3. Qualitätsprüfung in der Automobilindustrie (QP-A)	15
2.2. Synthetische Bilddaten	16
2.3. Stand der Forschung	18
2.3.1. Von Realbild-Rekomposition bis Full-Stack-Simulation	19
2.3.2. Domain Randomization	20
2.3.3. Domain Adaptation	24
2.3.4. Einordnung der vorliegenden Arbeit	27
3. Theoretische Grundlagen	29
3.1. Bildverarbeitung	29
3.1.1. Farbräume	29
3.1.2. Faltung	30
3.2. Deep Learning	31
3.2.1. Neuronale Netze	32
3.2.2. Convolutional Neural Network	34
3.2.3. Meta-Architekturen für Objekterkennung	36
3.2.4. Generative Adversarial Network	38
3.3. Bayes'sche Optimierung	42
3.3.1. Surrogatmodell und Gauß-Prozesse	43
3.3.2. Akquisitionsfunktion	44
3.4. Hypothesentests und Konfidenzintervalle	45
3.4.1. Hypothesentests	46

3.4.2.	Welch-Test	47
3.4.3.	Shapiro-Wilk-Test	48
3.4.4.	Konfidenzintervalle	49
3.5.	Bewertungsverfahren und Metriken	50
3.5.1.	Jaccard-Koeffizient (IoU)	50
3.5.2.	Genauigkeit, Präzision und Sensitivität	50
3.5.3.	Durchschnittspräzision (AP)	53
4.	Methoden	55
4.1.	Datengenerierung und Szenenerstellung	55
4.1.1.	SynthDet	56
4.1.2.	Scene Engineering	56
4.1.3.	Scene Adaptation	58
4.1.4.	Datengenerierung	61
4.1.5.	Datensatzanalyse	62
4.2.	Domain Randomization	63
4.2.1.	Zufallszahlen	65
4.2.2.	Farbe (RGB)	66
4.2.3.	Helligkeit	66
4.2.4.	Farbwert, Sättigung und Hellwert (HSV)	67
4.2.5.	Chromatische Aberration	67
4.2.6.	Kontrast	68
4.2.7.	Unschärfe	69
4.2.8.	Rauschen	70
4.3.	Domain Adaptation	71
4.3.1.	Bildstatistisch (ICT)	71
4.3.2.	Generativ (CycleGAN)	72
4.4.	Training und Evaluierung	73
4.4.1.	Trainingsprozess	73
4.4.2.	COCO Challenge Metriken	75
4.4.3.	Limited Selection-Metriken	77
5.	Experimente und Ergebnisse	79
5.1.	Voruntersuchungen	80
5.1.1.	Verteilung	81
5.1.2.	Stichprobenumfang	82
5.1.3.	Schwankungsbreite	83
5.2.	Datengenerierung und Szenenerstellung	85
5.2.1.	SynthDet und Scene Engineering	85

5.2.2.	Szenenstruktur	88
5.2.3.	Sichtfeld	93
5.2.4.	Beleuchtung	96
5.2.5.	Objektpositionierung	98
5.2.6.	Objektanzahl	101
5.2.7.	Scene Adaptation	103
5.3.	Domain Randomization	107
5.3.1.	Unschärfe und Rauschen	107
5.3.2.	Farbe und Helligkeit	111
5.3.3.	Chromatische Aberration	116
5.3.4.	Kontrast	117
5.3.5.	Kombination Augmentationen	119
5.4.	Domain Adaptation	122
5.4.1.	ICT	122
5.4.2.	GAN	127
5.4.3.	Kombination GAN und ICT	130
5.5.	Kombination Domain Adaptation und Domain Randomization	132
5.5.1.	SV und Kontrast	133
5.5.2.	Unschärfe und Rauschen	135
5.5.3.	Übertragung	137
6.	Zusammenfassung	147
6.1.	Zusammenfassung der Arbeit	147
6.2.	Ausblick	149
Literatur		151
Eigene Veröffentlichungen		167
Abbildungsverzeichnis		169
Tabellenverzeichnis		178
Abkürzungsverzeichnis		179
A. Anhang: Alternative Berechnungsmethoden		181
A.1.	HSV	181
A.2.	Kontrast	183
B. Anhang: Ergänzende Beispielbilder		185

C. Anhang: Ergänzende Abbildungen und Tabellen	191
C.1. Datengenerierung	191
C.2. Domain Randomization	201

1. Einleitung

Die Automatisierung industrieller Produktions- und Logistikprozesse ist und bleibt ein zentraler Treiber globaler Wertschöpfungsketten [91]. Da in modernen Automatisierungslösungen immer häufiger auch bildgebende Sensortechnik zum Einsatz kommt, wächst der Bedarf an zugehörigen Bildverarbeitungssystemen. Analysten erwarten in den kommenden Jahren eine Verdopplung des globalen Marktes für industrielle Bildverarbeitung („Machine Vision“). Je nach berücksichtigten Segmenten wird ein Wachstum des aktuellen Marktvolumens von rund 10 bis 20 Mrd. US-Dollar auf 20 bis 42 Mrd. US-Dollar bis 2030 bzw. 2035 prognostiziert. Dies entspricht jährlichen Wachstumsraten zwischen 7 und 13 % [50, 51, 92]. Zunehmende Bedeutung gewinnen dabei Bildverarbeitungssysteme auf Basis von maschinellem Lernen – für den zugehörigen globalen Markt wird ein jährliches Wachstum von rund 24 % bis 2030 bzw. 2033 erwartet [90, 70].

Im Kontext der industriellen Automatisierung kommen Bildverarbeitungssysteme unter anderem in den Anwendungsfeldern der Robotik (z. B. automatisierte Kommissionierung), Artikelverfolgung und Qualitätsprüfung zum Einsatz. Mithilfe dieser Systeme werden beispielsweise ganze Artikelinstanzen oder einzelne Artikelmerkmale erkannt und im Bild lokalisiert. Bildverarbeitung auf Basis von maschinellem Lernen ist eine vielversprechende und zunehmend eingesetzte Möglichkeit, die Systeme leistungsfähiger, flexibler und kostengünstiger zu gestalten und so neue Potenziale für den Einsatz in der Industrie zu schaffen. Gleichzeitig gehen mit dem maschinellen Lernen neue Herausforderungen einher, welche überwunden werden müssen – die vorliegende Arbeit versucht, einen Beitrag hierzu zu leisten.

Im Folgenden werden die Motivation für diese Arbeit aufgezeigt, die Problemstellung beschrieben und mit konkreten Forschungsfragen verknüpft sowie der Aufbau der vorliegenden Arbeit dargestellt.

1.1. Motivation

Neuronale Netze (NNs) sind heute eine Schlüsseltechnologie der modernen Bildverarbeitung. Als ein Verfahren des maschinellen Lernens werden diese anhand repräsentativer Beispieldaten trainiert, anstatt die Verarbeitungsalgorithmen für jeden Anwendungsfall manuell zu programmieren. Das Training von neuronalen Netzen für die Objekterkennung erfordert jedoch eine große Anzahl annotierter Bilddaten. In einigen Anwendungsfeldern, wie beispielsweise dem autonomen Fahren oder der Detektion von Menschen, Tieren und Alltagsobjekten, existieren bereits große, frei zugängliche Datensammlungen. Zu den bekanntesten zählen unter anderem ImageNet [20, 96], Microsoft COCO [63], LVIS [40] oder KITTI [31]. Im Kontext industrieller Anwendungen kann dagegen häufig nicht auf vorhandene Datensätze zurückgegriffen werden, da unternehmens- oder anwendungsfallspezifische Artikel oder Merkmale erkannt und unterschieden werden sollen. Darüber hinaus zeigen Untersuchungen, dass sich die Bilddaten großer öffentlicher Datensätze wie ImageNet systematisch von denen realer Bildverarbeitungsprobleme unterscheiden [27]. Daher müssen reale Trainingsdaten mit hohem Zeit- und Kostenaufwand manuell generiert und mit Klassen, Begrenzungsrahmen oder Masken annotiert werden.

Zur Vereinfachung der Datenerzeugung können synthetisch generierte Daten genutzt werden. Eine simulationsbasierte Bilderzeugung auf Basis von 3D-Modellen ist vergleichsweise kostengünstig, leicht skalier- und anpassbar und liefert exakte Annotationen. Darüber hinaus können synthetische Daten bereits vor der Einführung neuer Artikel oder vor der Inbetriebnahme einer neuen Produktionslinie zur Verfügung gestellt werden. Das Training mit synthetischen Daten stellt einen Ansatz des sogenannten Transferlernens (engl. transfer learning) dar. Allerdings führen Domänenunterschiede zwischen synthetischen und realen Daten, die sogenannte Domänenlücke (engl. domain gap) oder spezifischer die Sim2Real-Lücke (engl. simulation to reality gap), zu Herausforderungen bei der Übertragung der synthetisch trainierten neuronalen Netze in die reale Anwendung. Eine größere Domänenlücke kann beispielsweise zu einer niedrigeren Präzision und Sensitivität bei der Objekterkennung für alle oder für bestimmte Objektklassen führen.

In zahlreichen Forschungsarbeiten wurden verschiedene Methoden zur Überwindung der Domänenlücke entwickelt und untersucht. Ein gängiger Ansatz ist dabei die Feinabstimmung (engl. finetuning) von maschinellen Lernverfahren, die vorab mit Daten einer anwendungsfallfremden, jedoch realen Domäne trainiert wurden. Bereits erlerntes Wissen der Modelle kann

dabei auch in anderen Anwendungsfällen genutzt werden, wodurch die notwendige Trainingszeit reduziert und die erreichten Ergebnisse stark verbessert werden können. Dieser Ansatz kann jedoch auch in entgegengesetzter Richtung angewendet werden, indem ein synthetisch vortrainiertes Modell mit wenigen Realdaten auf die reale Anwendung feinabgestimmt wird. Auch die Anpassung der synthetisch erzeugten Daten an die reale Zieldomäne ist eine gängige Methode. Dies kann zum einen durch eine Optimierung der Simulation vor der Erzeugung der Daten (Scene Engineering) und zum anderen durch Nachbearbeitung und Anpassung bereits generierter Daten (Domain Adaptation) erfolgen, beispielsweise mithilfe von Generative Adversarial Networks (GANs) oder Autoencodern. Ein weiterer Ansatz ist die starke Vergrößerung der synthetischen Domäne durch eine breite Randomisierung diverser Szeneneigenschaften (Domain Randomization), wobei die so erzeugten Simulationsszenen häufig unrealistische Kamera-, Licht-, Textur- und Struktureigenschaften aufweisen. Die Idee der Domain Randomization besteht darin, die neuronalen Netze in einer Domäne zu trainieren, welche die Eigenschaften der realen Zieldomäne umfasst. Dadurch stellt die Realität für die trainierten Verfahren nur eine Ausprägung der synthetischen Domäne dar und der Einfluss der nicht entscheidungsrelevanten Eigenschaften, wie beispielsweise der realen Beleuchtung, wird reduziert.

Während die Domain Adaptation mit GANs meist eine größere Auswahl realer Referenzbilder erfordert, sind für die manuelle Szenenoptimierung nur einzelne Stichproben oder gar nur Domänenwissen durch Erfahrung eines Experten erforderlich. Allerdings ist die manuelle Anpassung der Simulationsszene an die Referenzdaten zeitaufwendig und erfordert Kenntnisse im Bereich der Simulationserstellung. Die Randomisierung hingegen ist sehr einfach anzuwenden und erfordert weder Expertenwissen noch Beispieldaten aus der realen Anwendung. Es ist jedoch fraglich, ob die reine Randomisierung eine Anpassung der Simulation bzw. der generierten Daten in realen Aufgaben vollständig ersetzen kann. Dies gilt insbesondere, da eine umfangreiche Randomisierung in der Regel die Komplexität der Bildverarbeitungsaufgabe erhöht.

In bisherigen Forschungsarbeiten konnte der positive Beitrag einzelner Techniken zur Reduktion der Domänenlücke demonstriert werden. Dagegen bestehen nach bestem Wissen des Autors keine Erkenntnisse zur systematischen Analyse der Kombination der verschiedenen Randomisierungs- und Adaptionstechniken. Bisherige Forschungsarbeiten konzentrieren sich in der Regel auf einzelne Methoden zur Verbesserung synthetischer Datensätze

1. Einleitung

oder auf die Mischung mit Realdaten. Es ist keine Forschungsarbeit bekannt, welche beim Training von tiefen neuronalen Netzen mit ausschließlich synthetischen Bilddaten eine Performanz erreicht, welche in realen Anwendungen nicht signifikant niedriger ist als beim Training mit Realdaten – einen sinnvollen Umfang und eine repräsentative Verteilung der Realdaten vorausgesetzt.

1.2. Problembeschreibung und Forschungsfragen

Im Rahmen der vorliegenden Arbeit werden die verschiedenen Methoden der synthetischen Bilddatengenerierung und Datennachbearbeitung jeweils einzeln und kombiniert untersucht. Ziel ist es, durch diese Kombination Datensätze zu erzeugen, welche eine geringere Domänenlücke aufweisen als die Datensätze, welche lediglich auf der Anwendung der einzelnen Ansätze beruhen. Überprüft werden die Datensätze mittels neuronaler Netze, welche für Objekterkennungsaufgaben trainiert und mit realen Testdatensätzen evaluiert werden. Dabei sollen die synthetisch trainierten neuronalen Netze eine Performanz – die mit einer passenden Metrik gemessene Leistung des Modells – erreichen, welche nicht signifikant niedriger ist als die beim Training mit realen Bilddaten. Der Gesamtprozess ist in Abbildung 1.1 dargestellt. Die forschungsleitende Fragestellung lautet:

Forschungsleitende Fragestellung

Lassen sich Objekterkennungssysteme auf Basis neuronaler Netze mittels **synthetischer Bilddaten** so trainieren, dass bei der Übertragung in die reale Zieldomäne eine **Performanz** erreicht wird, welche **nicht signifikant niedriger** ist als beim Training mit **realen Bilddaten**?

Die im Rahmen dieser Arbeit untersuchten und kombinierten Verfahren zur Reduktion der Domänenlücke umfassen die folgenden drei Ansätze mit spezifischen Untersuchungszielen:

1. **Scene Engineering:** Die manuelle oder automatisierte Anpassung einer Simulationsszene mithilfe von Domänenwissen und ggf. von wenigen realen Referenzdaten. Dabei werden unter anderem die Eigenschaften von Kamera, Objekten und Umgebung sowie die Beleuchtung angepasst (Szenenkontext). Neben der Ermittlung von passenden

Parameter-Mittelwerten findet die Definition von geeigneten Randomisierungsbereichen für die einzelnen Parameter statt.

→ Ziel hierbei ist es, die Relevanz der einzelnen Eigenschaften der Simulationsszene bzw. des Szenenkontextes sowie die Automatisierbarkeit der Simulationsoptimierung (Scene Adaptation) zu beurteilen.

2. **Domain Randomization:** Die breite Randomisierung ohne Berücksichtigung des Szenenkontextes weit über den zugehörigen realen Anwendungsfall hinaus. Die Randomisierung kann dabei entweder direkt während der Datengenerierung in der Simulation erfolgen oder im Nachhinein über Augmentationsverfahren auf bereits generierte Datensätze angewendet werden.

→ Ziel hierbei ist es, den Einfluss der Randomisierung relevanter Szeneneigenschaften und Bildstatistiken zu beurteilen.

3. **Domain Adaptation:** Die Anpassung von bereits generierten Bilddatensätzen an reale Referenzdatensätze durch Verfahren zur Adaption der Bildstatistik sowie durch generative Verfahren des maschinellen Lernens. Es können neben den Bildeigenschaften auch Informationen der zugehörigen Annotationen berücksichtigt werden.

→ Ziel hierbei ist es, den Einfluss der Methoden der Bildanpassung sowie die Auswirkungen ihrer Kombination zu beurteilen.

Somit ergibt sich aus der Kombination der drei Ansätze der folgende Teilaspekt der forschungsleitenden Fragestellung:

Teilaspekte der Fragestellung

Kann die Performanz der mit synthetischen Bilddaten trainierten Objekterkennungssysteme durch den **Einsatz** und die **Kombination** von (1) **Scene Engineering**, (2) **Domain Randomization** und (3) **Domain Adaptation** zur **Simulations- und Datenanpassung** verbessert werden?

Die vorliegende Arbeit behandelt diese Fragestellung in Bezug auf industrielle Anwendungsfälle aus den Bereichen der automatisierten Artikelkommissionierung und der Qualitätsinspektion. Zur Bewertung der Performanz werden primär Standardmetriken aus dem Bereich der Objekterkennung sowie spezifische Metriken für den Anwendungsfall der automatisierten Kommissionierung eingesetzt.

1. Einleitung

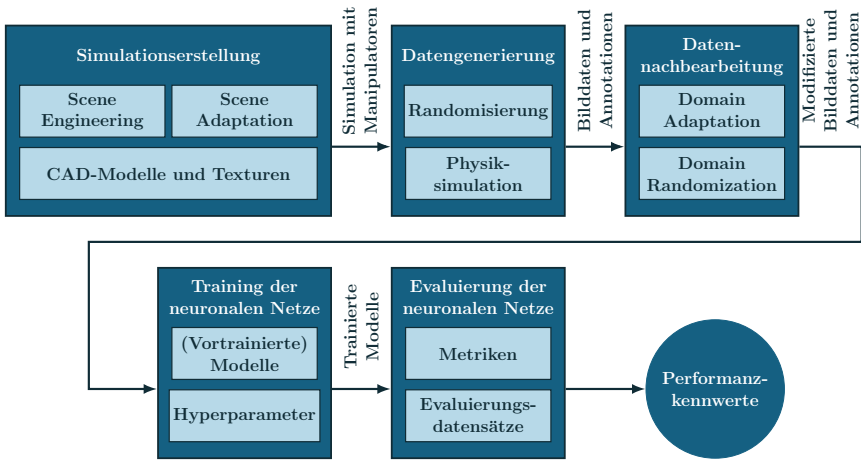


Abbildung 1.1.: Schematische Darstellung des Gesamtprozesses von der Simulationserstellung bis zu den ermittelten Performanzkennwerten der neuronalen Netze.

1.3. Aufbau der Arbeit

Kapitel 2 beschreibt den Hintergrund dieser Arbeit. Dabei wird ein Einblick in die Entwicklung moderner Bildverarbeitung auf Basis von maschinellem Lernen bis heute gegeben. Die im Rahmen dieser Arbeit untersuchten industriellen Anwendungsfälle werden eingeführt, abgegrenzt und detailliert beschrieben. Es werden Vor- und Nachteile synthetischer Daten für maschinelles Lernen diskutiert und ein Überblick über den Stand der Forschung im Bereich der synthetischen Datengenerierung gegeben.

Kapitel 3 erläutert die für Leser dieser Arbeit notwendigen theoretischen Grundlagen aus den Bereichen der Bildverarbeitung, der neuronalen Netze, der Bayes'schen Optimierung, Stochastik und der Bewertungsmetriken für Bildverarbeitungssysteme.

In **Kapitel 4** werden die für die Experimente genutzten Methoden eingeführt. Diese sind entlang der Verarbeitungskette des Trainings tiefer neuronaler Netze auf Basis synthetischer Bilddaten strukturiert. Dabei werden die Bereiche der Szenenerstellung und Datengenerierung in der Simulation, der Bildtransformation durch Randomisierung und Adaption sowie das Training und die Evaluierung der Bildverarbeitungssysteme abgedeckt.

Kapitel 5 umfasst alle im Rahmen dieser Arbeit durchgeführten Experimente und stellt deren Ergebnisse dar. Abschnittsweise werden die gewonnenen Erkenntnisse diskutiert, eingeordnet und bewertet. Diese sind nach den drei zentralen Methoden Szenenerstellung, Domain Randomization und Domain Adaptation sowie deren Kombination strukturiert.

Die Untersuchungen, Ergebnisse und gewonnenen Erkenntnisse der vorliegenden Arbeit werden in **Kapitel 6** zusammengefasst und eingeordnet sowie mit einem Ausblick auf zukünftig relevante Themenfelder im zugehörigen Forschungsgebiet abgeschlossen.

1.4. Konventionen

Im Folgenden werden spezifische Konventionen aufgeführt, denen diese Arbeit folgt, auch wenn sie nicht durchgängig als allgemein üblich gelten.

Dezimaltrennzeichen: Obwohl die vorliegende Arbeit in deutscher Sprache verfasst ist, wird zur Verbesserung der Lesbarkeit und zur Vereinheitlichung mit den in der internationalen Fachliteratur üblichen Darstellungsweisen durchgängig das englische Dezimaltrennzeichen (Punkt) verwendet.

Abbildungs- und Tabellenbeschreibungen: Die Abbildungsunterschriften und Tabellenüberschriften sind bewusst ausführlicher gestaltet, um eine weitgehend selbstständige Interpretierbarkeit der Abbildungen und Tabellen zu ermöglichen. Dadurch sollen wiederholte Rückverweise auf den Haupttext reduziert und die Nachvollziehbarkeit der dargestellten Inhalte erhöht werden.

2. Hintergrund

Im Folgenden wird der Hintergrund der Arbeit dargelegt. Dabei werden die untersuchten Anwendungsfälle beschrieben, die Vorteile des Einsatzes synthetischer Bilddaten aufgezeigt und der für diese Arbeit relevante Stand der Forschung dargestellt.

2.1. Automatisierung in der Industrie am Beispiel von Item Picking

Die im Rahmen dieser Arbeit untersuchten Anwendungsfälle betreffen Bildverarbeitungsaufgaben aus dem Bereich der Automatisierung industrieller Logistik- und Produktionsprozesse. Im Fokus der Untersuchungen steht die Aufgabe der automatisierten Artikelkommissionierung (engl. *automated item picking*) mithilfe von Kamera- und Robotiksystemen, im Weiteren als *Item Picking* bezeichnet. Die Bildverarbeitungssysteme basieren hierbei auf tiefen neuronalen Netzen, welche mit Beispieldaten trainiert werden.

Die Prozesse, Objekte, Anlagen und Umgebungen sind im industriellen Kontext meist detailliert geplant und leichter zu kontrollieren als in anderen Anwendungsfeldern wie beispielsweise dem autonomen Fahren. Es wird davon ausgegangen, dass realistische 3D-Modelle und Texturen der relevanten Szenenobjekte, insbesondere der Artikel und gegebenenfalls der Umgebungsobjekte (z. B. Ladungsträger), zur Verfügung stehen. Bei industriellen Eigenprodukten ist dies in der Regel durch die interne Entwicklungs- und Marketing-Abteilung gewährleistet. Für Fremdprodukte wird von der Möglichkeit zur Erfassung der 3D-Daten und Texturen über ein geeignetes Vermessungssystem ausgegangen. Zudem ist die Montage der Anlagenkomponenten und Kamerasysteme häufig statisch, die Lichtverhältnisse vergleichsweise stabil und die Umgebung meist bekannt – in einigen Fällen sind die Zellen sogar durch Schutzzäune von der Umgebung abgekapselt. Daher wird davon ausgegangen, dass sowohl die relative Pose zwischen Kamera und Umgebungsobjekten (z. B. Ladungsträger, Anlage

2. Hintergrund

oder Hintergrund), abgesehen von geringfügigen Änderungen aufgrund von Vibrationen, als auch die Position des Ladungsträgers weitgehend konstant sind.

Durch die Kenntnis der Szenenstruktur sowie die Verfügbarkeit von 3D-Modellen eignen sich solche industriellen Anwendungsfälle meist besonders gut für eine detaillierte Nachbildung in der Simulation und damit für die Generierung von synthetischen Trainingsdaten. Die zugehörigen Bilddaten weisen dabei vergleichsweise homogene Eigenschaften und wenig unvorhergesehene Abweichungen auf. Diese Besonderheit wird in der vorliegenden Arbeit explizit durch realitätsnahe Nachbildung (Scene Engineering) untersucht und genutzt, um die erreichte Performanz der trainierten NNs für die Aufgabe der Objekterkennung zu steigern.

Dennoch sind die Anforderungen an Präzision und Sensitivität häufig ähnlich hoch wie beim autonomen Fahren und wesentlich höher als in vielen anderen Anwendungsbereichen neuronaler Netze. So hat beispielsweise eine fehlerhafte Bildklassifikation in einer Foto-Anwendung (Verbrauchersektor) meist nur geringe negative Auswirkungen, während Fehler im Rahmen einer industriellen Qualitätsprüfung ein hohes Sicherheitsrisiko darstellen können. Bei den betrachteten Bildverarbeitungsaufgaben fallen Fehler 1. Art (falsch positiv, α -Fehler) jeweils schwerer ins Gewicht als Fehler 2. Art (falsch negativ, β -Fehler). Beim Item Picking haben nicht gefundene Objekte (falsch negativ) in der Regel keine relevanten Auswirkungen oder führen lediglich zu einer Wiederholung der Kameraaufnahme. Dagegen können durch eine falsch positive Detektion schwerwiegende Schäden beim folgenden Greifversuch entstehen. Auch bei der Prüfung notwendiger Objektmerkmale im Rahmen der Qualitätsinspektion können falsch positive Detektionen zum sicherheitskritischen Einbau von fehlerhaften Komponenten führen, während falsch negative Erkennungen lediglich zu einer zusätzlichen manuellen Nachkontrolle führen.

Die betrachteten Anwendungsfälle umfassen dabei eine geringe Anzahl unterschiedlicher Objekte bzw. Klassen im einstelligen Bereich (2 bis 9). Im Falle von verpackten Artikeln gilt entsprechend eine Einschränkung auf wenige Verpackungsarten mit eindeutig unterscheidbaren Texturen, Formen oder Größen. Dies steht im Gegensatz zu Anwendungsfällen des Online-Handels, bei welchen oft tausende Artikel in chaotischer Lagerhaltung detektiert werden müssen (z. B. Amazon). Grund für diese Einschränkung auf wenige Artikel ist, dass die Kapazität der neuronalen Netze, welche sich unter anderem über deren Größe definiert, explizit keine Rolle spielen soll. Das Training

2.1. Automatisierung in der Industrie am Beispiel von Item Picking

mit wesentlich mehr Objektklassen würde lediglich unnötig lange Trainingszeiten ohne relevanten Erkenntnisgewinn in Bezug auf die Forschungsfragen dieser Arbeit bedingen. Darüber hinaus kann in vielen industriellen Anwendungsfällen davon ausgegangen werden, dass mithilfe der Informationen aus einem zugehörigen Verwaltungssystem für jeden Detektionsprozess eines von mehreren spezifischen neuronalen Netzen ausgewählt werden kann, welches jeweils für eine Untermenge der Artikel trainiert wurde.

Neben der Anzahl der Klassen wird auch die Anzahl der Objekte pro Bild auf den einstelligen Bereich (1 bis 9, beispielsweise kein Schüttgut) eingeschränkt. Der Grund hierfür liegt in der Schwäche der betrachteten Objekterkennungsverfahren begründet, bei Bildern mit sehr vielen Objekten alle Instanzen zu lokalisieren und diese jeweils präzise zu separieren. Da es sich bei Kommissionieraufgaben mit Schüttgut in der Regel um eine sortenreine Befüllung handelt, ist eine Klassifizierung hier oft nicht notwendig und statt der Erkennung einzelner Objektinstanzen werden lediglich Greifflächen und -posen ermittelt. Die betrachteten Artikel sind weitgehend rigide, wobei eine geringfügige Verformung sowie eine Flexibilität über definierte Gelenke akzeptiert werden. Ausgeschlossen sind dagegen voll flexible Artikel wie beispielsweise lose Textilien.

Die Prämissen der im Rahmen dieser Arbeit untersuchten Anwendungsfälle lassen sich wie folgt zusammenfassen:

Anwendungsfall

Eine geringe Anzahl bekannter rigider Objekte (Artikel oder Merkmale) muss bei weitgehend konstanten Umgebungsbedingungen und Kamera-zu-Szene-Relation klassifiziert und lokalisiert werden (Objekterkennung oder Instanzsegmentierung).

Aus der expliziten Betrachtung industrieller Anwendungsfälle ergeben sich jedoch auch besondere Herausforderungen für die Bildverarbeitung. Zu diesen zählen unter anderem:

1. Dicht gepackte Szenen: Die Gegenstände werden typischerweise dicht im Ladungsträger gepackt und beim Transport indirekt bewegt, wodurch eine exakte Trennung der einzelnen Artikel erschwert wird.
2. Hohe Überdeckungsrate: Durch die enge Platzierung im Ladungsträger kommen Überdeckungen der zu detektierenden Artikel sehr häufig vor, wodurch insbesondere deren Klassifikation erschwert wird.

2. Hintergrund

3. Unterscheidung ähnlicher Objekte: Regelmäßig müssen mehrere ähnliche Artikel anhand nur weniger Merkmale unterschieden werden. Dies gilt insbesondere bei der Klassifikation von verschiedenen Zuständen desselben Objektes (z. B. geöffnete und geschlossene Kartons).
4. Keine oder wenige Daten: Für das Training von maschinellen Lernverfahren stehen häufig nur wenige Referenzdaten zur Verfügung. Dies gilt insbesondere bei der Einführung neuer Produkte und noch mehr vor der Inbetriebnahme neuer Logistik- oder Produktionsanlagen.

Im Rahmen dieser Arbeit werden die Experimente anhand eines Hauptanwendungsfalls aus der Logistik sowie zwei ergänzenden Anwendungsfällen aus der Logistik und der Produktion durchgeführt. Im Folgenden werden die drei untersuchten Anwendungsfälle dargestellt, die zugehörigen Bildverarbeitungsaufgaben erläutert und relevante Kennzahlen genannt. Exemplarische Darstellungen der Anwendungsfälle finden sich in Abbildung 2.1.

2.1.1. Item Picking Automobilindustrie (IP-A)

Der erste Anwendungsfall entstammt einer automatisierten Kommissionieranlage (Item Picking) in der Automobilindustrie. Im Weiteren wird für diesen Anwendungsfall die Abkürzung *IP-A* verwendet.

Im Rahmen der zugehörigen Bildverarbeitungsaufgabe müssen verschiedene Artikel in einem standardisierten Kartonladungsträger aus einer festen Kamerapose detektiert, d. h. nach Artikeltyp klassifiziert und lokalisiert werden (siehe Abbildung 2.1a). Das untersuchte Artikelspektrum umfasst sieben Artikel eines Automobilzulieferers: vier Boxen unterschiedlicher Größe und Textur, Keilriemen in zwei Größen sowie einen eng gepackten Beutel. Während die Boxen und der Beutel weitgehend rigide sind, treten bei den Keilriemen vereinzelt leichte Verformungen auf. Die Ladungsträger sind jeweils mit bis zu sieben Artikeln gemischt befüllt, wobei kein Artikel mehrfach vorkommt.

Es wurden fünf reale Vergleichsdatensätze manuell erzeugt und jeweils mit Masken annotiert (Instanzsegmentierung):

1. Voll: Trainingsdatensatz im Umfang von 1 016 annotierten Bilddaten
2. Midi: Trainingsdatensatz im Umfang von 141 annotierten Bilddaten, welche eine Teilmenge des vollen Trainingsdatensatzes darstellen

2.1. Automatisierung in der Industrie am Beispiel von Item Picking



(a) Beispielbilder des Anwendungsfalls IP-A (links und mitte: Voll, rechts: Pre).



(b) Beispielbilder des Anwendungsfalls IP-B.



(c) Beispielbilder des Anwendungsfalls QP-A.

Abbildung 2.1.: Reale Beispielbilder der drei untersuchten Anwendungsfälle (a) IP-A, (b) IP-B und (c) QP-A.

3. Mini: Trainingsdatensatz im Umfang von 21 annotierten Bilddaten, welche eine Teilmenge des Trainingsdatensatzes Midi darstellen
4. Pre: Evaluierungsdatensatz im Umfang von 93 annotierten Bilddaten, welche vor den Trainingsdaten erstellt wurden
5. Post: Evaluierungsdatensatz im Umfang von 253 annotierten Bilddaten, welche gemeinsam mit den Trainingsdaten erstellt wurden (nah am Trainingsdatensatz)

Die generierten synthetischen Trainingsdatensätze des Anwendungsfalls IP-A umfassen jeweils 5 000 annotierte Bilddaten. In Ausnahmefällen, in denen die Anzahl der Bilddaten davon abweicht, wird dies explizit erwähnt.

2.1.2. Item Picking Bekleidungsindustrie (IP-B)

Auch der zweite Anwendungsfall entstammt einer automatisierten Kommissionieranlage (Item Picking), jedoch ist dieser in der Bekleidungsindustrie angesiedelt und stellt somit eine Übertragung auf eine andere Branche dar. Der Anwendungsfall wurde bereits in einer früheren Veröffentlichung des Autors untersucht [119]. Im Weiteren wird für diesen Anwendungsfall die Abkürzung *IP-B* verwendet.

Im Rahmen der zugehörigen Bildverarbeitungsaufgabe müssen verschiedene Schuhkartons eines Sportartikelherstellers in einem standardisierten Kunststoffladungsträger aus einer fixen Kamerapose detektiert werden (siehe Abbildung 2.1b). Aufgrund der sortenreinen Befüllung der Ladungsträger mit bis zu zwölf Artikeln ist keine Unterscheidung nach einzelnen Schuhkartontypen notwendig. Stattdessen wird die Detektion des zu untersuchenden Artikelspektrums von sechs Schuhkartons unterschiedlicher Textur und Größe typagnostisch trainiert. Da die Deckel der Schuhkartons nicht versiegelt sind, können sich diese beim Transport öffnen, wodurch eine automatisierte Handhabung erheblich erschwert wird. Daher werden die Kartons nach deren Zustand (geöffnet oder geschlossen) klassifiziert und nur in geschlossenem Zustand automatisiert kommissioniert.

Besondere Herausforderungen sind hierbei der oft hohe Füllgrad mit typengleichen Artikeln, wodurch die Unterscheidung einzelner Instanzen erschwert wird, sowie die Unterscheidung des Zustands eines Artikels mit fließendem Übergang zwischen geöffnetem und geschlossenem Deckel [119].

Es wurden vier reale Vergleichsdatensätze manuell erzeugt und jeweils mit Begrenzungsrahmen annotiert (Objekterkennung):

1. Voll: Trainingsdatensatz im Umfang von 5 452 annotierten Bilddaten
2. Mini: Trainingsdatensatz im Umfang von 100 annotierten Bilddaten, welche eine Teilmenge des vollen Trainingsdatensatzes darstellen
3. Nah: Evaluierungsdatensatz mit bekannten Artikeln und Szenen im Umfang von 691 annotierten Bilddaten (nah am Trainingsdatensatz)
4. Fern: Evaluierungsdatensatz mit unbekannten Artikeln und Szenen im Umfang von 130 annotierten Bilddaten (fern vom Trainingsdatensatz)

Die generierten synthetischen Trainingsdatensätze des Anwendungsfalls IP-B umfassen jeweils 14 000 annotierte Bilddaten.

2.1.3. Qualitätsprüfung in der Automobilindustrie (QP-A)

Der dritte Anwendungsfall stellt eine Übertragung auf eine andere Anwendung dar und entstammt einem automatisierten Qualitätsprüfungssystem in der Automobilindustrie. Der Anwendungsfall wurde bereits in einer früheren Veröffentlichung des Autors untersucht [120]. Im Weiteren wird für diesen Anwendungsfall die Abkürzung *QP-A* verwendet.

Im Rahmen der zugehörigen Bildverarbeitungsaufgabe müssen sieben verschiedene Merkmale eines Automobilgetriebes aus fünf definierten Kameraposen detektiert, d. h. nach Merkmalstyp klassifiziert und lokalisiert werden (siehe Abbildung 2.1c). Pro Ansicht wird die Existenz von bis zu drei Merkmalen überprüft. Die anschließende Ableitung einer binären Qualitätsentscheidung wird in dieser Arbeit nicht betrachtet. Bei den Merkmalen handelt es sich um montierte Komponenten wie beispielsweise eine Blattfeder oder einen Getriebeflansch.

Besondere Herausforderungen ergeben sich aus der Verwendung mehrerer Kameraposen sowie aus verschiedenen Abweichungen der eingesetzten 3D-Modelle gegenüber den realen Objekten. Die zu detektierenden Merkmale sind dabei jedoch nicht von den Abweichungen der Modelle betroffen. Eine weitere Herausforderung stellt der Mangel an Negativbeispielen in der Simulation dar. Fehlt ein Merkmal im Realbild nicht nur, sondern wurde durch ein unbekanntes Merkmal ersetzt – beispielsweise durch den Einbau eines falschen Getriebeflanschs –, kann dieser Fehlerfall nicht in der Simulation nachgebildet werden.

Es wurden drei reale Vergleichsdatensätze manuell erzeugt und jeweils mit Masken annotiert (Instanzsegmentierung):

1. Voll: Trainingsdatensatz im Umfang von 1 000 annotierten Bilddaten (200 je Kamerapose)
2. Mini: Trainingsdatensatz im Umfang von 50 annotierten Bilddaten (10 je Kamerapose), welche eine Teilmenge des vollen Trainingsdatensatzes darstellen
3. Evaluierungsdatensatz im Umfang von 100 annotierten Bilddaten (20 je Kamerapose)

Die generierten synthetischen Trainingsdatensätze des Anwendungsfalls QP-A umfassen jeweils 5 000 annotierte Bilddaten (1 000 je Kamerapose).

2.2. Synthetische Bilddaten

Synthetische Daten stellen einen zunehmend relevanteren Ansatz dar, um dem großen Datenbedarf beim Training moderner Deep Learning-Modelle zu begegnen. Unter synthetischen Daten, vom altgriechischen „*sýnthesis*“ etwa Zusammensetzung oder Verknüpfung, werden künstliche Daten verstanden, welche entweder durch Modifikationen und Zusammensetzungen realer Daten erzeugt oder ohne direkte Verwendung realer Daten simuliert werden. Im Falle von synthetischen Bilddaten können dies beispielsweise Bilder sein, welche bildstatistisch verändert wurden, eine Zusammenstellung von Bildausschnitten mehrerer realer Bilder darstellen oder mithilfe einer Rendering-Software, im Weiteren als *Simulation* bezeichnet, erzeugt wurden. Synthetische Bilddaten werden in zahlreichen Anwendungsfeldern eingesetzt, so zum Beispiel im Bereich des autonomen Fahrens, in der Medizin oder in industriellen Anwendungen der Produktion und Logistik. Gegenüber der Verwendung realer Bilddaten bieten insbesondere die durch Simulation erzeugten synthetischen Daten zahlreiche Vorteile.

Hierbei ist zunächst die Reduktion des Aufwands und damit der Kosten zu nennen, welche im Falle realer Bilddaten durch die Erzeugung und deren Annotierung entstehen. Falls die Realdaten nicht indirekt aus einem existierenden Überwachungsprozess hervorgehen, müssen diese aufwendig über Kamerasysteme erzeugt werden, wobei eine repräsentative Verteilung der Szenen sicherzustellen ist. Hinzu kommt, dass die realen Anlagen, für die das Bildverarbeitungssystem benötigt wird, bei industriellen Anwendungen zum Zeitpunkt der Datengenerierung häufig noch gar nicht existieren. Hierbei ist eine Erzeugung von realen Bilddaten gar nicht bzw. nur durch eine aufwendige Umsetzung mit einer prototypischen Anlage möglich. Dasselbe gilt für neu entwickelte Produkte, welche bereits ab dem Produktionsstart automatisiert erkannt oder lokalisiert werden sollen. Einen noch größeren Aufwand stellt in der Regel die manuelle Erstellung der Annotationen, auch als Labels bezeichnet, für die realen Bilddaten dar. Über die Annotationen werden die, zumindest aus Sicht der Benutzers und für die jeweilige Anwendung, korrekten Ausgaben vorgegeben. Diese werden als Ground-Truth, übersetzt in etwa Grundwahrheit, bezeichnet und stellen in der Bildverarbeitung die Klasse eines Bildes (Bildklassifikation) oder die Klassen und Lokalisierungen von Objektinstanzen im Bild (Objekterkennung) dar. Der Annotationsaufwand ist bei Objekterkennungsaufgaben wesentlich höher als bei der Bildklassifikation. Hier muss jedes zu erkennende Objekt nicht nur klassifiziert, sondern auch über einen Begrenzungsrahmen (engl. *bounding box*) lokalisiert oder

im Falle einer Segmentierungsaufgabe sogar auf Bildpunktebene maskiert werden. Zusätzlich besteht bei diesem manuellen Prozess die Gefahr von ungenauen oder sogar fehlerhaften Annotationen, welche in der Regel zu Performanzeinbußen beim Training der Bildverarbeitungssysteme führen.

Im Gegensatz dazu können synthetische Bilddaten mithilfe einer Simulation nach der initialen Erstellung der Simulationsszene in kurzer Zeit und zu geringen Kosten erzeugt werden. Die zugehörigen Annotationen werden hierbei in der Regel automatisiert erstellt, sind hoch präzise und bedürfen, abgesehen von systematischen menschlichen Fehlern bei der Simulationserstellung, keiner weiteren Anpassung oder Überprüfung. Eine solche Simulation kann bereits erstellt werden, wenn die Anlagenplanung bzw. Produktentwicklung abgeschlossen ist und entsprechende Bau- und Konstruktionspläne sowie 3D-Modelle und Texturen existieren. Auf diese Weise können synthetische Datensätze bereits vor der Einführung neuer Anlagen oder Produkte und ohne die Umsetzung von Prototypen erstellt werden, wodurch der Zeitbedarf bis zur Einführung des Bildverarbeitungssystems stark reduziert werden kann. Darüber hinaus weist eine simulationsbasierte Datengenerierung eine hohe Flexibilität und Skalierbarkeit auf. Bei Bedarf können in vergleichsweise kurzer Zeit und zu geringen Kosten mehr Daten, Daten mit neuen Produkten oder angepassten Szenen (z. B. andere Hintergründe oder andere Ladungsträger) generiert werden.

Ein weiterer Vorteil der synthetischen Datengenerierung ist die einfache Steuerbarkeit von Annotations- und Bildeigenschaften und damit deren Verteilung im erzeugten Datensatz. So kann beispielsweise das Aufkommen der Objektklassen sowie deren Positionierung im Bild gezielt gesteuert werden. Einsatz findet dies häufig bei der Verstärkung von in der realen Anwendung unterrepräsentierten Klassen, wie beispielsweise Negativbeispielen in der Qualitätsprüfung. Dasselbe gilt für Bildeigenschaften, welche beispielsweise über Kamera- und Beleuchtungsparameter gezielt angepasst oder für eine höhere Robustheit in einem breiteren Spektrum randomisiert werden können.

Abschließend bestehen insbesondere im industriellen Kontext Vorteile in Bezug auf Privatsphäre und Datenschutz. Dies kann zum einen das Ausschließen von (unbeteiligten) Personen im Bild betreffen. Andererseits können mithilfe synthetischer Daten Unternehmensgeheimnisse gewahrt werden, welche beispielsweise durch die Abbildung von realen Produktionsanlagen, Produkten oder von Produkteigenschaften verletzt werden könnten.

2.3. Stand der Forschung

Die Methoden der synthetischen Datengenerierung für das Training tiefer neuronaler Netze für Bildverarbeitung reichen vom vergleichsweise einfachen Ausschneiden und Einfügen (Cut & Paste) realer oder gerenderter Objekte in reale Hintergrundbilder [32, 24, 33] bis zum fotorealistischen Rendering mit vollständigen 3D-Simulationen (Full-Stack Simulation) [2, 56]. Synthetische Trainingsdaten werden unter anderem in den folgenden Anwendungsfeldern eingesetzt:

1. Bildbasierte Greifposesbestimmung für Roboteranipulation [114, 113, 28, 129]
2. Robotersteuerung und -regelung [54, 55, 45, 98, 99]
3. Detektion von Alltagsgegenständen [32, 44, 43, 17]
4. Detektion industrieller Produkte oder deren Merkmale [44, 62, 77, 69]
5. Objekterkennung beim autonomen Fahren [2, 56, 29, 117, 116, 82]
6. Klassifizierung von Biomarkern in der medizinischen Diagnostik [49, 105, 1, 21]

Da die Übertragung neuronaler Netze in die Realität eine zentrale Herausforderung beim Training mit synthetischen Daten darstellt, werden unterschiedliche Ansätze verfolgt, um die sogenannte Domänenlücke zur realen Welt zu verringern. Die Domänenlücke stellt eine abstrakte Beschreibung der vielfältigen Unterschiede zwischen der Trainings- und der Anwendungsdomäne dar, welche sich sowohl auf bildstatistische als auch auf bildinhaltliche Aspekte der jeweiligen Datensätze bezieht. Ein vielversprechender Ansatz ist die Abstraktion der Erkennungsaufgabe durch Domain Randomization [114, 113, 55, 17, 116, 82, 76]. Dabei wird die Simulationsdomäne so stark randomisiert, dass die Eigenschaften der realen Anwendung nach Möglichkeit Teil des Randomisierungsspektrums sind. Für das trainierte Bildverarbeitungssystem, so die Idee, stellen die realen Daten damit keine fremde, sondern lediglich eine Ausprägung der synthetischen Domäne dar. Andere Verfahren setzen nach der initialen Datengenerierung an und nutzen Verfahren der Domain Adaptation wie beispielsweise der Bildtransformation mit generativen Modellen [55, 45, 1, 76, 8, 78, 64, 83], um die bereits erzeugten Bilddaten an die reale Domäne anzupassen.

Im Folgenden wird der Stand der Forschung mit Blick auf die Art der Datengenerierung sowie auf die Methoden der Domain Randomization und Domain Adaptation näher beleuchtet. Abschließend wird die vorliegende Arbeit in den aktuellen Stand der Forschung eingeordnet.

2.3.1. Von Realbild-Rekomposition bis Full-Stack-Simulation

Die Methoden der Realbild-Rekomposition, wie beispielsweise der Ansatz des Ausschneidens und Einfügens (engl. cut and paste), sind vergleichsweise einfach zu nutzen. Für die Anwendung werden lediglich segmentierte Realdaten der zu erkennenden Objekte sowie eine Sammlung von Hintergrundtexturen benötigt. Im Rahmen der Datengenerierung werden einzelne oder mehrere Objektinstanzen aus dem Quellbild ausgeschnitten und an einer zufälligen Position in ein zufällig gewähltes Hintergrundbild eingefügt [32, 24]. Dabei findet in der Regel keine Berücksichtigung der räumlichen Szenenstruktur statt, wobei die genutzten Hintergrundtexturen entweder kontextunabhängig oder passend zum Anwendungsfall ausgewählt sein können. Neben der Erzeugung von synthetischen Datensätzen wird dieser Ansatz auch als Augmentationsverfahren beim Training von Modellen für Instanzsegmentierung mit Realdaten genutzt [33].

Eine Alternative zur Verwendung von aufwendig zu erzeugenden segmentierten Realdaten stellt das Rendern von Objekten auf Basis von 3D-Modellen dar [79, 108, 94, 44, 127, 2, 69, 58], welche in reale Hintergrundtexturen eingefügt werden. Um eine stärkere Variation der Hintergrundtexturen zu erreichen, setzen andere Forschungsarbeiten auf die Füllung des Hintergrunds mit zufälligen, sich überschneidenden 3D-Objekten [43, 6]. Daneben werden neben den zu erkennenden Objekten auch Störobjekte im Vordergrund genutzt, um die Vielfalt der Daten zu erhöhen und die Erkennung von überlagerten Objekten zu trainieren [24, 94, 6].

Die Performanz der mit diesen Daten trainierten Objekterkennungssysteme kann jedoch durch die Nichtberücksichtigung der räumlichen Szenenstruktur, beispielsweise durch fehlende oder unrealistische Schatten und physikalisch falsch positionierte Elemente beeinträchtigt werden. Um die Ausschnitte oder gerenderten 3D-Objekte realistischer in das Bild einzufügen, werden verschiedene Methoden der Überblendung (engl. blending) und der intelligenten Objektpositionierung eingesetzt. Dazu zählen unter anderem einfache Blending-Verfahren wie das Unschärfezeichnen an den Rändern des Ausschnitts [24, 44], die Überlagerung mit Bildrauschen [44] oder das

2. Hintergrund

Poisson Blending [24]. Aufwendigere Methoden setzen beispielsweise auf das Rendern mit zum Anwendungsfall passenden Reflexionseigenschaften (Environment Maps) [2] oder auf eine realistischere Positionierung auf Basis einer semantischen Segmentierung der Hintergrundtexturen [32, 2].

Um die räumliche Szenenstruktur bei der Generierung der synthetischen Bilddaten direkt zu berücksichtigen, werden häufig Full-Stack-Simulationen eingesetzt [29, 56, 74, 82, 118, 116, 46]. Dafür werden neben den zu erkennenden Objekten auch Umgebung und Hintergrund der Szene durch 3D-Modelle dargestellt. Der Realismus der erzeugten Daten reicht dabei von fotorealistischen Darstellungen [118, 46] über die Verwendung von 3D-Spiele-Engines [56] und realitätsferneren Darstellungen [82] bis hin zu stark abstrakten Szenen [116]. Die Erstellung von hoch-realistischen Szenen erfordert einen aufwendigen Prozess der Modell- und Texturgenerierung sowie der Szenenkonstruktion. Der Vorteil dieses Ansatzes dagegen besteht darin, dass die Objekte realistisch positioniert und nahtlos in die Szene eingefügt werden können.

2.3.2. Domain Randomization

Die Randomisierung von Trainingsdaten wird bereits seit den Anfängen der Bildverarbeitung mit NNs eingesetzt und auch auf rein reale Datensätze angewendet (bspw. Augmentationstechniken). Die Randomisierung kann einer Überanpassung entgegenwirken, indem die Domäne erweitert und um fehlende Beispiele vervollständigt wird. So werden auch in allen gängigen Verfahren der synthetischen Datengenerierung Randomisierungen von Szeneneigenschaften, wie beispielsweise die Objektpositionierung oder -erscheinung sowie Beleuchtungs- oder Kameraeigenschaften, eingesetzt. Eine der ersten Forschungsarbeiten mit einer starken Randomisierung der Rendering- und Szeneneigenschaften zur Überwindung der Domänenlücke zwischen synthetischen und realen Daten ist die Arbeit von Sadeghi *et al.* [98]. Hierbei wurde bestärkendes Lernen (engl. reinforcement learning) eingesetzt, um den automatisierten Drohnenflug in der Simulation zu erlernen.

Tobin *et al.* führten den Begriff der Domain Randomization (DR) [114], im Deutschen etwa die Domänenrandomisierung, als eine Technik zur gezielten Abstraktion der Anwendungsfall-Domäne ein. Der Ansatz der Domain Randomization geht hierbei weiter als andere gängige Randomisierungsmethoden. Sie schlagen vor, die synthetische Domäne so stark und abstrakt zu vergrößern, dass die Unterschiede zwischen Simulation und Realität überwunden werden können, indem die reale Anwendung nur einen kleinen

Teilbereich der großen synthetischen Domäne darstellt. Sie gehen davon aus, dass synthetische Daten weder eine Übertragung des Szenenkontexts noch eine fotorealistische Darstellung benötigen, falls durch eine umfassende Randomisierung genügend Datenvielfalt erzeugt wird. In ihren Experimenten konnten sie zeigen, dass der Transfer von der Simulation in die Realität mithilfe vorab trainierter Modelle und einfacher Randomisierungsstrategien gelöst werden kann. Bei der von ihnen untersuchten bildbasierten Greifaufgabe waren insbesondere Texturfarben und -muster, Kamerapositionen und Störobjekte relevante Randomisierungsparameter. Die Untersuchungen beruhen dabei jedoch auf stark vereinfachten Szenen, beispielsweise mit geometrischen Grundformen statt realistischer Objekte als Greifziele. In einer Folgepublikation weiten sie den Ansatz auf die prozedurale, randomisierte Erzeugung von Objektmodellen aus [113]. Hierbei wurde eine Objektformagnostische Greifplanung auf Basis von Objekt-Modellen trainiert, welche stark randomisiert und realitätsfern generiert wurden.

Hinterstoisser *et al.* verwendeten DR zur Objekterkennung in stark überfüllten Szenen [44]. Anstatt die komplette Szene mit Umgebungsobjekten zu simulieren, wurde ein 3D-Objekt vor einem Hintergrundbild gerendert. Die Unschärfezeichnung der Objektumrisse ermöglichte ein weiches Einfügen in den Hintergrund und führte in ihren Experimenten zu besseren Ergebnissen. Ihre zentrale Erkenntnis war eine enorme Leistungssteigerung durch das Einfrieren des Feature-Extraktors während des Trainings. Der Erfolg dieses Ansatzes konnte jedoch von anderen nicht bestätigt werden (bspw. Tremblay *et al.* [116]). In einer Folgepublikation setzen sie auf eine stärkere Abstraktion der Hintergrundebene [43], wobei die besten Ergebnisse mit Unschärfe-Blending und der Randomisierung der Lichtfarbe erzielt werden konnte. Die Variationen der Kameraparameter dagegen war in ihrem Anwendungsfall mit geringem Szenenkontext weniger relevant.

James *et al.* evaluierten verschiedene Randomisierungsstrategien im Kontext des bildbasierten Robotergreifens [54]. Sie untersuchten synthetisch trainierte Convolutional Neural Networks (CNNs) auf Objektlokalisierung, ohne reale Daten oder vorab trainierte Gewichte zu verwenden. In ihrer Ablationsstudie konnten sie zeigen, dass die Kamera-Randomisierung und die Darstellung von Schatten wichtige Aspekte für die Übertragung der Simulation auf die Realität sind. Dagegen war beispielsweise die Verwendung komplexer Texturen anstelle einzelner Farben weniger relevant. Ablenkende Störobjekte waren nur von entscheidender Bedeutung, falls diese auch in den realen Daten enthalten waren.

2. Hintergrund

Tremblay *et al.* verwendeten DR bei der komplexeren Aufgabe der Fahrzeugerkennung für autonomes Fahren [116]. Die Leistung wurde anhand des realen Datensatzes KITTI [31] bewertet. Sie kombinierten mehrere Randomisierungsstrategien mit realistischen 3D-Modellen und hochwertigem 3D-Rendering. Mit ihren Experimenten konnten sie zeigen, dass gut ausgewählte Randomisierungsparameter, insbesondere für Beleuchtung und Texturen, eine erfolgreiche Übertragung auf komplexe reale Szenarien ermöglichen und sogar synthetische Replikat-Datensätze wie VKITTI [29] schlagen können. Ihre Untersuchungen beruhen jedoch auf der Detektion einer einzigen Klasse und insbesondere bei dicht positionierten Fahrzeugen zeigten sich aufgrund des fehlenden Szenenkontexts gravierende Mängel. Für die meisten Anwendungen aus dem Bereich der industriellen Artikelkommissionierung ist die Fähigkeit, dicht angeordnete Objekte präzise zu trennen, jedoch von entscheidender Bedeutung.

Basierend auf der Idee der Domain Randomization setzen Borkman *et al.* [6] das Unity Perception-Paket¹ sowie die zugehörige Pipeline SynthDet² für die Generierung synthetischer Trainingsdaten in der Spiele-Engine Unity um. Am Beispiel dieser kontextarmen Datengenerierungs-Pipeline, deren Konzept an die Arbeit von Hinterstoisser *et al.* [43] angelehnt ist, untersuchten sie das Training von Objekterkennungsverfahren für die Detektion von verpackten Lebensmitteln und weiteren Konsumgütern. Dabei verglichen sie die erreichte Performanz von CNNs beim Training mit Realdaten, mit stark randomisierten synthetischen Daten sowie mit Mischungen von realen und synthetischen Daten. Beim Training mit synthetischen Bilddaten alleine wurde eine klar niedrigere Performanz erreicht als beim Training mit realen Bilddaten. Bereits durch die Mischung mit 10 % der Realdaten ließ sich jedoch eine höhere Performanz erreichen als beim alleinigen Training mit dem vollen Realdatensatz mit 760 Bildern.

Im Kontext des maschinellen Lernens mit tiefen neuronalen Netzen werden die Einflüsse von Randomisierungseigenschaften üblicherweise mittels Ablationsstudien untersucht. Dabei werden die einzelnen Randomisierungsparameter durch die Definition von zwei Zuständen (aktiv und inaktiv) als binäre Eigenschaften betrachtet. Ausgehend von den Basiszuständen aller Eigenschaften als aktiv, wird jeweils der Einfluss der Deaktivierung einer einzelnen Eigenschaft gemessen. Die Untersuchung von Kombinationen, d. h. mehr als ein inaktiver Zustand zur selben Zeit, findet dabei in

¹GitHub Unity Perception: github.com/Unity-Technologies/com.unity.perception

²GitHub Unity SynthDet: github.com/Unity-Technologies/SynthDet

der Regel nicht statt. In Bezug auf die Realitätsnähe synthetischer Daten gegenüber der Abstraktion der Daten durch Domain Randomization wurden in veröffentlichten Ablationsstudien unter anderem folgende Randomisierungseigenschaften explizit untersucht:

- Szenenkontext: Verwendung von Objekteinbettungs- und Positionierungsstrategien [43, 2, 82, 58]
- Texturen: Verwendung realer Texturen, die auf Objekte angewendet und als Hintergrundbilder verwendet werden [94, 22, 43, 114, 2, 100, 108, 58, 116, 99]
- Farben und Muster: Verwendung zufälliger Farb-, Farbverlaufs-, Schachbrett- oder Perlin-Noise-Texturen für Objekte und Hintergrund [114, 54, 17, 38, 100, 116]
- Kamera: Variation der Ansicht durch Änderung der Kameraposition oder Brennweite [114, 54, 43, 17, 38, 99]
- Bildunschärfe: Einsatz von Weichzeichnungsfiltern, insbesondere an den Konturen von Objekten [44, 43]
- Rauschen: Hinzufügen von randomisiertem Bildrauschen [114, 44]
- Beleuchtung: Randomisierung der Anzahl und Position der Lichter in der Szene sowie ihrer Intensität und Farbe [44, 43, 38, 114, 82, 58, 116]
- Störobjekte (Distraktoren): Platzieren zufälliger Störobjekte im Vordergrund oder Hintergrund der Szene [114, 54, 38, 58, 116]
- Rendering-Qualität: Rendering mit unterschiedlicher Qualität, z.B. mit und ohne Schatten [54, 2]

Weitere Untersuchungsgegenstände waren unter anderem die Anzahl der Trainingsschritte und die Datensatzgröße [114, 113, 17, 116, 2, 62, 108, 54, 55], die Verwendung vortrainierter Netzwerke [116], die Feinabstimmung mit realen Daten [17, 6] oder das Einfrieren von Schichten des Feature-Extraktors [44, 116] und Datenerweiterungsstrategien [116, 94].

Neben der manuellen Definition der Randomisierungsgrade setzten einige Arbeiten auf den Einsatz von Meta-Algorithmen, um die relevanten Simulationsparameter zu identifizieren und deren Randomisierungsgrad zu

2. Hintergrund

optimieren. Bei einigen Ansätzen aus den Bereichen der Bildverarbeitung [59, 123, 95] und der Agentenregelung [11, 71] wird der Prozess der Randomisierungsoptimierung dabei in den Trainingsprozess integriert. Hierbei basiert die Verlustfunktion der Optimierung der Randomisierungsparameter direkt auf der jeweils aktuellen Güte des Modells der Hauptaufgabe (bspw. Bildklassifikation). Dieses Vorgehen ist jedoch sehr zeitintensiv, da die Optimierungsschleife jeweils die Trainings- und Evaluierungsprozesse der Hauptaufgabe enthält. Alternativ können die Randomisierungsparameter vor dem Training der Hauptaufgabe dahingehend angepasst werden, dass die erzeugten Bilder stärker mit einer realen Referenzdomäne übereinstimmen. Die Verlustfunktion kann dabei beispielsweise auf der Diskrepanz der internen Repräsentationen zwischen der Inferenz mit synthetischen und realen Daten eines vortrainierten Referenzmodells basieren [57]. Als Referenzmodelle kommen dabei neuronale Netze in Frage, welche auf einer möglichst breiten oder einer möglichst ähnlichen Domäne trainiert wurden. Für die Erkennung von Fahrzeugen kombinieren Prakash *et al.* die Domain Randomization mit an die Zieldomäne angepassten und damit vom Szenenkontext abhängigen Parameterverteilungen [82]. Bei der Untersuchung der von ihnen als Structured Domain Randomization (SDR) bezeichneten Technik konzentrierten sie sich auf die Platzierung von zu erkennenden Objekten und Störobjekten. In den durchgeführten Experimenten erzielten sie mit diesem Ansatz bessere Ergebnisse als mit kontextunabhängiger DR. Zakharov *et al.* schlagen ebenfalls eine Meta-Optimierung für den Ansatz der Domain Randomization auf Basis einer trainierten Encoder-Decoder-Architektur (DeceptionNet) vor [123]. DeceptionNet lernt unter anderem den Hintergrund, die Beleuchtung oder Verzerrung der Bilddaten so zu verändern, dass einem vortrainierten CNN die Erkennung möglichst stark erschwert wird. Dafür wird die Encoder-Decoder-Architektur trainiert, indem die Verluste des fixierten Erkennungsnetzes mit umgekehrten Gradienten auf DeceptionNet zurückgeführt werden. Anschließend werden die Gewichte des DeceptionNet fixiert und das Erkennungsnetz mit den jeweils transformierten Daten trainiert.

2.3.3. Domain Adaptation

Im Kontext des maschinellen Lernens beschreibt die Domain Adaptation (DA), im Deutschen etwa die Domänenanpassung, den Prozess der Anpassung eines Datensatzes aus einer Quelldomäne an eine Zieldomäne. In der Regel ist die Zieldomäne hierbei mit der Quelldomäne verwandt,

weicht jedoch in einigen Eigenschaften von dieser ab. Bei der unüberwachten Domänenanpassung (engl. unsupervised domain adaptation, UDA) findet diese Anpassung ohne den Einsatz von Annotationen der Zieldomäne statt. Ziel der Domain Adaptation ist es, die Domänenlücke zu verringern.

Die Domain Adaptation kann durch Veränderungen auf Datensatzebene (z. B. gezieltes Aussortieren von oder Erweitern um Datenpunkte) oder durch Veränderungen der bestehenden Datenpunkte erfolgen. Letzteres wird im Falle von Bildverarbeitungsaufgaben meist über eine Bildtransformation, auch als Bild-zu-Bild-Transformation (engl. image-to-image translation, I2I) bezeichnet, erreicht. Dabei findet eine Stilübertragung (engl. style transfer) statt, bei welcher ein einzelnes Quellbild an ein einzelnes Zielbild angepasst wird. Hierbei sollen zwar die für das Training relevanten und insbesondere annotierten Eigenschaften (Bildinhalt) der Quellbilder erhalten bleiben, die bildstatistischen Eigenschaften jedoch an die Zieldomäne angepasst werden. Die bildstatistischen Eigenschaften beziehen sich hierbei auf die Verteilung der Pixelwerte im Bild und damit auf die daraus resultierenden Farben, Muster und lokalen Bildmerkmale. Im Gegensatz dazu umfasst der Bildinhalt insbesondere die Art, Form und räumliche Anordnung der Objekte auf höherer Ebene. Im Falle der Anpassung synthetischer Daten an reale Referenzdaten wird auch der Begriff des Sim2Real-Transfers (engl. simulation to reality transfer) verwendet.

Für die Transformation von synthetischen Bilddaten werden häufig generative Verfahren und insbesondere Generative Adversarial Networks (GANs) eingesetzt (s. Unterabschnitt 3.2.4). Ein frühes erfolgreiches Beispiel hierfür ist die Arbeit von Shrivastava *et al.*, in welcher GANs eingesetzt wurden, um die synthetischen Bilddaten für eine Blickschätzung zu verbessern [103]. Andere Forschungsarbeiten setzen den Sim2Real-Transfer insbesondere in den Bereichen des autonomen Fahrens (semantische Segmentierung) und der Robotik (Greifplanung und -regelung) ein. Hier werden GANs und verwandte Architekturen meist zur direkten Übertragung synthetischer Bilder in die reale Domäne eingesetzt [7, 8, 47, 76, 78, 62, 45, 64, 83, 80]. So nutzten beispielsweise Bousmalis *et al.* ihre Architektur GraspGAN, um die Erkennung neuartiger Objekte im Kontext der Robotermanipulation zu verbessern [8]. Durch den Einsatz von GraspGAN konnten sie die für ein erfolgreiches Training notwendige Anzahl realer Stichproben um das bis zu 50-fache reduzieren. Neben der Transformation der synthetischen Daten durch das GAN ergänzten sie dafür die zugehörige Bildsemantik durch eine Greif-Erfolgsvorhersage.

2. Hintergrund

Mit der Veröffentlichung von CycleGAN [125] fokussierte sich der Sim2Real-Transfer stark auf den Einsatz und die Erweiterung dieser Architektur [47, 76, 62, 45, 64, 83]. Dabei beziehen sich die Erweiterungen insbesondere auf die Integration von zusätzlichen merkmals- und aufgabenspezifischen Verlustfunktionen aus den Bereichen der Bildklassifikation [64], der Objekterkennung [45], der semantischen Segmentierung [47] und des bestärkenden Lernens [83]. Durch diese Modifikationen kann die Transformation zum einen auf die aufgabenspezifischen Bildmerkmale und -eigenschaften fokussiert werden, wodurch die Performanz der mit diesen Daten trainierten maschinellen Lernverfahren verbessert werden kann. Mit anderen Weiterentwicklungen lässt sich dem Problem der Annotationsinkonsistenz, bei welcher der Inhalt des transformierten Bildes nicht mehr mit der ursprünglichen synthetischen Annotation übereinstimmt, entgegenwirken [73, 9, 60]. Jedoch werden für die Ermittlung der jeweiligen Verluste eine Vielzahl annotierter Realdaten oder bereits trainierte, aufgabenspezifische Modelle benötigt. Dass annotierte Realdaten oder vortrainierte Modelle für die jeweilige Aufgabe existieren und zugänglich sein müssen, stellt eine starke Einschränkung für die Anwendbarkeit dar.

Andere Ansätze setzen nicht bzw. nicht nur auf die Transformation der synthetischen Bilddaten. So nutzten Zhang *et al.* einen umgekehrten Ansatz für das Erlernen von Richtlinien (engl. policies) des bestärkenden Lernens (engl. deep reinforcement learning, DRL) in simulierten Umgebungen im Kontext der autonomen Navigation mobiler Roboter. Statt die synthetischen Daten anzupassen, transformierten sie die realen Daten mithilfe von CycleGAN in die Simulationsdomäne [124]. Durch die Entkopplung des Richtlinientrainings und der Domain Adaptation erreichen sie eine höhere Flexibilität und eine leichtere Übertragbarkeit auf andere Domänen. Bei einer Änderung der Domäne muss hierbei lediglich die Transformation der Realdaten neu erlernt werden, nicht jedoch die Richtlinien. Auch James *et al.* setzen für das Training der bildbasierten Roboteranipulation auf eine Anpassung realer Bilddaten [55]. Bei ihrem Ansatz RCAN wurden sowohl synthetische als auch reale Bilder mithilfe des konditionalen GANs Pix2Pix [52] in eine dritte, als kanonisch bezeichnete Domäne übertragen. Diese Domäne stellt die semantische Segmentierung der Szene dar. Sie zeigen die Überlegenheit ihres Ansatzes gegenüber unterschiedlichen Graden der Domain Randomization. Dem stehen insbesondere zwei Nachteile gegenüber. Zum einen die Voraussetzung semantisch segmentierter Realdaten für das Training des RCAN-Modells. Zum anderen die Notwendigkeit der

zusätzliche Ausführung der Bild-zu-Bild-Transformation vor jeder Inferenz in der realen Anwendung.

Eine Kombination von Domain Adaptation und Domain Randomization wurde von Nogues *et al.* betrachtet [76]. Die Domain Adaptation wurde hierbei für Bilder mit einfarbigen Hintergrund mit CycleGAN durchgeführt. Die transformierten Daten wurden in verschiedenen Verhältnissen mit randomisierten Daten gemischt. Dabei wurden die Beleuchtung (Anzahl Lichter und Farbe), die Objekte und der Hintergrund (Farbe und Textur) sowie die Kameraposition (Blickrichtung zur Szene) randomisiert. Eine Kombination der Randomisierungsverfahren für dieselben Bilder fand dabei nicht statt. Die höchste Performanz des trainierten Modells für Instanzsegmentierung wurde mit 20 % transformierten und 80 % randomisierten Daten erzielt.

2.3.4. Einordnung der vorliegenden Arbeit

Bisherige Forschungsarbeiten untersuchen die Auswirkungen von Domain Adaptation und Domain Randomization nur in wenigen Fällen im Kontext industrieller Anwendungen, sondern in der Regel im Kontext des autonomen Fahrens, der Objekterkennung von Alltagsobjekten mit inhomogenen Hintergründen sowie der Greifposenbestimmung und dem bestärkenden Lernen für Roboter. Es bleibt fraglich, ob die Erkenntnisse auf reale industrielle Anwendungen anwendbar sind, da sie die typischen Randbedingungen industrieller Anwendungsfälle vernachlässigen (s. Abschnitt 2.1). Dies gilt insbesondere für stark vereinfachte Szenen mit einfarbigen Texturen und geometrischen Grundformen (z. B. Pyramiden, Würfel, Zylinder) als Zielobjekte. Die besonderen Herausforderungen des im Rahmen dieser Arbeit untersuchten Anwendungsfalls des Item Pickings, insbesondere enge Packmuster mit Überlappung und die starke Ähnlichkeit der zu unterscheidenden Klassen, treten in den anderen genannten Anwendungsfeldern nicht oder zumindest seltener auf. Ausnahmen hiervon bilden Arbeiten, die Anwendungsfälle aus der Produktion und Logistik betrachten [69, 22], industrielle Objekte erkennen [127, 76, 62, 77] oder Szenen untersuchen, welche dem Item Picking nahe kommen [55, 28, 43, 16].

Nur eine der Arbeiten betrachtet sowohl Domain Adaptation als auch Domain Randomization [76]. Eine Kombination der Verfahren fand dabei jedoch nur indirekt durch die Mischung der Daten und nicht durch gemeinsame Anwendung auf dieselben Bilder statt. Darüber hinaus wurden keine unterschiedlichen Stärken der Randomisierung untersucht und verglichen. Im Rahmen der Untersuchungen der Domain Randomization wurde in den

2. Hintergrund

bisherigen Arbeiten Ablationstudien eingesetzt. Hierbei wurden zwar der Beitrag einer randomisierten Eigenschaft zum Gesamtergebnis ausgewertet, jedoch weder verschiedene Randomisierungsgrade noch die Kombinationen der einzelnen Parameter betrachtet. Eine Untersuchung des Szenenkontextes beschränkt sich auf Objekteinbettungs- und Positionierungsstrategien [43, 2, 82, 58]. Eine systematische Untersuchung des Einflusses relevanter Szeneigenschaften, welche insbesondere durch Scene Engineering reproduziert werden, existiert nach bestem Wissen nicht.

Viele der betrachteten Arbeiten verwenden den Begriff „signifikant“ zur Beschreibung ihrer Messwerte. Da jedoch überwiegend keine Versuchswiederholungen berichtet werden, ist davon auszugehen, dass der Begriff nicht als Terminus der induktiven Statistik, sondern eher umgangssprachlich im Sinne von „deutlich“ verwendet wird. Ausnahmen bilden eine Arbeit, bei welcher Hypothesentests angewendet und p-Werte angegeben wurden [6], sowie Arbeiten, bei welchen zumindest Standardabweichungen mit angegeben wurden [6, 71, 23, 41]. Andere Arbeiten führen zwar Versuchswiederholungen durch, mitteln diese jedoch und geben keine zugehörige Varianz an [3, 2, 33, 127, 83, 68].

Insbesondere durch die folgenden vier Aspekte unterscheidet sich die vorliegende Arbeit von der bisherigen Forschung im untersuchten Forschungsfeld:

1. Industrielle Anwendungsfälle stehen im Fokus der Untersuchungen. Dabei werden die zugehörigen Randbedingungen und Besonderheiten explizit betrachtet und die Relevanz der korrespondierenden Szeneigenschaften untersucht. Zudem werden die manuelle und die automatisierte Optimierung der Simulationsszene evaluiert.
2. Domain Adaptation und Domain Randomization werden sowohl getrennt als auch in Kombination untersucht. Dabei werden unterschiedliche Adaptionmethoden sowie Randomisierungseigenschaften und -grade betrachtet.
3. Die statistische Absicherung der Messergebnisse wird analysiert. Dazu werden die Versuche wiederholt durchgeführt, Konfidenzintervalle ermittelt und Hypothesentests zur Bestimmung der Signifikanz angewandt.
4. Das Training mit synthetischen Bilddaten wird für mehrere Anwendungsfälle mit realen Trainingsdaten verglichen. Die Untersuchungen erfolgen mit mehreren realen Evaluierungsdatensätzen, die unterschiedlich stark vom jeweiligen realen Trainingsdatensatz abweichen.

3. Theoretische Grundlagen

Im Folgenden werden die für diese Arbeit relevanten theoretischen Grundlagen zur Bildverarbeitung, zum Deep Learning und zur Bayes'schen Optimierung, zu Hypothesentests und Konfidenzintervallen sowie zu den eingesetzten Bewertungsverfahren und Metriken gelegt.

3.1. Bildverarbeitung

Im Folgenden werden die für diese Arbeit relevanten Grundlagen der Bildverarbeitung zu Farbräumen und zur Faltung von Matrizen erläutert.

3.1.1. Farbräume

Bilddaten können durch verschiedene Farbräume und damit unterschiedliche Kanalrepräsentationen dargestellt werden. Diese können in der Regel durch Transformationen ineinander überführt werden, wobei die Umwandlung häufig mit Informationsverlusten durch Diskretisierung sowie Präzisionsverlusten einhergeht. Im Rahmen dieser Arbeit werden die folgenden Farbräume eingesetzt:

1. RGB: Der RGB-Farbraum ist ein additiver Farbraum, der auf den drei Grundfarben Rot, Grün und Blau basiert. Er stellt einen De-facto-Standard bei der Speicherung und Verarbeitung von Bilddaten dar. Die Farbkanäle R , G und B sind dabei im Intervall $[0, 1]$ definiert, wobei sich für $R = G = B$ Grauwerte ergeben.
2. HSV: Beim HSV-Farbraum werden die Farben über den Farbwert H (engl. hue), die Sättigung S (engl. saturation) und den Hellwert bzw. die Helligkeit V (engl. value) definiert. Während für Sättigung S und Hellwert V Werte zwischen 0 % und 100 % verwendet werden, ist der Farbwert H als Farbwinkel definiert. Dabei stehen 0° für Rot, 120° für Grün und 240° für Blau, wobei gilt $360^\circ \equiv 0^\circ$. Der

3. Theoretische Grundlagen

HSV-Farbraum wird im Rahmen dieser Arbeit unter anderem bei der Farb-Augmentation eingesetzt (s. Unterabschnitt 4.2.4).

3. $L^*a^*b^*$ (CIELAB): Beim $L^*a^*b^*$ -Farbraum werden die Farben durch den Helligkeitswert L^* und die Farbebene (a^*, b^*) definiert. Die Koordinaten a^* und b^* geben jeweils die Farbart und Farbintensität zwischen Grün ($a^* < 0$) und Rot ($a^* > 0$) bzw. Blau ($b^* < 0$) und Gelb ($b^* > 0$) an. Hohe Werte $|a^*|$ und $|b^*|$ repräsentieren dabei intensivere Farbtöne, wobei sich für $a^* = 0$ und $b^* = 0$ Grauwerte ergeben. Der $L^*a^*b^*$ -Farbraum wird im Rahmen dieser Arbeit für die statistische Bildanpassung (ICT) genutzt (s. Unterabschnitt 4.3.1).

Im Rahmen der durchgeführten Experimente liegen die Bilder als RGB-Daten in einer 8-Bit-Repräsentation (je Kanal) vor, wobei die Bildpunkte, auch als Pixel bezeichnet, ganzzahlige Werte $v \in \{x \in \mathbb{N} \mid 0 \leq x \leq 255\}$ aufweisen. Verschiedene Bildtransformationen (insb. Augmentation oder Adaption) finden dagegen in einer 16-Bit-Fließkomma-Repräsentation (je Kanal) mit Pixelwerten $v' \in \{x \in \mathbb{R} \mid 0 \leq x \leq 1\}$ und teilweise in anderen Farbräumen statt. Die transformierten Bilddaten werden jedoch vor der Speicherung oder Weiterverarbeitung, insbesondere für das Training der neuronalen Netze, durch Diskretisierung und Intervall-Begrenzung wieder in die RGB-8-Bit-Repräsentation überführt.

3.1.2. Faltung

Faltung oder Konvolution (engl. convolution) beschreibt einen mathematischen Operator, der für zwei Funktionen das Integral des Produktes über mehrere relative Verschiebungen der Funktionen berechnet. Dabei findet eine gewichtete Mittelwertbildung statt. In der Bildverarbeitung wird die diskrete Form der Faltung von Matrizen eingesetzt. Hierbei wird ein Bild bzw. der Farbkanal eines Bildes als Eingabematrix I mit einer Faltungsmatrix K , auch als Filterkern oder Kernel bezeichnet, gefaltet. Beim Filterkern handelt es sich meist um eine quadratische Matrix mit ungerader Abmessung, welche in der Regel wesentlich kleiner als die Abmessung des Bildes ist – typische Größen sind 3×3 oder 5×5 .

Bei der Faltung wird der Filterkern iterativ und in beide Achsen über das Bild I bewegt. An jeder Position wird die Summe des Matrixproduktes der beiden Matrizen berechnet und an dieser Position in die Ergebnismatrix I' geschrieben. Dadurch wird jeder Wert in I durch das mit K gewichtete

Mittel der ihn räumlich umgebenden Werte ersetzt. Die Faltung einer Matrix $I \in \mathbb{R}^{w \times h}$ mit einem Filterkern $K \in \mathbb{R}^{m \times n}$ ist definiert als:

$$I'(x, y) = \sum_{i=1}^n \sum_{j=1}^m K(i, j) I\left(x - i + \frac{m-1}{2}, y - j + \frac{n-1}{2}\right) \quad (3.1)$$

Da die kleinere Faltungsmatrix bei einer Positionierung in den Randbereichen über die größere Eingabematrix I hinausragt, werden diese Bereiche typischerweise entweder ausgelassen oder aufgefüllt. Wird der Randbereich ausgelassen (engl. *valid convolution*), reduziert sich die Größe der Ergebnismatrix I' im Vergleich zu I . Wird I dagegen an den Rändern so aufgefüllt, dass jede Position in I erreicht wird (engl. *same convolution*), stimmen die Größen von I und I' überein. Die Auffüllung (engl. *padding*) geschieht üblicherweise mittels einer Reflexion der Randwerte von I oder mit konstanten Werten (z. B. Nullen). Die räumliche Größe des Filterkerns K , die Schrittweite der Filterverschiebung (engl. *stride*) und der Umgang mit den Randbereichen sind werden dabei fix definiert. Zusammen mit der Abmessung der Eingabematrix I bestimmen sie die Größe der Ergebnismatrix I' .

Die Faltung wird auf vielseitige Weise in der klassischen und modernen Bildverarbeitung eingesetzt, beispielsweise beim Weichzeichnen von Bildern (s. Unterabschnitt 4.2.7), bei der Hervorhebung von Kanten im Bild oder auch bei faltungsbasierten neuronalen Netzen (s. Unterabschnitt 3.2.2). Die Art des Filters und seine Transformationseigenschaften werden über die Filterparameter, also die Werte des Filterkerns, bestimmt.

3.2. Deep Learning

Das Training und die Ausführung von tiefen neuronalen Netzen (engl. *deep learning*) stellt einen Kernaspekt der vorliegenden Arbeit dar. Im Folgenden werden entsprechend die notwendigen Grundlagen zu neuronalen Netzen und zugehörigen Meta-Architekturen in der Bildverarbeitung geschaffen. Tieferer Einblicke erhalten interessierte Leser beispielsweise in *Goodfellow et al.* [36] oder *Chollet* [14].

3.2.1. Neuronale Netze

Ein neuronales Netz besteht aus Hunderten, Tausenden oder sogar Millionen von Neuronen, welche in aufeinander folgenden Schichten miteinander verbunden sind. Die vorab definierte Architektur entscheidet darüber, welche Art von Neuronen verwendet werden, wie die Schichten aufgebaut sind und welche Verbindungen zwischen den Schichten bestehen.

Die erste Ebene ist die sogenannte Eingabeschicht, welche die externen Eingaben (bspw. die Pixelwerte eines Bildes) empfängt. Die letzte Schicht ist die sogenannte Ausgabeschicht, an welcher das Ergebnis der Inferenz ausgelesen werden kann. Zwischen der Eingabe- und der Ausgabeschicht befinden sich die sogenannten versteckten Schichten, welche für den Nutzer in der Regel eine Black-Box darstellen. Insbesondere die Anzahl, die Größe und die Art der verborgenen Schichten bestimmen den Typ, die Fähigkeiten und die Leistungsfähigkeit eines neuronalen Netzes. Falls jedes Neuron einer Schicht mit jedem Neuron der vorherigen und der folgenden Schicht verbunden ist, wird die Schicht als dicht (engl. dense) oder vollständig verbunden (engl. fully connected) bezeichnet.

Neuronen und Aktivierungen: Neuronen sind die Grundbausteine neuronaler Netze. Ein Neuron empfängt, modifiziert und leitet Signale durch das Netzwerk. Diese Signale repräsentieren Wahrscheinlichkeiten des Vorhandenseins bzw. der Ausprägung von bestimmten Merkmalen. In der Bildverarbeitung kann es sich bei den Merkmalen beispielsweise um Helligkeits- und Farbverläufe, Kanten und Formen, Komponenten oder ganze Objekte handeln. Mit der Weiterleitung der Signale in spätere Schichten erhöht sich in der Regel der Informationsgehalt der Repräsentationen und die Signale werden zunehmend spezifischer.

Jedes Neuron empfängt n Signale oder Eingaben x_i von allen Neuronen der vorangehenden verbundenen Schicht. Diese Signale werden gewichtet, indem sie mit den während des Trainings erlernten Gewichten $w_{i,j}$ multipliziert und anschließend summiert werden. Zur Summe der gewichteten Signale wird ein zusätzlicher, erlernbarer Wert b_j addiert. Dieser wird als Bias bezeichnet und stellt einen eingabeunabhängigen Schwellenwert dar. Durch die Anwendung der Aktivierungsfunktion γ auf das Signal ergibt sich der endgültige Aktivierungswert o_j des Neurons, welcher an alle verbundenen Neuronen der nachfolgenden Schicht weitergegeben wird:

$$o_j = \gamma \sum_{i=0}^n (w_{i,j} x_i) + b_j \quad (3.2)$$

Aktivierungsfunktionen definieren die Neuronenausgabe bei einer Reihe von Eingaben. Nur nichtlineare Aktivierungsfunktionen ermöglichen es neuronalen Netzen, „nichttriviale Probleme“ mit nur wenigen Neuronen zu berechnen. Solche nichtlinearen Aktivierungsfunktionen werden Nichtlinearitäten genannt. In neuronalen Netzen sind die am häufigsten genutzten Aktivierungsfunktionen meist entweder Grat- (bspw. Sigmoid oder ReLU) oder nichtlineare Aggregatfunktionen (bspw. Softmax oder Max-Pooling).

Verlust, Gradientenverfahren und Fehlerrückführung: Beim überwachten Lernen überführt die Verlustfunktion die Ausgabe des neuronalen Netzes über den Abgleich mit der vorgegebenen Zielausgabe in einen Skalar. Dieser Wert wird als Verlust bezeichnet und repräsentiert die Fehlerhaftigkeit der aktuellen Ausgabe und damit der Entscheidung des neuronalen Netzes. Die Zielausgaben gehen hierbei aus der annotierten Ground-Truth hervor. Beispiele für Verlustfunktionen sind die mittlere quadratische Abweichung (engl. mean squared error) oder die Kreuzentropie (engl. cross-entropy).

Das Training neuronaler Netze ist rechenintensiv und nimmt daher viel Zeit in Anspruch. Aufgrund von Hardware-Einschränkungen konnten neuronale Netze über viele Jahre hinweg nur wenige Neuronen umfassen, wodurch diese für viele komplexe Aufgaben ungeeignet waren. Neben der Verbesserung der Hardware spielte insbesondere die Entwicklung des Verfahrens zur Fehlerrückführung (engl. backpropagation) eine zentrale Rolle auf dem Weg zu tieferen neuronalen Netzen. Diese ermöglicht über die Verkettung der partiellen Ableitungen eine gezielte Aktualisierung der einzelnen Parameter des neuronalen Netzes und verkürzt damit den Lernprozess erheblich. Dabei wird vorausgesetzt, dass die Ableitungen jedes einzelnen Neurons und der zugehörigen Aktivierungsfunktionen bekannt sind.

Für die Optimierung der Parameter eines neuronalen Netzes wird das Gradientenverfahren eingesetzt. Hierbei werden für eine oder mehrere (engl. batch) Eingaben der (mittlere) Verlust sowie die zugehörigen Gradienten aller Parameter mithilfe der Fehlerrückführung ermittelt. Die Anzahl der Datenpunkte, deren Verluste für eine Parameteranpassung gemittelt werden, wird als Batch Size bezeichnet. Da es das Ziel der Optimierung ist, den Verlust zu verringern, erfolgt die Anpassung der Parameter entgegen den ermittelten Gradienten (negative Richtung). Aufgrund der negativen Ab-

3. Theoretische Grundlagen

stiegsrichtung wird hierbei auch vom Verfahren des steilsten Abstiegs oder vom Gradientenabstieg (engl. gradient descent) gesprochen. Zusätzlich wird der Grad der Anpassung durch die Multiplikation des negativen Gradienten mit der Lernrate (engl. learning rate) reduziert. Aufgrund dieser Reduktion durch die Lernrate ist es in der Regel sinnvoll, die Datenpunkte jeweils mehrfach für Parameteranpassungen zu nutzen. Hierbei wird eine Iteration des vollen Datensatzes, wobei jeder Datenpunkt exakt einmal inferiert wird, als Trainingsepoche oder kurz Epoche bezeichnet.

3.2.2. Convolutional Neural Network

Die im Rahmen dieser Arbeit untersuchten Objekterkennungsverfahren sind Convolutional Neural Networks (CNNs). Im Vergleich zu vollständig verbundenen neuronalen Netzen werden bei diesen faltungsbasierten Netzen zusätzliche Typen von Schichten eingesetzt. Namensgebend sind hierbei die Faltungsschichten (engl. convolutinal layer), welche die Signale mittels Faltungsoperator verarbeiten (s. Unterabschnitt 3.1.2). Die Parameterwerte der zugehörigen Filter stellen dabei die erlernten Gewichte dar. Da diese Filter räumlich unabhängig eingesetzt und wiederverwendet werden, werden die Gewichte nicht für alle Bildbereiche separat erlernt. Da sich die relevanten Merkmale an vielen Stellen des Bildes befinden können, ist diese Form der Wiederverwendung sinnvoll und vereinfacht den Lernprozess durch die Reduktion der zu erlernenden Parameter enorm. Da sich die Bildgröße bei der Matrixfaltung im CNN, je nach Umgang mit den Randbereichen, nicht oder nur geringfügig reduziert, werden außerdem sogenannte Pooling-Schichten genutzt. Diese stellen Methoden der Bildgrößenreduktion auf Basis von Maximum- oder Mittelwertsfunktionen dar und enthalten keine erlernbaren Parameter. Mithilfe des Poolings wird ein ähnlicher Effekt der Informationskonzentration erreicht wie bei der Reduktion der Neuronenanzahl in späteren Schichten vollverbundener neuronaler Netze.

Die entwickelten CNN-Basis-Architekturen für Bildverarbeitung dienen in der Regel der Bildklassifikation, bei welcher einem Bild eine einzige Klasse zugewiesen wird. Exemplarisch werden im Folgenden einige der relevantesten Meilensteine in der Evolution der CNNs dargestellt. Das 1989 vorgestellte LeNet stellt eines der ersten veröffentlichten CNNs dar. Es wurde von einer Forschergruppe um Yann LeCun zur Klassifikation handgeschriebener Ziffern (MNIST) eingesetzt und über mehrere Jahre bis 1998 (LeNet-5) weiterentwickelt. Die erste Architektur, welche weltweite Aufmerksamkeit als konkurrenzfähiger Ansatz zur klassischen Bildverarbeitung auf sich

ziehen konnte, war das von Alexander Krizhevsky entwickelte AlexNet. Diese konnte 2012 als erstes Deep Learning-Verfahren die ImageNet Large Scale Visual Recognition Challenge (ILSVRC) gewinnen.

Mit zunehmender Rechenleistung wurden immer tiefere Netze mit mehr hintereinander geschalteten Schichten entwickelt, beispielsweise das 2014 veröffentlichte VGG mit 16 bzw. 19 Schichten [104] oder die 2015 veröffentlichte Inception-Architektur mit 22 Schichten [109, 111]. Eine weitere bedeutende Entwicklung stellt 2015 die Einführung von sogenannten residualen Verbindungen (engl. residual connections) in ResNet dar [41]. Durch diese Skip-Verbindungen können Signale auch mehrere Schichten überspringen. Da hierdurch dem Problem der verschwindenden Gradienten (engl. vanishing gradients) bei der Fehlerrückführung entgegengewirkt wird, ermöglichen diese Verbindungen ein effektives Training auch bei tiefen neuronalen Netzen.

In den Folgejahren war der Fortschritt im Bereich der CNNs primär durch die manuelle Suche nach neuen sogenannten Blöcken, also sich wiederholenden Kombinationen mehrerer Schichten, geprägt. Diese Entwicklung führte zu einer immer höheren Performanz der CNNs auf den großen Standarddatensätzen wie ImageNet, wobei als prominente Architekturen insbesondere Inception-ResNet [110], MobileNet für schwächere Hardware [48], ResNext [122] und Xception [15] zu nennen sind. Ab dem Jahr 2017 wurden zunehmend automatisierte Suchverfahren (engl. neural architecture search, NAS) für die Entwicklung neuer Blöcke und ganzer Architekturen eingesetzt, wobei NASNet eine der bekannteren NAS-Architekturen darstellt [128].

Nach der Veröffentlichung von Vision Transformer (ViT) [23] im Jahr 2020 verlagerte sich die Forschung im Bereich des Deep Learning für Bildverarbeitung zunehmend auf Transformer-Architekturen. ViT erzielte bessere Ergebnisse auf dem ImageNet-Datensatz als bisherige CNNs, obwohl die Transformer-Architektur ursprünglich für die Sprach- und Textverarbeitung entwickelt und eingesetzt wurde. Ein großer Nachteil der Transformer-Modelle ist deren zeit-, daten- und ressourcenintensives Training. So konnte das Team um Dosovitskiy ViT nur unter Verwendung eines ungewöhnlich großen (150 Mio. Daten) Google-internen Datensatzes erfolgreich trainieren. Spätere Veröffentlichungen fokussierten sich unter anderem auf die Reduktion des Datenbedarfs durch Destillation (engl. distillation) [115] und die Erhöhung der Performanz der Transformer-Modelle für Bildverarbeitungsaufgaben durch die Integration von Prinzipien der CNNs (bspw. Fenster-Operationen) [66]. 2022 konnten Liu *et al.* mit der Entwicklung von ConvNeXt [67] zeigen, dass sich die modernen Trainings- und Regu-

larisierungskonzepte der Transformer-Modelle auch erfolgreich auf CNNs übertragen lassen, wodurch sie signifikante Verbesserungen der Performanz auf das Niveau der Transformer-Architekturen erreichen konnten.

3.2.3. Meta-Architekturen für Objekterkennung

Bei den im Rahmen dieser Arbeit verwendeten neuronalen Netzen handelt es sich um Modelle für die Objekterkennung (engl. object detection) und Instanzsegmentierung (engl. instance segmentation). Diese beiden Aufgaben unterscheiden sich in der Art der Lokalisierung der Objekte: entweder über orthogonale Begrenzungsrahmen (engl. bounding box) um die Objekte (Objekterkennung) oder über Segmentierungen auf Bildpunktebene, als Masken bezeichnet (Instanzsegmentierung). Der Begriff der Objekterkennung wird in der Literatur jedoch auch als Oberbegriff für beide Aufgaben genutzt. Für die Umsetzung der Objekterkennung mit neuronalen Netzen werden Meta-Architekturen eingesetzt, welche in einem Bild für viele potenzielle Objektinstanzen Klassifizierungen und Lokalisierungen durchführen. Während Modell-Architekturen für die Bildklassifikation in der Regel den gesamten Bildverarbeitungsprozess von der Bildeingabe bis zur Klassifikationsentscheidung umfassen, klammern Meta-Architekturen die Merkmalsextraktion (engl. feature extraction) häufig aus. Stattdessen werden hierfür typischerweise etablierte und meist vortrainierte Klassifizierungsmodelle wie ResNet oder MobileNet eingesetzt, welche in diesem Kontext als Backbone-Modelle oder Merkmalsextraktoren bezeichnet werden. Die durch die Backbone-Modelle extrahierten Bildmerkmale (engl. feature maps) werden dabei in mehrere Detektionen überführt, indem für eine vordefinierte Menge an Anker-Boxen mit unterschiedlicher Position und Größe (Raster) Klassifikationsentscheidungen abgeleitet werden. Zusätzlich werden die initial grobe Lokalisierung der Anker-Boxen über eine Versatz-Regression für jede Objektinstanz feinabgestimmt und sich überschneidende Mehrfacherkennungen durch einen Vergleich der Aktivierungswerte unterdrückt (engl. non-max suppression). Beispiele für solche Meta-Architekturen sind SSD [65] und R-CNN [35, 34, 89, 42]. Andere Architekturen wie You Only Look Once (YOLO) [87] und seine Weiterentwicklungen [85, 86] bilden hierbei eine Ausnahme, da sie integrierte Gesamtarchitekturen inklusive Merkmalsextraktion darstellen.

Single Shot MultiBox Detector (SSD) ist ein einstufiger Detektor, der 2016 von Liu *et al.* entwickelt wurde. Einstufige Detektoren führen die Objekterkennung mithilfe einer einzelnen (z. B. YOLO) oder mehrerer (z. B. SSD) Feature-Maps in einem einzigen Schritt durch. Um mehrere

Feature-Maps abzuleiten, nutzt SSD am Ende des verkürzten Backbone-Netzes einige zusätzliche Faltungsschichten zur Merkmalsextraktion. Für jeden Anker wird ein Klassenwert einschließlich der Hintergrundklasse und ein relativer Versatz zur Anker-Box vorhergesagt. Einstufige Detektoren haben gegenüber zweistufigen Verfahren klare Vorteile in Bezug auf die Trainings- und Inferenzgeschwindigkeit. Jedoch weisen sie Nachteile in Bezug auf die erreichte Performanz auf, welche insbesondere in der starken Unausgewogenheit der Klassen beim einstufigen Entscheidungsprozess begründet liegt. Während pro Bild nur wenige positive Beispiele (tatsächliche Objektinstanzen) pro Klasse vorliegen, existieren abhängig von der Anzahl der Anker-Boxen sehr viele negative Beispiele (Hintergrund). Da jedoch hauptsächlich die positiven Beispiele für das Erlernen der Klassifikationsentscheidung relevant sind, führt diese Unausgewogenheit zu einer schlechteren Konvergenz des Optimierungsprozesses und damit in der Regel zu einer niedrigeren Performanz.

Diesem Problem versucht der Ansatz der zweistufigen Detektoren, wie beispielsweise das Region-based Convolutional Neural Network (R-CNN), entgegenzuwirken. Hierbei werden die Schritte der Hintergrunderkennung und der Objektklassifikation getrennt, indem für jede Anker-Box zunächst eine klassenunabhängige (engl. class agnostic) Entscheidung über Objekt oder Hintergrund durchgeführt wird. Auf Basis dieser Vorschläge über relevante Bildregionen (engl. region proposal) finden anschließend die eigentlichen Klassifizierungen der Objektinstanzen statt. Diese Trennung reduziert die Anzahl der negativen Beispiele im endgültigen Entscheidungsprozess erheblich. Die Anzahl der Klassifikationsentscheidungen im ersten Schritt bleibt abhängig von der Anzahl der Anker-Boxen hoch, jedoch stehen aufgrund der Klassenunabhängigkeit mehr positive Beispiele zur Verfügung und die Komplexität der Aufgabe reduziert sich auf die einfachere zu erlernende Trennung in Objekt und Hintergrund. Die Aufteilung in zwei Schritte verbessert in der Regel die Performanz der Objekterkennung, geht jedoch zu Lasten der Trainings- und Inferenzgeschwindigkeit.

Die wichtigsten Meilensteine in der Entwicklung der R-CNN-Architektur sowie die zugehörigen Anpassungen werden im Folgenden kurz dargestellt:

2013 R-CNN [35]: Selektive Suche zur Vorhersage von Regionen von Interesse (RoI) und separate Inferenz des neuronalen Netzes für jeden Regionsvorschlag.

3. Theoretische Grundlagen

- 2015 Fast R-CNN [34]: Einmalige Inferenz des neuronalen Netzes für das gesamte Bild und anschließendes Feature-Map-Slicing für jeden Regionvorschlag (engl. RoI pooling).
- 2015 Faster R-CNN [89]: Integration der Generierung von Regionenvorschlägen in das neuronale Netz statt des separaten Suchprozesses.
- 2017 Mask R-CNN [42]: Integration eines zusätzlichen Zweiges für die Maskenregression und Verbesserung des Feature-Map-Slicings (RoI align), wodurch eine pixelweise Instanzsegmentierung möglich wird.

Im Rahmen der Experimente dieser Arbeit werden hauptsächlich die Architekturen Faster R-CNN und Mask R-CNN, jeweils mit einem Inception-ResNet-V2 als Merkmalsextraktor, eingesetzt. Für die Überprüfung der Übertragbarkeit werden darüber hinaus spezifische Versuche mit der Architektur SSD MobileNet-V2 durchgeführt.

3.2.4. Generative Adversarial Network

Das Generative Adversarial Network (GAN) ist ein Rahmenwerk bzw. Trainingskonzept im Kontext des generativen maschinellen Lernens, welches 2014 von Goodfellow *et al.* veröffentlicht wurde [37]. Es kombiniert zwei gegeneinander arbeitende tiefe neuronale Netze, den Generator G und den Diskriminator D . Ziel des Trainings ist es, den Generator so zu optimieren, dass dieser für generative Bilderzeugungs- bzw. Bildtransformationaufgaben eingesetzt werden kann. Anwendung finden GANs unter anderem bei der Erzeugung menschlicher Gesichter (z. B. Deep Fakes), der Bildvervollständigung, der Steigerung der Bildauflösung (Super-Resolution) oder beim Sim2Real-Transfer.

Über den Trainingsprozess lernt der Generator neue Daten X' aus einer zufälligen Eingabe Z (Rauschen) so zu generieren, dass diese zunehmend besser in die Domäne eines Referenzdatensatzes X passen. Der Diskriminator dagegen lernt, die generierten Daten immer besser von den Referenzdaten zu unterscheiden, d. h. eine Entscheidung d zu treffen, ob das Bild aus X oder X' stammt. Damit nimmt der Diskriminator gegenüber dem Generator die Rolle des Überwachers ein. Für ihre Experimente verwendeten Goodfellow für Generator und Diskriminator jeweils sowohl vollständig verbundene als auch faltungsbasierte neuronale Netze. Die GAN-Architektur ist in Abbildung 3.1 dargestellt.

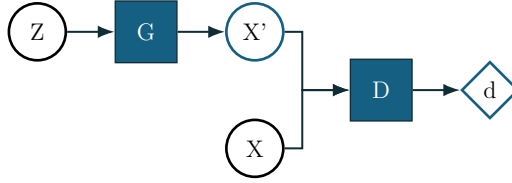


Abbildung 3.1.: GAN-Architektur mit Generator G , der aus Rauschen Z Bilder X' erzeugt und Diskriminator D , der die Entscheidung d trifft, ob Bilder aus der Referenzdomäne X stammen oder erzeugt wurden (X').

Das Training ist von der Idee des Nash-Gleichgewichts inspiriert und wird über eine gemeinsame Backpropagation durchgeführt. Für zufällige Eingaben (Rauschen) $z \sim p_z(z)$ und Referenzdaten $x \sim p_{data}(x)$ ist das Min-Max-Spiel des Trainingsprozesses wie folgt definiert:

$$\min_D \max_G V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (3.3)$$

Neben der Generierung zufälliger neuer Daten, werden GANs auch für die Bild-zu-Bild-Transformation eingesetzt, wie beispielsweise zur Erhöhung der Bildauflösung (engl. super resolution) oder bei der Stilübertragung. Im Gegensatz zum ursprünglichen GAN muss der Generator hierbei lernen, die grundlegenden Merkmale, welche Quell- und Zieldomäne gemeinsam haben, beizubehalten und lediglich die spezifischen Merkmale zu verändern, in welchen sich die Quell- und die Zieldomäne unterscheiden. Die ersten erfolgreichen Ansätze im Bereich der Bild-zu-Bild-Transformation nutzten eine direkte Paarung von Beispielen aus der Quell- und Zieldomäne. Ein prominentes Beispiel stellt hierbei die Pix2Pix-Architektur dar [52]. Hierbei setzen Isola *et al.* ein konditionales GAN (cGAN) [72] mit einem U-Net-Generator [93] und einem von ihnen als PatchGAN bezeichneten Diskriminator ein. Der PatchGAN-Diskriminator nimmt die Bewertung der Bilder weder auf Ebene einzelner Pixel noch auf Ebene des Gesamtbildes vor, sondern teilt das Bild zur Bewertung in mehrere Bereiche auf. Anhand der Bildpaarung kann im Trainingsprozess sichergestellt werden, dass der Generator lernt, lediglich die domänenspezifischen Merkmale zu verändern und die allgemeinen Bildinformationen beizubehalten. Ohne diese Zuordnung wäre es für

3. Theoretische Grundlagen

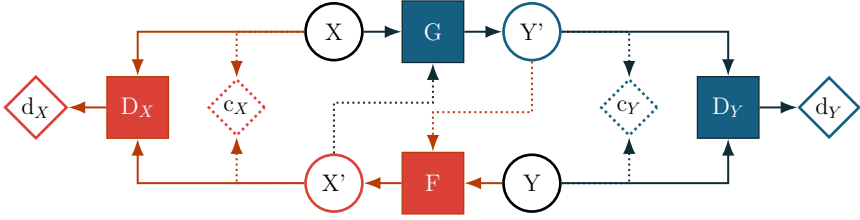


Abbildung 3.2.: CycleGAN-Architektur bestehend aus der Vorwärtstransformation mit Generator G und Diskriminator D_y (blau), der Rücktransformation mit Generator F und Diskriminator D_x (rot) sowie den Überprüfungen der Zykluskonsistenz c_x und c_y (gepunktet).

den Generator eine valide Lösung, die Informationen des Eingabebildes zu ignorieren und davon vollkommen unabhängige Bilder zu generieren.

Insbesondere im Bereich der Stilübertragung existieren jedoch häufig keine gepaarten Trainingsdaten. CycleGAN stellt eine der relevantesten Veröffentlichungen im Bereich der Stilübertragung ohne gepaarte Trainingsdaten dar [125]. Dabei schlugen Zhu *et al.* einen völlig neuen Ansatz vor, um die grundlegenden Bildinformationen bei der Transformation zu erhalten. CycleGAN erweitert das ursprüngliche GAN-Rahmenwerk um eine zyklische Rücktransformation der transformierten Daten Y in die Quelldomäne X . Statt zwei werden hierbei vier neuronale Netze, jeweils ein Generator und ein Diskriminator für die Hin- und für die Rücktransformation, gemeinsam trainiert (s. Abbildung 3.2). Für CycleGAN verwenden Zhu *et al.* einen ResNet-Generator [41] und einen PatchGAN-Diskriminator.

Die iterative Anwendung der Generatoren $G : X \rightarrow Y$ und $F : Y \rightarrow X$ erzeugen die zyklischen Transformationen $x \rightarrow G(x) \rightarrow F(G(x)) \approx x$ und $y \rightarrow F(y) \rightarrow G(F(y)) \approx y$, welche auf Konsistenz bzw. Übereinstimmung mit der ursprünglichen Eingabe überprüft werden. Zum Verlust des GAN der Originalveröffentlichung kommt der sogenannte Zykluskonsistenz-Verlust (engl. cycle consistency loss) hinzu, welcher wie folgt definiert ist:

$$L_{cyc}(G, F) = \mathbb{E}_{x \sim p_{data}(x)} [\|F(G(x)) - x\|_1] + \mathbb{E}_{y \sim p_{data}(y)} [\|G(F(y)) - y\|_1] \quad (3.4)$$

Viele nachfolgende Veröffentlichungen im Bereich der Stilübertragung basieren auf der CycleGAN-Architektur und erweitern das Konzept. So

ergänzte beispielsweise AttentionGAN [112] einen Aufmerksamkeitsverlust (engl. attention loss), welcher das Training auf besonders relevante Bildbereiche fokussiert und dadurch detailreichere Transformationen ermöglicht.

Trotz der subjektiv beeindruckenden Ergebnisse bei der Generierung und Transformation von Bilddaten weisen GANs verschiedene Instabilitäten im Trainingsprozess auf. Zu diesen zählen unter anderem:

1. Verschwindende Gradienten (engl. vanishing gradients): Häufig ist die Aufgabe des Diskriminators wesentlich einfacher zu erlernen als die des Generators. Da sich der Diskriminator dadurch im Verlauf des Trainingsprozesses zu schnell in der Unterscheidung der Daten verbessert, kann der Generator nicht mehr ausreichend aus dessen Entscheidungen lernen. Dies liegt darin begründet, dass die Gradienten eines optimalen Diskriminators gegen null gehen und somit kein weiteres Verbesserungspotenzial für die Optimierung des Generators mittels Gradientenverfahren bieten.
2. Moduszusammenbruch (engl. mode collapse): Statt anhängig von der Eingabe (Rauschen bzw. Eingabebild) eine große Vielfalt an Ausgaben zu erzeugen, kommt es beim GAN-Training vor, dass der Generator in einem lokalen Optimum verharret. Dabei wird dieser auf eine Ausgabe überoptimiert, welche der Diskriminator nicht von der Referenzdomäne unterscheiden kann. Dadurch erzeugt der Generator immer wieder eine sehr ähnliche Ausgabe und ignoriert die Informationen der Eingabe. Dieses Problem ist beim CycleGAN-Verfahren wesentlich geringer, da das System die ursprüngliche Eingabe aus der Transformation rekonstruieren muss und die Eingabeinformationen daher nicht komplett vernachlässigen kann. Dennoch kommt es auch hier vor, dass die notwendigen Informationen so kompakt im transformierten Bild kodiert sind, dass sich die Ausgaben des Generators kaum unterscheiden.
3. Annotationsinkonsistenz (engl. annotation inconsistency): Existieren zu den Bildern zugehörige Annotationen, kann es durch die Transformation zu Annotationsinkonsistenzen, auch als semantische Inkonsistenzen bezeichnet, kommen. Hierbei stimmen die transformierten Bilder semantisch nicht mehr mit den ursprünglichen Annotationen überein. Bei der Objekterkennung kann dies insbesondere das Vorhandensein einer Objektinstanz (Instanz entfernt oder hinzugefügt), deren Klasse (Erscheinung klassifikationsrelevant verändert) oder häufiger deren Lokalisierung (Instanz verschoben oder verformt) betreffen.

3. Theoretische Grundlagen

Existierende Verbesserungsansätze betreffen insbesondere das Erhalten des (Verlust-)Gleichgewichts beim Training von Generator und Diskriminator. So werden unter anderem die Verwendung anderer Verlustberechnungen auf Basis der Wasserstein-Metrik [4], die direkte Begrenzung des Diskriminator-Verlustes [5] oder die Bestrafung hoher Gradienten (engl. gradient penalty) des Diskriminator-Verlustes [39] vorgeschlagen. Hierdurch soll die Konvergenz des Diskriminators verlangsamt und damit das Problem der verschwindenden Gradienten im weiteren Trainingsprozess verhindert werden. Um Annotationsinkonsistenzen zu vermeiden, wurden Architekturen entwickelt, welche die zugehörigen Annotationen mitverarbeiten, um darauf basierend zusätzliche Verlustfunktionen abzuleiten [73, 102, 60, 106]. Dadurch kann die semantische Konsistenz der transformierten Bilder verbessert werden. Teilweise nutzen die Verfahren auch zusätzliche Prediktoren, welche teilweise mit externen Daten trainiert werden, um Pseudo-Annotationen zu generieren [12, 13, 9]. Nachteile sind, dass die Realdaten für die Sim2Real-Übertragung in beiden Fällen manuell zu annotieren sind und dass in letzterem Fall vortrainierte Prediktoren für die Zieldomäne benötigt werden. Einen alternativen Ansatz, welcher keine annotierten Realdaten benötigt, stellt die starke räumliche Randomisierung der Trainingsdaten dar [120], da der Einfluss der Lokalisierung auf den Zykluskonsistenz-Verlust hierdurch verstärkt wird.

3.3. Bayes'sche Optimierung

Die Bayes'sche Optimierung ist ein sequentielles globales Optimierungsverfahren für unbekannte Zielfunktionen, sogenannte Black-Box-Funktionen, über deren Struktur keine Kenntnis existiert, welche jedoch auswertbar sind. Im Vergleich zu anderen Optimierungsverfahren werden hierbei weder funktionale Annahmen noch Ableitungsfunktionen benötigt. Die Bayes'sche Optimierung wird bei aufwendig zu evaluierenden Zielfunktionen mit weniger als 20 Freiheitsgraden eingesetzt, beispielsweise bei der Hyperparameteroptimierung für tiefe neuronale Netze. Im Rahmen der vorliegenden Arbeit wird die Methode der Bayes'schen Optimierung zur automatisierten Optimierung von Simulationsparametern eingesetzt (s. Unterabschnitt 4.1.3). Tiefere Einblicke erhalten interessierte Leser beispielsweise in *Nguyen* [75] oder *Garnett* [30].

Die Grundidee der Bayes'schen Optimierung ist die Verwendung einer probabilistischen Ersatzfunktion für die unbekannte Zielfunktion, auch als

Surrogatmodell bezeichnet. Diese enthält zwar die bisher gewonnenen Erkenntnisse über die Zielfunktion, ist jedoch wesentlich einfacher auszuwerten. Das Surrogatmodell wird in jedem Optimierungsschritt mit den neu evaluierten Datenpunkten angepasst und dient der iterativen Ermittlung der jeweils nächsten zu evaluierenden Parameter. Mithilfe der Akquisitionsfunktion findet ein Abwägen zwischen der lokalen Maximierung der Zielfunktion in bekannten Regionen des Suchraums (Exploitation) und der Erkundung bisher weniger untersuchter Regionen des Suchraums statt (Exploration). Durch die Verwendung des Surrogatmodells zur Bestimmung der nächsten zu evaluierenden Punkte im Parameterraum gilt die Bayes'sche Optimierung in Bezug auf die Anzahl der ausgewerteten Datenpunkte als effizienter als andere Optimierungsverfahren für unbekannte Zielfunktionen.

Die Bezeichnung Bayes'sche Optimierung liegt in der Ableitung des Posteriori-Modells aus dem A-priori-Modell begründet, welche auf dem Satz von Bayes für die Berechnung bedingter Wahrscheinlichkeiten basiert. [10]

In Bezug auf ein Surrogatmodell M und beobachtete Daten D mit unbekannter Wahrscheinlichkeit $P(D)$ gilt, dass die A-posteriori Wahrscheinlichkeit des Modells $P(M|D)$ proportional zum Produkt der Likelihood der beobachteten Daten $P(D)$, falls das Modell gilt $P(D|M)$ und der A-priori Wahrscheinlichkeit des Modells $P(M)$ ist:

$$P(M|D)P(D) = P(D|M)P(M) \quad (3.5)$$

bzw.

$$P(M|D) \propto P(D|M)P(M) \quad (3.6)$$

3.3.1. Surrogatmodell und Gauß-Prozesse

Das Surrogatmodell muss eine probabilistische Funktion sein, welche die Unsicherheit über unbekannte Bereiche im Suchraum abbilden kann und sich an die bisherigen Beobachtungen anpassen lässt. Typischerweise kommen hierbei Gauß-Prozesse zum Einsatz, jedoch ist auch die Verwendung anderer Verfahren wie beispielsweise Parzen-Tree Estimatoren oder Random Forests möglich. Gauß-Prozesse sind stochastische Prozesse, deren endlichdimensionalen Verteilungen multivariate Normalverteilungen darstellen. Damit können sie als Generalisierung der mehrdimensionalen Normalverteilung verstanden werden. Ein Gauß-Prozess wird über die Erwartungswertfunktion m und die Kovarianzfunktion k definiert:

$$f(x) \sim GP(m(x), k(x, x')) \quad (3.7)$$

$$p(f) = GP(f; \mu, K) \quad (3.8)$$

Ein Gauß-Prozess kann damit als eine Funktion verstanden werden, welche für einen Wert x eine mit Erwartungswert μ und Varianz σ^2 normalverteilte Zufallsvariable $N(\mu, \sigma^2)$ liefert. Im Verlauf des Optimierungsprozesses wird der Gauß-Prozess so an die beobachteten Datenpunkte angepasst, dass hohe Werte der Zielfunktion mit hohen Erwartungswerten und unbekannte Bereiche der Zielfunktion mit hohen Varianzen im Gauß-Prozess korrespondieren.

3.3.2. Akquisitionsfunktion

Über die Akquisitionsfunktion wird die Wahl der zu überprüfenden Parameterwerte anhand des aktuellen Surrogatmodells und damit der aktuellen Datenlage bestimmt – also der nächste Punkt, an dem die Zielfunktion ausgewertet werden soll. Sie leitet die Suche nach dem Optimum, indem hohe Werte der Akquisitionsfunktion potenziell mit hohen Werten der Zielfunktion (Exploitation) oder einer großen Reduktion der Unsicherheit über die Zielfunktion (Exploration) korrespondieren. Dadurch bildet sie den Trade-off zwischen Exploration (Unsicherheit reduzieren) und Exploitation (Zielfunktion maximieren) ab. Die Gewichtung der Exploration und Exploitation kann in der Regel über einen Hyperparameter gesteuert werden. Bei der Bayes'sche Optimierung werden hauptsächlich drei Akquisitionsfunktionen eingesetzt:

1. Die Methode der Verbesserungswahrscheinlichkeit, engl. probability of improvement (PI), wählt den Punkt, der die größte Wahrscheinlichkeit für eine Verbesserung der Zielfunktion gegenüber dem aktuell besten ermittelten Wert μ_{max} aufweist. Da dieses Vorgehen primär zur Exploitation führt, wird die Exploration des Parameterraums über den Hyperparameter $\xi \geq 0$ gesteuert:

$$PI(x) = \Phi \left(\frac{\mu(x) - \mu_{max} - \xi}{\sigma(x)} \right) \quad (3.9)$$

3.4. Hypothesentests und Konfidenzintervalle

2. Die Methode der erwarteten Verbesserung, engl. expected improvement (EI), berücksichtigt für die Wahl des nächsten Punktes nicht nur die Wahrscheinlichkeit der Verbesserung, sondern bezieht auch die Stärke der Verbesserung mit ein:

$$EI(x) = d\Phi\left(\frac{d}{\sigma(x)}\right) + \sigma(x)\phi\left(\frac{d}{\sigma(x)}\right) \quad (3.10)$$

$$\text{mit } d = \mu(x) - \mu_{max} - \xi \quad (3.11)$$

3. Die Methode der oberen (bzw. unteren) Konfidenzgrenzen, engl. (upper) confidence bound (UCB oder CB), wählt den Punkt, der im aktuellen Surrogatmodell die höchste Konfidenzgrenze aufweist. Diese wird sowohl vom erwarteten Mittelwert als auch von den bestehenden Unsicherheiten bestimmt. Über den Hyperparameter κ lässt sich der Beitrag der aktuellen Unsicherheit zur Wahl des nächsten Punktes direkt als Gewichtungsfaktor festlegen:

$$UCB(x) = \mu(x) + \kappa\sigma(x) \quad (3.12)$$

Im Rahmen dieser Arbeit wird in der Regel die UCB-Akquisitionsfunktion mit $\kappa = 1$ für die Bayes'sche Optimierung eingesetzt. In wenigen Ausnahmefällen wird mit $\kappa = 4$ eine stärkere Exploration genutzt.

3.4. Hypothesentests und Konfidenzintervalle

Im Forschungsgebiet der Bildverarbeitung mit tiefen neuronalen Netzen wird die Überlegenheit neuer Modelle und Verfahren häufig mit einer Verbesserung der absoluten Performanz im Vergleich zu einem existierenden Ansatz begründet. Jedoch spielen im Kontext des Trainings von neuronalen Netzen häufig Zufallsprozesse eine zentrale Rolle, beispielsweise bei der Datenreihenfolge, der Randomisierung und Augmentation oder bei Funktionen innerhalb des neuronalen Netzes (bspw. Dropout-Schichten). Trotz der gängigen Verwendung des Begriffs der Signifikanz wird die Sicherheit bezüglich der gemessenen Kennzahlen dabei häufig nicht berücksichtigt.

Im Rahmen dieser Arbeit wird die Bestimmung der zugehörigen Sicherheit der Messergebnisse miteinbezogen. Durch mehrfache Versuchswiederholung wird die Signifikanz der Ergebnisse dabei mittels Hypothesentests und

3. Theoretische Grundlagen

Konfidenzintervallen bestimmt. Hierbei finden neben der absoluten Differenz der Stichprobenmittelwerte auch die Streuung (Varianz) und der Umfang der Stichprobe Berücksichtigung. Im Folgenden werden die verwendeten Methoden der Signifikanzbeurteilung, der Welch-Test, der Shapiro-Wilk-Test sowie Konfidenzintervalle eingeführt. Tiefere Einblicke erhalten interessierte Leser beispielsweise in *Dekking* [18].

3.4.1. Hypothesentests

Ein Hypothesentest ist ein statistischer Test, bei welchem eine Nullhypothese H_0 gegen eine Gegenhypothese H_1 getestet wird. Es findet eine statistische Überprüfung statt, wie wahrscheinlich es ist, die gemessene Stichprobe X zu erhalten, falls die Nullhypothese H_0 tatsächlich gilt ($P(X|H_0)$). Diese Wahrscheinlichkeit berechnet sich anhand der Teststatistik T des ausgewählten Tests. Ist diese Wahrscheinlichkeit sehr klein, fällt diese in den kritischen Bereich, auch als Ablehnungsbereich bezeichnet. In diesem Fall kann davon ausgegangen werden, dass die Nullhypothese H_0 nicht gilt, wodurch H_0 abgelehnt und die Gegenhypothese H_1 angenommen werden kann. Im gegenteiligen Fall kann die Nullhypothese H_0 nicht abgelehnt werden, wobei dies keine Bestätigung der Nullhypothese H_0 darstellt. Dabei treten zwei Arten von Fehlern auf: Um einen Fehler 1. Art oder α -Fehler handelt es sich, falls die Nullhypothese zurückgewiesen wird, obwohl sie in Wahrheit wahr ist. Um einen Fehler 2. Art oder β -Fehler handelt es sich, falls die Nullhypothese nicht abgelehnt wird, obwohl sie in Wahrheit nicht zutrifft.

Beim Signifikanztest stellt das Signifikanzniveau α eine obere Schranke für Fehler 1. Art dar. Hierbei wird der Ablehnungsbereich anhand des vorab definierten Signifikanzniveaus α aus den Quantilen der dem Test zugehörigen Wahrscheinlichkeitsdichtefunktion bestimmt. Alternativ kann der p-Wert aus der Teststatistik und der Wahrscheinlichkeitsdichtefunktion berechnet und direkt mit dem Signifikanzniveau α verglichen werden. Dabei repräsentiert der p-Wert die Wahrscheinlichkeit dafür, die gemessene Stichprobe X oder eine in Richtung der Gegenhypothese H_1 noch extremere Stichprobe zu erhalten, falls H_0 gilt. Für $p < \alpha$ wird H_0 abgelehnt und H_1 angenommen.

Hypothesentests können einseitig (z. B. Mittelwert $\bar{X}$ ist kleiner als Grenzwert A) oder zweiseitig (z. B. Mittelwert $\bar{X}$ liegt im Intervall $[A, B]$ mit $A < B$) formuliert sein und setzen normalverteilte Prüfgrößen voraus. Es existieren unter anderem Hypothesentests für die Überprüfung von Lageparametern wie Mittelwert und Median, von Streuungen oder von zugrundeliegenden Verteilungen.

3.4.2. Welch-Test

Als t-Test wird eine Gruppe von Hypothesentests mit t-verteilter Testprüfgröße bezeichnet. Die studentsche t-Verteilung, nach dem Pseudonym „Student“ des englischen Statistikers William Sealy Gosset benannt, ist eine Wahrscheinlichkeitsverteilung, welche über die Anzahl der Freiheitsgrade n bestimmt ist [107]. Sie ähnelt der Gauß'schen Normalverteilung, ist jedoch breiter als diese. Für eine zunehmende Anzahl an Freiheitsgraden wird diese schmaler und geht bei $n \rightarrow \infty$ in die Normalverteilung über. Die Berechnung der Anzahl der Freiheitsgrade unterscheidet sich je nach gewähltem Test, basiert jedoch immer auf dem Stichprobenumfang.

Es existieren unter anderem t-Tests für eine (Einstichproben-t-Test) oder mehrere abhängige oder unabhängige Stichproben (z. B. Zweistichproben-t-Test). Sowohl der Zweistichproben-t-Test als auch der Welch-Test überprüfen die Erwartungswertunterschiede zwischen zwei normalverteilten unabhängigen Grundgesamtheiten. Im Gegensatz zum Zweistichproben-Gauß-Test müssen die Varianzen der beiden Grundgesamtheiten beim Zweistichproben-t-Test nicht bekannt sein, jedoch dürfen sie sich nicht unterscheiden. Der Welch-Test [121] dagegen ist auch für zwei Stichproben aus Grundgesamtheiten mit ungleichen Varianzen geeignet, obwohl stark unterschiedliche Varianzen auch hier zu einer niedrigeren Güte des Tests führen können. Unabhängig von den Vorbedingungen wird die Wahl des Welch-Tests gegenüber dem t-Test empfohlen [97, 61, 19]. Die Teststatistik für zwei normalverteilte unabhängige Prüfgrößen X und Y mit den Standardabweichungen s_X und s_Y und den Stichprobenumfängen n und m ergibt sich durch:

$$T = \frac{\bar{X} - \bar{Y} - \omega_0}{\sqrt{\frac{S_X^2}{n} + \frac{S_Y^2}{m}}} \quad (3.13)$$

Der Parameter ω_0 stellt dabei einen optional zu definierenden Mindestabstand (Differenz) zwischen den Erwartungswerten dar und wird häufig zu 0 gesetzt. Da diese Teststatistik unter der Annahme ungleicher Varianzen keiner exakten t -Verteilung folgt, wird eine Korrektur über eine modifizierte Anzahl der Freiheitsgrade (Welch-Satterthwaite-Gleichung) vorgenommen. Hierbei gehen die Stichprobenumfänge gewichtet mit den jeweiligen Varianzen in die Berechnung ein.

3.4.3. Shapiro-Wilk-Test

Ein Verteilungstest oder Anpassungstest ist ein statistischer Signifikanztest, mit welchem geprüft wird, ob vorliegende Daten auf eine bestimmte Weise verteilt sind. Es existieren sowohl allgemeine Verteilungstests für die Prüfung der Übereinstimmung empirischer Daten mit einer beliebigen Wahrscheinlichkeitsverteilung (bspw. Kolmogorow-Smirnow-Test oder Chi-Quadrat-Test), als auch Verteilungstests auf eine fest definierte Wahrscheinlichkeitsverteilung, wie die Normalverteilung (bspw. Shapiro-Wilk-Test oder Lilliefors-Test). Um dieselbe Teststärke zu erreichen, werden bei allgemeinen Verteilungstests in der Regel größere Stichprobenumfänge benötigt als bei spezifischen Verteilungstests.

Der Shapiro-Wilk-Test [101] zeichnet sich im Vergleich zu anderen Verteilungstests durch seine hohe Teststärke aus – auch bei der Überprüfung von Stichproben mit einem Umfang $n < 50$, wobei die Teststärke bei geringeren Stichprobenumfängen im Allgemeinen nachlässt [84]. Die Teststatistik W wird hierbei als Verhältnis der erwarteten Varianz b^2 einer normalverteilten Grundgesamtheit und der tatsächlichen gemessenen (korrigierten) empirischen Varianz der Stichprobe s^2 berechnet:

$$W = \frac{b^2}{(n-1)s^2} \quad (3.14)$$

mit

$$b = \sum_{i=1}^{\lfloor n/2 \rfloor} a_i(x_{(n+1-i)} - x_{(i)}), \quad s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (3.15)$$

$$W = \frac{\left(\sum_{i=1}^n a_i x_{(i)} \right)^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (3.16)$$

$$W = \frac{\left(\sum_{i=1}^{\lfloor n/2 \rfloor} a_i (x_{(n+1-i)} - x_{(i)}) \right)^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (3.17)$$

Dabei sind $x_{(i)}$ die Werte der zugehörigen Ordnungsstatistiken. Die Koeffizienten a_i ergeben sich aus den erwarteten Ordnungsstatistiken einer Normalverteilung.

3.4.4. Konfidenzintervalle

Konfidenzintervalle sind eng mit den Hypothesentests verwandt und geben einen Wertebereich um den Stichprobenmittelwert an, welcher den wahren Erwartungswert der der Stichprobe zugrundeliegenden Grundgesamtheit mit einer vorgegebenen Wahrscheinlichkeit, dem Konfidenzniveau $1 - \alpha$, einschließt. Im Rahmen dieser Arbeit werden Konfidenzintervalle auf Basis der t-Verteilung genutzt, da diese besser für kleinere Stichprobenumfänge mit unbekannter Varianz geeignet ist als die Annahme einer Normalverteilung. Das Konfidenzintervall KI zum Signifikanzniveau α berechnet sich wie folgt aus dem Mittelwert $\bar{x}$, der korrigierten Standardabweichung s und dem Umfang n der Stichprobe:

$$KI_{\alpha} = \bar{x} \pm \left(t_{(1-\frac{\alpha}{2}; n-1)} \frac{s}{\sqrt{n}} \right) \quad (3.18)$$

Dabei stellt $t_{(1-\frac{\alpha}{2}; n-1)}$ das zugehörige Quantil der t-Verteilung mit $n - 1$ Freiheitsgraden dar. Die Verwendung des $(1 - \frac{\alpha}{2})$ -Quantils (statt $1 - \alpha$) liegt in der Zweiseitigkeit des Konfidenzintervalls begründet.

Da der wahre Erwartungswert der unbekannten Grundgesamtheit selbst keine Zufallsgröße darstellt, gibt das Konfidenzintervall keine Auskunft über die Wahrscheinlichkeit der Lage des Erwartungswertes. Stattdessen definiert das Konfidenzniveau $1 - \alpha$ die Wahrscheinlichkeit dafür, dass das Konfidenzintervall den Erwartungswert tatsächlich umschließt. Mit anderen Worten schließen 95 % aller 95 %-Konfidenzintervalle für unabhängige Stichproben derselben Grundgesamtheit den wahren Erwartungswert dieser Grundgesamtheit ein. Um diesem Umstand Rechnung zu tragen, wird in Bezug auf den Erwartungswert auch von einer Sicherheit über die Lage statt von einer Wahrscheinlichkeit gesprochen.

Falls die untere Grenze des $(1 - \alpha)$ -Konfidenzintervalls einer Stichprobe X größer oder gleich der oberen Grenze $(1 - \alpha)$ -Konfidenzintervalls einer anderen Stichprobe Y ist, führt der Welch-Test für $H_1 : \bar{X} \geq \bar{Y}$ mit Signifikanzniveau α immer zur Annahme der Alternativhypothese H_1 . Umgekehrt können sich die $(1 - \alpha)$ -Konfidenzintervalle für die Stichproben X und Y jedoch überschneiden, obwohl der Welch-Test für α signifikant ist. Daraus lässt sich folgern, dass Konfidenzintervalle die Signifikanz zweier unabhängiger Stichproben systematisch unterschätzen, falls sie für die Untersuchung der relativen Lage der Erwartungswerte eingesetzt werden.

3.5. Bewertungsverfahren und Metriken

Im Rahmen dieser Arbeit wird die Güte von unterschiedlich trainierten Bildverarbeitungssystemen auf Basis tiefer neuronaler Netze verglichen, im Weiteren auch als *Performanz* bezeichnet. Um die Performanz der Systeme beurteilen zu können, werden sowohl gängige Bewertungsmetriken als auch speziell für den untersuchten Anwendungsfall des Item Pickings entwickelte Metriken eingesetzt (s. Unterabschnitt 4.4.3. Im Folgenden werden die gängigen und für diese Arbeit relevanten Kennzahlen und Metriken aus dem Bereich der Bildverarbeitung eingeführt.

3.5.1. Jaccard-Koeffizient (IoU)

Der Jaccard-Koeffizient, nach Paul Jaccard [53], oder gängiger die IoU (engl. intersection over union), ist eine Kennzahl für den Grad der Übereinstimmung von Mengen. Sie wird als Quotient der Schnittmenge durch die Vereinigungsmenge berechnet. Im Kontext der Objekterkennung wird die IoU insbesondere zur Bestimmung der räumlichen Übereinstimmung einer Detektion mit der zugehörigen Ground-Truth eingesetzt. Für die zugehörigen Masken (oder Begrenzungsrahmen) D und G gilt:

$$IoU(D, G) = \frac{|D \cap G|}{|D \cup G|} \quad (3.19)$$

Die IoU kann Werte zwischen 0, falls keine Überschneidung vorliegt, und 1 bei voller Übereinstimmung annehmen. Bei der Bewertung von Objekterkennungsverfahren werden in der Regel Detektionen mit einer $IoU \geq 0.5$ zur Ground-Truth als erfolgreich lokalisiert angesehen. Detektionen mit einer geringen Übereinstimmung werden in den Metriken der PASCAL VOC und COCO Challenge als Fehldetektionen interpretiert (s. Unterabschnitt 3.5.3). Dieser Schwellenwert wurde mit 50 % bewusst niedrig angesetzt, um Ungenauigkeiten in den Ground-Truth-Daten durch subjektive Wahrnehmungsunterschiede bei der Annotation zu adressieren [26].

3.5.2. Genauigkeit, Präzision und Sensitivität

Zur Bewertung von Klassifikations- oder Objekterkennungsverfahren werden unterschiedliche Metriken eingesetzt, zu welchen insbesondere die Genauigkeit, Präzision und Sensitivität zählen. Diese beziehen sich auf unterschiedliche Aspekte der Performanz eines Bildverarbeitungssystems. Für

die Berechnung dieser Metriken findet zunächst für jedes Bild bzw. jede Objektinstanz eine Einordnung der Klassifikationsentscheidungen K_D des Bildverarbeitungssystems im Vergleich zur Klasse der zugehörigen Ground-Truth K_G statt. Im Falle einer Mehrklassen-Klassifikation (engl. multiclass classification) findet eine Transformation zu binären Entscheidungen statt. Für jede Klasse $k \in K$ und zusammengehörige Entscheidungen $k_D \in K_D$ und Vorgaben $k_G \in K_G$ gilt:

1. Korrekt-positiv (TP; von engl. true positiv): $k_G = k$ und $k_D = k$
2. Falsch-positiv, auch Fehler 1. Art (FP): $k_G \neq k$ und $k_D = k$
3. Korrekt-negativ (TN): $k_G \neq k$ und $k_D \neq k$
4. Falsch-negativ, auch Fehler 2. Art (FN): $k_G = k$ und $k_D \neq k$

Für Objekterkennungsverfahren spielt hierbei neben der Klasse noch die räumliche Übereinstimmung eine Rolle. Dabei muss die IoU zwischen Detektion und Ground-Truth über einem vorgegebenen Schwellwert liegen, um als korrekt-positiv angesehen zu werden. Falls mehrere Detektionen der korrekten Klasse eine ausreichend große Überschneidung mit der Ground-Truth haben, wird in der Regel nur die Detektion mit der höchsten Sicherheit als korrekt-positiv, die übrigen Detektionen als falsch-positiv angesehen.

Genauigkeit: Die Genauigkeit a (engl. accuracy) stellt die primäre Metrik bei der Bildklassifikation dar und gibt den Anteil der korrekten Entscheidungen an der Gesamtheit der Entscheidungen an:

$$a = \frac{|TP| + |TN|}{|TP| + |TN| + |FP| + |FN|} \quad (3.20)$$

Die Genauigkeit ist insbesondere dann geeignet, wenn das Auftreten der einzelnen Klassen ausgewogen ist. Falls die Anzahl der einzelnen Klassen sehr unterschiedlich ist, werden stattdessen oder zusätzlich die Metriken der Präzision und Sensitivität eingesetzt. Dies gilt insbesondere für die Bewertung von Objekterkennungssystemen. Hierbei ist die Gesamtzahl der Entscheidungen, insbesondere die Anzahl der korrekt-negativen Entscheidungen, theoretisch unbegrenzt ($|TN| \rightarrow \infty$). Daher geht die Genauigkeit immer gegen eins und kann nicht sinnvoll eingesetzt werden.

3. Theoretische Grundlagen

Präzision: Die Präzision p (engl. precision) gibt den Anteil der korrekt-positiven Entscheidungen (korrekte Detektionen) an der Gesamtheit der positiven Entscheidungen (alle Detektionen) an:

$$p = \frac{|TP|}{|TP| + |FP|} \quad (3.21)$$

Damit gibt die Präzision Auskunft über die Güte der getroffenen Positiv-Entscheidungen (Detektionen) und beantwortet die Frage: „Welcher Anteil der Detektionen ist korrekt?“.

Sensitivität: Die Sensitivität r (engl. recall) gibt den Anteil der korrekt-positiven Entscheidungen (korrekte Detektionen) an der Gesamtheit der möglichen positiven Entscheidungen (Ground-Truths) an:

$$r = \frac{|TP|}{|TP| + |FN|} \quad (3.22)$$

Damit gibt die Sensitivität Auskunft über die Fähigkeit, die tatsächlichen Positiv-Beispiele (Ground-Truths) korrekt zu klassifizieren, und beantwortet die Frage: „Welcher Anteil der tatsächlichen Objekte wird korrekt detektiert?“.

Im Falle der Mehrklassen-Klassifikation oder -Objekterkennung werden die Metriken für jede Klasse getrennt berechnet. Die einzelnen Kennzahlen werden anschließend entweder ohne Gewichtung (Makro) oder mit dem relativen Auftreten der Klasse gewichtet (Micro) gemittelt. Präzision und Sensitivität sind weitgehend unabhängig, d. h. Objekterkennungssysteme können eine hohe Präzision, aber eine geringe Sensitivität erreichen (z. B. die einzige Detektion bei 10 Ground-Truths ist korrekt) und umgekehrt (z. B. die 100 Detektionen umfassen alle 10 Ground-Truths).

Zur Vollständigkeit sollte hier auch das F-Maß (engl. auch F1-Score) Erwähnung finden. Es kombiniert Präzision p und Sensitivität r in einer Kennzahl, findet jedoch weniger häufig Anwendung. Es stellt das harmonische Mittel aus Präzision und Sensitivität dar:

$$F = \left(\frac{p^{-1} + r^{-1}}{2} \right)^{-1} = \frac{2|TP|}{2|TP| + |FP| + |FN|} \quad (3.23)$$

Interessierte Leser finden Untersuchungen zur Korrelation und Signifikanz der Metriken beispielsweise in *Powers* [81].

3.5.3. Durchschnittspräzision (AP)

Für die Bewertung von Objekterkennungsverfahren hat sich die Metrik der Durchschnittspräzision (engl. average precision, AP) [126] als De-facto-Standard durchgesetzt. Verwendung als primäre Metrik fand die AP beispielsweise bei der PASCAL Visual Object Classes (Pascal VOC) Challenge [25, 26] und der COCO Challenge [63].

Die Durchschnittspräzision AP verbindet die Präzision p mit der Sensitivität r (s. Unterabschnitt 3.5.2). Sie stellt die Fläche unter der Präzisions-Sensitivitäts-Kurve $p(r)$ dar und ist definiert als:

$$AP = \int_0^1 p(r) dr \quad (3.24)$$

Wird die Durchschnittspräzision als Objekterkennungsmetrik eingesetzt, müssen die Detektionen D zunächst in ein geeignetes Format überführt werden. Dabei werden zunächst alle Detektionen $D_c \in D$ einer Klasse $c \in C$ bildübergreifend und nach dem zugehörigen Sicherheitswert (engl. confidence score) absteigend sortiert. Es findet eine iterative Berechnung der Präzisionen und Sensitivitäten für die ersten m mit $\{m \in \mathbb{N} \mid 0 < m \leq n\}$ der insgesamt $n = |D_c|$ Detektionen statt. Dabei werden die Detektionen mit der zugehörigen Ground-Truth derselben Klassen abgeglichen. Mit Hilfe eines IoU-Grenzwertes $t \in T$ als minimal notwendige Überschneidung werden diese in TP-, FP- und FN-Entscheidungen überführt und darauf basierend die jeweilige Präzision p und Sensitivität r ermittelt. Wird die Präzisionen p_c über die zugehörige Sensitivität r_c aufgetragen, ergibt sich die Präzisions-Sensitivitäts-Kurve $p_c(r)$ für die Klasse c .

Um den Einfluss der Sortierung nach den Sicherheitswerten auf die Präzisions-Sensitivitäts-Kurve zu reduzieren, wobei bereits kleine Unterschiede zu einer anderen Ordnung und damit zu einer veränderten Kurve führen können, findet eine zusätzliche Interpolation bzw. Glättung der Kurve statt. Hierbei wird die Kurve so angepasst, dass die Präzision mit zunehmender Sensitivität nicht ansteigt, indem jeder Präzisionswert $p(r)$ durch das Maximum der folgenden Präzisionswerte $p(r')$ mit $r' \geq r$ ersetzt wird:

$$p'(r) = \max_{r'; r' \geq r} p(r') \quad (3.25)$$

3. Theoretische Grundlagen

Die Durchschnittspräzision $AP_{c,t}$ für Klasse $c \in C$ und IoU-Grenzwert $t \in T$ ergibt sich als Fläche unter der geglätteten Präzisions-Sensitivitäts-Kurve $p'_{c,t}(r_{c,t})$:

$$AP_{c,t} = \int_0^1 p'_{c,t}(r_{c,t}) dr_{c,t} \quad (3.26)$$

Zur Flächenbestimmung wird in der Praxis jedoch meist eine 11-Punkt-(Pascal VOC) oder 101-Punkt-Approximation (COCO) anstelle des Integrals eingesetzt. Im Gegensatz zu COCO verwendet Pascal VOC den Terminus der mittleren Durchschnittspräzision (engl. mean average precision, mAP), um die Mittelwertbildung über mehrere Klassen hervorzuheben. Im Rahmen dieser Arbeit werden die Bezeichnung AP und die Berechnung anhand der Definition der COCO Challenge verwendet. Hierbei stellt die Durchschnittspräzision AP den Mittelwert der Durchschnittspräzisionen $AP_{c,t}$ aller Klassen C und IoU-Grenzwerte T dar:

$$AP = \frac{1}{|C||T|} \sum_{i=1}^{|C|} \sum_{j=1}^{|T|} AP_{c_i,t_j} \quad (3.27)$$

Während Pascal VOC lediglich einen festen IoU-Grenzwert $T = 0.5$ definiert, werden bei der Definition nach COCO mehrere Grenzwerte $T = \{0.5, 0.55, \dots, 0.95\}$ genutzt. Weitere Metriken der COCO Challenge sind u.a. $AP^{IoU=.50}$ (entspricht Pascal VOC mAP) und $AP^{IoU=.75}$ mit festen IoU-Grenzwerten $T = 0.5$ bzw. $T = 0.75$ sowie APs für einzelne Klassen. Zur Vereinfachung der Darstellung werden diese Kennzahlen im Weiteren mit $AP50$ und $AP75$ abgekürzt.

4. Methoden

Im Folgenden werden die in dieser Arbeit eingesetzten Methoden und Verfahren zur Datengenerierung, zur Domain Randomization und Domain Adaptation sowie zum Training und zur Evaluierung der neuronalen Netze beschrieben.

4.1. Datengenerierung und Szenenerstellung

Für die Erstellung der Simulationsszenen und die anschließende Datengenerierung wurde die Spiel-Engine Unity eingesetzt. Diese plattformübergreifende Laufzeit- und Entwicklungsumgebung für Spiele umfasst sowohl eine moderne Rendering-Engine als auch eine Physiksimulation. Für die Erzeugung der für das Training der neuronalen Netze notwendigen Annotationen wurde das von Unity frei zur Verfügung gestellte Perception-Paket eingesetzt. Für die dynamische Veränderung (Randomisierung) der Szeneneigenschaften wurde im Rahmen dieser Arbeit ein Manipulations-Paket für Unity entwickelt. Ein Manipulator verändert dabei eine oder mehrere Eigenschaften der Simulationsszene zufällig in einem vom Nutzer definierten Bereich und mit einer vorgegebenen Wahrscheinlichkeitsverteilung. Es wurden unter anderem Manipulatoren für die randomisierte Veränderung der Sichtbarkeit, Posen, Größenskalierung, Textur- und Oberflächeneigenschaften der Objekte sowie für diverse Beleuchtungs- und Kameraeigenschaften umgesetzt.

Im Folgenden werden die drei untersuchten Methoden der Szenenerstellung vorgestellt: das kontextfreie Referenzverfahren SynthDet, die manuelle Erstellung und Optimierung durch Scene Engineering sowie die automatisierte Szenenanpassung mittels Scene Adaptation. Abschließend wird der eingesetzte Prozess zur Generierung von Datensätzen für Vergleichsstudien beschrieben.

4.1.1. SynthDet

Als *kontextfrei* werden in der vorliegenden Arbeit Verfahren zur synthetischen Bilddatengenerierung verstanden, welche auf dem Rendering in einer Simulation basieren und, abgesehen von den zu erkennenden Objekten selbst, nur wenige oder gar keine szenenspezifischen Elemente oder Parameter enthalten. Stattdessen fokussieren sich diese Verfahren auf eine starke Randomisierung der Szeneneigenschaften und setzen damit den Ansatz der Domain Randomization um. Randomisiert werden hierbei typischerweise Objektanzahl, -typ und -posen sowie Beleuchtungs-, Kamera- und Hintergrundeigenschaften [119].

Ein Beispiel für ein solches Verfahren ist die von Unity Computer Vision entwickelte SynthDet-Pipeline¹ [6]. Diese basiert auf dem Perception-Paket für die Erzeugung synthetischer Bilddaten in der Spiele-Engine Unity und erweitert dieses um Komponenten zur Objektplatzierung und zur Szenenrandomisierung. Hierbei werden die zu erkennenden Objekte als texturierte 3D-Modelle ohne Physiksimulation mit anwendungsfallfremden Objekten gemischt und vor einer randomisierten Hintergrundebene aus überlagerten Fremdobjekten gerendert.

Der große Vorteil kontextfreier Verfahren ist der geringe Aufwand für die Erstellung der Simulation und der geringe Zeitbedarf der Datengenerierung durch die fehlende Physiksimulation. Im Gegensatz zu synthetischer Datengenerierung mit stärkerem Bezug zum Anwendungskontext (Scene Engineering und Scene Adaptation) werden weder Hintergrundwissen und Informationen zum Anwendungsfall noch reale Referenzdaten benötigt.

4.1.2. Scene Engineering

Als *Scene Engineering* (SE) wird in der vorliegenden Arbeit der manuelle Anpassungsprozess einer Simulationsszene für synthetische Datengenerierung an den Anwendungskontext der realen Zieldomäne bezeichnet [119]. Die Anpassung kann anhand spezifischer Informationen zum realen Anwendungsfall erfolgen, wie beispielsweise zur Objektanzahl pro Bild, zur Entfernung der Kamera oder zu den Umgebungsobjekten. Daneben kann auch allgemeines Hintergrundwissen eines Experten einfließen, wie beispielsweise realistische Beleuchtungseigenschaften und Objektpositionierungen. Um die Simulation weiter an die Zieldomäne anzupassen, können auch reale Referenzdaten genutzt werden. Anhand dieser Referenz können unter anderem Farb-,

¹GitHub Unity SynthDet: github.com/Unity-Technologies/SynthDet

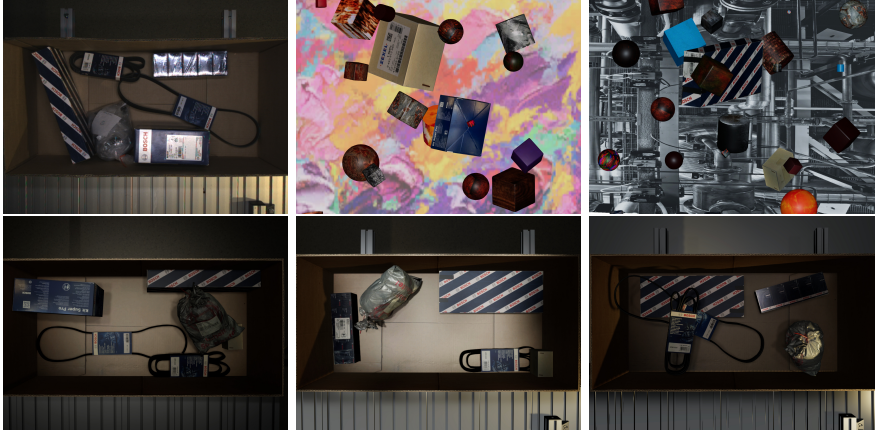


Abbildung 4.1.: Beispielbilder des realen Pre-Evaluierungsdatensatzes, den SynthDet-Datensätzen SD und SD-Opt (oben, v.l.n.r) sowie den Scene Engineering-Datensätzen SE-Preopt, SE-Opt und SE-Exp (unten, v.l.n.r).

Licht- und Kameraeigenschaften sowie die Randomisierungsbereiche für veränderliche Eigenschaften genauer abgeschätzt werden.

Im Gegensatz zu kontextarmen und -freien Ansätzen werden für das Scene Engineering im Allgemeinen wesentlich mehr Informationen benötigt. Die Optimierung profitiert vom Hintergrundwissen eines Simulationsexperten und verursacht einen großen zeitlichen Aufwand, wodurch wesentlich höhere Kosten für die Szenenerstellung entstehen. Die auf Basis dieser Simulationen erzeugten Bilddaten sind in der Regel jedoch wesentlich näher an der realen Domäne, wodurch sich die Performanz der damit trainierten Objekterkennungsverfahren deutlich verbessern kann. Die potenziellen Vorteile des Scene Engineering werden im Rahmen dieser Arbeit überprüft.

Für die eigentliche Datengenerierung wird auch im Falle der SE-Szenen das Perception-Paket eingesetzt. Die Randomisierung der Szeneneigenschaften basiert auf dem eigens entwickelten Manipulations-Softwarepaket. Im Gegensatz zum Standardverfahren Unity SynthDet werden die Objekte hier über zufällige Kräfte und Momente mithilfe der in Unity integrierten Physiksimulation positioniert.

4.1.3. Scene Adaptation

Als *Scene Adaptation* (SA) wird im Rahmen dieser Arbeit die automatisierte Anpassung einer Simulationsszene an reale Referenzdaten bezeichnet. Ein Optimierungsverfahren verändert dabei die Simulation über eine vorgegebene Auswahl an Simulationsparametern dahingehend, dass der Unterschied (Verlust) zwischen dem aktuellen Zustand der Simulationsszene und der Referenz minimiert wird. Die Scene Adaptation soll den großen manuellen Aufwand des Scene Engineering reduzieren oder sogar vollumfänglich ersetzen, wobei der Aufwand der initialen Simulationserstellung bestehen bleibt. Die automatisierte Anpassung kann jedoch auch als Ergänzung zum Scene Engineering eingesetzt werden, wie beispielsweise zur Feinabstimmung einzelner Szeneneigenschaften.

Im Rahmen dieser Arbeit wurde eine frei verfügbare Python-Implementierung der Bayes'sche Optimierung² zur Anpassung der Simulationsszene eingesetzt. Die Bayes'sche Optimierung ist hierbei besonders gut geeignet, da der genaue Zusammenhang zwischen den Parametern der nicht differenziellen Simulation und dem erzeugten Datenpunkt (gerendertes Bild mit Annotation) unbekannt ist (s. Abschnitt 3.3). Darüber hinaus ist die Auswertung eines Parametersatzes durch die Interaktion mit der Simulationssoftware und dem Rendering vergleichsweise aufwendig. Für die modulare Steuerbarkeit der Simulation wurde eine Web-Schnittstelle (RESTful-API) in das Manipulations-Softwarepaket integriert. Dadurch lassen sich die Eigenschaften der Simulationsszenen von externen Klienten, im vorliegenden Fall aus einer Python-Anwendung, verändern. Die Freiheitsgrade und die jeweiligen Optimierungsgrenzen, also der Suchraum der Optimierung, wurden dabei jeweils direkt über die existierenden Manipulator-Komponenten definiert.

Zu den verwendeten Verlusten zählen:

1. Pixel-Verlust: Der Pixel-Verlust L_P stellt die Summennorm (auch L1-Norm) der Differenz zwischen den Pixelwerten P des Referenzbildes R und des aktuellen Simulationsbildes S dar. Er ist unabhängig von Bildannotationen (Segmentierungsmasken) und berücksichtigt neben der Objektposition auch Farbeigenschaften. Dadurch ist er auch für die Optimierung der Beleuchtungsparameter sowie der Farb- und Reflexionseigenschaften der Objekte geeignet. Ein Nachteil des Pixel-Verlustes ist die fehlende monotone Abnahme bei Objektposi-

²GitHub Bayesian Opt.: github.com/bayesian-optimization/BayesianOptimization

tionierungen, welche sich nicht bereits nahe am Optimum befinden. Darüber hinaus führen bereits leichte Abweichungen der 3D-Modelle zum Ausbleiben der Konvergenz des Optimierungsverfahrens.

$$L_P = \sum_{x=0}^{|X|} \sum_{y=0}^{|Y|} |P_R(x, y) - P_S(x, y)| \quad (4.1)$$

2. Histogramm-Verlust: Der Histogramm-Verlust L_H berechnet sich als Summennorm der kumulierten Differenzen zwischen den Histogrammen H_R des Referenzbildes R und H_S des aktuellen Simulationsbildes S . Wie der Pixel-Verlust ist er für die Optimierung der Beleuchtungs-, Farb- und Reflexionseigenschaften geeignet. Durch die Histogrammbildung ist der Verlust jedoch weniger abhängig von der exakten Objektpositionierung.

$$L_H = \sum_{h=0}^{|H|} \left| \sum_{i=0}^h H_R(i) - H_S(i) \right| \quad (4.2)$$

3. IoU-Verlust: Der IoU-Verlust L_{IoU} berechnet sich als die mittlere Überlappung (auf Basis der IoU) aller zusammengehörigen annotierten Objektinstanzen I mit Segmentierungsmasken M zwischen Referenzbild R und dem aktuellen Simulationsbild S . Er ist lediglich für die Optimierung der Objektpositionen geeignet. Nachteil des IoU-Verlustes ist die fehlende monotone Abnahme vor einer Überschneidung der Segmentierungen.

$$L_{IoU} = \frac{1}{|I|} \sum_{i=0}^{|I|} \frac{|M_R(i) \cap M_S(i)|}{|M_R(i) \cup M_S(i)|} \quad (4.3)$$

4. Distanz-Verlust: Der Distanz-Verlust L_D stellt den mittleren Abstand der Mittelpunkte f_c aller zusammengehörigen annotierten Objektinstanzen I mit Segmentierungsmasken M zwischen dem Referenzbild R und dem aktuellen Simulationsbild S . Er ist wie der IoU-Verlust rein für die Optimierung der Objektpositionen geeignet. Im Gegensatz zum IoU-Verlust weist dieser jedoch bereits vor einer Überschneidung eine monotone Abnahme bei sich annähernden Segmentierungen auf, wobei sich die Instanzen zumindest partiell im Bild befinden müssen.

4. Methoden

Nachteil gegenüber dem IoU-Verlust ist dagegen die weitgehende Rotationsinvarianz des Verlustes.

$$L_D = \frac{1}{|I|} \sum_{i=0}^{|I|} |f_c(M_R(i)) - f_c(M_S(i))| \quad (4.4)$$

Dartüber hinaus wurden gewichtete Mischungen der Verluste genutzt, wie beispielsweise der IoU-Distanz-Verlust mit der IoU-Gewichtung w_{IoU} .

Um die im späteren Verlauf des Optimierungsprozesses weniger zielführende Exploration zu reduzieren, wurden sequenzielle Reduktionen des Suchraums (engl. domain reduction) eingesetzt. Nach einer initialen Aufwärmphase ohne Reduktion wurde der Suchraum S_r anhand einer vorab definierten Anzahl an Reduktionsvorgängen $r \in \mathbb{N}_0$ und einer relativen Zielgröße des Suchraums $z \in \mathbb{R}^+$ jeweils so angepasst, dass sich der neue Suchraum S_{r+1} gleichmäßig um die beste bisher gefundene Lösung x_r erstreckt:

$$S_{r+1} = x_r \pm \frac{|S_r| z^{\frac{1}{r}}}{2} \quad (4.5)$$

Durch diese Dynamik der Grenzen des Suchraums können auch Optima außerhalb der jeweils aktuell bestehenden Begrenzungen gefunden werden. Dieses Vorgehen reduziert die Relevanz der initial manuell gewählten Parameter und vereinfacht die Simulationserstellung somit noch weiter.

Für alle Experimente wurde eine Szenenoptimierung mittels Bayes'scher Optimierung eingesetzt, wobei sich die Optimierungsparameter und verwendeten Verlustfunktion je nach Optimierungsgegenstand unterscheiden. Die Parameter und die Reihenfolge der Optimierung wurden im Rahmen einer vorgelagerten Entwicklungsphase durch manuelle Exploration (Trial-and-Error-Methode) festgelegt. Dabei wurden, im Gegensatz zur Studie zur Simulationsoptimierung (s. Unterabschnitt 5.2.7), andere Bilddaten und eine andere Szene desselben Anwendungsfalls verwendet.

Im Rahmen dieser Arbeit soll die prinzipielle Machbarkeit einer automatisierten Simulationsoptimierung mittels Scene Adaptation in realistischen Anwendungsfällen belegt werden. Dabei wird explizit kein Anspruch auf eine optimale Parameterwahl in Bezug auf die Zeiteffizienz und die Generalisierbarkeit des Verfahrens erhoben. Beispielsweise wurde für die Kamera- und die Ladungsträgeroptimierung jeweils die folgenden Parameter genutzt:

- Explorationsphase: 4 500 Schritte (1 500 Aufwärmsschritte) mit $\kappa = 4$, 6 Reduktionen des Suchraums mit $z = 0.1$ (auf 10 % von S_0) und IoU-Distanz-Verlust mit $w_{IoU} = 0.2$
- Erweiterung des Suchraums mit $z = 5$ (auf 50 % von S_0)
- Exploitationsphase: 7 000 Schritte (1 000 Aufwärmsschritte) mit $\kappa = 1$, 20 Reduktionen des Suchraums mit $z = 0.1$ (auf 5 % von S_0) und IoU-Distanz-Verlust mit $w_{IoU} = 20$

Neben den quantitativen Experimenten zur Kamera- und Ladungsträgeroptimierung (s. Unterabschnitt 5.2.7), wurden weitere qualitative Untersuchungen zur Optimierung von Posen-, Farb- und Beleuchtungsparametern der Artikel und Umgebungsobjekten durchgeführt. Obwohl die Optimierung der Artikelposen für eine synthetische Datengenerierung unter Einsatz einer physiksimulationsbasierten Posenfindung keine Rolle spielt, ist diese für eine nachfolgende Farb- und Beleuchtungsoptimierung auf Basis von Pixel- oder Histogramm-Verlust von erheblicher Bedeutung. Diese weiteren qualitativen Untersuchungen umfassen unter anderem:

1. Optimierung der Positionen, Rotationen und Posen (Kombination Position und Rotation) der Einzelobjekte mit IoU-Distanz-Verlust
2. Optimierung der Posen über Spiegelungsoperationen mit IoU- und Pixel-Verlust
3. Optimierung der Beleuchtungs-, Objektfarb- und -reflexionseigenschaften mit Histogramm- und Pixel-Verlust

4.1.4. Datengenerierung

Im Rahmen der Experimente dieser Arbeit werden mehrere Studien zum Einfluss einzelner Szeneneigenschaften sowie einzelner Parameter der Datengenerierung durchgeführt. Durch die Wiederholung eines Datengenerierungsprozesses mit verändertem Zustand des Zufallszahlengenerators (engl. seed) sowie mit veränderter Anzahl oder Reihenfolge der Zufallszahlenabfrage entsteht ein veränderter Datensatz. Auch wenn die primären statistischen Eigenschaften des Datensatzes erhalten bleiben, haben Zufallseffekte beispielsweise Einfluss auf die Objektpositionierung oder das Auftreten von seltenen Randfällen. Dies kann die Performanz der damit trainierten neuronalen Netze signifikant verändern. Somit erschweren diese Zufallseffekte die

4. Methoden

Einordnung des Einflusses der untersuchten Eigenschaften und Parameter. Von besonderer Relevanz ist dies, falls die untersuchten Eigenschaften selbst nur einen geringen Einfluss auf die gemessene Performanz haben.

Um diesem Umstand zu begegnen und die Vergleichbarkeit der Datensätze zu erhöhen, wurden die Datensätze innerhalb einer Studie (z. B. Untersuchung des Einflusses eines einzelnen Szenen-Strukturelements) mittels eines doppel-schleifigen Datengenerierungsprozesses erzeugt. Bei der Erzeugung eines Datensatzes mit n Bilddaten und m Parametervariation p_j werden dabei iterativ Basiszustände B_i mit $i \in \{1, \dots, n\}$ der Simulationsszene zufallsbasiert ermittelt. Hierbei werden alle studienunabhängigen Randomisierungen angewandt, welche sich zwischen den Datensätzen der Studie nicht unterscheiden sollen. In den meisten der Studien dieser Arbeit umfassen diese die Positionierung von Kamera und Objekten sowie die Beleuchtungseigenschaften – jeweils mit Ausnahme der zugehörigen spezifischen Studien zum Sichtfeld, zur Objektpositionierung und zur Beleuchtung.

Zwischen dem Wechsel der Basiszustände B_i auf B_{i+1} findet die iterative Veränderung der eigentlichen Studienparameter p_j mit $j \in \{1, \dots, m\}$ statt. Dabei werden nur jene Manipulatoren angewandt, welche dem jeweiligen Studienparameter zugeordnet sind (z. B. Ausblenden eines Strukturelements). Nach jedem Iterationsschritt wird die Szene gerendert, das annotierte Bild gespeichert und die Simulation wieder auf den gemeinsamen Basiszustand B_i zurückgesetzt. Ein Datensatz D_j einer Studie wird damit über die simulationsbasierte Erzeugungsfunktion g wie folgt generiert:

$$D_j = \{g(B_i, p_j) | i \in \{1, \dots, n\}\} \quad (4.6)$$

Durch dieses Vorgehen kann sichergestellt werden, dass sich die Datensätze innerhalb einer Studie lediglich in den Studienparametern unterscheiden. Die Einflüsse von Zufallseffekten auf Eigenschaften, welche nicht Gegenstand der Studie sind, können somit stark reduziert und die Vergleichbarkeit innerhalb der Studie erhöht werden.

4.1.5. Datensatzanalyse

Um die Übereinstimmung zweier Datensätze zu ermitteln, können neben bildpunktbezogenen Merkmalen auch Annotationseigenschaften herangezogen werden. Ein Beispiel hierfür ist die im Rahmen dieser Arbeit überprüfte Übereinstimmung der Verteilungen der Objektanzahl pro Bild zwischen dem Trainings- und dem Evaluierungsdatensatz (s. Unterabschnitt 5.2.6).

Als Maßzahl für den Grad der Übereinstimmung wird die Histogramm-Differenz ΔH verwendet. Dieser Kennwert stellt den Mittelwert der absoluten Differenzen der Histogrammklassen zweier normierter Histogramme dar – hier die Histogramme der relativen Häufigkeit der Objektanzahl pro Bild. Für zwei normierte Histogramme H_a und H_b mit jeweils k Klassen (engl. bins) wird die Histogramm-Differenz ΔH wie folgt berechnet:

$$\Delta H = \frac{1}{n} \sum_{i=1}^k |H_a(i) - H_b(i)| \quad (4.7)$$

4.2. Domain Randomization

Als (Daten-)Augmentation werden im Bereich des maschinellen Lernens Techniken verstanden, welche die Erweiterung eines Datensatzes um neue Datenpunkte durch die Modifikation bestehender Datenpunkte ermöglichen. Augmentation wird beim maschinellen Lernen primär eingesetzt, um die Überanpassung (engl. overfitting) beim Training von Modellen zu reduzieren. Im Bereich der Bildverarbeitung werden hierbei meist zufallsbasierte bildstatistische Methoden genutzt. Durch die Augmentation wird in der Regel eine Vervollständigung der Trainingsdomäne um nicht vorhandene Zwischenstufen angestrebt. Dies ist sinnvoll, falls die Zieldomäne größtenteils der Trainingsdomäne entspricht oder eine Untermenge dieser ist – ihre Eigenschaften also weitgehend innerhalb der Grenzen der Eigenschaften der Trainingsdomäne liegen. Augmentationstechniken können jedoch auch im Sinne der Domain Randomization eingesetzt werden, um die Trainingsdomäne gezielt zu erweitern oder zu verschieben. Hierbei wird ein Datensatz um neue Eigenschaften ergänzt, welche explizit außerhalb der Spanne seiner bisherigen Eigenschaften liegen. Im Rahmen der Untersuchungen zur Domain Randomization werden die Augmentationen mit dem Ziel angewandt, die Trainingsdomäne so zu erweitern, dass die Domänenlücke zwischen synthetischen und realen Bilddaten reduziert wird. Besondere Relevanz hat dies auch aufgrund der ungewöhnlichen Homogenität der untersuchten Datensätze (s. Abschnitt 2.1).

Bei der typischen Anwendung von Augmentationstechniken werden die Datenpunkte beim Trainingsprozess vor jeder Inferenz auf Basis veränderter Zufallszahlen modifiziert. Dadurch wird derselbe Datenpunkt während des Trainings im Regelfall nie mehrfach verwendet und die Anzahl der einzigartigen Datenpunkte steigt. Bei den Untersuchungen zur Domain

4. Methoden

Randomization soll dagegen eine Vergrößerung der Trainingsdatensätze explizit verhindert werden, da diese ebenfalls potenziell Einfluss auf die Performanz der trainierten neuronalen Netze hätte und eine Vermischung der Effekte vermieden werden soll. Um dies zu erreichen, werden die Datensätze bereits vor dem Training so stark modifiziert, dass die Bildeigenschaften weit außerhalb ihrer bisherigen Grenzen liegen. Durch den vorgelagerten Augmentationsprozess umfasst der modifizierte Datensatz eine unveränderte Anzahl an Bildern, welche bei jeder Trainingsepoche wiederholt verwendet werden. Entsprechende Beispielbilder finden sich in den Anhängen B.1 und B.2.

Im Rahmen dieser Arbeit finden die folgenden Augmentationstechniken Anwendung:

1. RGB-Farbraum
 - a) Farbvariation
 - b) Helligkeit
2. HSV-Farbraum
 - a) Farbwert (Hue)
 - b) Sättigung (Saturation)
 - c) Hellwert (Value)
3. Chromatische Aberration
4. Kontrast
5. Unschärfe
 - a) Gauß'sche Unschärfe
 - b) Lineare (Box) Unschärfe
6. Gauß'sches Rauschen

Eigenschaften, welche sich auf den Sichtbereich beziehen, werden im Rahmen dieser Arbeit über die Kamerapose und -parameter untersucht. Selbiges gilt für das Einfügen von Störobjekten und -texturen, welches über randomisierte Hintergrundtexturen und Distraktoren in der Simulation gelöst wird. Zu den im Rahmen dieser Arbeit nicht untersuchten Augmentationstechniken zählen unter anderem:

- Löschen und Überschreiben von Bildbereichen (Erasing und Patching)
- Spiegeln (Flipping)
- Ausschneiden (Cropping) und Auffüllen (Padding)
- Vergrößern und Verkleinern (Resizing)
- Translation und Rotation

4.2.1. Zufallszahlen

Bei der Anwendung der randomisierten Augmentation wird die Implementierung des Zufallszahlen-Generators der NumPy-Bibliothek verwendet, um pseudozufällige Fließkommazahlen im Intervall $V = [0, 1] = \{x \in \mathbb{R} \mid 0 \leq x \leq 1\}$ zu erzeugen. Falls nicht anderweitig erläutert, werden hierbei stetig gleichverteilte Zufallszahlen $r \in V$ gezogen. Um den Einfluss anderer Verteilungen zu untersuchen, werden jedoch auch Versuche auf Basis normalverteilter Zufallszahlen $r_{norm} \in Z \sim \mathcal{N}(\mu, \sigma^2)$ mit $\mu = \frac{1}{2}$ und $\sigma = \frac{1}{6}$ durchgeführt. Da die generierten Zufallszahlen hierbei auch außerhalb des Intervalls V liegen können (ca. 0,27%), werden diese durch neue zufällige Ziehungen ersetzt, wodurch diese für das Intervall V zumindest näherungsweise normalverteilt sind.

Für jede Augmentation wird ein Parameter δ definiert, welche die Stärke der entsprechenden Augmentation bestimmt und im Weiteren als *Verstärkungsfaktor* bezeichnet wird. Im Rahmen jedes Versuchs wird über die Festlegung von δ_{min} und δ_{max} ein Variationsintervall der Augmentationstärke definiert. Der Verstärkungsfaktor $\delta(r)$ auf Basis einer Zufallszahl $r \in V$ wird damit folgendermaßen berechnet:

$$\delta(r) = (\delta_{max} - \delta_{min})r + \delta_{min} \quad (4.8)$$

Bei der Definition der im Folgenden dargestellten Augmentationen wird davon ausgegangen, dass die Bildpunkte p eines Bildes I bzw. p_c bei der Unterscheidung nach Farbkanälen $c \in \{R, G, B\}$ oder $c \in \{H, S, V\}$ auf das Intervall V skaliert sind. Die Pixel des augmentierten Bildes I' werden dementsprechend mit p' bzw. p'_c repräsentiert. Für jede Augmentation A wird dabei eine von $\delta(r)$ abhängige Transformation γ_A auf die Pixel p bzw. p_c angewandt:

$$p' = \gamma_A(p, \delta(r)) \quad \text{bzw.} \quad p'_c = \gamma_A(p_c, \delta(r)) \quad (4.9)$$

4.2.2. Farbe (RGB)

Bei der RGB-Farbaugmentation werden die Pixelwerte der drei Farbkanäle mit unabhängigen Zufallszahlen variiert. Dadurch werden primär Veränderungen des Farbtons, jedoch auch der Bildhelligkeit, der Sättigung und des Kontrastes erreicht. Ähnliche Veränderungen können in der realen Anwendung durch Kamera- (bspw. Blende und Belichtungszeit) und Beleuchtungseigenschaften (bspw. Wellenlänge und Intensität) beeinflusst werden.

Für die Variation der Farbe wird ein Verstärkungsfaktor $\delta \in \{x \in \mathbb{R} \mid -1 \leq x\}$ eingeführt und jeder Pixel eines Farbkanals mit dem Faktor $\delta + 1$ multipliziert. Damit führt ein Verstärkungsfaktor $\delta = 1$ zu einer Erhöhung der Pixelwerte um 100% und ein Verstärkungsfaktor $\delta = -1$ zu einer Reduktion der Pixelwerte um 100% (auf 0). Durch dieses Vorgehen entsprechen die mittleren Pixelwerte des Datensatzes weitgehend den Mittelwerten des Eingangsdatensatzes. Es ergibt sich für die Anpassung der Bildpunkte folgende Formel:

$$p'_c(r_c) = (1 + \delta(r_c))p_c \quad (4.10)$$

Im Rahmen der Studien wird jeweils der maximale Verstärkungsfaktor δ_{max} vorgegeben und $\delta_{min} = -\delta_{max}$ gewählt.

Alternative Ansätze zur gewählten Methode wären beispielsweise die Addition von zufälligen Pixelwerten $a \in \{x \in \mathbb{R} \mid -1 \leq x \leq 1\}$ oder die relative Änderung durch Multiplikation mit einem zufälligen Faktor $b \in \{x \in \mathbb{R} \mid 0 \leq x\}$.

4.2.3. Helligkeit

Bei der Helligkeits-Augmentation werden die drei Farbkanäle mit derselben Zufallszahl variiert. Da sich die Bildhelligkeit bei der Farbaugmentation im RGB-Farbraum als Summe der drei Bildkanäle ergibt, sind mittlere Helligkeitswerte bei gleichverteilter Randomisierung stark überrepräsentiert (zentraler Grenzwertsatz). Durch die direkte Augmentation der Helligkeit dagegen wird eine Gleichverteilung der Bildhelligkeit im Datensatz erreicht. Aufgrund der Anpassung im RGB-Farbraum können jedoch auch der Farbton, die Sättigung und der Kontrast des Bildes beeinflusst werden.

Für die Variation der Helligkeit wird ein Parameter $\delta \in \{x \in \mathbb{R} \mid -1 \leq x \leq 1\}$ eingeführt, welcher die absolute Veränderung der Helligkeit festlegt. Dabei wird jeder Pixel des Bildes mit dem Verstärkungswert δ addiert. Damit ergibt sich für die Anpassung der Bildpunkte folgende Formel:

$$p'(r) = \delta(r) + p \quad (4.11)$$

Im Rahmen der Studien wird jeweils der maximale Verstärkungswert δ_{max} vorgegeben und $\delta_{min} = -\delta_{max}$ gewählt.

4.2.4. Farbwert, Sättigung und Hellwert (HSV)

Die Augmentation im HSV-Farbraum (s. Unterabschnitt 3.1.1) erlaubt die gezielte Variation des Farbwertes (H), der Sättigung (S) und des Hellwertes (V) eines Bildes mit einer vorgegebenen Verteilung. Hierfür werden die Bilder vor der Augmentation in den HSV-Farbraum und nach der Augmentation zurück in den RGB-Farbraum konvertiert. Die HSV-Augmentation kann jedoch auch indirekte Auswirkungen auf den Kontrast des Bildes haben.

Das Vorgehen entspricht hierbei dem der RGB-Farbaugmentation, jedoch werden getrennte Verstärkungsfaktoren δ_c für die drei Kanäle $c \in \{H, S, V\}$ verwendet. Damit ergibt sich für die Anpassung der Bildpunkte folgende Formel:

$$p'_c(r_c) = (1 + \delta_c(r_c))p_c \quad (4.12)$$

Im Rahmen der Studien wird jeweils der maximale Verstärkungsfaktor δ_{max} vorgegeben und $\delta_{min} = -\delta_{max}$ gewählt. Darüber hinaus wurden alternative Berechnungsmethoden für die Variationsstärke der HSV-Augmentation untersucht, welche in Anhang A.1 ausgeführt sind.

4.2.5. Chromatische Aberration

Die chromatische Aberration ist ein Abbildungsfehler optischer Linsensysteme, welcher durch die Frequenzabhängigkeit (Dispersion) der Brechzahl verursacht wird, wobei kurzwelliges Licht (blau) stärker als langwelliges Licht (rot) gebrochen wird. Damit ergeben sich für ein Kamerasystem abhängig von der Farbe unterschiedliche Brennweiten, wodurch die Inhalte im resultierenden Bild je nach Farbkanal unterschiedlich groß abgebildet werden (chromatische Vergrößerungsdifferenz). Dieser Farbfehler tritt bei der im realen Anwendungsfall eingesetzten Kamera auf und ist in den realen Vergleichsdaten zu erkennen (s. Abbildung 4.2). Um diesen Fehler für die synthetischen Bilddaten nachzustellen, wird eine synthetische Nachbildung chromatischer Aberration mittels eines Farbdrifts auf die aberrationsfrei generierten Bilder angewendet.

4. Methoden

Für die Variation der chromatischen Aberration wird ein Verstärkungsfaktor δ eingeführt, welcher die Stärke des Farbdrifts der drei Farbkanäle im RGB-Farbraum bestimmt. Hierbei werden die einzelnen Farbkanäle als separate Monochrom-Bilder auf unterschiedliche Auflösungen vergrößert. Während der Rot-Kanal mit Auflösung (x, y) unverändert bleibt, werden die Monochrom-Bilder des Grün- und Blau-Kanals mit folgenden Faktoren in beide Dimensionen vergrößert:

$$\delta_G(r) = \frac{509}{500}\delta(r) \quad \text{und} \quad \delta_B(r) = \frac{261}{250}\delta(r) \quad (4.13)$$

Als Resampling-Methode wird hierbei der Lanczos-Filter verwendet. Dieser ist definiert als:

$$L(x) = \begin{cases} \text{sinc}(x) \text{sinc}\left(\frac{x}{a}\right) & \text{wenn } -a < x < a, \ a \neq 0 \\ 1 & \text{wenn } x = 0 \\ 0 & \text{ansonsten} \end{cases} \quad (4.14)$$

mit

$$\text{sinc}(x) = \frac{\sin(\pi x)}{\pi x}, \quad a = 3 \quad (4.15)$$

Abschließend werden die getrennten Monochrom-Bilder der Farbkanäle wieder mittelpunkzentriert zu einem Farbbild zusammengefügt, wobei die überstehenden Ränder des Grün- und des Blau-Kanals abgeschnitten werden. Im Rahmen der Studien wird jeweils der maximale Verstärkungsfaktor δ_{max} vorgegeben und $\delta_{min} = 0$ gewählt.

4.2.6. Kontrast

Bei der Kontrast-Augmentation wird der Bildkontrast, also die Spanne bzw. Differenz zwischen dem höchstem und dem niedrigstem Pixelwert des Bildes bzw. Farbkanals, variiert. Im Gegensatz zu RGB-Farb-, Helligkeits- und HSV-Augmentation kann hierbei eine vorgegebene Verteilung des Bildkontrastes im Datensatz erreicht werden.

Für die Variation des Kontrasts wird der Verstärkungsfaktor $\delta \in \{x \in \mathbb{R} \mid 0 \leq x\}$ eingesetzt. Auf Basis des Verstärkungsfaktors wird dabei die Differenz jedes Pixels p_c eines Bildkanals c vom Kanalmittelwert $\bar{p}_c$ variiert. Hierdurch wird zwar der Kontrast der einzelnen Kanäle und damit des Gesamtbildes verändert, die mittleren Pixelwerte jedes Kanals bleiben

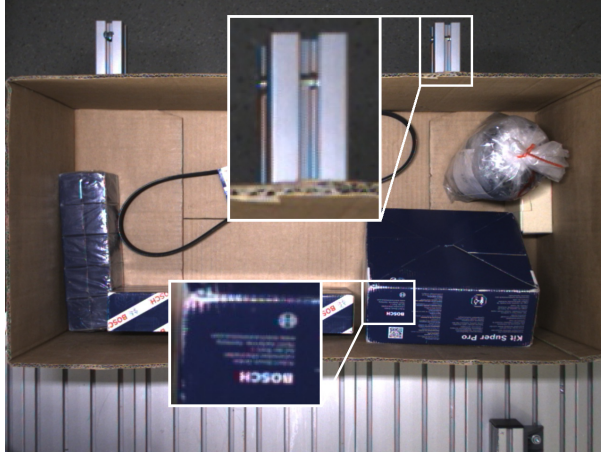


Abbildung 4.2.: Beispiel für chromatische Aberration in den realen Bilddaten. Hierbei sind verschiedene rote und blaue Einfärbungen im Bild zu erkennen.

jedoch stabil. Damit ergibt sich für die Anpassung der Bildpunkte folgende Formel:

$$p'_c(r) = (p_c - \bar{p}_c)\delta(r) + \bar{p}_c \quad (4.16)$$

mit

$$\bar{p}_c = \frac{1}{|P_c|} \sum_1^{|P_c|} p_c, \quad p_c \in P_c \in I \quad (4.17)$$

Im Rahmen der Studien wird jeweils der maximale Verstärkungsfaktor δ_{max} vorgegeben und $\delta_{min} = \frac{1}{\delta_{max}}$ gewählt. Darüber hinaus wurden alternative Berechnungsmethoden für die Variationsstärke der Kontrast-Augmentation untersucht, welche in Anhang A.2 ausgeführt sind.

4.2.7. Unschärfe

Bei der Unschärfe-Augmentation werden die Werte benachbarter Pixel gemittelt und das Bild somit geglättet. Dadurch reduzieren sich der Informationsgehalt des ursprünglichen Bildinhalts und die Varianz der Pixelwerte.

4. Methoden

Ähnliche Veränderungen können in der realen Anwendung durch die Brennweite der Kamera und deren Abstand zur Szene beeinflusst werden.

Für die Variation der Bild(un)schärfe wird als Verstärkungsfaktor $\delta \in \{k \mid k = 2x + 1 \mid x \in \mathbb{N}\}$ die Größe des jeweiligen Filterkerns eingesetzt. Im Rahmen der Studien werden mit Gauß- und Box-Filter zwei unterschiedliche Filterkerne bzw. Faltungsmatrizen eingesetzt.

Gauß-Filter: Bei der Glättung mit dem Gauß-Filter $G(\delta)$ ist die Gewichtung der benachbarten Pixel abhängig von deren Entfernung vom Zentrum der aktuellen Filterposition:

$$G(\delta) = \frac{\vec{g}(\delta) \cdot \vec{g}(\delta)^\top}{|\vec{g}(\delta) \cdot \vec{g}(\delta)^\top|} \quad (4.18)$$

mit

$$\vec{g}(\delta) = \begin{pmatrix} g_1(\delta) \\ \dots \\ g_\delta(\delta) \end{pmatrix}, \quad g_i(\delta) = e^{-\frac{\left(i - \frac{\delta+1}{2}\right)^2}{2\left(\frac{3\delta+7}{20}\right)^2}}, \quad i \in \{x \in \mathbb{N} \mid 1 \leq x \leq \delta\} \quad (4.19)$$

Box-Filter: Bei der Glättung mit dem Box-Filter $B(\delta)$ haben die benachbarten Pixel dieselbe Gewichtung:

$$K(\delta) = \frac{1}{\delta} \vec{k}(\delta) \cdot \vec{k}(\delta)^\top \quad (4.20)$$

mit

$$\vec{k}(\delta) = \begin{pmatrix} k_1 \\ \dots \\ k_\delta \end{pmatrix}, \quad k_i = 1, \quad i \in \{x \in \mathbb{N} \mid 1 \leq x \leq \delta\} \quad (4.21)$$

Im Rahmen der Studien wird jeweils der maximale Verstärkungsfaktor (Filtergröße) δ_{max} vorgegeben und $\delta_{min} = 1$ gewählt.

4.2.8. Rauschen

Bei der Rausch-Augmentation werden die einzelnen Pixelwerte zufällig verändert bzw. ausgelenkt, wodurch die Struktur und der Informationsgehalt des ursprünglichen Bildinhalts reduziert sowie die Varianz der Pixelwerte

erhöht werden. In der realen Anwendung wird Bildrauschen primär durch den Kamerasensor verursacht (bspw. elektrische Ladungsschwankungen oder Quantisierungseffekte).

Im Rahmen dieser Arbeit wird additives weißes Rauschen auf Basis der Normalverteilung, kurz Gauß'sches Rauschen, eingesetzt. Für die Variation des Bildrauschens wird ein Verstärkungsfaktor $\delta \in \{x \mid 0 \leq x\}$ eingeführt und als Varianz der zugrundeliegenden Zufallsvariablen $X(\sigma^2) \sim \mathcal{N}(0, \sigma^2)$ eingesetzt. Damit ergibt sich für die Anpassung der Bildpunkte folgende Formel:

$$p'(r) = X(\delta(r)) + p \quad (4.22)$$

Im Rahmen der Studien wird jeweils der maximale Verstärkungsfaktor (Varianz) δ_{max} vorgegeben und $\delta_{min} = 0$ gewählt.

4.3. Domain Adaptation

Durch die Domain Adaptation werden die Bilddaten an die Bildstatistik eines Referenzdatensatzes angepasst. Im Rahmen dieser Arbeit werden zwei unterschiedliche Ansätze zur Bildtransformation eingesetzt: (1) die bildstatistische Anpassung mittels ICT und (2) das generative Verfahren CycleGAN auf Basis tiefer neuronaler Netze, welches erst trainiert und anschließend für die Transformation inferiert wird.

4.3.1. Bildstatistisch (ICT)

Für die bildstatistische Anpassung eines Datensatzes an einen Referenzdatensatz wird exemplarisch eine Methode des Bildfarbtransfers, engl. image color transfer (ICT), eingesetzt. Hierbei findet für jedes Bild S des Quelldatensatzes eine Anpassung an die Statistik eines Bildes T des Referenzdatensatzes R statt. Für die Paarbildung wird neben einer rein zufälligen Zuordnung auch eine annotationsbasierte Auswahlmethode eingesetzt. Hierfür werden Histogramme für das Quellbild H_S und für das potenzielle Zielbild H_T mit der Anzahl des Auftretens einer Klasse $c \in C$ berechnet. Darauf basierend wird die Klassenhistogrammdistanz d wie folgt berechnet:

$$d(S, T) = \sum_c^C |H_{S,c} - H_{T,c}| \quad (4.23)$$

4. Methoden

Dabei bedeutet $d(S, T) = 0$, dass das Auftreten aller Klassen zwischen Quellbild S und Zielbild T übereinstimmt. Aus dem Referenzdatensatz R wird jenes Zielbild $T \in R$ ausgewählt, für welches sich die minimale Distanz $d_{S,min}$ zum Histogramm des Quellbildes ergibt:

$$d_{S,min} = \min_{\forall T \in R} d(S, T) \quad (4.24)$$

Falls sich für mehrere Zielbilder die minimale Distanz $d_{S,min}$ ergibt, findet unter diesen eine zufällige Auswahl statt.

Die Bildstatistik des Quellbildes wird, wie von Reinhard *et al.* beschrieben, für alle drei Bildkanäle $k \in L, a, b$ im $L^*a^*b^*$ -Farbraum getrennt an das Referenzbild angepasst [88]. Dabei werden Mittelwert μ_S und Standardabweichung σ_S jedes Bildkanals an die korrespondierende Statistik μ_T und σ_T des gepaarten Referenzbildes angepasst. Ein Pixel p an Position (x, y) und Kanal k wird dabei wie folgt zu p' transformiert:

$$p'(x, y, k) = \frac{\sigma_{T,k} (p(x, y, k) - \mu_{S,k})}{\sigma_{S,k}} + \mu_{T,k} \quad (4.25)$$

Falls die Annotation pixelweise Segmentierungsmasken enthält, kann alternativ eine Anpassung der Bildstatistik für einzelne Bildregionen (Pixel innerhalb der Segmentierung) vorgenommen werden. Hierbei wird jede Bildregion im Quellbild an die Bildstatistik einer passenden Bildregion mit übereinstimmender Klasse im Zielbild angepasst. Abschließend findet eine bildstatistische Anpassung aller unsegmentierten Pixel im Zielbild statt, inklusive der segmentierten Pixel ohne korrespondierendes Gegenstück.

Durch die pixelweise Anpassung einzelner Bildbereiche ergeben sich teilweise starke Farb- und Helligkeitssprünge an den Grenzen der Masken. Daher kann eine Glättung an den Begrenzungen der Segmentierungsmasken stattfinden. Im Rahmen dieser Arbeit wird hierbei ein Box-Unschärfe-Kernel mit Filtergröße 7 verwendet.

4.3.2. Generativ (CycleGAN)

Für die Domain Adaptation mittels generativer Verfahren wird CycleGAN eingesetzt (s. Unterabschnitt 3.2.4). Für die Anpassung eines synthetischen Quelldatensatzes werden Generator und Diskriminator zunächst gemeinsam für die Transformation auf den realen Referenzdatensatz trainiert. Nach dem Training wird der trainierte Generator für alle Bilder im Quelldatensatz inferiert, wobei die Ausgaben (transformierte Bilder) zusammen mit

den originalen Annotationen des Quelldatensatzes den neuen Datensatz bilden. Da aus Gründen begrenzter Hardware-Ressourcen und dem großen Grafikspeicherbedarf von CycleGAN eine reduzierte Bildgröße von 512 x 512 Pixeln verwendet wird, findet abschließend eine Skalierung der transformierten Bilder mit bikubischer Interpolation auf die originale Bildgröße 1280 x 960 statt. Das Training umfasst jeweils 2 bis 20 Epochen, wobei der kleinere Datensatz im Falle unterschiedlicher Datensatzgrößen für eine Epoche so lange wiederholt iteriert wird, bis eine volle Iteration auf dem größeren Datensatz abgeschlossen ist.

Im Rahmen einer Vorstudie konnte gezeigt werden, dass GANs zu annotationsinkonsistenten Transformationen neigen [120], da Objekte durch diese im Bild außerhalb ihrer ursprünglichen Umrisse reproduziert („verschoben“) werden können. Dies tritt insbesondere dann auf, falls mindestens einer der Trainingsdatensätze eine stark homogene Verteilung der Objektpositionen aufweist und diese zwischen Quell- und Referenzdatensatz nicht übereinstimmt. Um einer solchen Annotationsinkonsistenz entgegenzuwirken, werden beim Training Augmentationen zur Randomisierung von Merkmalspositionen angewendet. Diese umfassen das Ausschneiden eines zufälligen Bildbereichs (Anteil zwischen 85 % und 100 %) sowie horizontale und vertikale Spiegelungen.

4.4. Training und Evaluierung

Im Rahmen dieser Arbeit werden die unterschiedlichen synthetischen und realen Datensätze anhand der Performanz der damit trainierten Objekterkennungsverfahren auf Basis tiefer neuronaler Netze bewertet. Die Performanz wird dabei mithilfe von Metriken für Objekterkennungsaufgaben bewertet: die etablierten Metriken der COCO Challenge und spezifische Metriken für Aufgaben des Item Pickings, die Limited Selection-Metriken. Alle verwendeten Metriken eignen sich sowohl bei der Lokalisierung der Instanzen über Begrenzungsrahmen als auch über pixelweise Masken. Im Folgenden werden der Trainingsprozess erläutert, die Vor- und Nachteile der COCO Metriken diskutiert und die Limited Selection-Metriken eingeführt.

4.4.1. Trainingsprozess

Das Training und die Evaluierung der Objekterkennungsverfahren auf Basis von tiefen neuronalen Netzen (hier CNNs) stellt jeweils den Kern der im

4. Methoden

Rahmen dieser Arbeit durchgeführten Experimente dar, um die Güte der generierten und modifizierten Datensätze zu messen. Die Untersuchung der CNN-Modelle, der Trainingsprozesse und der zugehörigen Hyperparameter sollen dagegen nicht primärer Bestandteil dieser Arbeit sein. Daher werden hierfür etablierte Standardverfahren und feste Hyperparametersätze, welche weitgehend der Empfehlung der Autoren bzw. Entwickler folgen, genutzt. Bei den verwendeten neuronalen Netzen handelt es sich um Mask-RCNN (IP-A und QP-A) und Faster R-CNN (IP-B), jeweils mit einem Inception-ResNet-V2 als Backbone-Modell. Um die Übertragung der Ergebnisse auf andere Modelle exemplarisch zu überprüfen, werden zusätzlich Versuche mit der wesentlich kleineren Architektur SSD-MobileNet-V2 durchgeführt. Dies soll ausschließen, dass die Erkenntnisse bezüglich der Qualität der Datensätze nur für eine spezielle Architektur gelten, kann jedoch keinen Beleg für eine generelle Übertragungsfähigkeit auf alle Architekturen darstellen. Es wird die Implementierung der TensorFlow Object Detection API verwendet³.

Im Rahmen von Vorstudien wurde eine geeignete Lernrate von 0.002 ermittelt, welche über die durchgeführten Experimente nicht verändert wird. Dabei werden im Rahmen dieser Arbeit keine optimalen Parametersätze je Modell und Datensatz benötigt, sondern lediglich Parameter, welche ein sinnvolles, stabiles und vergleichbares Training der CNNs ermöglichen. Beim Training der Modelle werden die öffentlich verfügbaren (initialen) Gewichtungen des TensorFlow Models Object Detection Repositories übernommen, welche jeweils auf dem COCO-Datensatz vortrainiert wurden.

Da der Trainingsprozess selbst zufallsabhängig ist, beispielsweise die Reihenfolge der Trainingsdaten und die Drop-Out-Schichten in neuronalen Netzen, schwanken die Ergebnisse bei Wiederholungen des Trainings mit denselben Parametern. Da diese Schwankungen stark genug sind, um den Einfluss einzelner Datensatzveränderungen zu überdecken, wird jedes Training mehrfach durchgeführt und die Ergebnisse der Evaluierungen (Performanzkennwerte) gemittelt. So kann eine unerwünschte, indirekte Suche nach optimalen Zufallszahlen vermieden und stattdessen die Güte des Trainingsdatensatzes selbst besser bewertet werden. Neben der mittleren Performanz ermöglicht dieses Vorgehen auch die Angabe der Varianz bzw. der Standardabweichung und des Vertrauensbereichs (Konfidenzintervall). Damit kann zusätzlich die Sicherheit über die Lage des gemessenen Mittelwertes beurteilt werden.

³GitHub TensorFlow Object Detection API:
github.com/tensorflow/models/tree/master/research/object_detection

4.4.2. COCO Challenge Metriken

Die Metriken der COCO Challenge stellen heute einen De-facto-Standard im Bereich der Objekterkennung und Instanzsegmentierung dar. Die primäre Metrik ist hierbei die Durchschnittspräzision (AP), welche in Kapitel Unterabschnitt 3.5.3 definiert ist. Im Folgenden werden die Vor- und Nachteile der AP diskutiert, aus welchen sich die Entwicklung der Limited Selection-Metriken ableitet.

Vorteile Die AP ist für Anwendungsfälle geeignet, bei welchen eine Klassifizierung und Lokalisierung mehrerer Objektinstanzen in einem Bild erforderlich ist. Dabei ist die AP unabhängig von der Anzahl, Größe und dem Typ der Objekte sowie der jeweiligen Objekterkennungsmethode und dem jeweiligen Anwendungsfallkontext [25]. Da die AP ohne die manuelle Wahl von Parametern auskommt (bspw. Sicherheits- oder IoU-Grenzwerte), sind die Bewertungsergebnisse unterschiedlicher Bildverarbeitungssysteme innerhalb desselben Testdatensatzes gut miteinander vergleichbar. Dies ist insbesondere bei der Arbeit mit Standarddatensätzen (bspw. ImageNet oder COCO) von hoher Relevanz, welche von einer Vielzahl von Forschern und Entwicklern genutzt werden. Ein weiterer Vorteil der AP ist, dass sie unterschiedliche Aspekte wie die Präzision, die Sensitivität und die Güte der Lokalisierung in einem einzigen Kennwert vereint (s. Unterabschnitt 4.4.2). Für spezifische Auswertungszwecke sind im Rahmen der COCO Challenge jedoch zusätzliche Metriken wie die APs mit fixen IoU-Grenzwerten oder APs für einzelne Klassen definiert.

Nachteile Obwohl sich die AP als De-facto-Standard im Bereich der Objekterkennung etabliert hat, weist sie relevante Schwächen auf.

Insbesondere durch den Fokus auf breite Einsetzbarkeit und Parameterfreiheit ergeben sich Nachteile in Bezug auf die Untersuchung spezifischer Anwendungsfälle, wie in der vorliegenden Arbeit insbesondere das Item Picking. In vielen Fällen besteht keine lineare Korrelation des Kennwerts mit der späteren Leistung des Bildverarbeitungssystems in der realen Anwendung, wobei die tatsächliche Güte durch die AP meist unterschätzt wird. Dies liegt insbesondere darin begründet, dass spezifische Randbedingungen des Anwendungsfalls nicht berücksichtigt werden, wie unter anderem Gütekriterien für die Lokalisierung (bspw. $\text{IoU} > 75\%$ für die Anwendung nicht notwendig) oder eine begrenzte Anzahl an relevanten Objektinstanzen pro Bild (bspw. nur eine Detektion für die Anwendung notwendig).

4. Methoden

Eine zentrale Schwäche der Metrik ist auch die mangelhafte Interpretierbarkeit, insbesondere durch die Integralbildung und die mehrfache Mittelwertbildung über mehrere IoU-Grenzwerte und Klassen. Dadurch lässt sich beispielsweise die folgende zentrale Frage im Falle des Item Pickings nicht beantworten: „Wie viele Kommissioniervorgänge hätten mit dem Bildverarbeitungssystem erfolgreich durchgeführt werden können?“.

Während eine gute Vergleichbarkeit unterschiedlicher Bildverarbeitungssysteme für denselben Testdatensatz gegeben ist, gilt dieser Umstand nicht für den Vergleich über unterschiedliche Testdatensätze hinweg. So hängt das erreichte Performanzniveau unter anderen von der Anzahl und der Ähnlichkeit der Klassen sowie der Anzahl an Objekten pro Bild ab. Insbesondere reale industrielle Anwendungsfälle weichen hierbei meist stark von gängigen Objekterkennungsdatensätzen wie COCO ab. Eine präzise Einschätzung der Güte des Bildverarbeitungssystems in der späteren praktischen Anwendung ist mit der AP kaum möglich.

Ein weiterer Kritikpunkt stellt die Irrelevanz der absoluten Sicherheitswerte der Klassifikationen (engl. scores) der CNNs dar, da für die Berechnung der AP eine reine Rangbildung statt eines Grenzwertes eingesetzt wird. Dadurch ergibt sich für CNNs auch bei stark unterschiedlichen absoluten Sicherheitswerten dieselbe AP, falls sich die ordinale Rangfolge der Detektionen nicht unterscheidet. Beim Einsatz in der realen Anwendung würde ein Bildverarbeitungssystem mit höheren Sicherheitswerten jedoch eindeutig bevorzugt. Da hier in der Regel feste Schwellwerte eingesetzt werden, um unsichere Detektion auszuschließen, kämen die beiden Systeme auch nicht auf dieselbe reale Performanz.

Darüber hinaus bedingt der Verzicht auf einen Schwellwert für die Klassifikationssicherheit, eine geringe Anzahl an nicht detektierten Objekten (FN). Bei einer ausreichend hohen maximalen Anzahl an Detektionen pro Bild geht diese gegen null und es kommt nur noch zu korrekten (TP) und falschen (FP) Detektionen. In der praktischen Automatisierungsanwendung haben nicht detektierte Objekte jedoch einen relevanten Vorteil gegenüber Falschdetektionen. Während Falschdetektionen potenziell zu Fehlhandlungen und Prozessstörungen führen (beispielsweise Fehlgriffe und Kollisionsfahrten von Robotern), können bei nicht detektierten Objekten Alternativstrategien angewandt werden (beispielsweise Behandlung eines anderen Objektes oder die Wiederholung der Bildaufnahme).

4.4.3. Limited Selection-Metriken

Um den Schwächen der Durchschnittspräzision zu begegnen, wurde im Rahmen dieser Arbeit spezifische Objekterkennungsmetriken für den Anwendungsfall des Item Pickings entwickelt, welche im Folgenden als *Limited Selection*-Metriken bezeichnet werden. Im Gegensatz zu den Metriken der COCO Challenge sind diese in der Regel nur innerhalb dieses spezifischen Anwendungsfalls sinnvoll einsetzbar.

Die Limited Selection-Metriken basieren auf der Annahme, dass in vielen Anwendungsfällen zwar eine klassische Objekterkennung mit Klassifikation und Lokalisierung aller relevanter Objektinstanzen im Bild eingesetzt wird, für den nachgelagerten Prozess jedoch nur eine begrenzte Auswahl (engl. *limited selection*) an Detektionen genutzt wird. So wird bei der automatisierten Kommissionierung häufig nur ein einziges Objekt pro Bild gegriffen und bei mehreren aufeinanderfolgenden Griffen dementsprechend für jedes Objekt ein neues Bild aufgenommen. Dies ist insbesondere aufgrund von potenziellen Verschiebungen der verbleibenden Objekte durch den Handhabungsprozess notwendig.

Die zentralen Metriken der Limited Selection sind die Erfolgsrate (success rate) $\bar{s}_n$ und die Fehlerrate (error rate) $\bar{e}_n$ der Erkennung bei der Berücksichtigung von maximal n Detektionen pro Bild. Die Rate an nicht detektierten Objekten $\bar{u}_n$ ergibt dabei als $\bar{u}_n = 1 - (\bar{s}_n + \bar{e}_n)$.

Im Gegensatz zur AP-Berechnung findet hierbei keine Aufteilung nach Klassen statt. Stattdessen wird eine Teilmenge aller Klassen $C' \subseteq C$ definiert, welche für eine Selektion in Frage kommen. Alle Detektionen der Klasse $c \in C'$ werden pro Bild absteigend nach dem Klassifikationssicherheitswert sortiert. Alle Detektionen anderer Klassen oder Detektionen mit Sicherheitswerten unterhalb eines vorgegebenen Schwellwertes werden verworfen. Im Gegensatz zur AP findet dabei keine Bild-übergreifende Rangbildung der Detektionen statt. Stattdessen wird die Übereinstimmung der Detektionen mit der Annotation für jedes Bild einzeln ausgewertet und sich ergebenden Kennwerte anschließend zusammengefasst.

Die Detektionen werden anhand der Annotation in korrekt-positiv (TP) und falsch-positiv (FP) kategorisiert, wobei neben der ermittelten Klasse ein fester, sich aus den jeweiligen Rahmenbedingungen des Anwendungsfalls ergebender IoU-Schwellwert verwendet wird. Die Verwendung eines festen IoU-Schwellwertes dient nicht nur der Unterdrückung unsicherer Detektionen, sondern vermeidet darüber hinaus eine Mittelwertbildung über mehrere Schwellwerte hinweg und trägt so zur leichteren Interpretierbarkeit

4. Methoden

bei. Basierend auf dieser Kategorisierung wird für jedes Bild der Index der ersten falsch-positiven Detektion $i_{FP} \in \mathbb{N}_0 \cup \{\infty\}$ ermittelt, wobei für $|FP| = 0$ gilt: $i_{FP} = \infty$. Die Anzahl der korrekt-positiven Detektionen $|TP|$ wird auf i_{FP} begrenzt, da der Systemzustand ab der ersten falsch-positiven Detektion als fehlerhaft angesehen werden kann und nicht sichergestellt ist, dass folgende Detektionen korrekt gehandhabt werden können. Für die Berechnung mit maximal n Detektionen pro Bild werden nur die Datenpunkte $B' \subseteq B$ berücksichtigt, für welche die Anzahl der annotierten Objektinstanzen $|GT| \geq n$ ist. Dies liegt darin begründet, dass es für Bilder mit $|GT| < n$ unmöglich ist n Objektinstanzen zu finden.

Die Erfolgsrate $\bar{s}_n$ und Fehlerrate $\bar{e}_n$ für die ersten n Objektinstanzen ergeben sich wie folgt:

$$\bar{s}_n = \frac{1}{|B'|} \sum_{k=1}^{|B'|} s_n(k) \quad (4.26)$$

mit

$$s_n(k) = \begin{cases} 1, & \text{falls } \min(|TP(k)|, i_{FP}(k)) \geq n \\ 0, & \text{sonst} \end{cases} \quad (4.27)$$

$$\bar{e}_n = \frac{1}{|B'|} \sum_{k=1}^{|B'|} e_n(k) \quad (4.28)$$

mit

$$e_n(k) = \begin{cases} 1, & \text{falls } i_{FP}(k) < n \\ 0, & \text{sonst} \end{cases} \quad (4.29)$$

Die Raten $\bar{s}_n$, $\bar{e}_n$ und $\bar{u}_n$ stellen einfache Mittelwerte des Auftretens von erfolgreichen, fehlerhaften und ausgebliebenen Detektionsentscheidungen dar. Dadurch sind diese wesentlich leichter zu interpretieren als die Durchschnittspräzision, welche sich aus einer approximierten Integralbildung der Präzision über der Sensitivität ergibt.

5. Experimente und Ergebnisse

Im Folgenden werden die Experimente und Ergebnisse der Voruntersuchungen, der Untersuchung der Datengenerierung und der zugehörigen Szeneneigenschaften sowie der Studien zu den Methoden der Domain Randomization, der Domain Adaptation und deren Kombination dargestellt. Die Experimente werden primär anhand des Anwendungsfalls IP-A durchgeführt. An geeigneter Stelle und mit expliziter Nennung finden Übertragungen auf die Anwendungsfälle IP-B und QP-A statt. Teilergebnisse der Experimente zum Anwendungsfall IP-B wurden in ähnlicher Weise bereits vom Autor veröffentlicht [119]. Dies wird an den entsprechenden Stellen gesondert gekennzeichnet.

Es findet jeweils ein Vergleich der mit diesen Methoden erzeugten bzw. veränderten synthetischen Bilddaten mit auf andere Weise erzeugten bzw. unveränderten synthetischen Bilddaten statt. Dabei wird überprüft, mit welchen der jeweiligen Methoden sich synthetische Bilddaten generieren lassen, die zu einer signifikant höheren mittleren Performanz der mit diesen Daten trainierten neuronalen Netzen für Objekterkennung und Instanzsegmentierung führen. Die Performanz wird dabei im Sinne der relevanten Bildverarbeitungsmetriken definiert (s. Abschnitt 4.4). Zur besseren Lesbarkeit wird im Folgenden auf diese ausführliche Beschreibung bei der Formulierung der Hypothesen verzichtet und die Formulierung „bessere Bilddaten“ verwendet.

Die Performanz wird, falls nicht explizit abweichend benannt, durch die AP-Metrik gemessen (s. Unterabschnitt 3.5.3). Um die Vergleichbarkeit zwischen den unterschiedlichen Anwendungsfällen mit und ohne Segmentierung zu erhöhen, wird die AP jeweils auf Basis der Begrenzungsrahmen berechnet. Im Rahmen der Experimente werden jeweils Wiederholungen der Trainingsdurchläufe durchgeführt und das arithmetische Mittel sowie gegebenenfalls die korrigierte Standardabweichung der gemessenen Performanz-Werte angegeben. Aus Gründen der Vereinfachung wird im Weiteren jedoch auf diesen Zusatz verzichtet und die Bezeichnung AP verwendet.

Der notwendige Stichprobenumfang wurde vor der Durchführung der primären Experimente abgeschätzt (s. Unterabschnitt 5.1.3). Falls nicht

5. Experimente und Ergebnisse

anderweitig spezifiziert, werden Trainings mit 50 000 oder weniger Schritten 15 Mal und Trainings mit mehr als 50 000 Schritten 7 Mal wiederholt. In den zugehörigen Abbildungen werden jeweils die Konfidenzintervalle zum Konfidenzniveau 95 % durch Antennen (engl. whiskers) oder bei zusammenhängenden Mittelwerten durch semitransparent gefüllte Bereiche dargestellt.

Im Gegensatz zur Definition der COCO Challenge wird dabei nicht der Wertebereich $[0, 1]$, sondern eine Prozent-Darstellung $[0 \%, 100 \%]$ verwendet. Hierbei ist zu beachten, dass es sich bei dieser Angabe dennoch um einen absoluten Wert und nicht etwa eine relative Angabe bezüglich einer Basis-AP darstellt. Wird die Performanz zweier Datensätze direkt miteinander verglichen, wird neben der Angabe der jeweiligen AP auch die absolute Differenz beider APs verwendet und als ΔAP bezeichnet. Den Vergleichsdatsatz stellt hierbei entweder der vorher definierte Basis-Datsatz oder ein anderer explizit genannter Datsatz dar. Im Falle eines explizit genannten Vergleichsdatsatzes wird in der Regel die Formulierung „im direkten Vergleich“ gewählt.

Die Formulierung „signifikanter“ Unterschiede wird nur verwendet, falls die Prüfung mit einem Hypothesentest zum Signifikanzniveau 5 % zu einem signifikanten Ergebnis führt ($p\text{-Wert} < 5 \%$). Im Falle der Überprüfung von Performanz-Mittelwertsunterschieden wird im Folgenden der Welch-Test genutzt (s. Unterabschnitt 3.4.2).

5.1. Voruntersuchungen

Im Folgenden werden die Voruntersuchungen zur Prämisse der Normalverteilung innerhalb der Messreihen sowie zur Schwankungsbreite der Performanzkennwerte bei Wiederholungen des Trainings und der Datengenerierung vorgestellt. Dabei wurden jeweils Trainingsreihen mit mindestens 15 Wiederholungen (1 106 von 1 560 Messreihen) und mit einer AP von mindestens 10 % berücksichtigt (1 094 von 1 106 Messreihen). Falls nicht anderweitig spezifiziert wurden diese auf die ersten 15 Messwerte begrenzt. Messreihen mit sehr niedriger AP wurden nicht berücksichtigt, da Objekterkennungssystemen mit wenigen Klassen hierbei zunehmend rein zufällig zu erfolgreichen Detektionen kommen.

5.1.1. Verteilung

Um die in Abschnitt 3.4 vorgeschlagenen Hypothesen- und Signifikanztests anwenden zu können, muss die Prämisse erfüllt sein, dass die einzelnen Messwerte einer Messreihe (Trainingswiederholungen) zumindest näherungsweise normal verteilt sind. Die berücksichtigten Messreihen wurden bei einem Signifikanzniveau von 5 % jeweils getrennt mit dem Shapiro-Wilk-Test auf Normalverteilung überprüft. Entsprechend dem Signifikanzniveau ist hierbei zu erwarten, dass die Hypothese in ca. 5 % der Fälle (etwa 55 Trainingsreihen) abgelehnt wird, falls die Messreihen normalverteilt sind. Höhere Ablehnungsraten deuten darauf hin, dass über die Messreihen hinweg keine Normalverteilung vorliegt.

Die für die Messreihen ermittelte Ablehnungsrate von 6.4 % liegt knapp über der erwarteten Rate von 5 %. Für Messreihen mit einer hohen AP von 80 % oder höher (979 von 1094 Messreihen) liegt die Ablehnungsrate mit 5.82 % nochmals wesentlich näher an der Ablehnungsrate der Normalverteilung. Zur Einordnung wurden Vergleichsanalysen mit zufällig erzeugten Stichproben desselben Umfangs von 1094 Messreihen mit je 15 Messwerten durchgeführt (Monte-Carlo-Simulation). Die Stichproben wurden hierbei aus Zufallsvariablen gezogen, die der Normal-, der Student-t-, der Gleich- oder der Chi2-Verteilung folgen. Dieses Vorgehen wurde für jede Verteilung 1000 Mal wiederholt, wodurch sich jeweils 1000 Ablehnungsraten ergeben. Je nach Verteilung liegt die mittlere Ablehnungsrate für

- die Normalverteilung bei 4.96 % (3.20 % bis 7.31 %),
- die Student-t-Verteilung bei 7.18 % (5.12 % bis 9.60 %),
- die Gleichverteilung bei 12.80 % (9.96 % bis 16.00 %),
- die Chi2-Verteilung bei 14.03 % (9.69 % bis 17.09 %).

Es wird ersichtlich, dass die Ablehnungsrate der tatsächlichen Messreihen von 6.4 % nahe an der erwarteten Ablehnungsrate von 5 % und unterhalb der ermittelten Ablehnungsrate der Student-t-Verteilung liegt. Zur Veranschaulichung ist die Verteilungsdichte der Performanzkennwerte in Abbildung 5.1 dargestellt. Für die Bildung des Histogramms wurden die Messwerte jeweils innerhalb einer Trainingsreihe standardisiert.

Exemplarisch wurden vier Trainingsreihen, die der SE-Opt- und SE-Exp-Basisdatensätze im IP-A-Anwendungsfall bei 2 und 9 bzw. 18 und 40 Trainingsepochen, 50 Mal wiederholt und die Performanzkennwerte mit

5. Experimente und Ergebnisse

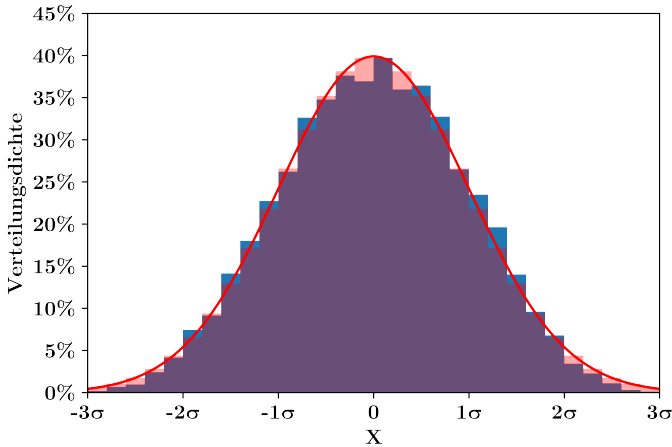


Abbildung 5.1.: Die gemessene Verteilungsdichte der Performanzkennwerte für 1 094 standardisierte Trainingsreihen (blau) sowie die erwartete Verteilungsdichte bei einer Standardnormalverteilung (rot).

dem Shapiro-Wilk-Test überprüft. Mit Signifikanzwerten (p-Werten) von 0.538 und 0.135 bzw. 0.661 und 0.798 weichen die beiden Messreihen nicht signifikant von der Normalverteilung ab.

Da die Hypothese der Normalverteilung somit für die tatsächlichen Messreihen nicht abgelehnt werden kann, wird im Weiteren davon ausgegangen, dass die Messwerte innerhalb der Trainingsreihen näherungsweise normalverteilt sind.

5.1.2. Stichprobenumfang

Grundsätzlich gilt, dass die Sicherheit bezüglich der Ergebnisse empirischer Experimente (Konfidenz der Hypothesen-Tests) mit zunehmendem Stichprobenumfang der Messreihen steigt. Da der Zeitbedarf für die Erzeugung einer Stichprobe beim maschinellen Lernen jedoch hoch und die benötigten Ressourcen begrenzt sind, muss ein geeigneter Kompromiss zwischen Konfidenz und Zeitbedarf gefunden werden. Vor der Durchführung der primären Experimente und der Kenntnis der sich daraus tatsächlich ergebenden mittleren Schwankungsbreiten (s. Unterabschnitt 5.1.3) wurden a priori Annahmen

zur Varianz auf Basis von zwei exemplarischen Trainingsreihen abgeschätzt (SE-Opt-Datensatz und SE-Exp-Datensatz des Anwendungsfalls IP-A). Als Prämissen für die Ermittlung des notwendigen Stichprobenumfangs wurden ein Konfidenzniveau $1 - \alpha = 95\%$ und ein minimaler Mittelwertunterschied ω_0 von 0.5% für einen signifikanten Welch-Test zugrunde gelegt.

Für zwei Stichproben X und Y mit demselben Stichprobenumfang $n = m$ und derselben Standardabweichung $s_X = s_Y = s$ ergibt sich der notwendige Stichprobenumfang n nach dem Welch-Test (s. Unterabschnitt 3.4.2; entspricht hier dem Zweistichproben-t-Test) als:

$$\sqrt{\frac{n}{2}} \frac{\omega_0}{s} > t_{1-\alpha; 2(n-1)} \quad (5.1)$$

$$n > 2 \left(\frac{s}{\omega_0} t_{1-\alpha; 2(n-1)} \right)^2 \quad (5.2)$$

Erste Schätzungen der Standardabweichung von 0.73% und 0.72% wurden durch 50 Wiederholungen des Trainings und der Evaluierung mit dem IP-A SE-Opt-Datensatz bei 2 und bei 9 Trainingsepochen ermittelt, wodurch sich ein notwendiger Stichprobenumfang von mindestens 13 ergibt. Aufgrund der vergleichsweise kurzen Trainingsdauer wurde die Anzahl der Trainingswiederholungen für 50 000 und weniger Schritte leicht erhöht und auf 15 festgelegt. Für Datensätze mit späterer Saturation wurden mit dem Datensatz IP-A SE-Exp nach 18 und 40 Trainingsepochen bei 50 bzw. 15 Wiederholungen Standardabweichungen von 0.49% und 0.45% ermittelt, wodurch sich ein notwendiger Stichprobenumfang von mindestens 7 bzw. 6 ergibt. Die Anzahl der Trainingswiederholungen für mehr als 50 000 Trainingsschritten wurde dementsprechend auf 7 festgelegt. Alternativ hätte der notwendige Stichprobenumfang auch über die Vorgabe einer gewünschten Teststärke ermittelt werden können (im vorliegenden Fall 56% bis 63%).

5.1.3. Schwankungsbreite

Die mittlere Standardabweichung je Messreihe beträgt 0.68% (0.24% bis 4.6%). Dabei besteht eine klare negative Korrelation zwischen dem Mittelwert und der Standardabweichung der Performanz (Korrelationskoeffizient -0.79). Im Mittel gilt: Je niedriger die erreichte Performanz, desto größer die Streuung der Stichprobe. Bei einem Stichprobenumfang von $n = 15$ weisen die Konfidenzintervalle zum Konfidenzniveau 95% entsprechend jeweils

5. Experimente und Ergebnisse

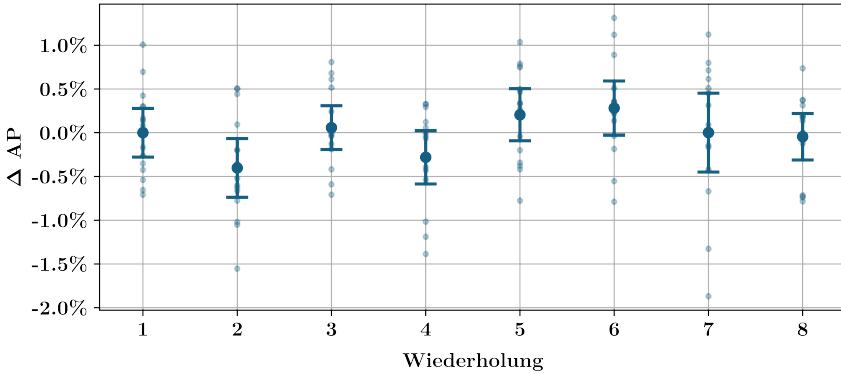


Abbildung 5.2.: Wiederholungen der Datengenerierung für dieselbe Simulationsszene (SE-Exp des Anwendungsfalls IP-A) mit jeweils anderen Zufallszahlen.

einseitig (halbe Breite) eine mittlere Spanne von 0.37 % auf.

Um die Konstanz der Ergebnisse bei einer Wiederholung der Datengenerierung (Wiederholbarkeit) zu untersuchen, wurde die Erzeugung mit derselben Simulationsszene (SE-Exp des Anwendungsfalls IP-A) exemplarisch 8-mal mit unterschiedlichen Zufallszahlen wiederholt. Das Training wurde für jeden Datensatz 15-mal mit 9 Epochen wiederholt, wodurch sich 8 Messreihen mit jeweils 15 Messwerten ergeben. Die Ergebnisse sind in Abbildung 5.2 dargestellt.

Werden die 120 Messwerte als eine einzige Stichprobe betrachtet, so liegt die Standardabweichung der APs mit 0.59 % nur knapp über der mittleren Standardabweichung der einzelnen Messreihen von 0.56 %. Die Spannweite der Mittelwerte liegt mit 0.68 % (zwischen Wdh. 2 und 6) weit unterhalb der mittleren Spannweite der Messwerte innerhalb der Messreihen (1.93 %). Diese Eigenschaften sprechen für die Ähnlichkeit der Messreihen. Dennoch zeigten sich bei der Überprüfung mittels wechselseitiger Welch-Tests und gepaarten t-Tests zum Konfidenzniveau von 95 % mehrere signifikante Mittelwertunterschiede, wobei die Wiederholungen 2 und 4 die signifikant niedrigsten Kennwerte aufwiesen.

Diese signifikanten Unterschiede bei den Wiederholungen der Datengenerierung belegen die Relevanz der Maßnahmen zur Sicherstellung der Vergleichbarkeit bei der Untersuchung einzelner Eigenschaften (s. Abschnitt 4.1).

5.2. Datengenerierung und Szenenerstellung

Im Folgenden werden unterschiedliche Methoden der synthetischen Datengenerierung miteinander und mit realen Bilddatensätzen verglichen. Dabei werden jeweils verschiedene Ausprägungen des kontextarmen Standardverfahrens „Unity SynthDet“ und der Simulationsoptimierung mittels Scene Engineerings betrachtet. Es wird untersucht, welche Methoden zu besseren Trainingsdatensätzen für die Objekterkennung führen und welche Szeneneigenschaften dabei signifikanten Einfluss haben. Während die neuronalen Netze im Rahmen des Vergleichs zwischen SynthDet und Scene Engineering mit bis zu 40 Epochen trainiert wurden, wurden die Untersuchungen zu einzelnen Szeneneigenschaften jeweils mit 9 Epochen durchgeführt.

5.2.1. SynthDet und Scene Engineering

Hypothese

Eine durch **Scene Engineering** optimierte Simulation erlaubt es im Vergleich zum **Standardverfahren** SynthDet, bessere synthetische Bilddaten zu erzeugen.

Im Rahmen der Studie zu den Methoden der Datengenerierung wurden die folgenden sechs Datensätze ohne weitere Modifikation verglichen:

1. Standardverfahren SynthDet (SD)
 - a) SD-Preopt: Hintergrund durch zufällig ausgewählte Texturen, welche sowohl abstrakte Muster als auch Realbilder umfassen
 - b) SD-Opt: Wie SD-Preopt, jedoch mit optimierten Objektgrößen
 - c) SD-Std: Wie SD-Opt, jedoch Hintergrund durch randomisierte 3D-Modelle (SynthDet Standardverfahren)
2. Scene Engineering (SE)
 - a) SE-Preopt: Simulation durch einen Anwender ohne Erfahrung im Bereich der synthetischen Datengenerierung erstellt
 - b) SE-Opt: Wie SE-Preopt, jedoch mit Optimierung anhand des direkten Vergleichs mit wenigen Referenzbildern
 - c) SE-Exp: Simulation durch einen Anwender mit Erfahrung im Bereich der synthetischen Datengenerierung (Experte) erstellt

5. Experimente und Ergebnisse

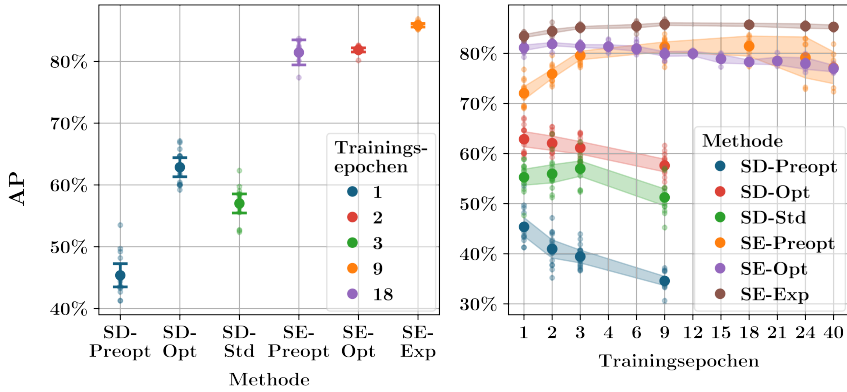


Abbildung 5.3.: Ergebnisse der Studie zu den Datengenerierungsmethoden SynthDet (SD) und Scene Engineering (SE) in jeweils drei Varianten.

Pre-Evaluierungsdatensatz: Wie aus der Abbildung 5.3 (links) hervorgeht, verbesserte sich die Performanz der SynthDet-Datengenerierung durch die manuelle Optimierung der relativen Objektgrößen signifikant von AP 45.4 % (SD-Preopt) auf AP 62.9 % mit dem SD-Opt-Datensatz (ΔAP 17.5 %). Die Verwendung des im Standardverfahren vorgesehenen stark randomisierten Hintergrunds aus 3D-Modellen statt Hintergrundtexturen (Datensatz SD-Std mit AP 57.0 %) führte dagegen im direkten Vergleich mit dem SD-Opt-Datensatz zu einem signifikanten Performanzrückgang um ΔAP -5.9 %. Die höchste Performanz des Datensatzes SD-Opt wurde bereits nach der ersten Trainingsepoche erreicht (s. Abbildung 5.3 rechts).

Die Performanz aller drei mittels Scene Engineering erzeugten synthetischen Trainingsdatensätze (SE-Preopt, SE und SE-Exp) war signifikant höher als die der SynthDet-Datensätze (s. Abbildung 5.3 links). In Bezug auf den manuellen Optimierungsschritt mit wenigen Referenzbildern zwischen SE-Preopt (AP 81.5 %) und SE-Opt (AP 81.9 %) war kein signifikanter Unterschied in Bezug auf die maximal erreichte Performanz zu verzeichnen. Abbildung 5.3 (rechts) zeigt jedoch, dass die höchste Performanz beim SE-Preopt-Datensatz nach 18 Epochen und damit wesentlich später als beim SE-Opt-Datensatz (nach zwei Epochen) erreicht wurde. Die signifikant besten Ergebnisse erzielte der SE-Exp-Datensatz mit einer Performanz von

AP 85.8 % nach 9 Trainingsepochen.

Die Schwankungen der Performanz bei der Wiederholung der Trainings reduzieren sich durch die jeweilige manuelle Optimierung der Datengenerierung deutlich. So waren die Standardabweichungen bei SE-Exp und SE-Opt (jeweils 0.5 %) trotz ähnlicher Performanz wesentlich geringer als beim SE-Preopt-Datensatz (2.2 %). Mit Standardabweichungen von 2.8 % (SD-Opt und SD-Std) und 3.4 % (SD-Preopt) weisen die SynthDet-Datensätze nochmals höhere Schwankungen auf. Aufgrund der einheitlichen Stichprobengröße und des selben Konfidenzniveaus sind diese Unterschiede direkt an den Konfidenzintervallen in Abbildung 5.3 ersichtlich.

Da die Saturation bei den SE-Datensätzen erst nach mehreren Epochen eintrat, wurden jeweils Versuche mit mehr Trainingsdaten und einer entsprechend reduzierten Anzahl an Trainingsepochen durchgeführt. Das Training umfasst dabei jeweils dieselbe Anzahl an Trainingsschritten, jedoch mit weniger Iterationen pro Bild. Weder bei einer Erhöhung des Datenumfangs von 5 000 auf 15 000 (SE-Preopt und SE-Opt) noch bei einer Erhöhung auf 45 000 Trainingsdaten (SE-Exp) konnten signifikante Unterschiede gemessen werden (s. Anhang C.1).

Post-Evaluierungsdatensatz: Bei der Evaluierung mit dem Post-Datensatz ergaben sich zwar keine Änderungen der ordinalen Rangfolge der Datensätze in Bezug auf deren Performanz, jedoch war die erreichte AP durchgängig höher als beim Pre-Datensatz (s. Anhang C.2). Auch hier erreichten der SD-Opt- (AP 67.2 %) und der SE-Exp-Datensatz (AP 87.7 %) die besten Ergebnisse der jeweiligen Datengenerierungsverfahren. Darüber hinaus war der Performanzunterschied zwischen den Datensätzen SE-Preopt (AP 84.2 %) und SE-Opt (AP 84.7 %) größer als beim Pre-Evaluierungsdatensatz und damit signifikant.

IP-B-Anwendungsfall: Die Überlegenheit von Scene Engineering gegenüber einer stark randomisierten, jedoch kontextarmen Simulation konnte für den IP-B-Anwendungsfall ebenfalls bestätigt werden (s. Anhang C.3) [119]. Hierbei erreichte der SE-Datensatz eine signifikant höhere Performanz als der kontextarme Datensatz (vergleichbar mit SynthDet beim Anwendungsfall IP-A) – sowohl bei der Evaluierung mit dem zum Trainingsdatensatz nahen Datensatz (Δ AP 56.6 % im direkten Vergleich) als auch mit dem fernen Datensatz (Δ AP 24.1 % im direkten Vergleich). Auch die Standardabweichungen der gemessenen Performanzkennwerte sind beim

5. Experimente und Ergebnisse

SE-Datensatz mit 1.1 % (nah) und 1.2 % (fern) wesentlich geringer als beim kontextarmen Datensatz mit jeweils 2.9 %. Damit sind die Performanzunterschiede beim IP-B-Anwendungsfall noch stärker ausgeprägt als beim IP-A-Anwendungsfall.

Zusammenfassung

Aufgrund der Studienergebnisse kann die **Hypothese** für die untersuchten Anwendungsfälle **bestätigt** werden. Die durch Scene Engineering optimierten Simulationen erreichten, unabhängig von der Expertise des Erstellers, signifikant bessere Ergebnisse als das Standardverfahren SynthDet. Die Optimierung der Scene Engineering-Szene mit realen Referenzdaten (SE-Opt) führte nicht zu signifikant besseren Ergebnissen als die Szene ohne diese Optimierung (SE-Preopt), jedoch konvergierte das Training mit diesen Daten früher.

5.2.2. Szenenstruktur

Hypothese

Eine **realitätsnahe Nachbildung** der Szene in Bezug auf das **Strukturelement** {1. Ladungsträger, 2. bekannte Umgebungsobjekte, 3. Hintergrund} erlaubt es, bessere synthetische Bilddaten zu erzeugen als **Szenen ohne** diese realitätsnahe Nachbildung.

Für die Studie zur Szenenstruktur wurden 16 Datensätze mit unterschiedlicher Sichtbarkeit der Struktur- und Randomisierungselemente erstellt. Dabei wurden alle Kombinationen der folgenden binären Eigenschaften untersucht:

1. Sichtbarkeit Ladungsträger
2. Sichtbarkeit realitätsnaher Umgebungsobjekte (Tisch)
3. Hintergrund realitätsnah einfarbig oder zufällig texturiert
4. Sichtbarkeit zufälliger Störobjekte im Hintergrund (randomisiert texturierte und positionierte geometrische Grundkörper)

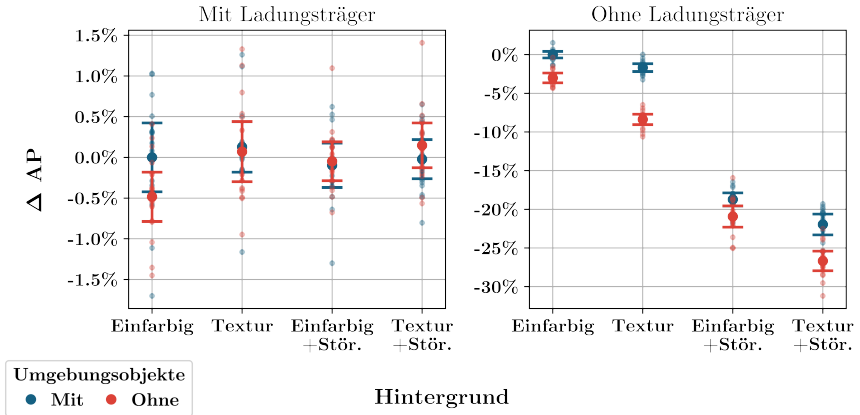


Abbildung 5.4.: Ergebnisse der Studie zur Szenenstruktur für unterschiedliche Hintergründe mit (links) und ohne (rechts) Ladungsträger. Es wird jeweils die Veränderung der Performanz gegenüber der Basis mit einfarbigem Hintergrund und sichtbaren Umgebungsobjekten angegeben.

Darüber hinaus wurden neun Datensätze zur spezifischen Untersuchung der Eigenschaften des Ladungsträgers (3D-Modell, Farbe und Textur) erstellt:

1. Ladungsträger, jeweils mit originalem 3D-Modell (Karton; Standard), mit Kartontextur mit (a) brauner Farbe (Standard), (b) reduzierter Helligkeit (45 %), (c) schwach randomisierter Farbe (90-100 % Helligkeit), (d) blauer Farbe und (e) voll randomisierter Farbe sowie mit (f) Holztextur (ca. 80 % Helligkeit)
2. Ladungsträger, jeweils mit abweichendem 3D-Modell (Kunststoffbehälter) mit (a) Kartontextur, (b) blauer Farbe und (c) randomisierter Farbe.

Die Vergleichsbasis stellen jeweils die Datensätze mit einfarbigem Hintergrund und sichtbaren Umgebungsobjekten dar.

Pre-Evaluierungsdatensatz: Die Studienergebnisse zeigen die hohe Relevanz der Sichtbarkeit des Ladungsträgers für die Performanz der trainierten

5. Experimente und Ergebnisse

Objekterkennungssysteme. Der Unterschied zwischen der niedrigsten Performanz mit sichtbarem Ladungsträger und der höchsten Performanz ohne Ladungsträger ist mit ΔAP 2.7 % signifikant (nicht abgebildet).

Bei sichtbarem Ladungsträger (s. Abbildung 5.4 links) konnten mit Ausnahme einer Variante (einfarbiger Hintergrund ohne Umgebung mit ΔAP -0.5 % zur Basis) keine signifikanten Performanzunterschiede in Abhängigkeit von Umgebung und Hintergrund gemessen werden (maximaler Performanzunterschied ΔAP 0.2 %). Für die Ausnahme kann vermutet werden, dass das Objekterkennungssystem durch die geringere Komplexität der Umgebung und des Hintergrunds im Trainingsdatensatz reale Umgebungselemente im Pre-Datensatz häufiger fälschlicherweise als Objekte klassifiziert (falsch-positive Detektionen).

Beim direkten Vergleich zwischen Varianten ohne und mit einer bestimmten Eigenschaft konnte bei den Datensätzen ohne sichtbaren Ladungsträger (s. Abbildung 5.4 rechts) Folgendes festgestellt werden:

1. Sichtbare Umgebungsobjekte führten jeweils zu einer signifikant höheren Performanz
2. Einfarbige Hintergründe führten jeweils zu einer signifikant höheren Performanz als zufällige Hintergrundtexturen
3. Sichtbare Störobjekte führten jeweils zu einer signifikant niedrigeren Performanz

Im Rahmen der Untersuchung der Ladungsträgererscheinung stellt der Datensatz mit realistischem 3D-Modell und realitätsnaher Textur (Braunton) des Ladungsträgers die Basis dar. Wie Abbildung 5.5 (links) zeigt, reduzierte sich die Performanz bei Verwendung eines abweichenden statt des realistischen 3D-Modells jeweils signifikant um ΔAP -1.6 % bei randomisierter Kolorierung bzw. um ΔAP -1.5 % bei realitätsnaher Färbung des Kartons. Die Wahl eines fix abweichenden Farbtons (blau) führte jeweils zu signifikant schlechteren Ergebnissen, wobei die Verwendung des realistischen 3D-Modell hierbei keine signifikante Rolle mehr spielte und keine signifikanten Unterschiede zum Datensatz ohne sichtbaren Ladungsträger aufwies (Performanzunterschied ΔAP -3.8 % bis ΔAP -4.1 % zur Basis).

Die Ergebnisse der detaillierten Untersuchung der Farbe und Textur des sichtbaren Ladungsträgers mit korrektem 3D-Modell sind in Abbildung 5.5 (rechts) dargestellt. Die signifikant besten Ergebnisse wurden mit der Originaltextur bei leicht randomisiertem braunem Farbton (ΔAP

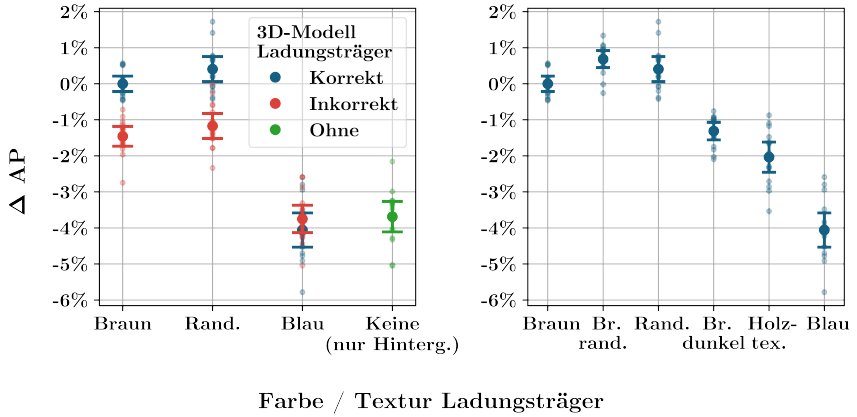


Abbildung 5.5.: Ergebnisse der Studie zum Ladungsträger mit unterschiedlichen Farben und 3D-Modellen (links) sowie mit unterschiedlichen Farben und Texturen (rechts).

0.6 % zur Basis) sowie bei voll randomisierter Farbe erreicht (ΔAP 0.4 % zur Basis). Die Verwendung einer abweichenden Textur (braunes Holzmuster mit ca. 80 % Helligkeit) führte dabei mit ΔAP -2.1 % zu wesentlich stärkeren Performanzeinbußen als ein dunklerer Brauntönen (45 % Helligkeit) unter Beibehaltung der Originaltextur mit ΔAP -1.3 % (jeweils zur Basis). Die größte Performanzreduktion im Vergleich zur Basis ergab sich für den statisch abweichenden blauen Farbton (ΔAP -4.1 %).

Post-Evaluierungsdatensatz: Die Evaluierungen mit dem realen Post-Datensatz bestätigen die Ergebnisse der Szenen mit Ladungsträger (insignifikant mit derselben Ausnahme) sowie den Einfluss der Umgebung und der Störobjekte für Szenen ohne Ladungsträger (s. Anhang C.4). Im Unterschied zum Pre-Evaluierungsdatensatz erhöhte sich die Performanz jedoch signifikant, falls bei Szenen ohne Ladungsträger und Störobjekte eine zufällige Hintergrundtextur statt einem einfarbigen Hintergrund genutzt wurde (im direkten Vergleich ΔAP 5.8 % mit und ΔAP 5.0 % ohne sichtbaren Maschinentisch). Es kann hierbei vermutet werden, dass der Post-Datensatz mehr ablenkende Umgebungselemente aufweist und das Objekterkennungssystem durch die geringere Komplexität des einfarbigen

5. Experimente und Ergebnisse

Hintergrunds anfälliger für falsch-positive Detektionen wird. Die Ergebnisse in Bezug auf die Erscheinung des Ladungsträgers konnten weitgehend bestätigt werden (s. Anhang C.5). Eindeutige Unterschiede waren zum einen der stärkere negative Einfluss eines abweichenden 3D-Modells bei abweichender Farbe. Zum anderen war der negative Einfluss durch eine abweichende Holztextur des Ladungsträgers zwar signifikant, jedoch wesentlich schwächer ausgeprägt als beim Pre-Datensatz ($\Delta AP -0.5\%$ zur Basis).

IP-B-Anwendungsfall: Die hohe Relevanz des Ladungsträgers als Strukturelement zeigte sich auch beim IP-B-Anwendungsfall (s. Anhang C.6) [119]. Während das Entfernen der Umgebungsobjekte (Tisch) bei sichtbarem Ladungsträger mit $\Delta AP -0.6\%$ keinen signifikanten Einfluss auf die Performanz hatte, reduzierte sich die Performanz ohne die Darstellung des Ladungsträgers signifikant um $\Delta AP -41.7\%$ mit bzw. $\Delta AP -45.6\%$ ohne Umgebungsobjekte (jeweils zur Basis). Die Darstellung von Störobjekten im Hintergrund führte jeweils zu einer Reduktion der Performanz. Diese Ergebnisse des IP-B-Anwendungsfalls bestätigen damit die des IP-A-Anwendungsfalls.

Zusammenfassung

Aufgrund der Studienergebnisse kann die **Hypothese** für die untersuchten Anwendungsfälle **bestätigt** werden. Eine realitätsnahe Nachbildung der untersuchten Strukturelemente führte zu signifikant besseren synthetischen Trainingsdaten. Die höchste Relevanz hatten hierbei die Sichtbarkeit und die realitätsnahe Erscheinung des Ladungsträgers. Als weniger relevant hat sich die Nachbildung der Umgebungsobjekte erwiesen. Dagegen hatte der Einsatz zufälliger Störobjekte keinen signifikanten oder sogar negativen Einfluss auf die Performanz der trainierten Objekterkennungssysteme.

5.2.3. Sichtfeld

Hypothese

Simulationsszenen mit einer stark an den **realen Anwendungsfall angepassten** (realistischen) **Kamerapose** erlauben es, bessere synthetische Bilddaten zu erzeugen als Szenen, bei denen die **fehlende Anpassung** der Kamerapose **durch Randomisierung ausgeglichen** wird.

Für die Studie zum Sichtfeld wurden insgesamt 37 Datensätze auf Basis der SE-Simulation generiert. Die Datensätze unterscheiden sich zum einen in Bezug auf den Kameraabstand und -blickwinkel und auf die Zentriertheit des Ladungsträgers im Bild. Zum anderen weisen die Datensätze jeweils bis zu drei unterschiedliche Randomisierungsgrade der 6D-Pose auf (Translation und Rotation mit jeweils drei Freiheitsgraden). Weitere Kameraparameter wie Öffnungswinkel oder Brennweite wurden hierbei nicht verändert. Exemplarische Abbildungen finden sich in Anhang B.3 und Anhang B.4.

1. Basis: Manuelle Nachbildung der Kamerapose mittels Scene Engineering, wobei die Kamera zentriert senkrecht von oben aus einer Entfernung ca. 1 m auf den Ladungsträger gerichtet ist
2. Abstand: Kamerapose wie Basis, jedoch im Abstand von jeweils ca. 0.75 m, 1.25 m, 1.5 m, 2.0 m und 2.5 m zum Ladungsträger
3. Zentriertheit: Kamerapose wie Basis, jedoch mit zur Blickrichtung horizontaler und vertikaler Verschiebung im Abstand von ca. 1.5 m und 2.0 m (Ladungsträger verbleibt im Bild)
4. Fixe Blickwinkelveränderung: Blickwinkelveränderung auf zwei, in allen sechs Dimensionen stark von der Realität abweichende, Posen im Abstand von ca. 1.0 m und 2.0 m (Ladungsträger verbleibt im Bild)
5. Dynamische Blickwinkelveränderung: Randomisierte sphärische Blickwinkelveränderung (Positionen auf Kugelhülle mit $\pm 30^\circ$ von senkrechter Pose mit Blickrichtung zur Mitte des Ladungsträgers) im Abstand von ca. 1.0 m sowie Abstandsvariation im Bereich 0.75 m bis 1.5 m

5. Experimente und Ergebnisse

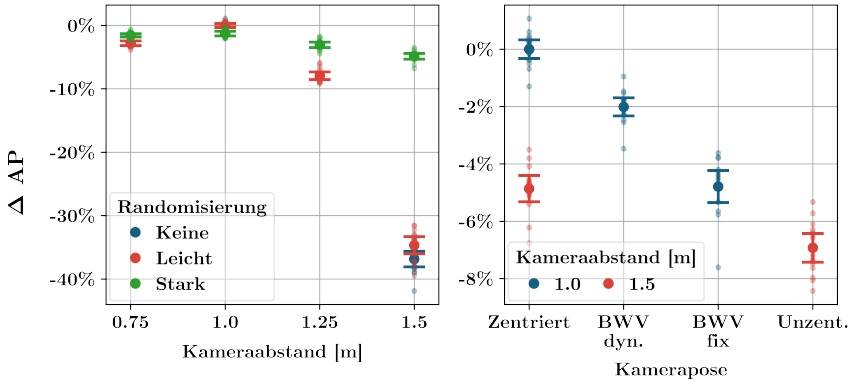


Abbildung 5.6.: Ergebnisse der Studie zum Sichtfeld für unterschiedliche Kameraabstände und drei Randomisierungsgrade (links) sowie für 6D-Blickwinkelveränderungen (BWV) und die (Un-)Zentriertheit des Ladungsträgers im Bild (rechts).

Pre-Evaluierungsdatensatz: Abbildung 5.6 (links) zeigt den signifikanten Rückgang der Performanz mit steigender Entfernung vom realen Kameraabstand (Reduktion oder Steigerung). Die signifikant besten Ergebnisse wurden mit dem Basis-Datensatz bei zentrierter Kamerapose und einem realistischen Abstand von 1.0 m erreicht. Der Rückgang der Performanz bei abweichenden Kameraabständen im Vergleich zur Basis (1 m) lag hierbei zwischen $\Delta AP -1.4\%$ bei 0.75 m und $\Delta AP -4.8\%$ bei 1.5 m (jeweils starke Randomisierung). Eine Erhöhung des Abstands auf 2.0 m und mehr führte unabhängig von der Randomisierung zu absoluten APs unter 6% ($\Delta AP -80.6\%$ und weniger zur Basis; nicht abgebildet).

Beim Abstand von 1.0 m konnten keine signifikanten Unterschiede zwischen einer leichten Randomisierung (± 0.02 m) und einer statischen Kamerapose gemessen werden. Eine stärkere Randomisierung (hier ± 0.12 m) führte zu einem signifikanten Rückgang der Performanz um $\Delta AP -1.2\%$ im Vergleich zur leichten Randomisierung. Bei zunehmendem Kameraabstand zeigte sich die Relevanz einer entsprechend stärkeren Randomisierung, deren Randomisierungsintervall jeweils nah an dem realen Kameraabstand liegt oder diesen einschließt. So verbesserte sich die Performanz für einen Abstand von 1.5 m durch eine starke Randomisierung (hier ± 0.6 m) im direkten Vergleich zur leichten Randomisierung (± 0.02 m) um $\Delta AP 29.8\%$.

Ein von der Realität abweichender Blickwinkel führte für alle untersuchten Entfernungen zu einer signifikanten Reduktion der Performanz (s. Abbildung 5.6 rechts). Bei derselben Entfernung erreichte die dynamische Veränderung des Blickwinkels ($\Delta AP -2.0\%$ zur Basis bei 1 m Entfernung) ein signifikant besseres Ergebnis als die fixe Veränderung der Kamerapose ($\Delta AP -4.8\%$ zur Basis bei 1 m Entfernung). Eine zum Ladungsträger unzentrierte Kamerapose führte unabhängig vom Randomisierungsgrad zu einer signifikant niedrigeren Performanz als eine zentrierte Kamerapose ($\Delta AP -2.1\%$ zur Basis bei 1.5 m).

Post-Evaluierungsdatensatz: Die Evaluierung mit dem Post-Datensatz bestätigte die Ergebnisse des Pre-Datensatzes (s. Anhang C.7). Insgesamt lag die Performanz auf einem leicht erhöhten Niveau. In Bezug auf Kameraabstand, -blickwinkel und Zentriertheit blieb die ordinale Rangfolge jedoch bestehen. Damit konnten alle Erkenntnisse aus den Experimenten mit dem Pre-Datensatz bestätigt werden.

IP-B-Anwendungsfall: Die hohe Relevanz einer realitätsnahen Kamerapose konnte für den IP-B-Anwendungsfall bestätigt werden (s. Anhang C.8) [119]. Eine unzentrierte Kamerapose mit erhöhtem Abstand (2.25 m statt 1.65 m) und einer Rotation von 80° um die Blickachse führte auch mit starker Randomisierung zu einem signifikanten Rückgang der Performanz um mindestens $\Delta AP -14.5\%$ zu einer realistischen Kamerapose.

Zusammenfassung

Aufgrund der Studienergebnisse kann die **Hypothese** für die untersuchten Anwendungsfälle **bestätigt** werden. Ein abweichender Blickwinkel oder Kameraabstand sowie die Unzentriertheit des Ladungsträgers im Kamerabild führten zu signifikanten Reduktionen der Performanz. Durch stärker abweichende Abstände wurden hohe Performanzreduktionen bis hin zum kompletten Scheitern des trainierten Objekterkennungssystems verursacht. Die Nachteile durch die von der Realität abweichenden Kameraposen konnten durch eine stärkere Randomisierung zwar reduziert, jedoch nicht ausgeglichen werden. Im Falle eines unbekannten Blickwinkels zeigte sich die dynamische Posenfindung (Kugelhülle) gegenüber einer fixen Abweichung überlegen.

5.2.4. Beleuchtung

Hypothese

Simulationsszenen mit einer stark an den **realen Anwendungsfall angepassten** (realistischen) **Beleuchtung** erlauben es, bessere synthetische Bilddaten zu erzeugen als Szenen, bei denen die **fehlende Anpassung** der Beleuchtung **durch Randomisierung ausgeglichen** wird.

Im Rahmen der Beleuchtungsstudie wurden acht Datensätze mit unterschiedlichen Beleuchtungsvarianten auf Basis der SE-Simulation erstellt. Dabei wurde die Anzahl der simulierten Lichter aus Gründen der Vereinfachung auf eine Lichtquelle reduziert. Als Vergleichsdatensatz für die Festlegung der Lichtintensität diente eine unveränderte SE-Szene (Basis) mit mehreren simulierten Lichtern, welche in Bezug auf die mittlere Helligkeit und den Randomisierungsgrad manuell an ein reales Referenzbild angepasst wurde. Für die beiden Beleuchtungsarten direktionales und kegelförmiges Licht wurden jeweils vier Datensätze mit unterschiedlicher mittlerer Helligkeit und Randomisierungsgrad erstellt:

1. SE: Szene mit nur einer Lichtquelle, jedoch mit ähnlicher resultierender Helligkeit und Randomisierungsgrad wie Basis
2. Statisch: Mittlere Helligkeit wie SE, jedoch ohne Randomisierung
3. Randomisiert: Mittlere Helligkeit wie SE, jedoch unrealistische starke Randomisierung
4. Dunkel: Randomisierungsgrad wie SE, jedoch mittlere Helligkeit unrealistisch reduziert

Pre-Evaluierungsdatensatz: Die Studie zeigt, dass eine unrealistisch starke Randomisierung der Beleuchtung sowohl bei direktionalem (ΔAP 0.7 % zu SE direktional) und kegelförmigem Licht (ΔAP 1.2 % zu SE kegelförmig) zu einer signifikant höheren Performanz führte als eine manuell optimierte oder statische Beleuchtung (s. Abbildung 5.7). Die signifikant besten Ergebnisse wurden unabhängig von der Beleuchtungsart mit unrealistisch reduzierter Helligkeit (dunkel) erreicht (direktional ΔAP 1.1 % und kegelförmig ΔAP 1.3 % zur Basis).

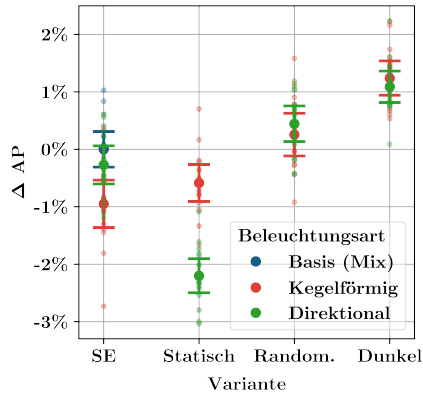


Abbildung 5.7.: Ergebnisse der Studie zur Beleuchtung mit drei Beleuchtungsarten sowie für unterschiedliche Randomisierungsgrade und Helligkeiten.

Um die Überlegenheit der dunklen Simulationsszene gegenüber der randomisierten zu überprüfen, wurde eine nachträgliche Randomisierung der Bildfarbe und -helligkeit (HSV-Augmentation) des Basisdatensatz und des dunklen Datensatzes (direktional) durchgeführt (nicht abgebildet). Hierbei lag die Performanz des augmentierten Basisdatensatzes signifikant über der des augmentierten dunklen (ΔAP 0.9 %) sowie der des in der Simulation randomisierten Datensatzes (ΔAP 1.7 %).

Ein möglicher Erklärungsansatz für die Überlegenheit der nachträglichen Randomisierung (Augmentation) gegenüber der Randomisierung der simulierten Beleuchtung besteht darin, dass erstere eine homogenere Verteilung der Bildhelligkeiten über den Datensatz hinweg ermöglicht. Die daraus resultierende höhere Frequenz an für die Objekterkennung herausfordernden und zum Evaluierungsdatsatz passenden Randfällen könnte im vorliegenden Fall zu einem effizienteren Training und dadurch zu einer höheren Performanz führen. Der höhere Anteil an herausfordernden Randfällen könnte ebenfalls die ursprüngliche Überlegenheit des dunklen Datensatzes erklären.

Post-Evaluierungsdatsatz: Für den realen Post-Datsatz ergibt sich ein etwas anderes Bild (s. Anhang C.9). Die Randomisierung des Lichts führte hierbei zu den besten Ergebnissen der trainierten Objekterkennungs-

5. Experimente und Ergebnisse

systeme je Lichtart (kegelförmig ΔAP 1.1 % zur Basis). Dagegen wiesen die dunklen Datensätze jeweils keinen Vorteil gegenüber der randomisierten Variante auf. Wie beim Pre-Evaluierungsdatsatz führt eine nachträgliche HSV-Augmentation des Basisdatsatzes auch hier zu einer signifikant höheren Performanz als die Augmentation des dunklen Datensatzes (ΔAP 1.3 % im direkten Vergleich) und als die Randomisierung in der Simulation (ΔAP 1.0 % im direkten Vergleich).

IP-B-Anwendungsfall: Auch bei den Untersuchungen mit dem IP-B-Anwendungsfall führten starke Randomisierungen der Beleuchtung zu signifikant besseren Ergebnissen als eine manuelle Nachbildung der realen Beleuchtungssituation (s. Anhang C.10) [119]. Hierbei verbesserte sich die Performanz durch die Randomisierung um ΔAP 2.5 % für die SE-Szene und um ΔAP 8.7 % für eine stark vereinfachte Szene.

Zusammenfassung

Aufgrund der Studienergebnisse kann die **Hypothese** für die untersuchten Anwendungsfälle **nicht bestätigt** werden. Eine unrealistisch starke Randomisierung der Beleuchtung führte zu einer signifikant höheren Performanz als eine manuell an den realen Anwendungsfall angepasste Beleuchtung. Dies könnte sowohl in einer generellen Überlegenheit randomisierter Beleuchtungseigenschaften als auch in einer potenziell unpräzisen Nachbildung begründet liegen. Eine nachträgliche HSV-Randomisierung führte zu signifikant besseren Ergebnissen als die Randomisierung in der Simulation.

5.2.5. Objektpositionierung

Hypothese

Simulationen mit einer **physikalischen Kollisionssimulation** erlauben es, bessere synthetische Bilddaten zu erzeugen als Simulationen mit **randomisierter Objektpositionierung ohne Kollisionskontrolle**.

Im Rahmen der Studie zur Objektpositionierung wurden acht Datensätze mit und ohne physikalische Kollisionssimulation für die Positionierung mit-

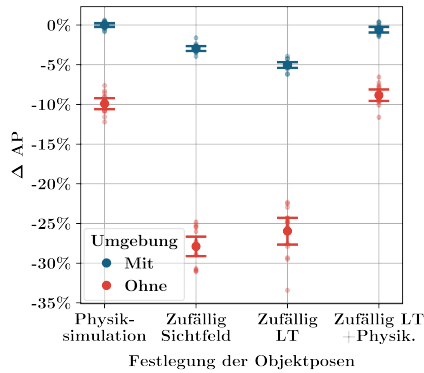


Abbildung 5.8.: Ergebnisse der Studie zur Objektpositionierung mit physikalischer Kollisionssimulation und zufallsbasierter Posenbestimmung ohne Kollisionsprüfung sowie mit und ohne Umgebungsobjekte.

einander verglichen. Die folgenden vier Varianten der Objektpositionierung wurden jeweils entweder mit realitätsnahen Umgebungsobjekten (Ladungsträger und Tisch) oder mit zufälliger Hintergrundtextur untersucht:

1. Physikalische Kollisionssimulation über zufällige auf die Objekte wirkende Kräfte
2. Festlegung einer zufälligen Pose ohne Kollisionskontrolle, deren Position auf das Volumen des Ladungsträgers begrenzt ist
3. Festlegung einer zufälligen Pose ohne Kollisionskontrolle, deren Position auf das gesamte Sichtfeld begrenzt ist
4. Kombination aus einer zufälligen Pose im Ladungsträger (wie 2.) und anschließender Physiksimation mit zufälligen Kräften (wie 1.)

Der Datensatz mit Physiksimation (1.) und sichtbaren Umgebungsobjekten stellt im Folgenden den Basisdatensatz dar.

Pre-Evaluierungsdatensatz: Die in Abbildung 5.8 dargestellten Studienergebnisse zeigen eine signifikante Überlegenheit der Objektpositionierung mittels Physiksimation gegenüber einer zufälligen Posenfindung ohne

5. Experimente und Ergebnisse

Kollisionsprüfung. Obwohl die Kombination aus zufälliger Pose und Physiksimulation zu einer gleichmäßigeren Verteilung der Objekte führt (s. Anhang C.11 bis C.13), erbrachte sie keine Verbesserung gegenüber der reinen Physiksimulation ($\Delta AP -0.6\%$). Die rein zufällige Positionierung im Ladungsträger ($\Delta AP -5.0\%$) und im Sichtfeld ($\Delta AP -2.9\%$) führten jeweils zu signifikanten Reduktionen der Performanz gegenüber dem Datensatz mit Physiksimulation. Als Erklärung für die signifikant schlechteren Ergebnisse der Positionierung im Ladungsträger gegenüber der im Sichtfeld kommt die höhere Rate an sich überschneidenden Objekten durch die engere Raumbegrenzung in Frage.

Ohne sichtbare Umgebungsobjekte sank die Performanz durchgängig signifikant (vgl. Unterabschnitt 5.2.2). Durch das Ausblenden der Umgebungsobjekte (direkter Vergleich mit und ohne) fiel die Reduktion der Performanz bei der zufälligen Positionierung (Ladungsträger $\Delta AP -20.9\%$ und Sichtfeld $\Delta AP -25.0\%$) wesentlich stärker aus als bei der Variante mit Physiksimulation ($\Delta AP -9.9\%$) und deren Kombination ($\Delta AP -8.2\%$). Ohne sichtbare Umgebungsobjekte führte die Kombination aus einer zufälligen Pose und anschließender Physiksimulation mit $\Delta AP 1.1\%$ im direkten Vergleich zu signifikant besseren Ergebnissen als die reine Physiksimulation.

Post-Evaluierungsdatensatz: Bei der Evaluierung mit dem Post-Datensatz ergab sich lediglich eine Änderung in Bezug auf die ordinale Rangfolge der Datensätze nach deren Performanz (s. Anhang C.14). Ohne sichtbare Umgebungsobjekte erreichte der Datensatz mit Physiksimulation eine signifikant höhere Performanz als die kombinierte Variante ($\Delta AP 1.8\%$ im direkten Vergleich). Damit konnten die Erkenntnisse der Experimente mit dem Pre-Datensatz weitgehend bestätigt werden.

Zusammenfassung

Aufgrund der Studienergebnisse kann die **Hypothese** für die untersuchten Anwendungsfälle **bestätigt** werden. Die Untersuchungen zeigten die Überlegenheit der Objektpositionierung mittels Physiksimulation, unabhängig von einer realitätsnahen Nachbildung der Umgebung inklusive Ladungsträger. Die Randomisierung der initialen Pose vor der Physiksimulation führte bei sichtbaren Umgebungsobjekten zu keiner Verbesserung der Performanz.

5.2.6. Objektanzahl

Hypothese

Ein höherer **Übereinstimmungsgrad** in der Verteilung der **Objektanzahl pro Bild** zwischen Realität und Simulation erlaubt es, bessere synthetische Bilddaten zu erzeugen.

Im Rahmen der Studie zur Objektanzahl wurden sieben Datensätze mit unterschiedlich verteilter Objektanzahl pro Bild miteinander verglichen:

1. Drei Datensätze mit einer unimodalen und symmetrischen Verteilung der Objektanzahl pro Bild (Glockenkurve), welche über die zufällige Ausblendung einzelner Objektinstanzen erreicht wurde.
 - a) A 10 %: Ausblendewahrscheinlichkeit 10 % mit im Mittel 6.2 (tatsächlich 6.24) Objekten pro Bild
 - b) A 30 %: Ausblendewahrscheinlichkeit 30 % mit im Mittel 4.9 (tatsächlich 4.87) Objekten pro Bild
 - c) A 50 %: Ausblendewahrscheinlichkeit 50 % mit im Mittel 3.5 (tatsächlich 3.49) Objekten pro Bild
2. Vier Datensätze mit näherungsweise gleichverteilter oder fixer Objektanzahl pro Bild.
 - a) G 1: Ein einziges Objekt pro Bild
 - b) G 3-4: Drei oder vier Objekte pro Bild (Mittel tatsächlich 3.48)
 - c) G 1-7: Zwischen einem und sieben Objekten pro Bild (Mittel tatsächlich 3.98)
 - d) G 7: Sieben Objekte pro Bild (Mittel tatsächlich 6.9)

Der Datensatz A 30 % entspricht der für die SE-Datensätze verwendeten Verteilung und stellt im Folgenden den Basis-Datensatz dar. Näherungsweise ist die Anzahl der Objekte des realen Pre-Datensatzes gleichverteilt und die des realen Post-Datensatzes glockenförmig verteilt. Die tatsächliche Übereinstimmung der Datensätze in Bezug auf die Verteilung der Objektanzahl pro Bild wird mit dem Kennwert ΔH gemessen (s. Unterabschnitt 4.1.5).

Wie Tabelle 5.1 zeigt, wurden die besten Ergebnisse für beide Evaluierungsdatensätze jeweils mit den Trainingsdatensätzen mit der höchsten

5. Experimente und Ergebnisse

Tabelle 5.1.: Abweichungen der Verteilung der Anzahl an Objekten pro Bild (ΔH) und resultierende Veränderungen der Performanz (ΔAP) zwischen synthetischen und realen Datensätzen (Pre und Post). Die geringsten (ΔH) bzw. signifikant höchsten Kennwerte (ΔAP) je Metrik sind hervorgehoben.

Trainings- datensatz	Pre-Datensatz		Post-Datensatz	
	ΔH	ΔAP [%]	ΔH	ΔAP [%]
A 30 %	0.12	-	0.19	-
A 10 %	0.31	0.3	0.38	-0.5
A 50 %	0.08	-0.2	0.02	0.5
G 1	0.43	-32.9	0.36	-27.0
G 3-4	0.08	0.6	0.02	-0.1
G 1-7	0.03	0.6	0.06	0.7
G 7	0.41	-0.6	0.48	-0.7

Übereinstimmung der Verteilung der Objektanzahl pro Bild erzielt. Für den Pre-Datensatz war dies der Datensatz G 1-7 (G 3-4 mit insignifikantem Unterschied) und für den Post-Datensatz der Datensatz A 50 % (G 1-7 insignifikant höhere Performanz). Dennoch lässt sich keine lineare oder ordinale Korrelation zwischen dem Übereinstimmungsgrad und der Performanz der trainierten CNNs erkennen. Der gleichverteilte Datensatz G 1-7 gehörte für beide Evaluierungsdatsätze zu den Trainingsdatensätzen mit den signifikant besten Ergebnissen (ΔAP 0.6 % und ΔAP 0.7 %). Die jeweils signifikant geringste Performanzkennwerte erzielte der Datensatz G 1 mit nur einem Objekt pro Bild (ΔAP -32.9 % und ΔAP -27.0 %).

Zusammenfassung

Aufgrund der Studienergebnisse kann die **Hypothese** für die untersuchten Anwendungsfälle **nicht bestätigt** werden. Zwar wurden die höchsten Performanzkennwerte je Evaluierungsdatsatz mit dem jeweils höchsten Übereinstimmungsgrad erreicht, jedoch führten teilweise auch stärker abweichende Verteilungen zu keinem signifikanten Rückgang der Performanz. Mit einer fixen Objektanzahl reduzierte sich die Performanz dagegen durchgängig signifikant.

5.2.7. Scene Adaptation

Hypothese

Simulationsszenen, welche mittels **Scene Adaptation** automatisiert an reale Referenzbilder angepasst wurden, erlauben es, nicht signifikant schlechtere synthetische Bilddaten zu erzeugen als Szenen, welche **manuell mittels Scene Engineering** angepasst wurden.

Die automatisierte Simulationsoptimierung (Scene Adaptation) wurde für unterschiedliche reale Zielbilder überprüft. Da das entwickelte Verfahren auf der Bayes'schen Optimierung basiert und an verschiedenen Prozessschritten von Zufallszahlen abhängt, wurde auch die Wiederholgenauigkeit für dasselbe Zielbild untersucht. Es fand dabei zunächst eine qualitative und anschließend eine quantitative Bewertung des Verfahrens statt.

Bei subjektiver Bewertung führten die Anpassungen der Kamera- und Objektposen, der Objekt- und Ladungsträgererscheinung (Farbe und Reflexionseigenschaften) sowie der Beleuchtung für verschiedene Realbilder zu einer hohen Übereinstimmung zwischen Simulations- und Zielbild. Die Wiederholgenauigkeit war insbesondere in Bezug auf die Objektpositionierung und Beleuchtung hoch und führte bei optischem Bildvergleich zu kaum erkennbaren Unterschieden. Bei der Objektrotation lagen dagegen beim direkten Vergleich geringe, jedoch klar erkennbare Abweichungen zwischen den Wiederholungen vor. Bei der Optimierung der Farb- und Reflexionseigenschaften wurde eine hohe Abhängigkeit von der vorherigen Festlegung der Objektposen festgestellt. Fehlerhafte Objektposen führten dabei zu starken Abweichungen der Objektfarben. Darüber hinaus konnte festgestellt werden, dass das Verfahren durch eine hohe Überlappung der Objekte im Zielbild und die dadurch resultierende Verdeckung beeinträchtigt wurde.

Um eine objektive Bewertung der automatisierten Simulationsoptimierung auf Basis der Performanz eines trainierten Objekterkennungssystems zu ermöglichen, wurde eine Studie mit 11 Datensätzen zu den besonders relevanten Eigenschaften der Kamera- und Ladungsträgerpose durchgeführt (s. Unterabschnitt 5.2.2 und Unterabschnitt 5.2.3). Neben einer manuell mittels Scene Engineering optimierten Szene (Basis-Datensatz) umfasst diese jeweils 5 Wiederholungen des Scene Adaptation (SA) Optimierungsprozesses für eine korrekte und eine leicht abweichende Modellierung der Umgebung:

5. Experimente und Ergebnisse

1. SA korrekt: automatisiert optimierte Szene auf Basis von weitgehend mit der realen Anwendung übereinstimmenden 3D-Modellen
2. SA inkorrekt: automatisiert optimierte Szene auf Basis eines im Vergleich zur realen Anwendung leicht veränderten 3D-Modells des Maschinentischs, wobei die Halterungen des Ladungsträgers um 2.2 cm horizontal zum Bild und 4.5 cm nach oben (in Richtung der Kamera) sowie der Anschlag um 3.3 cm vertikal zum Bild versetzt wurden

Dabei wurde zunächst die Kamerapose relativ zum Maschinentisch und anschließend die Ladungsträgerpose mit fixer Kamera optimiert. Startzustand des Optimierungsprozesses war jeweils eine Kamerapose mit starker Blickwinkelveränderung (s. Unterabschnitt 5.2.3) sowie ein verschobener und rotierter Ladungsträger. Für die Durchläufe wurden jeweils unterschiedliche Startzustände des Zufallszahlengenerators verwendet. Entsprechende Beispielsbilder sind in Abbildung 5.9 zu finden.

Pre-Evaluierungsdatensatz: Abbildung 5.10 zeigt, dass die signifikant besten Ergebnisse mit den mittels Scene Adaptation und korrektem 3D-Modell optimierten Datensätzen erzielt wurden. Gegenüber der manuell optimierten Szene verbesserte sich die Performanz um ΔAP 0.5 % bis ΔAP 1.1 %. Zwischen den Wiederholungen zeigten sich mit einem maximalen Unterschied von ΔAP 0.6 % kaum signifikante Unterschiede (Ausnahme Wdh. 1 signifikant besser als Wdh. 3, 4 und 5).

Dagegen zeigten sich größere Schwankungen der Performanz zwischen den Wiederholungen der Scene Adaptation mit leicht abweichendem 3D-Modell. Hier lag der maximale Performanzunterschied zwischen den Wiederholungen bei ΔAP 1.9 %. Mit einer Ausnahme (Wdh. 2 insignifikant) führten die Simulationsoptimierungen hierbei zu signifikant niedrigeren Performanzkennwerten als die manuell optimierte SE-Szene. Dennoch erreichten auch hier vier der fünf Durchläufe signifikant bessere Ergebnisse (eine Wdh. insignifikant) als der beste Datensatz mit randomisierter Kamerapose (ΔAP 1.7 % im direkten Vergleich mit der dynamischen Blickwinkelveränderung, s. Unterabschnitt 5.2.3).

Post-Evaluierungsdatensatz: Für den Post-Datensatz ergibt sich ein leicht anderes Bild (s. Anhang C.15). Zwar liegen die Performanzkennwerte für die SA-Datensätze mit korrektem 3D-Modell weiterhin signifikant über dem Basis-Datensatz, jedoch fällt die Verbesserung geringer aus (ΔAP

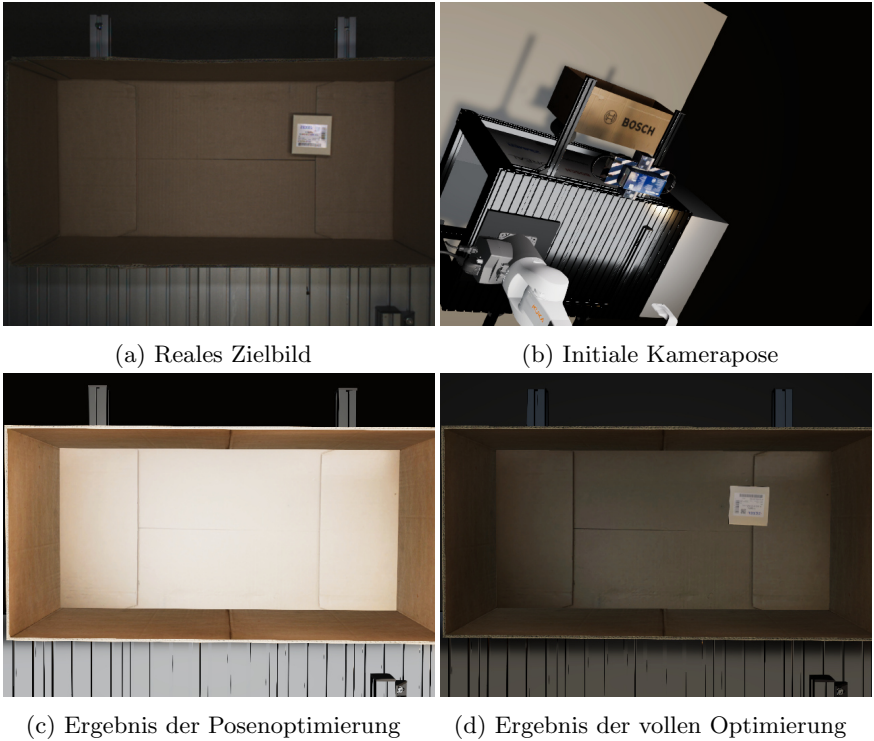


Abbildung 5.9.: Beispielbilder der Simulationsoptimierung (Scene Adaptation) mit (a) Ziel- und (b) Startzustand der Optimierung sowie den Ergebnissen (c) der Posenoptimierung für Kamera und Ladungsträger und (d) der vollen Optimierung inklusive Objekteigenschaften und Beleuchtung.

0.3 % bis ΔAP 0.6 %). Mit dem leicht veränderten 3D-Modell wurde ein höherer Performanzunterschied zwischen den Wiederholungen gemessen (bis zu ΔAP 3.1 %). Darüber hinaus führten alle Wiederholungen zu einer signifikant niedrigeren Performanz als der beste Datensatz mit randomisierter Kamerapose (ΔAP -1.0 % bis ΔAP -4.1 % im direkten Vergleich mit der dynamischen Blickwinkelveränderung).

5. Experimente und Ergebnisse

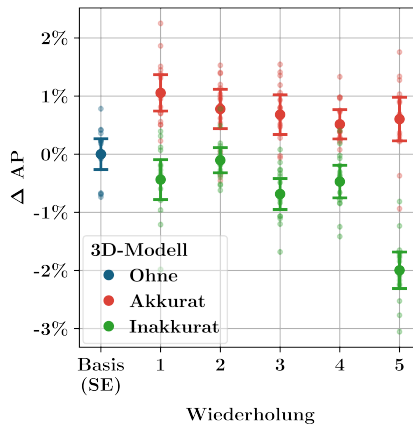


Abbildung 5.10.: Ergebnisse der Studie zur Scene Adaptation mit jeweils fünf Wiederholungen sowie akkuratem und inakkuratem 3D-Modell der Szene. Beim inakkuraten 3D-Modell stimmt die relative Positionierung zwischen Ladungsträger und Tisch nicht mit den realen Referenzdaten überein.

Zusammenfassung

Aufgrund der Studienergebnisse kann die **Hypothese** für die untersuchten Anwendungsfälle **bestätigt** werden. Mit einer akkuraten 3D-Modellierung ließen sich Simulationsszenen durch Scene Adaptation wiederholbar so parametrisieren, dass die generierten Bilddaten sogar zu einer höheren Performanz führten als die manuell optimierten Simulationsszenen (Scene Engineering). Abweichende 3D-Modelle führten dagegen zu einer geringeren Wiederholgenauigkeit der Scene Adaptation und zu einer Reduktion der resultierenden Performanz im Vergleich zur SE-Basiszene.

5.3. Domain Randomization

Im Folgenden werden die Methoden der Domain Randomization über unterschiedliche Augmentationstechniken verglichen. Es wird untersucht, ob sich die synthetischen Datensätze durch diese Randomisierungen der Bildstatistik sowie deren Kombination für das Training von Objekterkennungsverfahren verbessern lassen. Dabei findet jeweils der Vergleich zum unveränderten SE-Opt-Basisdatensatz statt. Alle Trainings wurden zunächst mit bis zu 3 Epochen durchgeführt und mit dem Pre-Datensatz evaluiert. Zur Überprüfung der Saturation wurden jeweils die Datensätze mit der höchsten erreichten Performanz zusätzlich mit bis zu 40 Epochen trainiert. Im Nachgang fand eine zusätzliche Überprüfung mit dem Post-Datensatz statt. Die zugehörige Hypothese lautet wie folgt:

Hypothese

Synthetische Bilddaten können durch die **Augmentation** mit {1. Unschärfe, 2. Rauschen, 3. Farbvariation, 4. Helligkeitsvariation, 5. chromatischer Aberration, 6. Kontrastvariation} im Vergleich zu den ursprünglichen synthetischen Bilddaten verbessert werden.

5.3.1. Unschärfe und Rauschen

Im Rahmen der Studie zum Einfluss von Unschärfe und Rauschen auf die Performanz des trainierten Objekterkennungssystems wurden die folgenden Augmentationsmethoden und -parameter im Vergleich mit dem unveränderten Basisdatensatz untersucht:

1. Unschärfe Gauß-Filter (U) mit neun Filtergrößen
 $\delta_U = \{3, 7, 15, 21, 33, 45, 61, 77, 99\}$
2. Unschärfe Box-Filter (X) mit neun Filtergrößen
 $\delta_X = \{3, 5, 7, 9, 11, 15, 21, 33, 45\}$
3. Gauß'sches Rauschen (N) mit neun Varianzen
 $\delta_N = \{8, 16, 32, 64, 128, 256, 512, 1024, 2048\}$
4. Gauß'sches Rauschen und Unschärfe (Gauß-Filter) mit zwölf Kombinationen
 $\delta_{N,U} = \{\delta_N \in \{256, 512, 1024\}, \delta_U \in \{21, 33, 45, 61\}\}$

5. Experimente und Ergebnisse

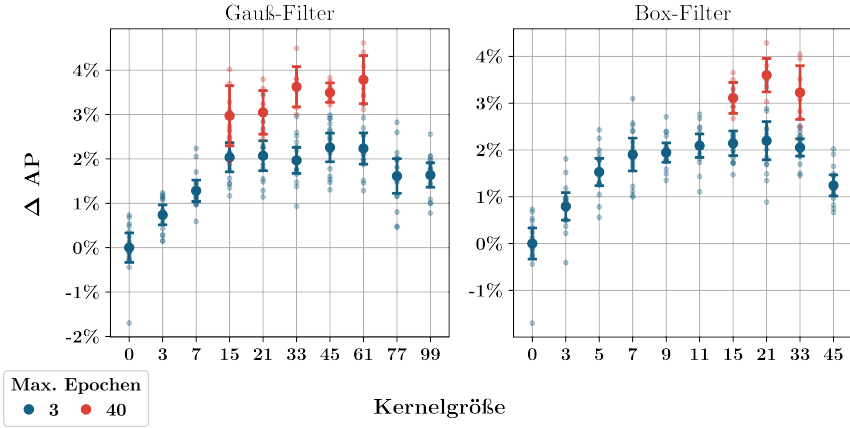


Abbildung 5.11.: Ergebnisse der Augmentation mit Unschärfe auf Basis des Gauß- (links) und des Box-Filters (rechts) jeweils für unterschiedliche Variationsstärken.

Unschärfe Gauß-Filter (Pre-Eval.): Wie Abbildung 5.11 (links) zeigt, wurden die höchsten Performanzkennwerte beim Training mit bis zu 3 Epochen für die Kernelgrößen $15 \leq \delta_U \leq 61$ erreicht (insignifikante Unterschiede). Die besten Ergebnisse wurden mit $\delta_U = 45$ erzielt (ΔAP 2.2 % zur Basis). Alle getesteten Filtergrößen (ΔAP 0.7 % zur Basis bei $\delta_U = 3$) führten zu einer signifikanten Verbesserung der Performanz gegenüber dem Basisdatensatz ohne Unschärfe.

Die Datensätze mit $15 \leq \delta_U \leq 61$ wurden zur Überprüfung der Saturation zusätzlich länger trainiert. Die insgesamt höchste Performanz erreichte der Datensatz $\delta_U = 61$ mit ΔAP 3.7 % zur Basis nach 24 Trainingsepochen ($\delta_U \geq 33$ mit insignifikanten Unterschieden).

Unschärfe Box-Filter (Pre-Eval.): Aus Abbildung 5.11 (rechts) geht hervor, dass die höchsten Performanzkennwerte bei bis zu 3 Trainingsepochen für die Kernelgrößen $7 \leq \delta_X \leq 33$ erreicht (insignifikante Unterschiede). Die besten Ergebnisse wurden mit $\delta_X = 21$ erzielt (ΔAP 2.2 % zur Basis). Alle getesteten Filtergrößen (ΔAP 0.8 % zur Basis bei $\delta_X = 3$) führten zu einer signifikanten Performanzsteigerung gegenüber dem Basisdatensatz.

Die Datensätze mit $7 \leq \delta_X \leq 33$ wurden zusätzlich länger trainiert. Die

insgesamt höchste Performanz erreichte der Datensatz $\delta_X = 21$ mit ΔAP 3.6 % zur Basis nach 40 Epochen (insignifikanter Unterschied zu $\delta_X = 33$).

Bei der Betrachtung der maximal erreichten Performanz zeigen die Ergebnisse damit keine signifikanten Unterschiede zwischen der Unschärfe mit Gauß- und Box-Filter. Die höchsten Performanzkennwerte wurden beim Box-Filter jedoch mit geringeren Kernel-Größen erreicht als beim Gauß-Filter. Dies lässt sich dadurch erklären, dass weiter entfernte Pixel beim Gauß-Filter weniger stark gewichtet werden und die resultierende Weichzeichnung bei gleicher Filtergröße daher insgesamt schwächer ausfällt. Da sich in der maximal erreichten Performanz keine signifikanten Unterschiede zeigten, wird im weiteren Verlauf der Experimente ausschließlich die mit dem Gauß-Filter erzeugte Unschärfe verwendet und im Folgenden kurz als *Unschärfe* bezeichnet.

Gauß'sches Rauschen (Pre-Eval.): Wie Abbildung 5.12 (links) zeigt, wurden die höchsten Performanzkennwerte beim Training mit bis zu 3 Epochen mit $256 \leq \delta_N \leq 1024$ erreicht (insignifikante Unterschiede). Die insgesamt besten Ergebnisse wurden mit $\delta_N = 512$ erzielt (ΔAP 1.0 % zur Basis). Keine der getesteten Variationsstärken $\delta_N < 256$ führte zu einer signifikanten Steigerung der Performanz gegenüber dem Basisdatensatz ohne Rauschen.

Die Datensätze mit $256 \leq \delta_N \leq 1024$ wurden zur Überprüfung der Saturation zusätzlich länger trainiert. Nach 6 Epochen konnten keine weiteren Verbesserungen mehr gemessen werden. Die insgesamt höchste Performanz wurde mit ΔAP 1.3 % zum Basisdatensatz mit der Variationsstärke $\delta_N = 1024$ erreicht (insignifikanter Unterschied zu $\delta_N = 512$).

Kombination (Pre-Eval.): Wie Abbildung 5.12 (rechts) zeigt, wurden die besten Ergebnisse mit insignifikanten Unterschieden für sechs der acht Kombinationen $\delta_{N,U} = \{\delta_N \in \{512, 1024\}, \delta_U \in \{21, 33, 45, 61\}\}$ erzielt (jeweils nach 40 Trainingsepochen). Die insgesamt höchste Performanz erreichte der Datensatz $\delta_{N,U} = \{1024, 61\}$ mit ΔAP 5.0 % gegenüber dem Basisdatensatz ohne Augmentation. Dabei lag die Performanz der Kombination signifikant über der höchsten Performanz der einzelnen Augmentationen (ΔAP 1.3 % im direkten Vergleich zu $\delta_U = 61$). Eine Anwendung der Unschärfe vor dem Rauschen führte im direkten Vergleich zu einer signifikanten Reduktion der Performanz (ΔAP -1.5 % im direkten Vergleich), wodurch eine allgemeingültige Kommutation ausgeschlossen werden kann.

5. Experimente und Ergebnisse

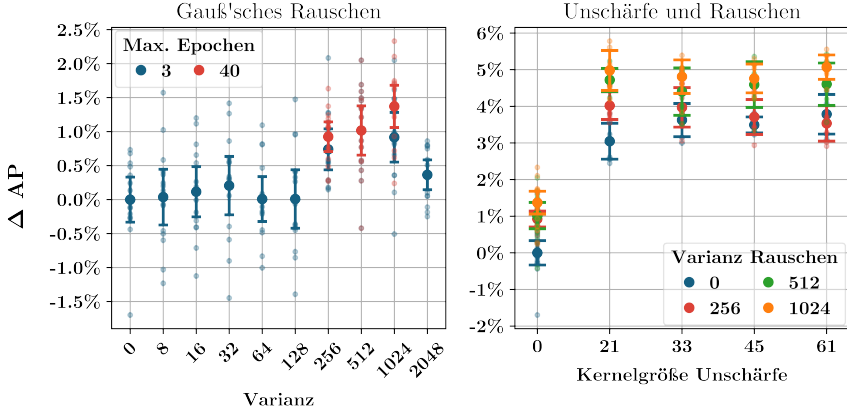


Abbildung 5.12.: Ergebnisse der Augmentation mit Gauß'schem Rauschen für unterschiedlichen Variationsstärken (links) sowie der kombinierten Augmentation mit Unschärfe (Gauß-Filter) und Gauß'schem Rauschen (rechts).

Post-Evaluierungsdatensatz: Die Untersuchungen der Augmentation mit Unschärfe zeigten für den realen Post-Evaluierungsdatensatz eine wesentlich geringere Verbesserung und eine frühere Saturation als beim Pre-Evaluierungsdatensatz (s. Anhang C.16).

Die insgesamt besten Ergebnisse wurden mit $\delta_U = 3$ nach 3 Trainingsepochen bzw. $\delta_X = 9$ nach 2 Trainingsepochen erreicht (jeweils ΔAP 0.4%). Sowohl größere Filter ($\delta_U \geq 61$ und $\delta_X \geq 33$) als auch ein längeres Training führten zu einer Reduktion der Performanz gegenüber dem Datensatz ohne Augmentation. Die Augmentation mit Gauß'schem Rauschen zeigte keinerlei signifikante Verbesserung und teilweise sogar Verschlechterungen gegenüber dem Basisdatensatz (s. Anhang C.17 links).

Auch mit den Kombinationen aus Unschärfe und Rauschen wurden wesentlich geringere Verbesserungen der Performanz gegenüber dem Basisdatensatz als beim Pre-Evaluierungsdatensatz erreicht (s. Anhang C.17 rechts). Hierbei wurden die besten Ergebnisse mit $\delta_{N,U} = \{1024, 21\}$ nach 9 Trainingsepochen erzielt (ΔAP 1.1% zum Basisdatensatz).

Zusammenfassung

Aufgrund der Studienergebnisse kann die **Hypothese** in Bezug auf die Augmentation mit Unschärfe und Rauschen für den untersuchten Anwendungsfall **bestätigt** werden. Durch die einzelnen Augmentationen konnten signifikante Verbesserungen der Performanz der trainierten Objekterkennungssysteme erreicht werden, wobei die maximale Verbesserung bei Unschärfe signifikant stärker ausfiel als bei Rauschen. Die signifikant höchste Performanz wurde durch die Kombination von Unschärfe und Rauschen erreicht.

5.3.2. Farbe und Helligkeit

Im Rahmen der Studie zum Einfluss der Farb- und Helligkeitsvariation auf die Performanz des trainierten Objekterkennungssystems wurden die folgenden Augmentationsmethoden im Vergleich zum unveränderten Basisdatensatz untersucht:

1. RGB-Farbe mit sieben Variationsstärken
 $\delta_{RGB} = \{0.05, 0.1, 0.2, 0.4, 0.6, 0.8, 1\}$
2. Helligkeit (L) mit neun Variationsstärken
 $\delta_L = \{0.01, 0.02, 0.05, 0.1, 0.2, 0.4, 0.6, 0.8, 1\}$
3. RGB-Farbe und Helligkeit mit 15 Kombinationen
 $\delta_{RGB,L} = \{\delta_{RGB} \in \{0.2, 0.4, 0.6, 0.8, 1\}, \delta_L \in \{0.4, 0.6, 0.8\}\}$
4. HSV mit 12 Kombinationen
 $\delta_{HSV} = \{\delta_H, \delta_S, \delta_V\}$ mit $\delta_H, \delta_S \in \{0, 1\}, \delta_V \in \{0, 0.5, 1\}$

RGB (Pre-Eval.): Wie Abbildung 5.13 zeigt, wurden die höchsten Performanzkennwerte bei der RGB-Farbvariation für $\delta_{RGB} \geq 0.8$ erreicht (insignifikante Unterschiede). Die besten Ergebnisse beim Training mit bis zu 3 Epochen wurden mit $\delta_{RGB} = 1$ erzielt (ΔAP 2.8% zur Basis). Alle getesteten Variationsstärken (ΔAP 1.0% zur Basis bei $\delta_{RGB} = 0.05$) führten zu einer signifikanten Performanzsteigerung gegenüber dem Basisdatensatz.

Die Datensätze mit $\delta_{RGB} \geq 0.8$ wurden zur Überprüfung der Saturation zusätzlich länger trainiert. Die signifikant höchste Performanz erreichte der Datensatz $\delta_{RGB} = 1$ mit ΔAP 3.3% zum Basisdatensatz nach 6 Trainingsepochen.

5. Experimente und Ergebnisse

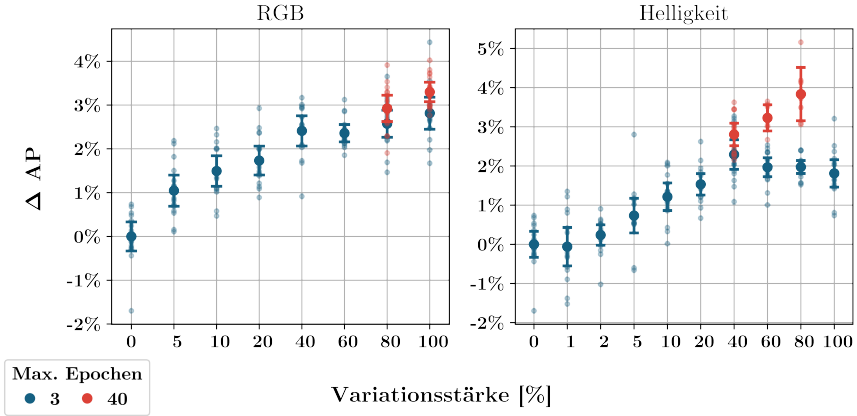


Abbildung 5.13.: Ergebnisse der Augmentation mit RGB-Farb- (links) und mit Helligkeitsvariation (rechts) jeweils für unterschiedliche Variationsstärken.

Helligkeit (Pre-Eval.): Die höchsten Performanzkennwerte bei der Helligkeitsvariation wurden für $0.4 \leq \delta_L \leq 0.8$ erreicht (insignifikante Unterschiede, s. Abbildung 5.13). Die besten Ergebnisse beim Training mit bis zu 3 Epochen wurden mit $\delta_L = 0.4$ erzielt (ΔAP 2.3 % zur Basis). Alle getesteten Variationsstärken mit $\delta_L \geq 0.05$ (ΔAP 0.7 % zur Basis bei $\delta_L = 0.05$) führten zu einer signifikanten Steigerung der Performanz gegenüber dem Basisdatensatz ohne Augmentation.

Die Datensätze mit $0.4 \leq \delta_L \leq 0.8$ wurden zur Überprüfung der Sättigung zusätzlich länger trainiert. Der Datensatz $\delta_L = 0.8$ erreichte hierbei mit ΔAP 3.8 % zur Basis nach 40 Epochen die signifikant höchste Performanz.

Kombination (Pre-Eval.): Wie Abbildung 5.14 zeigt, wurden die besten signifikant Ergebnisse der kombinierten RGB-Farb- und Helligkeitsvariation mit $\delta_{RGB,L} = \{\delta_{RGB} = 1, \delta_L \in \{0.4, 0.6, 0.8\}\}$ erreicht (jeweils nach 40 Trainingsepochen). Dabei erreichten alle Kombinationen mit $\delta_{RGB} \geq 0.4$ jeweils signifikant bessere Ergebnisse als die Anwendung der einzelnen Augmentationen. Die insgesamt höchste Performanz wurde mit $\delta_{RGB,L} = \{1, 0.4\}$ erreicht: ΔAP 5.8 % zur Basis und ΔAP 2.0 % zur besten Einzelaugmentation ($\delta_L = 0.8$). Bezüglich der Kommutation der Augmentationen konnten keine signifikanten Performanzunterschiede gemessen werden.

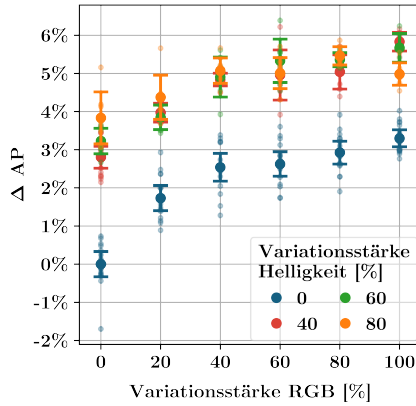


Abbildung 5.14.: Ergebnisse der kombinierten Augmentationen mit RGB-Farb- und Helligkeitsvariation für unterschiedliche Variationsstärken und bis zu 40 Trainingsepochen.

Zwar führt die reine RGB-Farbvariation bereits zu Veränderungen der Bildhelligkeit, jedoch ergibt sich die Helligkeit hierbei als Mittelwert der Farbkanäle. Dadurch ist die Bildhelligkeit bei gleichverteilten Zufallszahlen über den Datensatz hinweg nicht gleichverteilt, sondern bildet eine Glockenkurve (zentraler Grenzwertsatz). Die Ergebnisse deuten darauf hin, dass sich eine direkte, gleichverteilte Variation der Bildhelligkeit zusätzlich zur Farbvariation positiv auswirkt.

HSV (Pre-Eval.): Da die Kombination aus RGB-Farb- und Helligkeitsaugmentation signifikante Vorteile ergab, wurde zusätzlich die Variation im HSV-Farbraum untersucht. Im Gegensatz zur RGB-Farbvariation wird der Hellwert V hierbei explizit (gleichverteilt) augmentiert. Die Trainings wurden jeweils mit bis zu 9 Epochen durchgeführt.

Wie Abbildung 5.15 zur Untersuchung der Kombination des Farbwerts (H), der Sättigung (S) und des Hellwerts (V) zeigt, erreichten die Kombinationen mit $\delta_V = 1$ und $\delta_H + \delta_S \geq 0$ die signifikant besten Ergebnisse. Für die jeweils selben Parameter δ_H und δ_S führte die Erhöhung von δ_V jeweils zu einer signifikanten Steigerung der Performanz, wodurch sich eine hohe Relevanz der Hellwertvariation zeigt. Die Variation des Farbwerts und/oder der Sättigung führte für alle Variationsstärken des Hellwerts zu

5. Experimente und Ergebnisse

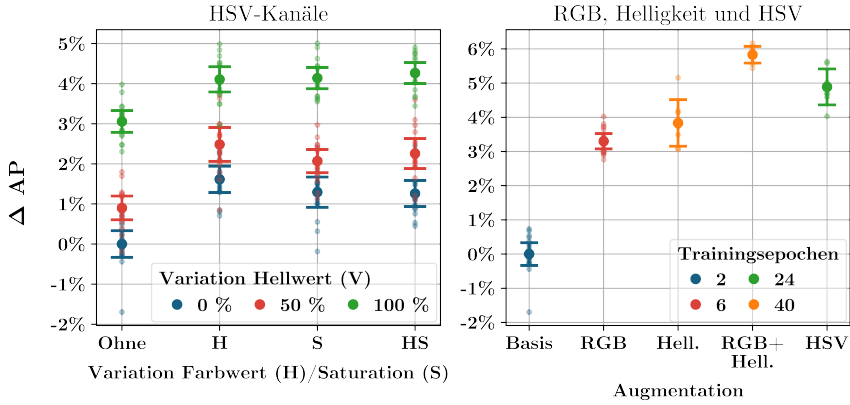


Abbildung 5.15.: Ergebnisse der kombinierten Augmentation der Kanäle im HSV-Farbraum (links) sowie der Vergleich der RGB- und Helligkeits- mit der HSV-Augmentationen beim Training mit bis zu 40 Epochen (rechts).

einer signifikanten Erhöhung der Performanz gegenüber der reinen Hellwertvariation. Dabei waren die Unterschiede zwischen den Variationen H, S und HS gering ausgeprägt und meist insignifikant, solange zumindest einer der beiden Kanäle variiert wurde. Die insgesamt höchste Performanz beim Training mit bis zu 40 Epochen erzielte die volle HSV-Augmentation $\delta_{HSV} = \{1, 1, 1\}$ mit ΔAP 4.8 % zur Basis.

Bei der Untersuchung der alternativen Berechnungsmethoden für die Variationsstärke der HSV-Augmentation erwies sich die gewählte Variante mit „stabilem Mittelwert“ als am besten geeignet (s. Anhang A.1).

Die Performanzunterschiede zwischen HSV-Augmentation und der Kombination aus RGB-Farb- und Helligkeitsvariation waren beim Training mit bis zu 9 Epochen insignifikant. Bei mehr Trainingsschritten zeigte sich die Kombination aus RGB-Farb- und Helligkeitsvariation als überlegen und führte bei 40 Trainingsepochen zu einer Verbesserung der Performanz um ΔAP 1.0 % im direkten Vergleich mit der HSV-Augmentation. Insbesondere aufgrund der einfacheren Anwendung der HSV-Augmentation als singulärer Augmentationsschritt werden die Varianten HSV- und SV-Variation dennoch in die spätere Untersuchung der Kombinationen einbezogen (s. Unterabschnitt 5.3.5).

Post-Evaluierungsdatensatz: Die Erhöhung der erreichten Performanz fiel bei der Evaluierung mit dem Post-Datensatz für alle Farb- und Helligkeitsvariationen jeweils geringer aus.

Die besten Ergebnisse der RGB-Farbvariation wurden ebenfalls für die stärkste Variation $\delta_{RGB} = 1$ erreicht (s. Anhang C.18). Die höchste Performanzsteigerung von ΔAP 2.5 % zum Basisdatensatz wurde dabei nach 18 Trainingsepochen erzielt. Wie beim Pre-Evaluierungsdatensatz lag die höchste erreichte Performanz bei der Helligkeitsvariation (hier für $\delta_L = 1$) mit ΔAP 3.4 % zur Basis über der Performanz der RGB-Farbvariation. Die signifikant besten Ergebnisse wurden mit der Kombination der Augmentationen erreicht, wobei die Kombination $\delta_{RGB,L} = \{0.8, 0.8\}$ die insgesamt höchste Performanz von ΔAP 4.3 % zum Basisdatensatz erzielte (s. Anhang C.19).

Im HSV-Farbraum erreichte ebenfalls die volle HSV-Variation mit ΔAP 2.0 % zum Basisdatensatz die signifikant höchste Performanz (s. Anhang C.20). Im Gegensatz zum Pre-Evaluierungsdatensatz fielen die Performanzunterschiede zwischen den Variationen H, S, HS stärker aus, wobei insbesondere die Variation des Farbwerts H bessere Ergebnisse erzielte als die Variation der Sättigung S.

Zusammenfassung

Aufgrund der Studienergebnisse kann die **Hypothese** in Bezug auf die Augmentation mit RGB-Farb-, Helligkeits- und HSV-Variation für den untersuchten Anwendungsfall **bestätigt** werden. Durch alle einzelnen Augmentationen konnten signifikante Verbesserungen der Performanz der trainierten Objekterkennungssysteme erreicht werden, wobei die Verbesserung bei der RGB-Farbvariation schwächer ausfiel als bei der Helligkeitsvariation. Die höchste Performanz wurde durch die Kombination von RGB-Farb- und Helligkeitsvariation erreicht. Die HSV-Augmentation konnte diese Kombination nicht ohne signifikante Performanzverluste ersetzen.

5. Experimente und Ergebnisse

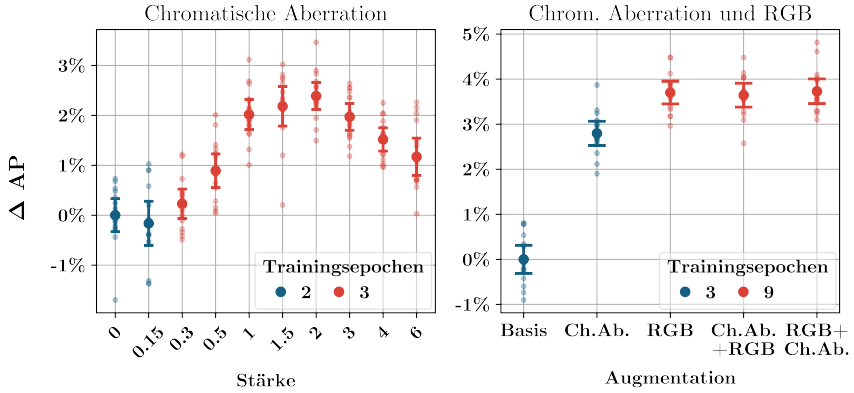


Abbildung 5.16.: Ergebnisse der Augmentation mit chromatischer Aberration für unterschiedliche Variationsstärken (links) sowie der kombinierten Augmentation von RGB-Farbvariation und chromatischer Aberration (rechts).

5.3.3. Chromatische Aberration

Im Rahmen der Studie zur Augmentation mit chromatischer Aberration wurden die neun Variationsstärken $\delta_C = \{0.15, 0.3, 0.5, 1, 1.5, 2, 3, 4, 6\}$ sowie die Kombination mit RGB-Farbvariation untersucht.

Pre-Evaluierungsdatensatz: Wie Abbildung 5.16 (links) zeigt, wurden die höchsten Performanzkennwerte für die Variationsstärken $1.5 \leq \delta_C \leq 2$ erreicht (insignifikante Unterschiede). Die besten Ergebnisse wurden mit $\delta_C = 2$ bereits nach 3 Trainingsepochen erzielt (ΔAP 2.3% zur Basis) – ein längeres Training erbrachte keine höhere Performanz. Alle getesteten Variationsstärken $\delta_C \geq 0.5$ (ΔAP 0.9% bei $\delta_C = 0.5$) führten zu einer signifikanten Verbesserung der Performanz gegenüber dem Basisdatensatz ohne chromatische Aberration.

Kombination (Pre-Eval.): Bei der Untersuchung der Augmentationskombination von chromatischer Aberration und RGB-Farbvariation $\delta_{RGB,C} = \{1, 2\}$ konnte mit $\Delta AP -0.1\%$ im direkten Vergleich keine Verbesserung gegenüber der reinen Farbvariation $\delta_{RGB} = 1$ gemessen werden (rechts).

Somit kann für den untersuchten Anwendungsfall davon ausgegangen werden, dass eine Augmentation mit chromatischer Aberration keinen über die RGB-Farbvariation hinausgehenden Beitrag zur Verbesserung des CNN-Trainings leistet. Im weiteren Verlauf der Experimente wurde daher auf eine weitere Untersuchung der chromatischen Aberration verzichtet.

Post-Evaluierungsdatensatz: Für den Post-Evaluierungsdatensatz wurden für die Augmentation mit chromatischer Aberration nur geringe Verbesserungen gemessen (s. Anhang C.21). Die signifikant besten Ergebnisse wurden mit $1 \leq \delta_C \leq 2$ erreicht, wobei $\delta_C = 1$ die insgesamt höchste Performanz erzielte (ΔAP 0.5 % zur Basis). Die Kombination der chromatischen Aberration mit der RGB-Farbvariation erreichte auch hier keine höhere Performanz als die reine Farbvariation (ΔAP -0.4 % im direkten Vergleich).

Zusammenfassung

Aufgrund der Studienergebnisse kann die **Hypothese** in Bezug auf die Augmentation mit chromatischer Aberration für den untersuchten Anwendungsfall **bestätigt** werden. Durch diese Augmentation wurden signifikante Verbesserungen der Performanz der trainierten Objekterkennungssysteme erreicht. Da die Augmentation mit chromatischer Aberration in Kombination mit der RGB-Farbvariation für den getesteten Fall jedoch keine zusätzliche Verbesserung erbrachte, wird diese im weiteren Verlauf der Experimente nicht berücksichtigt.

5.3.4. Kontrast

Im Rahmen der Studie zum Einfluss der Kontrastvariation auf die Performanz des trainierten Objekterkennungssystems wurden die sieben Variationsstärken $\delta_K = \{1.2, 1.5, 3, 5, 10, 20, 40\}$ im Vergleich zum unveränderten Basisdatensatz (entspricht $\delta_K = 1$) untersucht.

Pre-Evaluierungsdatensatz: Wie Abbildung 5.17 zeigt, wurden die signifikant höchsten Performanzkennwerte beim Training mit bis zu 3 Epochen für die Variationsstärken $\delta_K = \{10, 20\}$ erreicht (insignifikante Unterschiede). Dabei erzielte der Datensatz $\delta_K = 20$ mit ΔAP 4.4 % zum Basisdatensatz die besten Ergebnisse. Alle getesteten Variationsstärken führten zu

5. Experimente und Ergebnisse

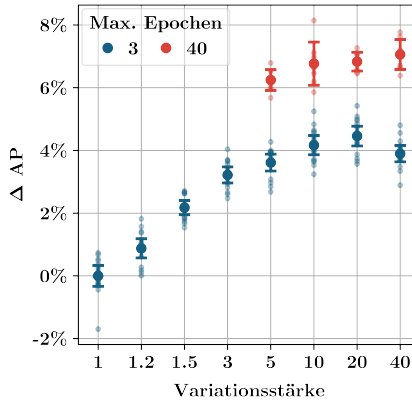


Abbildung 5.17.: Ergebnisse der Augmentation mit Kontrastvariation für unterschiedliche Variationsstärken.

einer signifikanten Verbesserung der Performanz um mindestens ΔAP 0.8 % gegenüber dem Basisdatensatz ohne Kontrastvariation.

Die Datensätze mit $\delta_K \geq 5$ wurden zur Überprüfung der Saturation zusätzlich länger trainiert. Die insgesamt höchste Performanz erreichte der Datensatz $\delta_K = 40$ mit ΔAP 7.0 % zum Basisdatensatz nach 24 Epochen, wobei die Unterschiede für $\delta_K \geq 10$ insignifikant waren.

Bei der Untersuchung alternativer Berechnungsmethoden für die Stärke der Kontrastvariation erwies sich die gewählte Variante mit „stabilem Bildkanalmittelwert“ als am besten geeignet (s. Anhang A.2).

Post-Evaluierungsdatensatz: Für den Post-Evaluierungsdatensatz ergibt sich ein ähnliches Bild für das Training mit bis zu 3 Epochen (s. Anhang C.22), jedoch mit einer weniger stark ausgeprägten Verbesserung durch die Kontrastvariation um maximal ΔAP 1.4 % für $\delta_K = 10$ (insignifikanter Unterschied zu $\delta_K = 20$). Bei längerem Training mit bis zu 40 Epochen erreichte der Datensatz mit der Kontrastvariation $\delta_K = 20$ die höchste Performanz (ΔAP 4.5 % zur Basis), wobei die Unterschiede für $\delta_K \geq 10$ auch hier insignifikant waren.

Zusammenfassung

Aufgrund der Studienergebnisse kann die **Hypothese** in Bezug auf die Kontrastaugmentation für den untersuchten Anwendungsfall **bestätigt** werden. Durch die Kontrastvariation wurden signifikante Verbesserungen der Performanz der trainierten Objekterkennungssysteme erreicht. Diese stellen die stärksten Performanzsteigerungen unter den untersuchten einzelnen Augmentationstechniken dar.

5.3.5. Kombination Augmentationen

Im Rahmen der Studie zur Kombination der Augmentationstechniken wurden die Augmentationen Unschärfe und Rauschen, Farbe und Helligkeit, HSV- und SV sowie Kontrast untersucht, um Rückschlüsse auf deren Wechselwirkung zu gewinnen und potenziell weitere Verbesserungen der synthetischen Trainingsdaten zu erreichen. Die zugehörige Hypothese lautet:

Hypothese

Synthetische Bilddaten können durch eine **Kombination** von **Augmentationstechniken** im Vergleich zu synthetischen Bilddaten auf Basis einzelner Augmentationen verbessert werden.

Auf Basis der vorangegangenen Studien wurden die folgenden Augmentationsparameter für die zusätzlichen Kombinationen ausgewählt, wobei im Falle insignifikanter Unterschiede jeweils geringere Variationsstärken bevorzugt wurden:

1. Rauschen und Unschärfe (N+U) mit
 $\delta_{N,U} = \{512, 21\}$
2. Farb- und Helligkeitsvariation (RGB+L) mit
 $\delta_{RGB+L} = \{1, 0.4\}$
3. HSV-Variation (HSV) und SV-Variation (SV) mit
 $\delta_{HSV} \in \{\{1, 1, 1\}, \{0, 1, 1\}\}$
4. Kontrastvariation (K) mit
 $\delta_K = 10$

5. Experimente und Ergebnisse

Auf weitere Untersuchungen der Unschärfe mit Box-Filter (s. Unterabschnitt 5.3.1) und der chromatischen Aberration (s. Unterabschnitt 5.3.3) wird dabei, wie in den jeweiligen Abschnitten beschrieben, verzichtet. Alle im Folgenden ausgewerteten Objekterkennungssysteme wurden mit bis zu 40 Epochen trainiert. Die Ergebnisse sind in Tabelle 5.2 aufgeführt.

Rauschen und Unschärfe (Pre-Eval.): Die Kombination der Augmentation von Rauschen und Unschärfe mit der HSV- (ΔAP 1.3 %) und der SV-Variation (ΔAP 1.4 %) führten im direkten Vergleich zu der reinen Augmentation mit Rauschen und Unschärfe zu signifikanten Verbesserungen der Performanz. Dagegen wurden durch die Kombinationen von Rauschen und Unschärfe mit RGB-Farb- und Helligkeitsvariation (ΔAP -0.8 %) sowie mit Kontrastvariation (ΔAP -1.4 %) jeweils im direkten Vergleich signifikant niedrigere Performanzkennwerte erreicht als ohne die Augmentation mit Rauschen und Unschärfe. In allen untersuchten Fällen war die Kombination mit Rauschen und Unschärfe mit weiteren Augmentationen nicht kommutativ, wobei die vorherige Anwendung von Rauschen und Unschärfe jeweils zu signifikant besseren Ergebnissen führte.

Farb- und Kontrastvariation (Pre-Eval.): Weder durch die Kombination der Kontrastaugmentation mit der RGB- und Helligkeitsvariation (ΔAP -0.2 % im direkten Vergleich) noch mit der HSV-Variation (ΔAP -0.03 % im direkten Vergleich) konnten Verbesserungen gegenüber der reinen Kontrastvariation gemessen werden. Die Kombination aus SV- und Kontrastvariation führte im direkten Vergleich zu einer Steigerung der Performanz um ΔAP 0.3 %, jedoch ohne signifikanten Unterschied. Dies stellt die insgesamt höchste Performanz aller Augmentationen dar und liegt um ΔAP 7.3 % über der Performanz des Basisdatensatzes ohne Augmentation.

Post-Evaluierungsdatensatz: Für den Post-Evaluierungsdatensatz erreichte die Kontrastvariation mit ΔAP 4.5 % zum Basisdatensatz ebenfalls die signifikant besten Ergebnisse unter den nicht kombinierten Augmentationen (s. Anhang Tabelle C.1). Im Gegensatz zum Pre-Evaluierungsdatensatz schnitt die HSV- im direkten Vergleich zur SV-Variation signifikant besser ab (ΔAP 1.4 %). Die insgesamt höchste Performanz wurde mit der Kombination aus HSV- und Kontrastvariation (ΔAP 4.9 % zur Basis) erreicht, wobei die Unterschiede zur Kontrastvariation hier signifikant waren (ΔAP 0.4 % im direkten Vergleich).

Tabelle 5.2.: Maximale Verbesserungen der Performanz (ΔAP [%]) für alle einzelnen Augmentationen und ausgewählte Kombinationen. Die signifikant höchsten Kennwerte sind hervorgehoben.

Augmentation	Max. Perf. $\Delta AP \pm \text{Std.}$	Direkter Vergl. ΔAP (Aug.)	Epochen max. Perf.
Basis (SE-Opt)	0.0 ± 0.6		2
Gauß. Rauschen (N)	1.3 ± 0.6		6
Chrom. Aberration (C)	2.3 ± 0.5		3
RGB-Variation (RGB)	3.3 ± 0.4		6
Unschärfe Box (X)	3.6 ± 0.4		40
Unschärfe Gauß (U)	3.7 ± 0.6		24
Helligkeitsvariation (L)	3.8 ± 0.7		40
SV-Variation (SV)	4.1 ± 0.5		12
HSV-Variation (HSV)	4.8 ± 0.6		24
Kontrastvariation (K)	7.0 ± 0.5		24
RGB + C	3.2 ± 0.5	-0.1 (RGB)	18
N + U	5.0 ± 0.3	1.3 (U)	40
N + U + RGB + L	5.0 ± 0.4	-0.8 (RGB + L)	40
N + U + K	5.6 ± 0.3	-1.4 (K)	40
RGB+L	5.8 ± 0.3	2.5 (RGB)	40
N + U + HSV	6.3 ± 0.5	1.3 (N + U)	40
N + U + SV	6.4 ± 0.4	1.4 (N + U)	40
RGB + L + K	6.5 ± 0.4	-0.5 (K)	40
HSV + K	7.0 ± 0.5	0.0 (K)	40
SV + K	7.3 ± 0.6	0.3 (K)	40

Zusammenfassung

Aufgrund der Studienergebnisse kann die **Hypothese** in Bezug auf die Kombination der Augmentationstechniken für den untersuchten Anwendungsfall **bestätigt** werden. Zwar konnten beim Pre-Evaluierungsdatensatz durch Kombinationen lediglich insignifikante Steigerungen der Performanz (SV und Kontrast) gegenüber der reinen Kontrastvariation erzielt werden. Bei der Evaluierung mit dem Post-Datensatz wurde hingegen die signifikant höchste Performanz mit der Kombination aus HSV- und Kontrastvariation gemessen.

5.4. Domain Adaptation

Im Folgenden werden die gewählten Methoden der Domain Adaptation, die Bildanpassung mit Image Color Transfer (ICT) sowie die Transformation mit GANs untersucht und miteinander verglichen. Es wird überprüft, ob sich die synthetischen Datensätze durch diese Verfahren sowie deren Kombination für das Training von Objekterkennungsverfahren (hier CNNs) verbessern lassen. Die Domain Adaptation wird dabei jeweils auf den SE-Opt-Datensatz angewendet. Entsprechende Beispielbilder finden sich in Abbildung 5.18.

5.4.1. ICT

Im Rahmen der Studie zur Domain Adaptation mittels ICT wurde der Einfluss des Referenzdatensatzes und der Zuordnungsmethode des Referenzbildes untersucht. Dabei wurden die Objekterkennungsverfahren jeweils mit bis zu 9 Epochen trainiert. Die zugehörige Hypothese lautet wie folgt:

Hypothese

Synthetische Bilddaten können durch die **Anpassung** an reale Referenzdaten mit Image Color Transfer (**ICT**) im Vergleich zu den ursprünglichen synthetischen Bilddaten verbessert werden.

Referenzdatensatz: Für die Untersuchung des Einflusses des Referenzdatensatzes wurden für die Anpassung mit klassenbasiertem Matching alle in Unterabschnitt 2.1.1 aufgeführten Realdatensätze verwendet.

Dabei wurde der spätere Evaluierungsdatensatz jeweils auch als Referenz für die bildstatistische Anpassung verwendet. Dies stellt einen Verstoß gegen das Konzept des getrennten Testdatensatzes dar, wurde aber dennoch unternommen, um die Auswirkungen von perfekt übereinstimmenden Referenz- und Evaluierungsdatensätzen zu überprüfen. Es wurde erwartet, dass die Anpassung und Evaluierung mit demselben Datensatz zu den besten Ergebnissen führt.

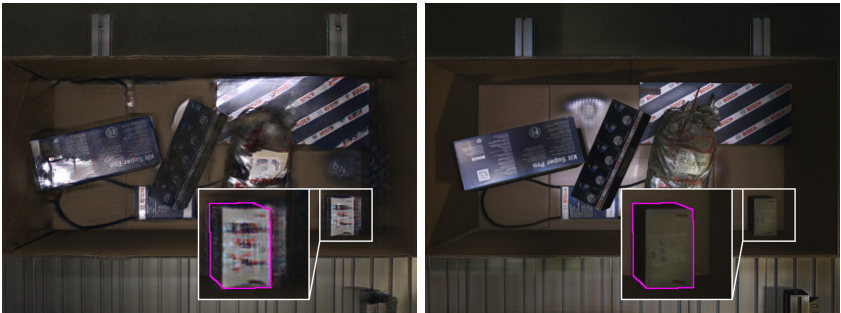
Zunächst wurde das Potenzial von ICT auf Basis des realen Pre-Datensatzes untersucht (als Referenz- und Evaluierungsdatensatz). Wie Abbildung 5.19 (links) zeigt, erreichte das Training die signifikant höchste Performanz nach 6 Epochen, wobei die Performanz um ΔAP 3.1 % zum Basisdatensatz erhöht werden konnte. Die Verwendung des Midi-Referenzdatensatzes



(a) Synthetisches Quellbild (links) und reales Referenzbild (rechts).



(b) ICT-Anpassung ohne (links) und mit (rechts) maskenbasierter Anwendung.



(c) GAN-Transformation ohne (links) und mit (rechts) Gradient-Penalty-Verlust. Ohne den Gradient-Penalty-Verlust (links) ist eine Annotationsinkonsistenz an der rechten Außenkante der braunen Box zu erkennen (Vergrößerung).

Abbildung 5.18.: Beispielbilder für die untersuchten Methoden der Domain Adaptation: (a) Eingabebilder, (b) ICT und (c) GAN.

5. Experimente und Ergebnisse

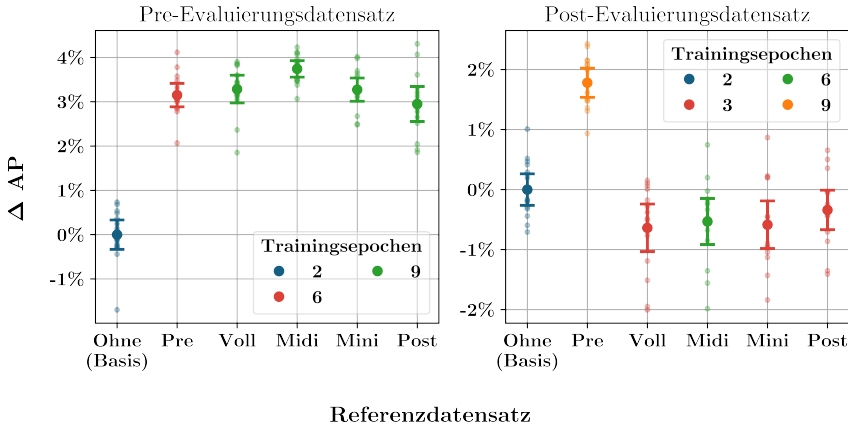


Abbildung 5.19.: Ergebnisse der ICT-Bildanpassung mit unterschiedlichen realen Referenzdatensätzen bei Evaluierung mit dem Pre- (links) und dem Post-Datensatz (rechts).

führte mit ΔAP 3.7 % zum Basisdatensatz nach 9 Trainingsepochen zu den signifikant besten Ergebnissen bei der Evaluierung mit dem Pre-Datensatz. Damit lag die Performanz signifikant über der des Pre-Referenzdatensatzes (ΔAP 0.6 % im direkten Vergleich). Für alle Referenzdatensätze führte die bildstatistische Anpassung zu einer signifikanten Verbesserung gegenüber dem Basis-Datensatz ohne Anpassung.

Abbildung 5.19 (rechts) zeigt, dass die signifikant besten Ergebnisse bei der Evaluierung mit dem Post-Datensatz mit dem Pre-Referenzdatensatzes erreicht wurden (ΔAP 1.8 % zur Basis nach 9 Epochen). Die Anpassungen mit den zum Post-Datensatz eng verwandten Datensätzen (Voll, Midi und Mini) führten zu keiner Verbesserung gegenüber dem Basis-Datensatz. Damit lag die Performanz auch hier bei unterschiedlichen Referenz- und Evaluierungsdatensätzen signifikant höher als bei Verwendung desselben Datensatzes als Referenz und für die Evaluierung.

Die Ergebnisse widersprechen damit der These der höchsten Performanz bei voller Übereinstimmung zwischen Referenz- und Evaluierungsdatensatz. Dies deutet an, dass nicht die eigentliche Anpassung an die Bildstatistik der Referenz, sondern beispielsweise die durch die Anpassung bedingten Farb-, Helligkeits- oder Kontrastvariationen zur Erhöhung der Performanz führen.

Zuordnungsmethode: Bei der Untersuchung des Einflusses der Art der Bildzuordnung wurden die folgenden drei Methoden miteinander verglichen:

1. Klassen (Basis): Zuordnung anhand höchster Übereinstimmung der annotierten Klassen und Bildanpassung auf Ebene des Gesamtbildes
2. Randomisiert: Zufällige Zuordnung ohne Berücksichtigung der Annotationen und Bildanpassung auf Ebene des Gesamtbildes
3. Maske: Zuordnung anhand höchster Übereinstimmung der annotierten Klassen und getrennte Anpassung für die einzelnen Objektinstanzen mit übereinstimmender Klasse auf Ebene der Masken

Als Referenz wurde der Pre-Datensatz verwendet und die Evaluierung mit dem Post-Datensatz durchgeführt. Beispielbilder der klassen- und der maskenbasierten Zuordnung finden sich in Abbildung 5.18b.

Wie Abbildung 5.20 (links) zeigt, führten alle ICT-Datensätze zu einer signifikanten Steigerung der Performanz gegenüber dem Basis-Datensatz. Beim Pre-Evaluierungsdatsatz bestanden dabei keine signifikanten Unterschiede zwischen den Zuordnungsmethoden. Die höchste Performanz wurde hierbei mit der randomisierten Zuordnung erreicht (ΔAP 3.3 % zur Basis). Bei der Evaluierung mit dem Post-Datensatz zeigte sich die Zuordnung auf Maskenebene dagegen den anderen Verfahren als signifikant überlegen (ΔAP 2.7 % zur Basis und ΔAP 0.7 % zur randomisierten Zuordnung).

Darüber hinaus wurde bei der Anwendung von ICT auf einzelne Objektinstanzen (Maskenebene) eine Unschärfmaskierung der Außenkanten der einzelnen Objekte mit unterschiedlicher Stärke untersucht. Hierfür wurden Box-Weichzeichenfilter mit den Größen 3, 5 und 7 eingesetzt. Die Messergebnisse in Abbildung 5.20 (rechts) zeigen jedoch keinen positiven Einfluss der Unschärfmaskierung auf die Performanz der trainierten Objekterkennungssysteme für den Pre-Evaluierungsdatsatz: ΔAP -0.2% bis ΔAP -0.4% zum Datensatz ohne Unschärfmaskierung (Filtergröße 0). Bei der Evaluierung mit dem Post-Datensatz kam es durch die Steigerung der Unschärfmaskierung sogar zu signifikanten Reduktionen der Performanz.

5. Experimente und Ergebnisse

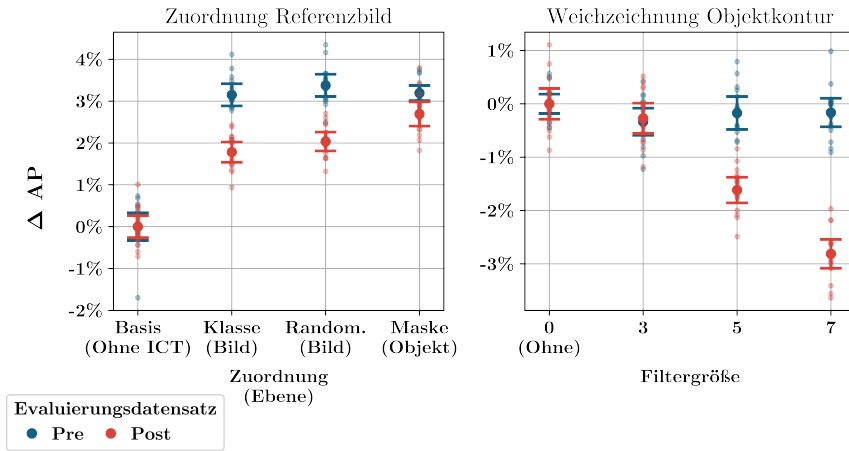


Abbildung 5.20.: Ergebnisse der ICT-Bildanpassung mit unterschiedlichen Zuordnungsmethoden der Referenzbilder (links) und mit unterschiedlich starker Weichzeichnung der Objektkonturen für die Zuordnung auf Maskenebene (rechts).

Zusammenfassung

Aufgrund der Studienergebnisse kann die **Hypothese** für den untersuchten Anwendungsfall **bestätigt** werden. Eine Transformation durch Image Color Transfer (ICT) führte zu signifikant besseren synthetischen Trainingsdaten. Der Grad der Übereinstimmung mit den Evaluierungsdaten sowie die Größe des realen Referenzdatensatzes korrelierten in den durchgeführten Experimenten jedoch nicht mit der Performanz des Objekterkennungssystems. Dies deutet darauf hin, dass die Steigerungen der Performanz nicht unmittelbar durch die Anpassung selbst, sondern durch Effekte auf die Bildeigenschaften bedingt sind, die mit der ICT-Transformation einhergehen. Darüber hinaus zeigte sich die Anwendung der statistischen Farbanpassung auf einzelne Objektinstanzen ohne Unschärfmaskierung als einer einheitlichen Anpassung des Gesamtbildes überlegen.

5.4.2. GAN

Im Rahmen der Studie zur Domain Adaptation mit GANs wurden die Anzahl Trainingsschritte, die Verwendung eines Gradient-Penalty-Verlustes (GP-Verlust) und die Wahl des Referenzdatensatzes beim Training der GANs betrachtet. Das anschließende Training der Objekterkennungsverfahren wurde mit bis zu 9 Epochen durchgeführt. Beispielbilder der GAN-Transformation finden sich in Abbildung 5.21. Die zugehörige Hypothese lautet wie folgt:

Hypothese

Synthetische Bilddaten können durch die **Anpassung** an reale Referenzdaten mit Generative Adversarial Networks (**GANs**) im Vergleich zu den ursprünglichen synthetischen Bilddaten verbessert werden.

Trainingsumfang: Bei der Untersuchung des Einflusses des Trainingsumfangs wurden die GANs mit 10 000, 14 000, 16 000, 30 000 und 100 000 Trainingsschritten zunächst ohne GP-Verlust trainiert. Die Ergebnisse in Abbildung 5.22 (links) zeigen, dass alle transformierten Trainingsdatensätze zu signifikanten Steigerungen der Performanz führten, wobei mit ΔAP 4.7 % nach 100 000 GAN-Trainingsschritten die signifikant höchste Steigerung der Performanz erreicht wurde.

Das GAN-Training mit 16 000 Trainingsschritten stellt unter den GAN-transformierten Datensätzen in Bezug auf die erreichte Performanz einen Ausreißer dar (ΔAP 1.8 % zur Basis). Jedoch weist dieser Datensatz weder bei subjektiver Beurteilung (s. Abbildung 5.21b) noch bei statistischer Analyse Auffälligkeiten gegenüber den anderen GAN-Datensätzen auf.

Gradient-Penalty-Verlust: Bei Verwendung des GP-Verlustes beim GAN-Training ergeben sich transformierte Bilddaten, welche bei subjektiver Beurteilung eindeutig weniger Annotationsinkonsistenzen aufweisen. Ein Beispiel für eine solche Inkonsistenz zwischen dem transformierten Bild und der ursprünglichen Annotation findet sich in Abbildung 5.18c. Beim Training ohne GP-Verlust ist eine „Erweiterung“ des Objektes über die ursprüngliche Maske hinaus zu erkennen (pinkfarben umrandet).

5. Experimente und Ergebnisse

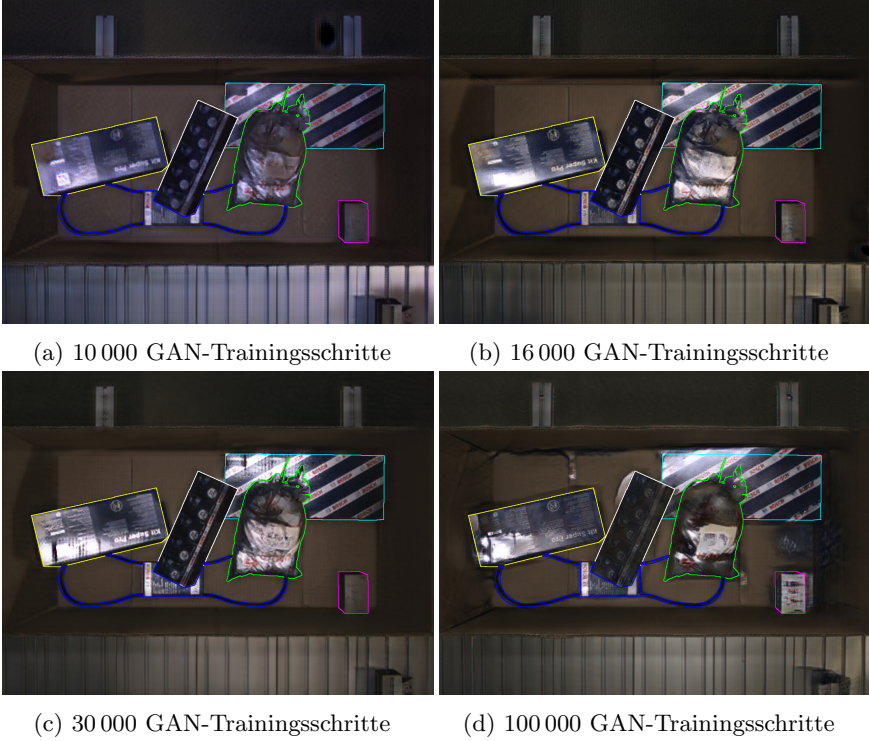


Abbildung 5.21.: Beispielbilder für die GAN-Transformation mit dem Pre-Referenzdatensatz und unterschiedlicher Anzahl an Schritten beim Training des GANs.

Wie aus Abbildung 5.22 (links, letzter Eintrag) hervorgeht, wurden trotz der Annotationsinkonsistenzen keine signifikanten Unterschiede für den Pre-Evaluierungsdatensatz zwischen dem GAN-Training mit und ohne GP-Verlust gemessen (jeweils ΔAP 4.7 % zur Basis). Bei Evaluierung auf dem Post-Datensatz dagegen erreichten die transformierten Datensätze mit GP-Verlust mit ΔAP 1.3 % im direkten Vergleich eine signifikant höhere Performanz als die transformierten Daten ohne GP-Verlust.

Referenzdatensatz: Für die Untersuchung des Einflusses des Referenzdatensatzes wurden für das GAN-Training alle in Unterabschnitt 2.1.1 auf-

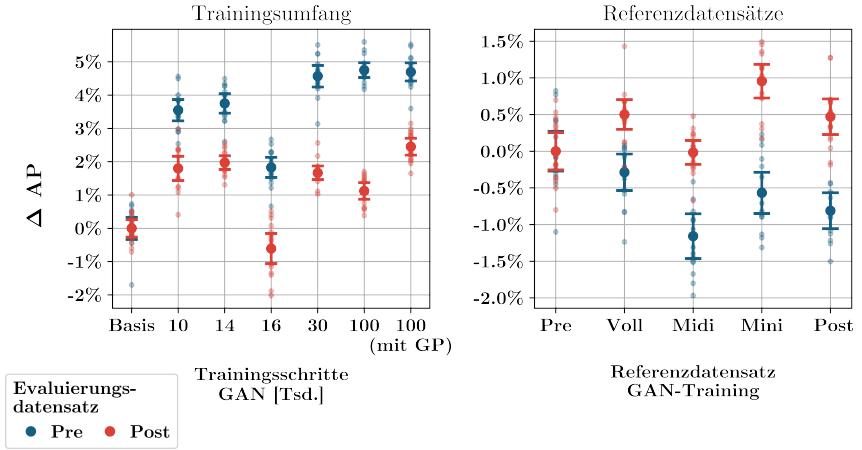


Abbildung 5.22.: Ergebnisse der GAN-Transformation mit unterschiedlichem Trainingsumfang (links) ohne und mit Gradient-Penalty-Verlust (letzter Eintrag links) sowie unterschiedlichen Referenzdatensätzen (rechts).

geführten Realdatensätze verwendet. Das GAN-Training wurde jeweils mit 100 000 Schritten durchgeführt und ein GP-Verlust verwendet. Die trainierten CNNs wurden jeweils sowohl mit dem Pre- als auch dem Post-Datensatz evaluiert. Mit derselben Begründung wie in Absatz 5.4.1 (Referenzdatensatz) wurden beide Datensätze auch als Referenz für das GAN-Training verwendet. Entsprechende Beispielbilder finden sich in Anhang B.5.

Im Gegensatz zur Bildanpassung mittels ICT bestätigte sich die These der höchsten Performanz bei voller Übereinstimmung zwischen Referenz- und Evaluierungsdatensatz bei der GAN-Transformation. Wie Abbildung 5.22 (rechts) zeigt, führte die Verwendung des Pre-Referenzdatensatz zu den besten Ergebnissen (ΔAP 4.7% zur Basis ohne Domain Adaptation) bei der Evaluierung mit dem Pre-Datensatz.

Für den Post-Evaluierungsdatensatz wurden ebenso mit den eng verwandten Referenzdatensätzen (Voll, Mini und Post) signifikant bessere Ergebnisse als mit dem weiter entfernten Pre-Referenzdatensatz erzielt (ΔAP 2.4% zur Basis). Eine Ausnahme bildet der Midi-Referenzdatensatz, der für beide Evaluierungsdatensätze die niedrigste Performanz erreichte. Die höchste Performanz wurde mit ΔAP 3.4% zum Basisdatensatz bei Verwendung

5. Experimente und Ergebnisse

des Mini-Referenzdatensatzes erreicht. Die volle Übereinstimmung zwischen Referenz- und Evaluierungsdatensatz (jeweils Post-Datensatz) führte dagegen zu einer signifikant niedrigeren Performanz als die Anpassung an den Mini-Referenzdatensatz ($\Delta AP -0.5\%$ im direkten Vergleich).

Zusammenfassung

Aufgrund der Studienergebnisse kann die **Hypothese** für den untersuchten Anwendungsfall **bestätigt** werden. Durch die Transformation der synthetischen Trainingsdaten mit Generative Adversarial Networks (GANs) wurden signifikant bessere Ergebnisse erzielt als ohne Domain Adaptation. Eine höhere Übereinstimmung zwischen Referenz- und Evaluierungsdatensatz führte dabei zu einer höheren Performanz der trainierten CNNs. Darüber hinaus konnten zwar keine signifikanten Unterschiede zwischen dem GAN-Training mit und ohne Gradient-Penalty-Verlusts gemessen werden, jedoch verlief das Training mit GP-Verlust auch bei mehr Schritten stabiler und mit erkennbar weniger Annotationsinkonsistenzen.

5.4.3. Kombination GAN und ICT

Im Rahmen der Studie zur Kombination der beiden Domain Adaptation-Verfahren wurden sechs Datensätze jeweils mit oder ohne GAN-Transformation sowie mit klassenbasierter, mit maskenbasierter oder ohne ICT-Bildanpassung miteinander verglichen. Die Vergleichsbasis stellt der unveränderte SE-Opt-Datensatz dar. Die zugehörige Hypothese lautet wie folgt:

Hypothese

Synthetische Bilddaten können durch die **Kombination** von **GAN- und ICT-Transformation** im Vergleich zur einzelnen Anwendung der beiden Verfahren verbessert werden.

Dabei wurden die GANs jeweils mit einem GP-Verlust trainiert, die GAN- vor der ICT-Transformation angewandt und entweder der Pre- oder der Mini-Realdatensatz als Referenz verwendet. Das Training der Objekterkennungsverfahren wurde im Gegensatz zu den Einzelstudien mit bis zu 40 Epochen durchgeführt und mit Pre- und dem Post-Datensatz evaluiert.

Wie Tabelle 5.3 zeigt, wurde die höchste Performanz bei Evaluierung mit

Tabelle 5.3.: Maximal erreichte Performanzverbesserungen durch die Domain Adaptation mit ICT (Klasse ✓ oder Maske M), GANs sowie deren Kombination. Dabei wird die ΔAP jeweils im Vergleich zum SE-Opt-Datensatz ohne Modifikation angegeben (erste Zeile) und jeweils mit dem Pre- und dem Post-Datensatz evaluiert. Die signifikant höchsten Kennwerte je Referenz- und Evaluierungsdatensatz sind hervorgehoben.

Methode		Referenz- datensatz	Max. Perform. $\Delta AP \pm \text{Std.} [\%]$		Epochen bis max. Perform.	
ICT	GAN		Pre	Post	Pre	Post
-	-	-	0.0 ± 0.6	0.0 ± 0.5	2	2
✓		Pre	3.1 ± 0.5	1.9 ± 0.5	6	12
M			3.5 ± 0.8	2.9 ± 0.3	40	24
	✓		5.7 ± 0.5	3.8 ± 0.5	40	40
✓	✓		6.2 ± 0.7	3.6 ± 0.4	40	40
M	✓		2.9 ± 0.6	2.6 ± 0.4	9	9
✓		Mini	4.0 ± 0.6	-0.6 ± 0.7	18	3
M			4.7 ± 0.5	3.0 ± 0.4	24	24
	✓		4.4 ± 0.7	4.2 ± 0.3	24	40
✓	✓		5.1 ± 0.6	4.0 ± 0.3	40	40
M	✓		4.5 ± 0.7	4.0 ± 0.3	24	24

dem Pre-Datensatz für beide Referenzdatensätze mit der Kombination aus GAN-Transformation und klassenbasierter ICT erreicht. Die Verbesserung mit den Pre-Referenzdaten (ΔAP 6.2 % zur Basis) war dabei signifikant höher als mit den Mini-Referenzdaten (ΔAP 5.1 % zur Basis).

Bei der Evaluierung mit dem Post-Datensatz wurden jeweils die besten Ergebnisse bei alleiniger GAN-Anpassung erreicht. Die Steigerungen der Performanz zum Basisdatensatz beliefen sich auf ΔAP 3.8 % mit dem Mini-Referenzdatensatz und ΔAP 4.2 % Pre-Referenzdatensatz. Die mit der Kombination aus GAN und ICT erreichte Performanz war dabei für beide Evaluierungsdatensätze nicht signifikant niedriger.

Während die Anpassungen auf Maskenebene bei der alleinigen Anwendung von ICT jeweils zu signifikant besseren Ergebnissen gegenüber der Anpassung auf Bildebene führten, konnte bei der Kombination mit einer vorherigen GAN-Transformation keine Verbesserung durch die Anpassung auf

5. Experimente und Ergebnisse

Maskenebene gemessen werden. Die Ergebnisse der GAN-transformierten Datensätze verbesserten sich beim Pre-Evaluierungsdatsatz jeweils durch die Anwendung von ICT – beim Post-Evaluierungsdatsatz dagegen reduzierte sich die Performanz jeweils.

Zusammenfassung

Aufgrund der Studienergebnisse kann die **Hypothese** für den untersuchten Anwendungsfall **bestätigt** werden. Die Kombination aus GAN-Transformation und ICT führte für den Mini-Referenz- und den Pre-Evaluierungsdatsatz zu signifikant besseren synthetischen Trainingsdaten als die Verwendung der einzelnen Anpassungsverfahren. Bei der Evaluierung mit dem Post-Datsatz konnte durch die Kombination keine signifikanten Unterschiede zur alleinigen GAN-Transformation gemessen werden. Bei der Kombination mit der GAN-Transformation erzielte die klassenbasierte ICT-Anpassung signifikant bessere Ergebnisse als die Anpassung auf Maskenebene.

5.5. Kombination Domain Adaptation und Domain Randomization

Im Folgenden werden die Methoden der Domain Adaptation und Domain Randomization kombiniert auf SE-Opt-Datsatz des IP-A-Anwendungsfalls angewendet. Dabei werden die Auswirkungen verschiedener Kombinationen auf die Performanz der CNNs beim Training mit bis zu 40 Epochen untersucht. Die zugehörige Hypothese lautet:

Hypothese

Synthetische Bilddaten können durch die **Kombination** von **Domain Adaptation** und **Domain Randomization** im Vergleich zur alleinigen Anwendung der Verfahren verbessert werden.

Anschließend finden Übertragungen der Methoden auf weitere Datensätze des IP-A-Anwendungsfalls, weitere gängige und spezifische Metriken, weitere Anwendungsfälle sowie andere CNN-Architekturen statt.

5.5.1. SV und Kontrast

Im Folgenden wird der Einfluss der Augmentation mit SV- und Kontrastvariation sowie deren Kombination (Domain Randomization) auf Datensätze untersucht, welche zuvor mit den Methoden der Domain Adaptation angepasst wurden.

Pre-Evaluierungsdatensatz: Wie Abbildung 5.23 (links) im vertikalen Vergleich zeigt, gehörte die erreichte Performanz der Datensätze mit GAN-Transformation (inkl. GAN + ICT) für alle Augmentationen zu den signifikant höchsten und lag jeweils über der Performanz der reinen ICT-Anpassung. Bei der reinen und der mit SV kombinierten Kontrastvariation (jeweils $\delta_K = 10$; K) waren die Performanzunterschiede zwischen den Datensätzen mit GAN-Transformation (inkl. GAN + ICT) und ohne Domain Adaptation jedoch jeweils nicht signifikant.

Aufgrund dieser Ergebnisse kann vermutet werden, dass sich die Transformationseigenschaften der Domain Adaption und der Kontrastvariation auf die Bildeigenschaften zumindest teilweise überschneiden und sich die beiden Verfahren daher nur wenig ergänzen. Aus diesem Grund wurde zusätzlich eine SV- und Kontrastkombination mit geringerer Variationsstärke $\delta_K = 5$ (K5) überprüft. Bei dieser ergaben sich durch die GAN-Transformation (inkl. GAN + ICT) signifikante Verbesserungen von ΔAP 0.6 % bis ΔAP 0.9 % im direkten Vergleich zum Datensatz SV + K5 ohne Domain Adaptation.

Beim horizontalen Vergleich zeigte sich für jede Domain Adaptation-Methode eine signifikante Steigerung der Performanz durch die zusätzliche Anwendung von Domain Randomization (alle betrachteten Augmentationen) im Vergleich zu den Datensätzen mit derselben Adaptationsmethode jedoch ohne Augmentation.

Der Datensatz mit reiner GAN-Transformation und Augmentation mit SV + K5 erreichte mit ΔAP 7.8 % zum Basisdatensatz die insgesamt höchste Performanz. Es konnten jedoch keine signifikanten Unterschiede zur Performanz des Datensatzes SV + K ohne Domain Adaptation gemessen werden (ΔAP -0.5 % im direkten Vergleich).

Post-Evaluierungsdatensatz: Für den Post-Evaluierungsdatensatz ergibt sich ein ähnliches Bild (s. Abbildung 5.23 rechts). Im Gegensatz zum Pre-Evaluierungsdatensatz erbrachte zusätzliche die Kontrastvariation (K) gegenüber den Datensätze ohne Augmentation keine signifikante Verbesserung mehr bei einer vorherigen GAN-Transformation (inkl. GAN + ICT).

5. Experimente und Ergebnisse

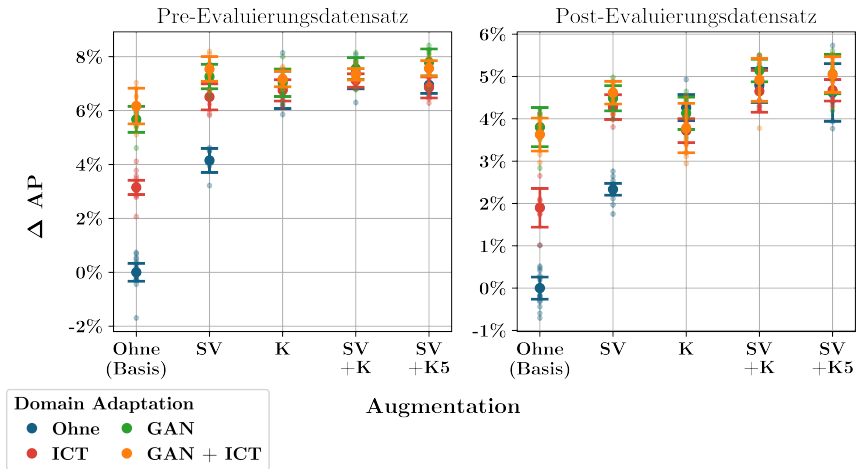


Abbildung 5.23.: Ergebnisse der Kombination von Domain Adaptation und Domain Randomization mit der SV- und der Kontrastaugmentation und deren Kombination im Vergleich zum unveränderten Basisdatensatz SE-Opt.

Die insgesamt höchste Performanz wurde mit der GAN-Transformation (ohne ICT) und der Augmentation SV + K erreicht (ΔAP 5.1 % zur Basis), jedoch waren die Unterschiede zum Datensatz SV + K ohne Domain Adaptation wie beim Pre-Evaluierungsdatensatz insignifikant (ΔAP -0.3 % im direkten Vergleich).

Zusammenfassung

Aufgrund der Studienergebnisse kann die **Hypothese** für den untersuchten Anwendungsfall **nicht bestätigt** werden. Zwar konnten die synthetischen Trainingsdaten mit Domain Adaptation durch die zusätzliche Domain Randomization verbessert werden. Die Trainingsdaten mit Domain Randomization (SV- und Kontrastvariation) erreichten jedoch keine signifikante Steigerung der Performanz durch die Kombination mit vorheriger Domain Adaptation.

5.5.2. Unschärfe und Rauschen

Im Folgenden wird der Einfluss der Augmentation mit Rauschen und Unschärfe sowie die Kombination mit weiteren Augmentationstechniken (Domain Randomization) auf Datensätze untersucht, welche zuvor mit den Methoden der Domain Adaptation angepasst wurden.

Pre-Evaluierungsdatensatz: Wie Abbildung 5.24 (links) im vertikalen Vergleich zeigt, gehörte keiner der Datensätze mit Domain Adaptation für jede Augmentation zu den signifikant besten (bspw. $N + U + SV$).

Beim horizontalen Vergleich zeigte sich nicht für jede Domain Adaptation-Methode eine signifikante Steigerung der Performanz durch die zusätzliche Anwendung von Domain Randomization. Während der Datensatz mit ICT-Anpassung durch alle Augmentationen verbessert werden konnte, wurde bei der GAN-Transformation (ohne ICT) lediglich mit der Augmentation $N + U + SV + K$ eine signifikante Erhöhung der Performanz erreicht (ΔAP 0.7 % im direkten Vergleich zu GAN ohne Augmentation). Die Performanz der kombinierten Domain Adaptation ($GAN + ICT$) reduzierte sich durch die Hinzunahme jeder der untersuchten Augmentationen signifikant.

Die insgesamt höchste Performanz wurde mit ΔAP 6.4 % zum Basisdatensatz für die Augmentation $N + U + SV$ ohne vorherige Domain Adaptation erreicht, wobei die Unterschiede beispielsweise zum Datensatz mit einer GAN-Transformation und der Augmentation $N + U + SV + K$ insignifikant waren (ΔAP -0.1 % im direkten Vergleich).

Post-Evaluierungsdatensatz: Bei der Evaluierung mit dem Post-Datensatz (s. Abbildung 5.24 rechts) zeigten sich im vertikalen Vergleich andere Ergebnisse. Im Gegensatz zum Pre-Evaluierungsdatensatz gehörte die Performanz der GAN-transformierten Datensätze für jede der untersuchten Augmentationen zu den signifikant höchsten und lag durchgängig signifikant über der Performanz der Datensätze mit ICT (inkl. $GAN + ICT$). Dagegen konnte bei der Augmentation $N + U + SV$ lediglich eine insignifikante Verbesserung durch die GAN-Transformation gemessen werden (ΔAP 0.1 % im direkten Vergleich zu $N + U + SV$ ohne Domain Adaptation).

Bei keiner der Domain Adaptation-Methoden (horizontaler Vergleich) zeigte sich für alle untersuchten Augmentationen ein durchgängig positiver Einfluss. Die Performanz der Datensätze mit GAN-Transformation (inkl. $GAN + ICT$) reduzierte sich sogar für alle Augmentationen signifikant im Vergleich zur Variante ohne Augmentation.

5. Experimente und Ergebnisse

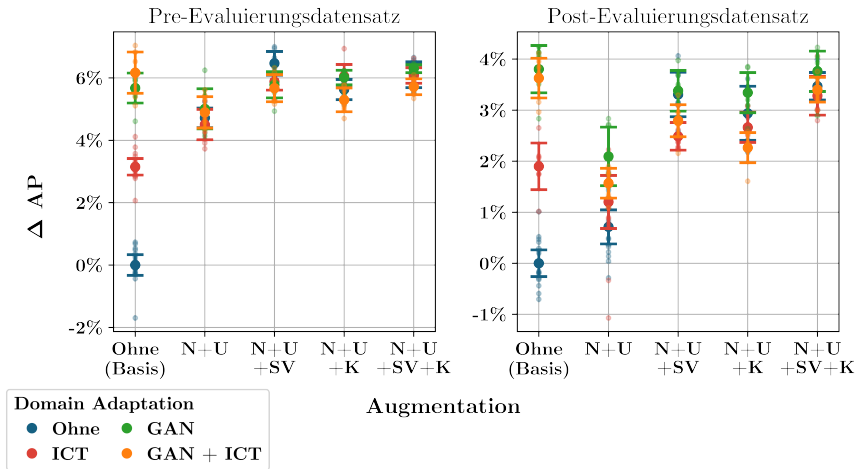


Abbildung 5.24.: Ergebnisse der Kombination von Domain Adaptation und Domain Randomization mit kombinierten Augmentationen mit Rauschen und Unschärfe im Vergleich zum unveränderten Basisdatensatz SE-Opt bei Evaluierung mit dem Pre- (links) und dem Post-Datensatz (rechts).

Die insgesamt höchste Performanz für den Post-Evaluierungsdatensatz wurde mit der GAN-Transformation ohne anschließende Augmentation erreicht (ΔAP 3.8% zur Basis), wobei die Unterschiede beispielsweise zum Datensatz mit einer GAN-Transformation und der Augmentation N + U + SV + K insignifikant waren (ΔAP -0.1% im direkten Vergleich).

Zusammenfassung

Aufgrund der Studienergebnisse kann die **Hypothese** für den untersuchten Anwendungsfall **nicht bestätigt** werden. Für die Augmentation der synthetischen Trainingsdaten mit Rauschen und Unschärfe (Domain Randomization) führte die Kombination mit einer vorherigen Domain Adaptation zu keiner signifikanten Verbesserung der Performanz im Vergleich zur alleinigen Anwendung von Domain Adaptation oder Domain Randomization.

5.5.3. Übertragung

Im Folgenden werden die Methoden der Domain Adaptation und der Domain Randomization auf weitere (1) Datensätze, (2) Metriken, (3) CNN-Architekturen und (4) Anwendungsfälle übertragen. Die erreichte Performanz der trainierten Objekterkennungssysteme wird dabei jeweils mit der durch das Training mit Realdaten erreichten Performanz verglichen. Die zugehörige Hypothese lautet:

Hypothese

Mit durch **Domain Adaptation** und **Domain Randomization** modifizierten **synthetischen Bilddaten** kann eine nicht signifikant niedrigere Performanz erreicht werden als mit **realen Trainingsdaten**.

Die realen und synthetischen Datensätze wurden dabei jeweils in zwei bzw. drei Varianten miteinander verglichen:

1. Ohne Veränderung (Basis-Datensatz)
2. Mit Domain Randomization (DR)
3. Mit Domain Adaptation und anschließender Domain Randomization (DADR; nur bei synthetischen Datensätzen)

Die Domain Adaptation umfasst jeweils eine GAN-Transformation und eine anschließende ICT-Anpassung, wobei jeweils der reale Mini-Datensatz als Referenz verwendet wurde. Für die Domain Randomization wurde jeweils eine SV- und Kontrastvariation ($\delta_K = 10$ für DR und $\delta_K = 5$ für DADR) angewendet. Alle CNNs wurden mit bis zu 40 Epochen trainiert und sowohl mit dem Pre- als auch mit dem Post-Datensatz evaluiert.

Diese Übertragungen stellen nur einen kleinen Ausschnitt und keine repräsentative Untermenge aller hier relevanter Anwendungsfälle und CNN-Architekturen dar. Daher kann damit keine Allgemeingültigkeit der Verbesserung oder generelle Übertragbarkeit der untersuchten Methoden belegt werden. Stattdessen soll an Einzelbeispielen gezeigt werden, ob sich die gewonnenen Erkenntnisse prinzipiell erfolgreich übertragen lassen oder die Übertragungen zu abweichenden Ergebnissen führen.

Datensätze

Bei der Übertragung auf weitere Datensätze wurden neben den drei Realdatensätzen Voll, Midi und Mini auch die synthetischen Datensätze SD-Opt, SE-Preopt, SE-Opt und SE-Exp verwendet (s. Unterabschnitt 5.2.1). Die Ergebnisse sind in Tabelle 5.4 aufgeführt, wobei im Folgenden zunächst die Kennwerte der Durchschnittspräzision (Spalten „AP“) besprochen werden.

Pre-Evaluierungsdatensatz: Bei der Evaluierung mit dem Pre-Datensatz erreichte der SE-Exp-Datensatz in der Variante DADR mit AP 90.2 % die signifikant höchste Performanz und lag um Δ AP 3.6 % über der Performanz, welche mit dem vollen Realdatensatz mit Domain Randomization erreicht wurde. Ohne Modifikation erreichte der SE-Exp-Datensatz lediglich eine um Δ AP 0.4 % höhere Performanz als der volle Realdatensatz (insignifikanter Unterschied).

Mit den modifizierten Varianten DR und DADR wurden sowohl mit dem SE-Exp- als auch mit dem SE-Opt-Datensatz signifikant höhere Performanzkennwerte als mit den realen Trainingsdaten erreicht. Dabei reduzierten sich die Unterschiede in der erreichten Performanz zwischen dem SE-Exp- und SE-Opt-Datensatz durch die Modifikation deutlich.

Die Ergebnisse des SynthDet-Datensatzes (SD-Opt) lagen auch mit Domain Adaptation und Domain Randomization weit unterhalb der Ergebnisse der anderen synthetischen Datensätze und erreichten lediglich in der Variante DADR eine signifikante Verbesserung um Δ AP 4.1 % gegenüber dem Mini-Realdatensatz mit nur 22 Bilddaten.

Post-Evaluierungsdatensatz: Bei der Evaluierung mit dem Post-Datensatz erreichte der volle Realdatensatz die signifikant höchste Performanz von AP 95.8 %. Unter den synthetischen Trainingsdaten erzielte der Datensatz SE-Exp in der Variante DADR mit AP 90.7 % die besten Ergebnisse.

Im Gegensatz zur Evaluierung mit dem Pre-Datensatz reduzierte sich die Performanz der mit Realdaten trainierten CNNs jeweils durch den Einsatz der Randomisierungstechniken (DR). Für fast alle synthetischen Datensätze wurden dagegen Steigerungen der Performanz zwischen der Basis- und der DR-Variante sowie zwischen der DR- und der DADR-Variante gemessen. Eine Ausnahme bildet hierbei der SE-Preopt-Datensatz, bei welchem sich die Performanz durch die Domain Randomization für keinen der Evaluierungsdatensätze steigerte.

Tabelle 5.4.: Maximal erreichte Performanz (AP, AP50, AP75, SR1 und ER1) mit DA (GAN- und ICT-Transformation jeweils mit Mini-Referenzdatensatz) und DR (SV- und Kontrastvariation). Die signifikant höchsten bzw. niedrigsten (ER1) Kennwerte je Metrik sind hervorgehoben.

Datensatz	DA	DR	Max. Performanz Pre [%]					Max. Performanz Post [%]				
			AP	AP50	AP75	SR1	ER1	AP	AP50	AP75	SR1	ER1
Real Voll		✓	85.4	99.8	95.4	100.0	0.0	95.8	99.9	99.7	99.9	0.0
			86.6	99.7	95.2	100.0	0.0	95.2	99.9	99.8	100.0	0.0
Real Midi		✓	81.7	99.2	94.0	100.0	0.0	91.5	99.5	99.1	99.7	0.0
			82.3	98.9	92.8	100.0	0.0	90.5	99.5	98.8	99.9	0.0
Real Mini		✓	71.2	96.5	85.6	100.0	0.0	82.7	98.8	96.3	99.5	0.1
			71.0	96.3	85.6	99.7	0.0	78.8	97.6	93.9	99.4	0.2
SD-Opt	✓	✓	62.9	83.2	72.4	96.3	0.5	67.2	83.4	78.6	97.4	0.3
			71.7	90.2	83.0	100.0	0.0	73.2	90.6	86.3	98.7	0.2
			75.3	91.8	84.9	99.2	0.1	76.6	92.8	88.6	98.6	0.3
SE-Preopt	✓	✓	81.5	96.2	93.8	99.4	0.0	84.2	97.8	96.4	99.0	0.9
			81.0	94.0	91.8	99.9	0.1	84.2	97.0	95.5	98.9	1.1
			82.9	95.6	93.2	100.0	0.0	85.0	97.0	95.6	99.4	0.6
SE-Opt	✓	✓	81.9	99.6	95.8	99.9	0.0	84.7	99.5	98.2	99.3	0.5
			89.2	100.0	98.7	100.0	0.0	89.5	99.9	99.5	99.9	0.1
			89.6	99.9	98.3	100.0	0.0	90.5	99.8	99.4	100.0	0.0
SE-Exp	✓	✓	85.8	99.7	97.0	99.7	0.0	87.7	99.4	98.8	99.6	0.4
			89.2	100.0	98.3	100.0	0.0	90.0	99.8	99.4	99.6	0.4
			90.2	99.8	98.1	100.0	0.0	90.7	99.7	99.3	99.9	0.0

Zusammenfassung

Aufgrund der Studienergebnisse kann die **Hypothese** für den untersuchten Anwendungsfall **bestätigt** werden. Für den Pre-Evaluierungsdatensatz wurde mit dem SE-Exp-Datensatz ohne Modifikation eine insignifikant höhere Performanz als mit realen Trainingsdaten erreicht. Bei der Anwendung von Domain Randomization und Domain Adaptation konnten sogar signifikante Verbesserungen gegenüber dem Training mit Realdaten erzielt werden. Bei hoher Übereinstimmung zwischen den realen Trainings- und Evaluierungsdaten (Post) lag die Performanz der synthetischen Daten jedoch trotz Modifikation signifikant unter der Performanz der Realdaten.

Metriken

Für die Übertragung auf weitere Metriken wurden die Durchschnittspräzisionen mit festen IoU-Grenzwerten (AP50 und AP75) sowie die Erfolgs- (SR1) und die Fehlerrate (ER1) für die jeweils erste Detektion der Limited Selection genutzt (s. Unterabschnitt 4.4.3). Die Ergebnisse sind in Tabelle 5.4 aufgeführt. Im Folgenden wird die Formulierung „bessere Ergebnisse“ verwendet, um die Einordnung der Kennwerte der Erfolgsrate (höher) und der Fehlerrate (niedriger) zusammenzufassen.

AP50 und AP75: Aufgrund der niedrigeren IoU-Grenzwerte erreichten alle Objekterkennungssysteme für die AP50 und die AP75 höhere Kennwerte als für die AP. Bei Evaluierung mit dem Pre-Datensatz erreichten sowohl der SE-Opt- als auch der SE-Exp-Datensatz mit Domain Randomization (DR) signifikant höhere Performanzkennwerte als die realen Datensätze. Für den Post-Evaluierungsdatensatz lagen die Performanzkennwerte des SE-Exp-Datensatzes zwar geringfügig, jedoch signifikant unter denen der vollen Realdaten. Im Gegensatz zur AP-Metrik schnitt der SE-Opt-Datensatz hier deutlich besser ab und erreichte mit Modifikation (DR und DADR) jeweils gleich hohe oder höhere Performanzkennwerte als der SE-Exp-Datensatz mit Modifikation. Es konnten keine signifikanten Unterschiede zwischen den vollen Real- und SE-Opt-Daten (mit DR) gemessen werden. Die Performanz des SynthDet-Datensatzes (SD-Opt) lag unabhängig von Evaluierungsdatensatz und Modifikationen signifikant unter der Performanz des kleinsten Realdatensatzes (Mini).

5.5. Kombination Domain Adaptation und Domain Randomization

Für die meisten synthetischen Datensätze führte mindestens eine der Modifikationen zu einer Verbesserung gegenüber dem jeweiligen Basisdatensatz ohne Modifikation. Ausnahmen waren der SE-Preopt- und der SE-Exp-Datensatz (Post). Bei den Datensätzen SE-Opt und SE-Exp wurde die jeweils höchste Performanz in der Variante DR ohne zusätzliche Domain Adaptation erzielt.

SR1 und ER1: Bei der Evaluierung mit dem Pre-Datensatz erreichten alle Datensätze mindestens für eine Variante eine Fehlerrate ER1 von 0 % (SD-Opt nur für DR, SE-Preopt nicht für DR). Für den Post-Evaluierungsdatsatz war dies dagegen neben den Realdatsätzen nur mit den durch DADR modifizierten Datensätzen SE-Opt und SE-Exp der Fall.

Ein ähnliches Bild ergibt sich entsprechend für die Erfolgsrate SR1. Während beim Pre-Evaluierungsdatsatz alle modifizierten synthetischen Datensätze mindestens in einer Variante eine Erfolgsrate von 100 % erreichten, war dies beim Post-Datensatz nur für die DADR-Variante des SE-Opt-Datensatzes der Fall.

Mindestens eine der Modifikationen führte für alle synthetischen Datensätze zu selben oder besseren Ergebnissen gegenüber dem jeweiligen Basisdatensatz ohne Modifikation. Dabei erreichte die Variante DADR für die Datensätze SE-Opt und SE-Exp dasselbe oder ein besseres Ergebnis als die Variante DR. Für die realen, SD-Opt und SE-Preopt-Datensätze war dagegen keine klare Unter- oder Überlegenheit zwischen den Modifikationsvarianten erkennbar.

Zusammenfassung

Aufgrund der Studienergebnisse kann die **Hypothese** für den untersuchten Anwendungsfall **bestätigt** werden. Mit den synthetischen Datensätzen wurden sowohl für die AP-Metriken mit fixen IoU-Grenzwerten als auch für die spezifischen Limited Selection-Metriken signifikant höhere Performanzkennwerte oder insignifikante Unterschiede gegenüber den realen Trainingsdaten gemessen. Die Kennwerte die mit dem Datensatz SE-Opt mit Domain Randomization über alle Metriken hinweg erreicht wurden, waren jeweils die signifikant besten oder nicht signifikant schlechter als die Kennwerte anderer Datensätze (inklusive Realdaten).

CNN-Architekturen

Im Rahmen der Übertragung auf weitere CNN-Architekturen wurden Mask R-CNN mit Inception-ResNet-v2, sowohl mit Metrikberechnung auf Basis der Begrenzungsrahmen (Rahmen) als auch auf Basis der Masken, sowie SSD mit MobileNetV2 (nur Rahmen) untersucht. Die CNNs wurden jeweils mit dem vollen Real-, dem SE-Opt- und dem SE-Exp-Datensatz trainiert. Die Ergebnisse sind in Abbildung 5.25 dargestellt.

SSD: Alle SSD-Modelle erreichten eine signifikant niedrigere Performanz als die Mask R-CNN-Modelle (Rahmen), jedoch besteht mit einem Korrelationskoeffizienten von 0.7 ein klarer Zusammenhang zwischen den Performanzkennwerten der beiden Modelle. Für beide Evaluierungsdatsätze konnten bei den synthetischen Datsätzen signifikante Steigerungen der Performanz vom jeweiligen Basisdatensatz zur DR-Variante sowie von der DR- zur DADR-Variante gemessen werden. Während der SE-Exp-Basisdatensatz eine signifikant niedrigere Performanz als der SE-Opt-Basisdatensatz erzielte, erreichte SE-Exp mit den modifizierten Varianten jeweils eine höhere Performanz als dieselben Varianten bei SE-Opt.

Bei der Evaluierung mit dem Pre-Datensatz wurde die insgesamt höchste Performanz von AP 74.9 % mit dem SE-Exp-Datensatz in der DADR-Variante erreicht (Δ AP 3.1 % zum realen Basisdatensatz). Beim Post-Evaluierungsdatsatz erreichte der Realdatsatz ohne Modifikation die signifikant besten Ergebnisse (AP 81.2 %) und lag damit mit Δ AP 3.4 % signifikant über dem besten synthetischen Datsatz (SE-Exp mit DADR).

Mask R-CNN auf Maskenbasis: Das Mask R-CNN (maskenbasiert) erzielte bei der Auswertung ein mit dem SSD-Modell vergleichbares Performanzniveau. Es besteht ein klarer Zusammenhang zwischen den Performanzkennwerten der Varianten Begrenzungsrahmen und Masken (Korrelationskoeffizient 0.93). Mit einer Ausnahme (SE-Opt in DADR-Variante) konnten für beide Evaluierungsdatsätze signifikante Steigerungen der Performanz mit zunehmender Modifikation gemessen werden.

Die besten Ergebnisse beim Pre-Evaluierungsdatsatz wurden mit dem SE-Exp-Datensatz mit DADR-Modifikation erreicht (AP 73.7 %, Δ AP 1.2 % zum DR-Realdatsatz). Beim Post-Evaluierungsdatsatz erreichte der Realdatsatz mit DR-Modifikation die signifikant besten Ergebnisse (AP 81.4 %) und lag damit signifikant über dem besten synthetischen Datsatz (Δ AP 3.7 % zu SE-Exp in DADR-Variante).

5.5. Kombination Domain Adaptation und Domain Randomization

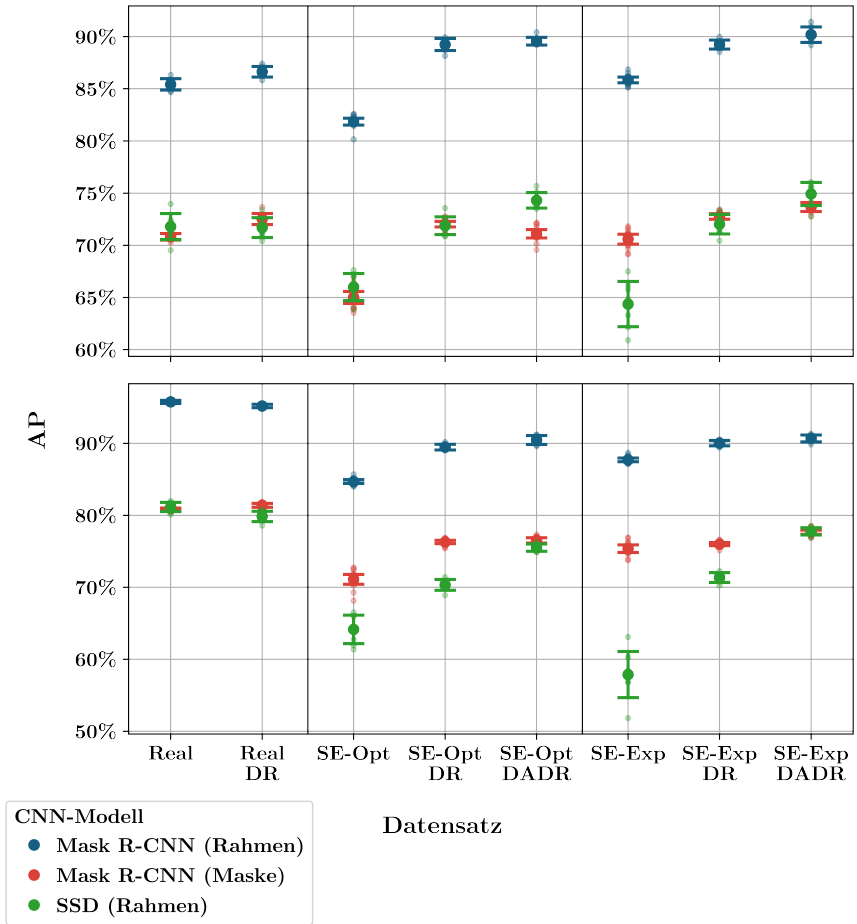


Abbildung 5.25.: Ergebnisse der Übertragung auf die CNN-Architekturen Mask R-CNN (maskenbasiert) und SSD für reale und synthetische Datensätze des Anwendungsfalls IP-A ohne und mit Modifikation (DR und DADR). Die Evaluierung wurde mit dem Pre- (oben) und dem Post-Datensatz (unten) durchgeführt.

Zusammenfassung

Aufgrund der Studienergebnisse kann die **Hypothese** für den untersuchten Anwendungsfall **bestätigt** werden. Sowohl für SSD MobileNetV2 als auch für Mask R-CNN auf Maskenbasis wurden signifikant höhere Performanzkennwerte mit synthetischen als mit realen Daten erreicht (Pre-Evaluierungsdatensatz). Für den Post-Evaluierungsdatensatz dagegen erzielte der Realdatensatz die signifikant besten Ergebnisse. Die insgesamt besten Ergebnisse mit synthetischen Trainingsdaten wurden mit dem SE-Exp-Datensatz und der Kombination aus Domain Adaptation und Domain Randomization erreicht.

Anwendungsfälle

Für die Übertragung auf weitere Anwendungsfälle wurden neben dem SE-Exp-Datensatz (IP-A) auch die Scene Engineering-Datensätze (SE) der zweiten Item Picking-Aufgabe (IP-B) und der Qualitätsprüfungsaufgabe (QP-A) genutzt. Für den abschließenden Vergleich zu realen Daten wurde jeweils zusätzlich mit dem zugehörigen vollen Realdatensatz trainiert. Evaluiert wurde die Performanz der trainierten CNNs jeweils mit einem Datensatz, der nahe an der realen Trainingsdomäne liegt (Post für IP-A und Nah für IP-B), sowie, falls verfügbar, mit einem fernerem Datensatz (Pre für IP-A und Fern für IP-B). Alle CNN-Trainings wurden mit bis zu 40 Epochen (200 000 Trainingsschritte bei IP-A und QP-A) bzw. 15 Epochen (210 000 Trainingsschritte bei IP-B) trainiert. Die Ergebnisse sind in Tabelle 5.5 aufgeführt.

IP-B: Für den fernen Evaluierungsdatensatz des IP-B-Anwendungsfalls wurden durch Modifikation der synthetischen Datensätze jeweils schlechtere Ergebnisse erreicht als mit den unveränderten Basisdatensätzen (Ausnahme ER1). Dagegen verbesserte sich die Performanz des realen Trainingsdatensatzes durch die DR-Modifikation für die AP-Metriken signifikant. Der SE-Basisdatensatz erreichte zwar signifikant höhere Kennwerte (AP, AP50 und SR1) als der reale Basisdatensatz, jedoch eine signifikant niedrigere AP als die reale DR-Variante. Bei der AP50 (78.1 %) und der Erfolgsrate SR1 (97.5 %) erreichte der SE-Basisdatensatz dennoch eine Performanzsteigerung gegenüber dem modifizierten Realdatensatz (95.8 %). Im Falle der

5.5. Kombination Domain Adaptation und Domain Randomization

Erfolgsrate war die Verbesserung sogar signifikant. Die Fehlerrate ER1 lag deutlich höher als beim Anwendungsfall IP-A, wobei die Unterschiede zwischen dem SE-Datensatz mit DADR-Modifikation und den Realdatensätzen insignifikant waren.

Für den nahen Evaluierungsdatensatz wurden mit den Realdaten durchweg signifikant bessere Ergebnisse als mit den synthetischen Trainingsdaten erzielt. Im Gegensatz zum fernen Datensatz erreichten die SE-Daten durch die DR-Modifikation jeweils bessere Kennwerte als in der unveränderten Basisvariante, jedoch führte die Anwendung von Domain Adaptation (DADR) zu keinen Verbesserungen gegenüber reiner Domain Randomization.

QP-A: Für den QP-A-Anwendungsfall steigerte sich die Performanz (oder selbe Performanz) jeweils von den Basisdatensätzen zur DR-Variante (inkl. Realdaten) sowie von der DR- zur DADR-Variante. Während die insgesamt höchste Performanz für die AP-Metrik mit dem Realdatensatz (DR-Variante) erreicht wurde (AP 98.8%), konnten keine signifikanten Unterschiede zu den modifizierten SE-Datensätzen gemessen werden (DADR-Variante mit $\Delta AP -1.8\%$ im direkten Vergleich).

Bei der AP50-Metrik ergaben sich keinerlei signifikante Unterschiede zwischen den Datensätzen, jedoch wurde die insgesamt höchste AP50 mit 100 % für die modifizierten synthetischen Trainingsdaten erreicht.

Zusammenfassung

Aufgrund der Studienergebnisse kann die **Hypothese** für die untersuchten Anwendungsfälle **bestätigt** werden. Abhängig von den gewählten Metriken und Evaluierungsdatensätzen wurden beim Training mit synthetischen Scene Engineering-Daten sowohl für den IP-B (AP50, SR1 und ER1 bei Evaluierung mit dem fernen Datensatz) als auch für den QP-A-Anwendungsfall (AP und AP50) signifikant bessere oder nicht signifikant schlechtere Performanzkennwerte als mit Realdaten erreicht.

Tabelle 5.5.: Maximal erreichte Performanzkennwerte für die drei Anwendungsfälle IP-A, IP-B und QP-A. Es werden die Metriken AP, AP50, SR1 und ER1 für bis zu zwei Evaluierungsdatensätze, welche der jeweiligen realen Trainingsdomäne ferner (Pre und Fern) oder näher (Post und Nah) sind, angegeben. Die Trainingsdatensätze umfassen jeweils auch Varianten mit DR (SV- und Kontrastvariation) und DADR (GAN- und ICT-Anpassung mit Mini-Referenzdatensatz und DR). Die signifikant höchsten Kennwerte je Anwendungsfall und Metrik sind hervorgehoben.

Anw.	Datensatz	Max. Performanz fern [%]				Max. Performanz nah [%]			
		AP	AP50	SR1	ER1	AP	AP50	SR1	ER1
IP-A	Real	85.4	99.8	100.0	0.0	95.8	99.9	99.9	0.0
	Real DR	86.6	99.7	100.0	0.0	95.2	99.9	100.0	0.0
	SE	85.8	99.7	99.7	0.0	87.7	99.4	99.6	0.4
	SE DR	89.2	100.0	100.0	0.0	90.0	99.8	99.6	0.4
	SE DADR	90.2	99.8	100.0	0.0	90.7	99.7	99.9	0.0
IP-B	Real	57.1	71.9	95.8	2.1	90.5	98.4	99.3	0.6
	Real DR	62.6	77.2	95.8	2.5	89.3	98.2	99.1	0.6
	SE	60.1	78.1	97.5	2.3	76.3	93.6	97.1	1.4
	SE DR	58.6	75.5	95.6	3.0	79.3	94.4	97.7	1.1
	SE DADR	56.6	72.7	95.1	2.3	79.4	94.3	97.5	1.1
QP-A	Real	-	-	-	-	98.3	99.4	-	-
	Real DR	-	-	-	-	98.8	99.9	-	-
	SE	-	-	-	-	92.5	99.4	-	-
	SE DR	-	-	-	-	96.5	100.0	-	-
	SE DADR	-	-	-	-	97.0	100.0	-	-

6. Zusammenfassung

Im Folgenden wird die Arbeit mit Blick auf die forschungsleitende Fragestellung und ihre Teilaspekte zusammengefasst. Abschließend wird ein Ausblick auf potenzielle zukünftige Forschungsfragen im Bereich des Transfer Learnings mit synthetischen Bilddaten im industriellen Kontext gegeben.

6.1. Zusammenfassung der Arbeit

Im Rahmen der vorliegenden Arbeit wurden verschiedene Methoden zur Verbesserung von synthetischen Bilddaten, das Scene Engineering, die Domain Randomization und die Domain Adaptation sowie deren Kombination untersucht. Die mit diesen Daten trainierten Objekterkennungssysteme auf Basis tiefer neuronaler Netze (CNNs) wurden über verschiedene Metriken, allgemein verbreitete Objekterkennungsmetriken und spezifische Metriken für den Anwendungsfall des Item Pickings miteinander sowie mit CNNs verglichen, welche mit realen Bilddaten trainiert wurden. Dabei wurde untersucht, ob die Performanz der mit synthetischen Bilddaten trainierten Objekterkennungssysteme durch den Einsatz der drei Verbesserungsmethoden sowie deren Kombination signifikant erhöht werden kann.

Teilaspekt Scene Engineering: Die Methode des Scene Engineerings wurde mit dem kontextarmen Referenzverfahren SynthDet verglichen. Dabei führten die mittels Scene Engineering erzeugten Bilddaten – unabhängig von zusätzlichen Anpassungen und Modifikationen – zu signifikant höheren Performanzkennwerten der trainierten Objekterkennungssysteme als die Daten des Referenzverfahrens SynthDet.

Darüber hinaus wurden die folgenden Aspekte des Scene Engineerings getrennt untersucht: die manuelle Anpassung der Szenenstruktur, des Sichtfelds, der Beleuchtung sowie der Objektanzahl und -positionierung an wenige reale Referenzdaten. Von besonderer Relevanz waren dabei die korrekte Darstellung des Ladungsträgers (Szenenstruktur) sowie die exakte Nachbildung der realen Kamerapose in Relation zum Ladungsträger (Sichtfeld).

6. Zusammenfassung

Zudem konnte gezeigt werden, dass eine realistische Objektpositionierung mittels Physiksimulation sowie realitätsnah verteilte oder gleichverteilte Objektanzahlen pro Bild signifikante Vorteile gegenüber anderen Ansätzen aufweisen. Die untersuchten Konfigurationen der manuell optimierten Beleuchtung waren dagegen einer starken und unrealistischen Randomisierung unterlegen und erzielten signifikant schlechtere Ergebnisse.

Abschließend wurde belegt, dass eine Automatisierung des aufwendigen manuellen Anpassungsprozesses der Simulationsszene durch Scene Adaptation, also die automatisierte Simulationsoptimierung, möglich ist und wiederholbar zu gleichen oder sogar signifikant besseren Ergebnissen führen kann als die manuelle Optimierung.

Teilaspekt Domain Randomization: Die Domain Randomization wurde durch die nachträgliche Augmentierung der Bilddaten mit Unschärfe (Box- und Gauß-Filter), Gauß'schem Rauschen, chromatischer Aberration sowie RGB-, Helligkeits-, HSV- und Kontrastvariation untersucht.

Mit allen untersuchten Augmentierungen konnten signifikante Verbesserungen gegenüber dem Basisdatensatz erzielt werden. Die höchste Performanz wurde dabei durch die kombinierte Variation von Farbwert, Saturation, Hellwert (HSV) und Kontrast erreicht. Dabei war die durch diese Kombination erzielte Steigerung gegenüber der reinen Augmentierung des Kontrasts lediglich für einen der beiden Evaluierungsdatensätze signifikant. Alle weiteren untersuchten Augmentierungen führten zu einer signifikant niedrigeren Performanz als die reine Kontrastvariation.

Teilaspekt Domain Adaptation: Die Domain Adaptation wurde durch die nachträgliche Anpassung der Bilddaten mittels ICT- und GAN-Transformation jeweils mit verschiedenen realen Referenzdaten untersucht.

Durch beide Adaptionsverfahren konnten signifikante Verbesserungen gegenüber dem Basisdatensatz erzielt werden. Die besten Ergebnisse wurden dabei mit der kombinierten Domain Adaptation aus GAN-Transformation und anschließender ICT-Anpassung erreicht, jedoch waren die Unterschiede zur reinen GAN-Transformation für einen der Evaluierungsdatensätze nicht signifikant. Für die untersuchten Anwendungsfälle konnte gezeigt werden, dass eine größere Übereinstimmung zwischen Referenz- und Evaluierungsdaten bei der Domain Adaptation mit GANs zu signifikant besseren Ergebnissen führt. Entgegen der Erwartung konnte dieser Zusammenhang für die ICT-Anpassung nicht nachgewiesen werden.

Teilaspekt Kombination: Die Kombination von Domain Adaptation und anschließender Domain Randomization für die mittels Scene Engineering erzeugten Bilddaten führte beim primären Anwendungsfall zu keiner signifikanten Verbesserung gegenüber der reinen Domain Randomization.

Bei der Übertragung auf weitere Datensätze (AP bei SD-Opt, SE-Preopt und SE-Exp), CNN-Architekturen (SSD) und Anwendungsfälle (QP-A) konnten dagegen teilweise signifikant bessere Ergebnisse durch die Kombination als durch die reine Domain Randomization erzielt werden. Eine generelle Überlegenheit der Kombination über die durchgeführten Experimente hinweg ließ sich dabei jedoch nicht nachweisen.

Forschungsleitende Fragestellung: Basierend auf diesen Ergebnissen wurde abschließend die übergeordnete, forschungsleitende Fragestellung zum Vergleich mit realen Trainingsdaten beantwortet und die damit verbundene Hypothese bestätigt. Abhängig vom Evaluierungsdatensatz und den genutzten Metriken konnten mit synthetischen Trainingsdaten auf Basis von Scene Engineering durch die Modifikation mit Domain Randomization und Domain Adaptation für alle Anwendungsfälle Performanzkennwerte erreicht werden, welche nicht signifikant niedriger und teilweise sogar signifikant höher sind als beim Training mit realen Bilddaten. Die Ergebnisse waren dabei auch auf weitere Datensätze, Metriken, CNN-Architekturen und Anwendungsfälle übertragbar.

6.2. Ausblick

In der vorliegenden Arbeit wurden die verschiedenen Methoden der Datengenerierung und -anpassung mit Blick auf eine möglichst hohe Performanz der trainierten Objekterkennungssysteme für definierte Evaluierungsdatensätze untersucht. Zwar wurden dabei jeweils mehrere Trainingsdurchläufe gemittelt, Konfidenzintervalle berechnet und Signifikanztests durchgeführt, jedoch wurde die Robustheit der Objekterkennungssysteme nicht explizit betrachtet. Eine solche gezielte Untersuchung der Robustheit synthetisch trainierter CNNs im Bereich industrieller Anwendungsfälle könnte helfen, den Entscheidungsprozess der CNNs im Falle von Anomalien und Systemfehlern besser einschätzen zu können und potenzielle Schwachstellen zu identifizieren. Dies könnte durch eine gezielte Anpassung der Evaluierungsdatensätze anstatt nur der Trainingsdatensätze geschehen, beispielsweise durch randomisierte Modifikationen und das explizite Einfügen von Anomalien. Darauf basierend

6. Zusammenfassung

könnten Optimierungsschritte für die Datengenerierung abgeleitet werden, welche es ermöglichen, die CNNs direkt auf Robustheit zu trainieren.

Die vorgestellten Experimente wurden anhand von drei industriellen Anwendungsfällen aus Logistik und Produktion durchgeführt. Trotz dieser Übertragung der identifizierten Methoden der Domain Adaptation und Domain Randomization auf weitere Anwendungsfälle sind keine allgemeingültigen Aussagen für Aufgaben der industriellen Bildverarbeitung möglich. Durch eine Erweiterung auf weitere Anwendungsfälle und Industriebereiche wäre jedoch eine Ableitung weithin gültiger oder sogar allgemeingültiger Aussagen möglich. Darüber hinaus würde ein breites Spektrum an synthetischen und realen Bilddaten ein GAN-Training für den generischen Sim2Real-Transfer ermöglichen. Dabei könnten die GANs erlernen, die typischen Unterschiede zwischen synthetischen und realen Daten zu reduzieren, und müssten nicht für jeden Anwendungsfall mit spezifischen Realdaten neu trainiert werden.

Im Rahmen dieser Arbeit konnte die Automatisierbarkeit des aufwendigen manuellen Anpassungsprozesses der Simulationsszene durch Scene Adaptation exemplarisch belegt werden. Dennoch verbleibt hierbei noch manueller Aufwand für die initiale Einrichtung der Simulation, die passende Parametrisierung und die Kontrolle des laufenden Optimierungsprozesses. Dieser bisher noch für die Scene Adaptation notwendige manuelle Aufwand könnte dabei beispielsweise durch die Optimierung der Prozessabfolgen und -wiederholungen, die Weiterentwicklung der Verlustfunktionen sowie die Verbesserung der Standardparameterauswahl weiter reduziert werden. Im Kombination mit einer breiten Auswahl an industriellen Umgebungsmodellen und -texturen wäre eine vollautomatisierte Simulationsoptimierung ab dem ersten 3D-Modell potenziell möglich.

Literatur

- [1] Ibrahim Saad Aly Abdelhalim, Mamdouh Farouk Mohamed und Yousef Bassyouni Mahdy. „Data augmentation for skin lesion using self-attention based progressive generative adversarial network“. In: *Expert Systems with Applications* 165 (2021), S. 113922. issn: 09574174. DOI: 10.1016/j.eswa.2020.113922.
- [2] Hassan Abu Alhaija et al. „Augmented Reality Meets Computer Vision: Efficient Data Generation for Urban Driving Scenes“. In: *International Journal of Computer Vision* 126.9 (2018), S. 961–972. issn: 0920-5691. DOI: 10.1007/s11263-018-1070-x.
- [3] Hassan Abu Alhaija et al. „Augmented Reality meets Deep Learning“. In: *Proceedings of the British Machine Vision Conference 2017*. Hrsg. von Tae-Kyun Kim et al. British Machine Vision Association, 2017. ISBN: 1-901725-60-X. DOI: 10.5244/C.31.81.
- [4] Martin Arjovsky, Soumith Chintala und Léon Bottou. „Wasserstein generative adversarial networks“. In: *Proceedings of the 34th International Conference on Machine Learning - Volume 70*. ICML'17. Sydney, NSW, Australia: JMLR.org, 2017, S. 214–223. URL: <https://proceedings.mlr.press/v70/arjovsky17a/arjovsky17a.pdf>.
- [5] David Berthelot, Thomas Schumm und Luke Metz. „BEGAN: Boundary Equilibrium Generative Adversarial Networks“. In: *ArXiv abs/1703.10717* (31. März 2017). URL: <https://arxiv.org/pdf/1703.10717>.
- [6] Steve Borkman et al. „Unity Perception: Generate Synthetic Data for Computer Vision“. In: *ArXiv abs/2107.04259* (9. Juli 2021). URL: <https://arxiv.org/pdf/2107.04259>.
- [7] Konstantinos Bousmalis et al. „Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks“. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Hrsg. von IEEE Staff. Piscataway: IEEE, Juli 2017, S. 95–104. ISBN: 978-1-5386-0457-1. DOI: 10.1109/CVPR.2017.18.

- [8] Konstantinos Bousmalis et al. „Using Simulation and Domain Adaptation to Improve Efficiency of Deep Robotic Grasping“. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 52018, S. 4243–4250. ISBN: 978-1-5386-3081-5. DOI: 10.1109/ICRA.2018.8460875.
- [9] Stephan Brehm, Sebastian Scherer und Rainer Lienhart. „Semantically Consistent Image-to-Image Translation for Unsupervised Domain Adaptation“. In: *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022)*. Hrsg. von Ana Paula Rocha, Luc Steels und Jaap den van Herik. Setúbal: SciTePress - Science and Technology Publications Lda, 2022, S. 131–141. ISBN: 978-989-758-547-0. DOI: 10.5220/0010786000003116.
- [10] Eric Brochu, Vlad M. Cora und Nando de Freitas. *A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning*. 12. Dez. 2010. URL: <http://arxiv.org/pdf/1012.2599>.
- [11] Yevgen Chebotar et al. „Closing the Sim-to-Real Loop: Adapting Simulation Randomization with Real World Experience“. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 52019, S. 8973–8979. ISBN: 978-1-5386-6027-0. DOI: 10.1109/ICRA.2019.8793789.
- [12] Anoop Cherian und Alan Sullivan. „Sem-GAN: Semantically-Consistent Image-to-Image Translation“. In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2019, S. 1797–1806. DOI: 10.1109/WACV.2019.00196.
- [13] Anoop Cherian et al. „InSeGAN: A Generative Approach to Segmenting Identical Instances in Depth Images“. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, S. 10003–10012. DOI: 10.1109/ICCV48922.2021.00987.
- [14] François Chollet. *Deep Learning with Python, Second Edition*. Shelter Island, New York: Manning Publications, 2021. ISBN: 9781638350095. URL: <https://www.manning.com/books/deep-learning-with-python-second-edition>.
- [15] Francois Chollet. „Xception: Deep Learning with Depthwise Separable Convolutions“. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Hrsg. von IEEE Staff. Pis-

- cataway: IEEE, 2017, S. 1800–1807. ISBN: 978-1-5386-0457-1. DOI: 10.1109/CVPR.2017.195.
- [16] Michael Danielczuk et al. „Segmenting Unknown 3D Objects from Real Depth Images using Mask R-CNN Trained on Synthetic Data“. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, S. 7283–7290. ISBN: 978-1-5386-6027-0. DOI: 10.1109/ICRA.2019.8793744.
 - [17] Atabak Dehban et al. „The Impact of Domain Randomization on Object Detection: A Case Study on Parametric Shapes and Synthetic Textures“. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2019, S. 2593–2600. DOI: 10.1109/IROS40897.2019.8968139.
 - [18] Frederik Michel Dekking. *A Modern Introduction to Probability and Statistics: Understanding Why and How*. Springer Texts in Statistics. London: Springer-Verlag London Limited, 2005. ISBN: 978-1-85233-896-1. DOI: 10.1007/1-84628-168-7.
 - [19] Marie Delacre, Daniël Lakens und Christophe Leys. „Why psychologists should by default use Welch’s t-test instead of student’s t-test“. English. In: *International Review of Social Psychology* 30.1 (Apr. 2017), S. 92–101. ISSN: 2397-8570. DOI: 10.5334/irsp.82.
 - [20] Jia Deng et al. „ImageNet: A large-scale hierarchical image database“. In: *Computer Vision and Pattern Recognition, 2009, CVPR 2009, IEEE Conference on : dates: 20-25 June 2009*. Piscataway, N.J.: IEEE, 2009, S. 248–255. ISBN: 978-1-4244-3992-8. DOI: 10.1109/CVPR.2009.5206848.
 - [21] Kexin Ding et al. „A Large-scale Synthetic Pathological Dataset for Deep Learning-enabled Segmentation of Breast Cancer“. In: *Scientific data* 10.1 (2023), S. 231. DOI: 10.1038/s41597-023-02125-y.
 - [22] Laura Dörr et al. „Lean Training Data Generation for Planar Object Detection Models in Unsteady Logistics Contexts“. In: *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. 2019, S. 329–334. DOI: 10.1109/ICMLA.2019.00062.
 - [23] Alexey Dosovitskiy et al. „An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale“. In: *9th International Conference on Learning Representations, ICLR 2021, Virtual Event,*

- Austria, May 3-7, 2021*. OpenReview.net, 2021. URL: <https://openreview.net/forum?id=YicbFdNTTy>.
- [24] Debidatta Dwibedi, Ishan Misra und Martial Hebert. „Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection“. In: *ICCV 2017*. Proceedings (IEEE International Conference on Computer Vision. Online). Los Alamitos, Washington und Tokyo: CPS und IEEE Computer Society, 2017, S. 1310–1319. ISBN: 978-1-5386-1032-9. DOI: 10.1109/ICCV.2017.146.
 - [25] Mark Everingham et al. „The Pascal Visual Object Classes (VOC) Challenge“. In: *International Journal of Computer Vision* 88.2 (2010), S. 303–338. ISSN: 0920-5691. DOI: 10.1007/s11263-009-0275-4.
 - [26] Mark Everingham et al. „The Pascal Visual Object Classes Challenge: A Retrospective“. In: *International Journal of Computer Vision* 111.1 (2015), S. 98–136. ISSN: 0920-5691. DOI: 10.1007/s11263-014-0733-5.
 - [27] Alex Fang, Simon Kornblith und Ludwig Schmidt. „Does progress on ImageNet transfer to real-world datasets?“ In: *Proceedings of the 37th International Conference on Neural Information Processing Systems*. NIPS ’23. New Orleans, LA, USA: Curran Associates Inc., 2024. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/4eb33c53ed5b14ce9028309431f565cc-Paper-Datasets_and_Benchmarks.pdf.
 - [28] Kuan Fang et al. „Multi-Task Domain Adaptation for Deep Learning of Instance Grasping from Simulation“. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, S. 3516–3523. ISBN: 978-1-5386-3081-5. DOI: 10.1109/ICRA.2018.8461041.
 - [29] Adrien Gaidon et al. „VirtualWorlds as Proxy for Multi-object Tracking Analysis“. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*. Piscataway, NJ: IEEE, 2016, S. 4340–4349. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.470.
 - [30] Roman Garnett. *Bayesian optimization*. 1st ed. Cambridge, United Kingdom: Cambridge University Press, 2023. ISBN: 9781108348973. DOI: 10.1017/9781108348973.

- [31] Andreas Geiger et al. „Vision meets robotics: The KITTI dataset“. In: *The International Journal of Robotics Research* 32.11 (2013), S. 1231–1237. ISSN: 0278-3649. DOI: 10.1177/0278364913491297.
- [32] Georgios Georgakis et al. „Synthesizing Training Data for Object Detection in Indoor Scenes“. In: *Proceedings of Robotics: Science and Systems XIII*. Cambridge, Massachusetts, Juli 2017. DOI: 10.15607/RSS.2017.XIII.043.
- [33] Golnaz Ghiasi et al. „Simple Copy-Paste is a Strong Data Augmentation Method for Instance Segmentation“. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Piscataway, NJ: IEEE, 2021, S. 2917–2927. ISBN: 978-1-6654-4509-2. DOI: 10.1109/CVPR46437.2021.00294.
- [34] Ross Girshick. „Fast R-CNN“. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, S. 1440–1448. ISBN: 978-1-4673-8391-2. DOI: 10.1109/ICCV.2015.169.
- [35] Ross Girshick et al. „Rich feature hierarchies for accurate object detection and semantic segmentation“. In: *ArXiv abs/1311.2524* (11. Nov. 2013). URL: <https://arxiv.org/pdf/1311.2524>.
- [36] Ian Goodfellow, Yoshua Bengio und Aaron Courville. *Deep Learning*. <http://www.deeplearningbook.org>. Cambridge, Massachusetts: The MIT Press, 2016.
- [37] Ian J. Goodfellow et al. „Generative Adversarial Nets“. In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. NIPS’14. Cambridge, MA, USA: MIT Press, 2014, S. 2672–2680.
- [38] Silas Grün et al. „Evaluation of Domain Randomization Techniques for Transfer Learning“. In: *2019 19th International Conference on Advanced Robotics (ICAR)*. IEEE, 2019, S. 481–486. ISBN: 978-1-7281-2467-4. DOI: 10.1109/ICAR46387.2019.8981654.
- [39] Ishaan Gulrajani et al. „Improved Training of Wasserstein GANs“. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS’17. Red Hook, NY, USA: Curran Associates Inc, 2017, S. 5769–5779. ISBN: 9781510860964.

- [40] Agrim Gupta, Piotr Dollár und Ross Girshick. „LVIS: A Dataset for Large Vocabulary Instance Segmentation“. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, S. 5351–5359. DOI: 10.1109/CVPR.2019.00550.
- [41] Kaiming He et al. „Deep Residual Learning for Image Recognition“. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*. Piscataway, NJ: IEEE, 2016, S. 770–778. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.90.
- [42] Kaiming He et al. „Mask R-CNN“. In: *ICCV 2017*. Proceedings (IEEE International Conference on Computer Vision. Online). Los Alamitos, Washington und Tokyo: CPS und IEEE Computer Society, 2017, S. 2980–2988. ISBN: 978-1-5386-1032-9. DOI: 10.1109/ICCV.2017.322.
- [43] Stefan Hinterstoisser et al. „An Annotation Saved is an Annotation Earned: Using Fully Synthetic Training for Object Detection“. In: *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*. IEEE, 102019, S. 2787–2796. ISBN: 978-1-7281-5023-9. DOI: 10.1109/ICCVW.2019.00340.
- [44] Stefan Hinterstoisser et al. „On Pre-trained Image Features and Synthetic Images for Deep Learning“. In: *Computer Vision – ECCV 2018 Workshops*. Hrsg. von Laura Leal-Taixé und Stefan Roth. Bd. 11129. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, S. 682–697. ISBN: 978-3-030-11008-6. DOI: 10.1007/978-3-030-11009-3_42.
- [45] Daniel Ho et al. „RetinaGAN: An Object-aware Approach to Sim-to-Real Transfer“. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021, S. 10920–10926. DOI: 10.1109/ICRA48506.2021.9561157.
- [46] Tomas Hodan et al. „Photorealistic Image Synthesis for Object Instance Detection“. In: *2019 IEEE International Conference on Image Processing (ICIP)*. Piscataway, NJ: IEEE, 2019, S. 66–70. ISBN: 978-1-5386-6249-6. DOI: 10.1109/ICIP.2019.8803821.
- [47] Judy Hoffman et al. „CyCADA: Cycle-Consistent Adversarial Domain Adaptation“. In: *Proceedings of the 35th International Conference on Machine Learning*. Hrsg. von Jennifer Dy und Andreas Krause. Bd. 80. Proceedings of Machine Learning Research. PMLR,

- Juli 2018, S. 1989–1998. URL: <https://proceedings.mlr.press/v80/hoffman18a.html>.
- [48] Andrew G. Howard et al. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. 17. Apr. 2017. URL: <https://arxiv.org/pdf/1704.04861>.
 - [49] Zilong Hu et al. „Deep learning for image-based cancer detection and diagnosis – A survey“. In: *Pattern Recognition* 83 (2018), S. 134–149. ISSN: 00313203. DOI: 10.1016/j.patcog.2018.05.014.
 - [50] Fortune Business Insights. *Machine Vision Market*. 2025. URL: <https://www.fortunebusinessinsights.com/machine-vision-market-105188> (besucht am 18.11.2025).
 - [51] Future Market Insights. *Industrial Machine Vision Market*. 2025. URL: <https://www.futuremarketinsights.com/reports/industrial-machine-vision-market> (besucht am 18.11.2025).
 - [52] Phillip Isola et al. „Image-to-Image Translation with Conditional Adversarial Networks“. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Hrsg. von IEEE Staff. Piscataway: IEEE, 2017, S. 5967–5976. ISBN: 978-1-5386-0457-1. DOI: 10.1109/CVPR.2017.632.
 - [53] Paul Jaccard. „Étude comparative de la distribution florale dans une portion des Alpes et du Jura“. In: *Bulletin de la Société Vaudoise des Sciences Naturelles* 37.142 (1901), S. 547. ISSN: 0037-9603. DOI: 10.5169/seals-266450.
 - [54] Stephen James, Andrew J Davison und Edward Johns. „Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task“. In: *Conference on Robot Learning*. 2017.
 - [55] Stephen James et al. „Sim-To-Real via Sim-To-Sim: Data-Efficient Robotic Grasping via Randomized-To-Canonical Adaptation Networks“. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, S. 12619–12629. ISBN: 978-1-7281-3293-8. DOI: 10.1109/CVPR.2019.01291.
 - [56] Matthew Johnson-Roberson et al. „Driving in the Matrix: Can virtual worlds replace human-generated annotations for real world tasks?“. In: *ICRA2017*. Danvers (MA): IEEE Computer Society, 2017, S. 746–753. ISBN: 978-1-5090-4633-1. DOI: 10.1109/ICRA.2017.7989092.

- [57] Amlan Kar et al. „Meta-Sim: Learning to Generate Synthetic Datasets“. In: *2019 International Conference on Computer Vision*. Piscataway, NJ: IEEE, 2019, S. 4550–4559. ISBN: 978-1-7281-4803-8. DOI: 10.1109/ICCV.2019.00465.
- [58] Rawal Khirodkar, Donghyun Yoo und Kris Kitani. „Domain Randomization for Scene-Specific Car Detection and Pose Estimation“. In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2019, S. 1932–1940. ISBN: 978-1-7281-1975-5. DOI: 10.1109/WACV.2019.00210.
- [59] Rawal Khirodkar, Donghyun Yoo und Kris M. Kitani. „VADRA: Visual Adversarial Domain Randomization and Augmentation“. In: *ArXiv abs/1812.00491* (3. Dez. 2018). URL: <https://arxiv.org/pdf/1812.00491>.
- [60] Soohyun Kim et al. „InstaFormer: Instance-Aware Image-to-Image Translation with Transformer“. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Piscataway, NJ: IEEE, 2022, S. 18300–18310. ISBN: 978-1-6654-6946-3. DOI: 10.1109/CVPR52688.2022.01778.
- [61] Klaus D. Kubinger, Dieter Rasch und Karl Moder. „Zur Legende der Voraussetzungen des t-Tests für unabhängige Stichproben“. In: *Psychologische Rundschau* 60.1 (2009), S. 26–27. DOI: 10.1026/0033-3042.60.1.26.
- [62] Yu-Hui Lee et al. „Automatic Generation of Photorealistic Training Data for Detection of Industrial Components“. In: *2019 IEEE International Conference on Image Processing (ICIP)*. Piscataway, NJ: IEEE, 2019, S. 2751–2755. ISBN: 978-1-5386-6249-6. DOI: 10.1109/ICIP.2019.8803339.
- [63] Tsung-Yi Lin et al. „Microsoft COCO: Common Objects in Context“. In: *Computer vision - ECCV 2014*. Hrsg. von David Fleet. Bd. 8693. Lecture Notes in Computer Science. Cham: Springer, 2014, S. 740–755. ISBN: 978-3-319-10601-4. DOI: 10.1007/978-3-319-10602-1_48.
- [64] Y. Lin et al. „GENERATING SYNTHETIC TRAINING DATA FOR OBJECT DETECTION USING MULTI-TASK GENERATIVE ADVERSARIAL NETWORKS“. In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* V-2-2020 (2020), S. 443–449. DOI: 10.5194/isprs-annals-V-2-2020-443-2020.

- [65] Wei Liu et al. „SSD: Single Shot MultiBox Detector“. In: *Computer vision - ECCV 2016*. Hrsg. von Bastian Leibe et al. Bd. 9905. LNCS sublibrary: SL6 - Image processing, computer vision, pattern recognition, and graphics. Cham, Switzerland: Springer, 2016, S. 21–37. ISBN: 978-3-319-46447-3. DOI: 10.1007/978-3-319-46448-0_2.
- [66] Ze Liu et al. „Swin Transformer: Hierarchical Vision Transformer using Shifted Windows“. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE Computer Society, 2021, S. 9992–10002. ISBN: 978-1-6654-2812-5. DOI: 10.1109/ICCV48922.2021.00986.
- [67] Zhuang Liu et al. „A ConvNet for the 2020s“. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, S. 11966–11976. DOI: 10.1109/CVPR52688.2022.01167.
- [68] Antonio Loquercio et al. „Deep Drone Racing: From Simulation to Reality With Domain Randomization“. In: *IEEE Transactions on Robotics* 36.1 (2020), S. 1–14. ISSN: 1552-3098. DOI: 10.1109/TR0.2019.2942989. URL: <https://arxiv.org/pdf/1905.09727>.
- [69] Alejandro Magana et al. „PoseNetwork: Pipeline for the Automated Generation of Synthetic Training Data and CNN for Object Detection, Segmentation, and Orientation Estimation“. In: *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. Piscataway, NJ: IEEE, 2020, S. 587–594. ISBN: 978-1-7281-8956-7. DOI: 10.1109/ETFA46521.2020.9212064.
- [70] MarketsandMarkets. *AI Vision Market*. 2025. URL: <https://www.marketsandmarkets.com/Market-Reports/ai-vision-market-33244653.html> (besucht am 18.11.2025).
- [71] Bhairav Mehta et al. „Active Domain Randomization“. In: *ArXiv abs/1904.04762* (9. Apr. 2019). URL: <https://arxiv.org/pdf/1904.04762>.
- [72] Mehdi Mirza und Simon Osindero. „Conditional Generative Adversarial Nets“. In: *ArXiv abs/1411.1784* (6. Nov. 2014). URL: <https://arxiv.org/pdf/1411.1784>.
- [73] Sangwoo Mo, Minsu Cho und Jinwoo Shin. „Instance-aware Image-to-Image Translation“. In: *International Conference on Learning Representations*. 2019. URL: <https://openreview.net/forum?id=ryxwJhC9YX>.

- [74] Yair Movshovitz-Attias, Takeo Kanade und Yaser Sheikh. „How Useful Is Photo-Realistic Rendering for Visual Learning?“. In: *Computer vision - ECCV 2016 workshops*. Hrsg. von Gang Hua und Hervé Jégou. Bd. 9915. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, S. 202–217. ISBN: 978-3-319-49408-1. DOI: 10.1007/978-3-319-49409-8_18.
- [75] Quan Nguyen. *Bayesian Optimization in Action*. Shelter Island, New York: Manning Publications, 2023. ISBN: 9781633439078. URL: <https://www.manning.com/books/bayesian-optimization-in-action>.
- [76] Fernando Camaro Nogues, Andrew Huie und Sakyasingha Dasgupta. „Object Detection using Domain Randomization and Generative Adversarial Refinement of Synthetic Images“. In: *ArXiv abs/1805.11778* (30. Mai 2018). URL: <https://arxiv.org/pdf/1805.11778>.
- [77] Sangjun Noh et al. „Automatic Detection and Identification of Fasteners with Simple Visual Calibration using Synthetic Data“. In: *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. Piscataway, NJ: IEEE, 2020, S. 38–44. ISBN: 978-1-7281-8956-7. DOI: 10.1109/ETFA46521.2020.9212155.
- [78] Taesung Park et al. „Contrastive Learning for Unpaired Image-to-Image Translation“. In: *Computer Vision – ECCV 2020*. Hrsg. von Andrea Vedaldi et al. Bd. 12354. Springer eBook Collection. Cham: Springer International Publishing und Imprint Springer, 2020, S. 319–345. ISBN: 978-3-030-58544-0. DOI: 10.1007/978-3-030-58545-7_19.
- [79] Xingchao Peng et al. „Learning Deep Object Detectors from 3D Models“. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, S. 1278–1286. ISBN: 978-1-4673-8391-2. DOI: 10.1109/ICCV.2015.151.
- [80] Fabio Pizzati et al. „Domain Bridge for Unpaired Image-to-Image Translation and Unsupervised Domain Adaptation“. In: *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2020, S. 2979–2987. DOI: 10.1109/WACV45572.2020.9093540.
- [81] David Powers. „Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation“. In: *Journal of Machine Learning Technologies* 2 (Jan. 2011), S. 2229–3981. DOI: 10.9735/2229-3981.

- [82] Aayush Prakash et al. „Structured Domain Randomization: Bridging the Reality Gap by Context-Aware Synthetic Data“. In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, S. 7249–7255. ISBN: 978-1-5386-6027-0. DOI: 10.1109/ICRA.2019.8794443.
- [83] Kanishka Rao et al. „RL-CycleGAN: Reinforcement Learning Aware Simulation-to-Real“. In: *Proceedings, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Los Alamitos, California: IEEE Computer Society, Conference Publishing Services, 2020, S. 11154–11163. ISBN: 978-1-7281-7168-5. DOI: 10.1109/CVPR42600.2020.01117.
- [84] Nornadiah Mohd Razali und Yap Bee Wah. „Power comparisons of Shapiro-Wilk , Kolmogorov-Smirnov , Lilliefors and Anderson-Darling tests“. In: *Journal of Statistical Modeling and Analytics*. Bd. Vol.2 No.1. 2011, S. 21–33. URL: <https://api.semanticscholar.org/CorpusID:18639594>.
- [85] Joseph Redmon und Ali Farhadi. „YOLO9000: Better, Faster, Stronger“. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Hrsg. von IEEE Staff. Piscataway: IEEE, 2017, S. 6517–6525. ISBN: 978-1-5386-0457-1. DOI: 10.1109/CVPR.2017.690.
- [86] Joseph Redmon und Ali Farhadi. „YOLOv3: An Incremental Improvement“. In: *ArXiv abs/1804.02767* (9. Apr. 2018). URL: <https://arxiv.org/pdf/1804.02767>.
- [87] Joseph Redmon et al. „You Only Look Once: Unified, Real-Time Object Detection“. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*. Piscataway, NJ: IEEE, 2016, S. 779–788. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.91.
- [88] Erik Reinhard et al. „Color transfer between images“. In: *IEEE Computer Graphics and Applications* 21.4 (2001), S. 34–41. ISSN: 02721716. DOI: 10.1109/38.946629.
- [89] Shaoqing Ren et al. „Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks“. In: *IEEE transactions on pattern analysis and machine intelligence* 39.6 (2017), S. 1137–1149. DOI: 10.1109/TPAMI.2016.2577031.

- [90] Grand View Research. *AI Vision Market (2025 - 2033)*. 2025. URL: <https://www.grandviewresearch.com/industry-analysis/ai-vision-market-report> (besucht am 18.11.2025).
- [91] Grand View Research. *Industrial Automation And Control Systems Market (2025 - 2033)*. 2025. URL: <https://www.grandviewresearch.com/industry-analysis/industrial-automation-market> (besucht am 18.11.2025).
- [92] Grand View Research. *Machine Vision Market (2025 - 2030)*. 2025. URL: <https://www.grandviewresearch.com/industry-analysis/machine-vision-market> (besucht am 18.11.2025).
- [93] Olaf Ronneberger, Philipp Fischer und Thomas Brox. „U-Net: Convolutional Networks for Biomedical Image Segmentation“. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Hrsg. von Nassir Navab et al. Bd. 9351. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015, S. 234–241. ISBN: 978-3-319-24573-7. DOI: 10.1007/978-3-319-24574-4_28.
- [94] Martin Rudorfer, Lukas Neumann und Jörg Krüger. „Towards Learning 3d Object Detection and 6d Pose Estimation from Synthetic Data“. In: *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. 2019, S. 1540–1543. DOI: 10.1109/ETFA.2019.8869318.
- [95] Nataniel Ruiz, Samuel Schuster und Manmohan Chandraker. „Learning To Simulate“. In: *International Conference on Learning Representations*. 2019. URL: <https://openreview.net/forum?id=HJgkx2Aqt7>.
- [96] Olga Russakovsky et al. „ImageNet Large Scale Visual Recognition Challenge“. In: *International Journal of Computer Vision* 115.3 (2015), S. 211–252. ISSN: 0920-5691. DOI: 10.1007/s11263-015-0816-y.
- [97] Graeme D. Ruxton. „The unequal variance t-test is an underused alternative to Student’s t-test and the Mann–Whitney U test“. In: *Behavioral Ecology* 17.4 (Mai 2006), S. 688–690. ISSN: 1045-2249. DOI: 10.1093/beheco/ark016.

- [98] Fereshteh Sadeghi und Sergey Levine. „CAD2RL: Real Single-Image Flight Without a Single Real Image“. In: *Proceedings of Robotics: Science and Systems XIII*. Cambridge, Massachusetts, USA, 2017. ISBN: 978-0-9923747-3-0. DOI: 10.15607/RSS.2017.XIII.034.
- [99] Fereshteh Sadeghi et al. „Sim2Real Viewpoint Invariant Visual Servoing by Recurrent Control“. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2018)*. Piscataway, NJ: IEEE, 2018, S. 4691–4699. ISBN: 978-1-5386-6420-9. DOI: 10.1109/CVPR.2018.00493.
- [100] Kripasindhu Sarkar, Kiran Varanasi und Didier Stricker. „Trained 3D Models for CNN based Object Recognition“. In: *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. SCITEPRESS - Science and Technology Publications, 2017, S. 130–137. ISBN: 978-989-758-225-7. DOI: 10.5220/0006272901300137.
- [101] S. S. Shapiro und M. B. Wilk. „An analysis of variance test for normality (complete samples)“. In: *Biometrika* 52.3-4 (1965), S. 591–611. ISSN: 0006-3444. DOI: 10.1093/biomet/52.3-4.591.
- [102] Zhiqiang Shen et al. „Towards Instance-Level Image-To-Image Translation“. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, S. 3678–3687. ISBN: 978-1-7281-3293-8. DOI: 10.1109/CVPR.2019.00380.
- [103] Ashish Shrivastava et al. „Learning from Simulated and Unsupervised Images through Adversarial Training“. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Hrsg. von IEEE Staff. Piscataway: IEEE, 2017, S. 2242–2251. ISBN: 978-1-5386-0457-1. DOI: 10.1109/CVPR.2017.241.
- [104] Karen Simonyan und Andrew Zisserman. „Very Deep Convolutional Networks for Large-Scale Image Recognition“. In: *ArXiv abs/1409.1556* (4. Sep. 2014). URL: <https://arxiv.org/pdf/1409.1556>.
- [105] Nripendra Kumar Singh und Khalid Raza. „Medical Image Generation Using Generative Adversarial Networks: A Review“. In: *Health Informatics: A Computational Perspective in Healthcare*. Hrsg. von Ripon Patgiri, Anupam Biswas und Pinki Roy. Bd. 932. Studies in Computational Intelligence. Singapore: Springer Singapore

- und Imprint Springer, 2021, S. 77–96. ISBN: 978-981-15-9734-3. DOI: 10.1007/978-981-15-9735-0_5.
- [106] Mikhail Sokolov et al. „High-Resolution Semantically Consistent Image-to-Image Translation“. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 16 (2023), S. 482–492. DOI: 10.1109/JSTARS.2022.3226705.
 - [107] Student. „THE PROBABLE ERROR OF A MEAN“. In: *Biometrika* 6.1 (1908), S. 1–25. ISSN: 0006-3444. DOI: 10.1093/biomet/6.1.1.
 - [108] Hao Su et al. „Render for CNN: Viewpoint Estimation in Images Using CNNs Trained with Rendered 3D Model Views“. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, S. 2686–2694. ISBN: 978-1-4673-8391-2. DOI: 10.1109/ICCV.2015.308.
 - [109] Christian Szegedy et al. „Going deeper with convolutions“. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [Piscataway (NJ): IEEE Computer Society], 2015, S. 1–9. ISBN: 978-1-4673-6964-0. DOI: 10.1109/CVPR.2015.7298594.
 - [110] Christian Szegedy et al. „Inception-v4 Inception-ResNet and the Impact of Residual Connections on Learning“. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 31.1 (2017). ISSN: 2159-5399. DOI: 10.1609/aaai.v31i1.11231.
 - [111] Christian Szegedy et al. „Rethinking the Inception Architecture for Computer Vision“. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*. Piscataway, NJ: IEEE, 2016, S. 2818–2826. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.308.
 - [112] Zheng Tang et al. „PAMTRI: Pose-Aware Multi-Task Learning for Vehicle Re-Identification Using Highly Randomized Synthetic Data“. In: *2019 International Conference on Computer Vision*. Piscataway, NJ: IEEE, 2019, S. 211–220. ISBN: 978-1-7281-4803-8. DOI: 10.1109/ICCV.2019.00030.
 - [113] Josh Tobin et al. „Domain Randomization and Generative Models for Robotic Grasping“. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, S. 3482–3489. ISBN: 978-1-5386-8094-0. DOI: 10.1109/IROS.2018.8593933.

- [114] Josh Tobin et al. „Domain randomization for transferring deep neural networks from simulation to the real world“. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 92017, S. 23–30. ISBN: 978-1-5386-2682-5. DOI: 10.1109/IROS.2017.8202133.
- [115] Hugo Touvron et al. „Training data-efficient image transformers & distillation through attention“. In: *ArXiv abs/2012.12877* (23. Dez. 2020). URL: <https://arxiv.org/pdf/2012.12877>.
- [116] Jonathan Tremblay et al. „Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization“. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW 2018)*. Piscataway, NJ: IEEE, 2018, S. 1082–10828. ISBN: 978-1-5386-6100-0. DOI: 10.1109/CVPRW.2018.00143.
- [117] Apostolia Tsirikoglou, Gabriel Eilertsen und Jonas Unger. „A Survey of Image Synthesis Methods for Visual Machine Learning“. In: *Computer Graphics Forum* 39.6 (2020), S. 426–451. ISSN: 0167-7055. DOI: 10.1111/cgf.14047.
- [118] Apostolia Tsirikoglou et al. „Procedural Modeling and Physically Based Rendering for Synthetic Data Generation in Automotive Applications“. In: *ArXiv abs/1710.06270* (17. Okt. 2017). URL: <https://arxiv.org/pdf/1710.06270>.
- [119] Moritz Weisenböhler, Björn Hein und Christian Wurll. „On Scene Engineering and Domain Randomization: Synthetic Data for Industrial Item Picking“. In: *Intelligent Autonomous Systems 17*. Hrsg. von Emanuele Petrovic Ivanand Menegatti und Ivan Marković. Cham: Springer Nature Switzerland, 2023, S. 643–660. ISBN: 978-3-031-22216-0. DOI: 10.1007/978-3-031-22216-0_43.
- [120] Moritz Weisenböhler et al. „Sim2Real When Data Is Scarce: Image Transformation for Industrial Applications“. In: *Intelligent Autonomous Systems 18*. Hrsg. von Soon-Geul Lee et al. Cham: Springer Nature Switzerland, 2024, S. 65–76. ISBN: 978-3-031-44981-9. DOI: 10.1007/978-3-031-44981-9_6.
- [121] B. L. Welch. „THE GENERALIZATION OF ‘STUDENT’S’ PROBLEM WHEN SEVERAL DIFFERENT POPULATION VARIANCES ARE INVOLVED“. In: *Biometrika* 34.1-2 (1947), S. 28–35. ISSN: 0006-3444. DOI: 10.1093/biomet/34.1-2.28.

- [122] Saining Xie et al. „Aggregated Residual Transformations for Deep Neural Networks“. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Hrsg. von IEEE Staff. Piscataway: IEEE, 2017, S. 5987–5995. ISBN: 978-1-5386-0457-1. DOI: 10.1109/CVPR.2017.634.
- [123] Sergey Zakharov, Wadim Kehl und Slobodan Ilic. „DeceptionNet: Network-Driven Domain Randomization“. In: *2019 International Conference on Computer Vision*. Piscataway, NJ: IEEE, 2019, S. 532–541. ISBN: 978-1-7281-4803-8. DOI: 10.1109/ICCV.2019.00062.
- [124] Jingwei Zhang et al. „VR-Goggles for Robots: Real-to-Sim Domain Adaptation for Visual Control“. In: *IEEE Robotics and Automation Letters* 4.2 (2019), S. 1148–1155. DOI: 10.1109/LRA.2019.2894216.
- [125] Jun-Yan Zhu et al. „Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks“. In: *ICCV 2017. Proceedings (IEEE International Conference on Computer Vision*. Online). Los Alamitos, Washington und Tokyo: CPS and IEEE Computer Society, 2017, S. 2242–2251. ISBN: 978-1-5386-1032-9. DOI: 10.1109/ICCV.2017.244.
- [126] Mu Zhu. *Recall, precision and average precision*. Aug. 2004. URL: <https://datascience-intro.github.io/1MS041-2022/Files/AveragePrecision.pdf> (besucht am 18.11.2025).
- [127] Xiaomeng Zhu et al. „Towards Sim-to-Real Industrial Parts Classification with Synthetic Dataset“. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2023, S. 4454–4463. ISBN: 979-8-3503-0249-3. DOI: 10.1109/CVPRW59228.2023.00468.
- [128] Barret Zoph et al. „Learning Transferable Architectures for Scalable Image Recognition“. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2018)*. Piscataway, NJ: IEEE, 2018, S. 8697–8710. ISBN: 978-1-5386-6420-9. DOI: 10.1109/CVPR.2018.00907.
- [129] Guoyu Zuo et al. „Low-Quality Rendering-Driven 6D Object Pose Estimation from Single RGB Image“. In: *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 72020, S. 1–8. ISBN: 978-1-7281-6926-2. DOI: 10.1109/IJCNN48605.2020.9207286.

Eigene Veröffentlichungen

- 2018 Moritz Weisenböhler und Christian Wurll. “Dynamic Route Planning for Area Processing Autonomous Robots”. In: ISR 2018; 50th International Symposium on Robotics, Munich, Germany, 2018, S. 1-7. ISBN: 978-3-8007-4699-6.
- 2019 Moritz Weisenböhler, Michael Walz und Christian Wurll. “Automated Item Picking - Machine Learning Approach to Recognize Sports and Fashion Articles”. In: Artificial Intelligence: from Research to Application; the Upper-Rhine Artificial Intelligence Symposium (UR-AI 2019), 2019, S. 71-75. DOI: 10.48550/arXiv.1903.08495
- 2020 Moritz Weisenböhler und Christian Wurll. “Automated Item Picking for fashion articles using Deep Learning”. In: ISR 2020; 52th International Symposium on Robotics, Munich, Germany, 2020, S. 1-8. ISBN: 978-3-8007-5428-1.
- 2023 Moritz Weisenböhler, Björn Hein und Christian Wurll. “On Scene Engineering and Domain Randomization: Synthetic Data for Industrial Item Picking”. In: Intelligent Autonomous Systems 17. Hrsg. von Emanuele Petrovic Ivan and Menegatti und Ivan Marković. Cham: Springer Nature Switzerland, 2023, S. 643–660. ISBN: 978-3-031-22216-0. DOI: 10.1007/978-3-031-22216-0_43.
- 2024 Moritz Weisenböhler, Philipp Augenstein, Björn Hein, Christian Wurll und Kai Furmans. “Sim2Real When Data Is Scarce: Image Transformation for Industrial Applications”. In: Intelligent Autonomous Systems 18. Hrsg. von Soon-Geul Lee u. a. Cham: Springer Nature Switzerland, 2024, S. 65–76. ISBN: 978-3-031-44981-9. DOI: 10.1007/978-3-031-44981-9_6.
- 2025 Philipp Augenstein, Luisa Pfreundschuh, Till Weber, Moritz Weisenböhler, Christian Wurll und Björn Hein. “The DeepFrame Concept: Improving AI Robustness for Industrial Applications”. In: 2025 IEEE 30th International Conference on Emerging Technologies and

Eigene Veröffentlichungen

Factory Automation (ETFA), Porto, Portugal, 2025, S. 1-4. ISBN:
979-8-3315-5383-8. DOI: 10.1109/ETFA65518.2025.11205533.

Abbildungsverzeichnis

1.1.	Schematische Darstellung des Gesamtprozesses von der Simulationserstellung bis zu den ermittelten Performanzkennwerten der neuronalen Netze.	6
2.1.	Reale Beispielbilder der drei untersuchten Anwendungsfälle (a) IP-A, (b) IP-B und (c) QP-A.	13
3.1.	GAN-Architektur mit Generator G , der aus Rauschen Z Bilder X' erzeugt und Diskriminator D , der die Entscheidung d trifft, ob Bilder aus der Referenzdomäne X stammen oder erzeugt wurden (X').	39
3.2.	CycleGAN-Architektur bestehend aus der Vorwärtstransformation mit Generator G und Diskriminator D_y (blau), der Rücktransformation mit Generator F und Diskriminator D_x (rot) sowie den Überprüfungen der Zykluskonsistenz c_x und c_y (gepunktet).	40
4.1.	Beispielbilder des realen Pre-Evaluierungsdatensatzes, den SynthDet-Datensätzen SD und SD-Opt (oben, v.l.n.r) sowie den Scene Engineering-Datensätzen SE-Preopt, SE-Opt und SE-Exp (unten, v.l.n.r).	57
4.2.	Beispiel für chromatische Aberration in den realen Bilddaten. Hierbei sind verschiedene rote und blaue Einfärbungen im Bild zu erkennen.	69
5.1.	Die gemessene Verteilungsdichte der Performanzkennwerte für 1 094 standardisierte Trainingsreihen (blau) sowie die erwartete Verteilungsdichte bei einer Standardnormalverteilung (rot).	82
5.2.	Wiederholungen der Datengenerierung für dieselbe Simulationsszene (SE-Exp des Anwendungsfalls IP-A) mit jeweils anderen Zufallszahlen.	84

5.3. Ergebnisse der Studie zu den Datengenerierungsmethoden SynthDet (SD) und Scene Engineering (SE) in jeweils drei Varianten.	86
5.4. Ergebnisse der Studie zur Szenenstruktur für unterschiedliche Hintergründe mit (links) und ohne (rechts) Ladungsträger. Es wird jeweils die Veränderung der Performanz gegenüber der Basis mit einfarbigem Hintergrund und sichtbaren Umgebungsobjekten angegeben.	89
5.5. Ergebnisse der Studie zum Ladungsträger mit unterschiedlichen Farben und 3D-Modellen (links) sowie mit unterschiedlichen Farben und Texturen (rechts).	91
5.6. Ergebnisse der Studie zum Sichtfeld für unterschiedliche Kameraabstände und drei Randomisierungsgrade (links) sowie für 6D-Blickwinkelveränderungen (BWV) und die (Un-)Zentriertheit des Ladungsträgers im Bild (rechts).	94
5.7. Ergebnisse der Studie zur Beleuchtung mit drei Beleuchtungsarten sowie für unterschiedliche Randomisierungsgrade und Helligkeiten.	97
5.8. Ergebnisse der Studie zur Objektpositionierung mit physikalischer Kollisionssimulation und zufallsbasierter Posenbestimmung ohne Kollisionsprüfung sowie mit und ohne Umgebungsobjekte.	99
5.9. Beispielbilder der Simulationsoptimierung (Scene Adaptation) mit (a) Ziel- und (b) Startzustand der Optimierung sowie den Ergebnissen (c) der Posenoptimierung für Kamera und Ladungsträger und (d) der vollen Optimierung inklusive Objekteigenschaften und Beleuchtung.	105
5.10. Ergebnisse der Studie zur Scene Adaptation mit jeweils fünf Wiederholungen sowie akkuratem und inakkuratem 3D-Modell der Szene. Beim inakkuraten 3D-Modell stimmt die relative Positionierung zwischen Ladungsträger und Tisch nicht mit den realen Referenzdaten überein.	106
5.11. Ergebnisse der Augmentation mit Unschärfe auf Basis des Gauß- (links) und des Box-Filters (rechts) jeweils für unterschiedliche Variationsstärken.	108

5.12. Ergebnisse der Augmentation mit Gauß'schem Rauschen für unterschiedlichen Variationsstärken (links) sowie der kombinierten Augmentation mit Unschärfe (Gauß-Filter) und Gauß'schem Rauschen (rechts).	110
5.13. Ergebnisse der Augmentation mit RGB-Farb- (links) und mit Helligkeitsvariation (rechts) jeweils für unterschiedliche Variationsstärken.	112
5.14. Ergebnisse der kombinierten Augmentationen mit RGB-Farb- und Helligkeitsvariation für unterschiedliche Variationsstärken und bis zu 40 Trainingsepochen.	113
5.15. Ergebnisse der kombinierten Augmentation der Kanäle im HSV-Farbraum (links) sowie der Vergleich der RGB- und Helligkeits- mit der HSV-Augmentationen beim Training mit bis zu 40 Epochen (rechts).	114
5.16. Ergebnisse der Augmentation mit chromatischer Aberration für unterschiedliche Variationsstärken (links) sowie der kombinierten Augmentation von RGB-Farbvariation und chromatischer Aberration (rechts).	116
5.17. Ergebnisse der Augmentation mit Kontrastvariation für unterschiedliche Variationsstärken.	118
5.18. Beispielbilder für die untersuchten Methoden der Domain Adapatation: (a) Eingabebilder, (b) ICT und (c) GAN.	123
5.19. Ergebnisse der ICT-Bildanpassung mit unterschiedlichen realen Referenzdatensätzen bei Evaluierung mit dem Pre- (links) und dem Post-Datensatz (rechts).	124
5.20. Ergebnisse der ICT-Bildanpassung mit unterschiedlichen Zuordnungsmethoden der Referenzbilder (links) und mit unterschiedlich starker Weichzeichnung der Objektkonturen für die Zuordnung auf Maskenebene (rechts).	126
5.21. Beispielbilder für die GAN-Transformation mit dem Pre-Referenzdatensatz und unterschiedlicher Anzahl an Schritten beim Training des GANs.	128
5.22. Ergebnisse der GAN-Transformation mit unterschiedlichem Trainingsumfang (links) ohne und mit Gradient-Penalty-Verlust (letzter Eintrag links) sowie unterschiedlichen Referenzdatensätzen (rechts).	129

5.23. Ergebnisse der Kombination von Domain Adaptation und Domain Randomization mit der SV- und der Kontrastaugmentation und deren Kombination im Vergleich zum unveränderten Basisdatensatz SE-Opt.	134
5.24. Ergebnisse der Kombination von Domain Adaptation und Domain Randomization mit kombinierten Augmentationen mit Rauschen und Unschärfe im Vergleich zum unveränderten Basisdatensatz SE-Opt bei Evaluierung mit dem Pre- (links) und dem Post-Datensatz (rechts).	136
5.25. Ergebnisse der Übertragung auf die CNN-Architekturen Mask R-CNN (maskenbasiert) und SSD für reale und synthetische Datensätze des Anwendungsfalls IP-A ohne und mit Modifikation (DR und DADR). Die Evaluierung wurde mit dem Pre- (oben) und dem Post-Datensatz (unten) durchgeführt.	143
A.1. Maximal erreichte Performanz mit alternativen Berechnungsmethoden für die HSV-Augmentation mit Variationsstärke 100 % (links) sowie für die Kontrastaugmentation mit verschiedenen Variationsstärken (rechts).	182
B.1. Beispielbilder (I) für verschiedene Augmentationstechniken mit ausgewählten Variationsstärken (in Klammern).	186
B.2. Beispielbilder (II) für verschiedene Augmentationstechniken mit ausgewählten Variationsstärken (in Klammern).	187
B.3. Beispielbilder (I) für verschiedene Kameraposen und sich ergebende Sichtfelder.	188
B.4. Beispielbilder (II) für verschiedene Kameraposen und sich ergebende Sichtfelder.	189
B.5. Beispiele für transformierte Bilder bei unterschiedlichen realen Referenzdatensätzen für das GAN-Training.	190
C.1. Ergebnisse der Studie zur Anzahl an Trainingsdaten für drei Scene Engineering-Datensätze mit einer Verdrei- oder Verneunfachung der Datensatzgröße mit zusätzlichen Bilddaten bei derselben Anzahl an Trainingsschritten.	191
C.2. Ergebnisse der Studie zu den Datengenerierungsmethoden SynthDet (SD) und Scene Engineering (SE) in jeweils drei Varianten und bei Evaluierung mit dem Post-Datensatz.	192

C.3. Ergebnisse der Studie zur Datengenerierung mit Scene Engineering sowie mit vereinfachter und mit kontextarmer Szene (ähnlich SynthDet beim IP-A Anwendungsfall) für den IP-B Anwendungsfall bei Evaluierung mit dem zum Trainingsdatensatz nahen und dem fernen Datensatz. (Modifiziert nach [119])	192
C.4. Ergebnisse der Studie zur Szenenstruktur für unterschiedliche Hintergründe (links) mit und (rechts) ohne Ladungsträger bei Evaluierung mit dem Post-Datensatz. Es wird jeweils die Veränderung der Performanz gegenüber der Basis mit einfarbigem Hintergrund und sichtbaren Umgebungsobjekten angegeben. (Modifiziert nach [119])	193
C.5. Ergebnisse der Studie zum Ladungsträger mit unterschiedlichen Farben und 3D-Modellen (links) sowie mit unterschiedlichen Farben und Texturen (rechts) bei Evaluierung mit dem Post-Datensatz.	193
C.6. Ergebnisse der Studie zur Szenenstruktur mit und ohne Ladungsträger sowie mit unterschiedlicher Umgebungsobjekten für den Anwendungsfall IP-B. (Modifiziert nach [119]) . . .	194
C.7. Ergebnisse der Studie zum Sichtfeld für unterschiedliche Kameraabstände und drei Randomisierungsgrade (links) sowie für 6D-Blickwinkelveränderungen (BWV) und die (Un-)Zentriertheit des Ladungsträgers im Bild (rechts) bei Evaluierung mit dem Post-Datensatz.	194
C.8. Ergebnisse der Studie zum Sichtfeld mit zentrierter und unzentrierter Kamerapose sowie mit drei Randomisierungsgraden der Kameraposen für den Anwendungsfall IP-B. (Modifiziert nach [119])	195
C.9. Ergebnisse der Studie zur Beleuchtung mit drei Beleuchtungsarten sowie für unterschiedliche Randomisierungsgrade und Helligkeiten bei Evaluierung mit dem Post-Datensatz. .	195
C.10. Ergebnisse der Studie zur Beleuchtung mit drei Szenenarten sowie mit drei Randomisierungsgraden der Beleuchtung für den Anwendungsfall IP-B. (Modifiziert nach [119])	196
C.11. Auftreten der Objekte im Bild (normierte Überlagerung der Masken) je Klasse für die realen Datensätze des IP-A-Anwendungsfalls.	197

C.12. Auftreten der Objekte im Bild (normierte Überlagerung der Masken) je Klasse für ausgewählte synthetische Datensätze des IP-A-Anwendungsfalls.	198
C.13. Auftreten der Objekte im Bild (normierte Überlagerung der Masken) je Klasse für synthetische Datensätze der Studie zur Objektpositionierung.	199
C.14. Ergebnisse der Studie zur Objektpositionierung mit physikalischer Kollisionssimulation und zufallsbasierter Posenbestimmung ohne Kollisionsprüfung sowie mit und ohne Umgebung bei Evaluierung mit dem Post-Datensatz.	200
C.15. Ergebnisse der Studie zur Scene Adaptation mit jeweils fünf Wiederholungen sowie akkuratem und inakkuraten 3D-Modell bei Evaluierung mit dem Post-Datensatz.	200
C.16. Ergebnisse der Augmentation mit Unschärfe auf Basis des Gauß- (links) und des Box-Filters (rechts) jeweils für unterschiedliche Variationsstärken bei Evaluierung mit dem Post-Datensatz.	201
C.17. Ergebnisse der Augmentation mit Gauß'schem Rauschen für unterschiedliche Variationsstärken (links) sowie der kombinierten Augmentation mit Unschärfe (Gauß-Filter) und Gauß'schem Rauschen (rechts) bei Evaluierung mit dem Post-Datensatz.	202
C.18. Ergebnisse der Augmentation mit RGB-Farb- (links) und mit Helligkeitsvariation (rechts) für unterschiedliche Variationsstärken bei Evaluierung mit dem Post-Datensatz.	202
C.19. Ergebnisse der kombinierten Augmentationen mit RGB-Farb- und Helligkeitsvariation für unterschiedliche Variationsstärken und bis zu 40 Trainingsepochen bei Evaluierung mit dem Post-Datensatz.	203
C.20. Ergebnisse der kombinierten Augmentation der Kanäle im HSV-Farbraum (links) sowie der Vergleich der RGB- und Helligkeits- mit der HSV-Augmentationen beim Training mit bis zu 40 Epochen (rechts) bei Evaluierung mit dem Post-Datensatz.	203

C.21. Ergebnisse der Augmentation mit chromatischer Aberration für unterschiedliche Variationsstärken (links) sowie der kombinierten Augmentation von RGB-Farbvariation und chromatischer Aberration (rechts) bei Evaluierung mit dem Post-Datensatz.	204
C.22. Ergebnisse der Augmentation mit Kontrastvariation für unterschiedliche Variationsstärken bei Evaluierung mit dem Post-Datensatz.	204

Tabellenverzeichnis

5.1.	Abweichungen der Verteilung der Anzahl an Objekten pro Bild (ΔH) und resultierende Veränderungen der Performanz (ΔAP) zwischen synthetischen und realen Datensätzen (Pre und Post). Die geringsten (ΔH) bzw. signifikant höchsten Kennwerte (ΔAP) je Metrik sind hervorgehoben.	102
5.2.	Maximale Verbesserungen der Performanz (ΔAP [%]) für alle einzelnen Augmentationen und ausgewählte Kombinationen. Die signifikant höchsten Kennwerte sind hervorgehoben. . .	121
5.3.	Maximal erreichte Performanzverbesserungen durch die Domain Adaptation mit ICT (Klasse ✓ oder Maske M), GANs sowie deren Kombination. Dabei wird die ΔAP jeweils im Vergleich zum SE-Opt-Datensatz ohne Modifikation angegeben (erste Zeile) und jeweils mit dem Pre- und dem Post-Datensatz evaluiert. Die signifikant höchsten Kennwerte je Referenz- und Evaluierungsdatensatz sind hervorgehoben. .	131
5.4.	Maximal erreichte Performanz (AP, AP50, AP75, SR1 und ER1) mit DA (GAN- und ICT-Transformation jeweils mit Mini-Referenzdatensatz) und DR (SV- und Kontrastvariation). Die signifikant höchsten bzw. niedrigsten (ER1) Kennwerte je Metrik sind hervorgehoben.	139
5.5.	Maximal erreichte Performanzkennwerte für die drei Anwendungsfälle IP-A, IP-B und QP-A. Es werden die Metriken AP, AP50, SR1 und ER1 für bis zu zwei Evaluierungsdatensätze, welche der jeweiligen realen Trainingsdomäne ferner (Pre und Fern) oder näher (Post und Nah) sind, angegeben. Die Trainingsdatensätze umfassen jeweils auch Varianten mit DR (SV- und Kontrastvariation) und DADR (GAN- und ICT-Anpassung mit Mini-Referenzdatensatz und DR). Die signifikant höchsten Kennwerte je Anwendungsfall und Metrik sind hervorgehoben.	146

C.1. Maximale Verbesserungen der Performanz (ΔAP [%]) für alle einzelnen Augmentationen und ausgewählte Kombinationen bei Evaluierung mit dem Post-Datensatz. Die signifikant höchsten Kennwerte je Kategorie sind hervorgehoben. . . . 205

Abkürzungsverzeichnis

AP Average Precision

API Application Programming Interface

CNN Convolutional Neural Network

DA Domain Adaptation

DADR Kombination Domain Adaptation und Domain Randomization

DR Domain Randomization

GAN Generative Adversarial Network

GP Gradient Penalty

HSV Hue, Saturation, Value

ICT Image Color Transfer

IP Item Picking

IP-A Anwendungsfall Item Picking in der Automobilindustrie

IP-B Anwendungsfall Item Picking in der Bekleidungsindustrie

NN Neuronales Netz

QP-A Anwendungsfall Qualitätsprüfung in der Automobilindustrie

RGB Rot, Grün, Blau

SA Scene Adaptation

SD SynthDet

SE Scene Engineering

A. Anhang: Alternative Berechnungsmethoden

Im Folgenden werden zusätzlich untersuchte alternative Berechnungsansätze für die Augmentationsmethoden HSV- und Kontrastvariation vorgestellt.

A.1. HSV

Die in der Arbeit standardmäßig verwendete Berechnungsmethode für die HSV-Augmentation wird als „stabiler Mittelwert“ bezeichnet (s. Unterabschnitt 4.2.4). Die alternativ getesteten Berechnungsmethoden für die HSV-Variation wurden nach folgender Formel berechnet:

Zielverteilung:

$$p'_c(r_c) = \frac{\frac{1}{2} + \delta_c(r_c)}{\bar{p}_c} p_c \quad (\text{A.1})$$

mit

$$\bar{p}_c = \frac{1}{|P_c|} \sum_1^{|P_c|} p_c, \quad p_c \in P_c \in I \quad (\text{A.2})$$

Absolut:

$$p'_c(r_c) = \delta_c(r_c) + p_c \quad (\text{A.3})$$

Relativ:

$$p'_c(r_c) = \delta_c(r_c)) p_c \quad (\text{A.4})$$

Für alle Alternativen wurde jeweils der maximale Verstärkungsfaktor δ_{max} mit $\delta_{min} = \frac{1}{\delta_{max}}$ („relativ“) bzw. $\delta_{min} = -\delta_{max}$ („Zielverteilung“ und „absolut“) vorgegeben. Für die Vergleichsuntersuchung wurde jeweils die Variationsstärke $\delta_{max} = 1$ gewählt.

A. Anhang: Alternative Berechnungsmethoden

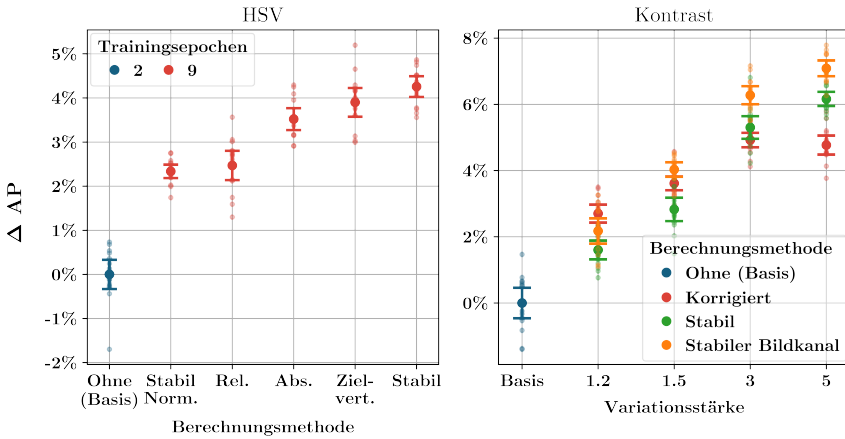


Abbildung A.1.: Maximal erreichte Performanz mit alternativen Berechnungsmethoden für die HSV-Augmentation mit Variationsstärke 100 % (links) sowie für die Kontrastaugmentation mit verschiedenen Variationsstärken (rechts).

Bei der Untersuchung der Berechnungsmethoden der Variationsstärke für die HSV-Augmentation (siehe Kapitel Unterabschnitt 4.2.4) wurde mit der Methode „stabiler Mittelwert“ (gewählter Standard) die signifikant besten Ergebnisse erzielt (s. Abbildung A.1 links). Die Berechnungsmethoden relativ ($\Delta AP -0.6\%$), absolut ($\Delta AP -1.7\%$) und „Zielverteilung“ ($\Delta AP -0.3\%$) führten jeweils im direkten Vergleich zur Methode „stabiler Mittelwert“ zu einer signifikant niedrigeren Performanz. Auch die Verwendung von normal- statt gleichverteilten Zufallszahlen als Berechnungsgrundlage für die Methode „stabiler Mittelwert“ führte zu einer signifikanten Performanzreduktion um $\Delta AP -2.0\%$. Für den untersuchten Anwendungsfall kann vermutet werden, dass die bei normalverteilten Zufallszahlen weniger häufig auftretenden Extremfälle zu einem schlechteren Trainingsergebnis führten.

A.2. Kontrast

Die in der Arbeit standardmäßig verwendete Berechnungsmethode für die Kontrastaugmentation wird als „stabiler Bildkanalmittelwert“ bezeichnet (s. Unterabschnitt 4.2.6). Die alternativ getesteten Berechnungsmethoden für die Kontrastvariation wurden nach folgender Formel berechnet:

Stabil:

$$p'_c(r) = (p_c - \bar{p})\delta(r) + \bar{p} \quad (\text{A.5})$$

mit

$$\bar{p} = \frac{1}{|P|} \sum_1^{|P|} p, \quad p \in I \quad (\text{A.6})$$

Korrigiert:

$$p'_c(r) = \delta(r)p_c + \frac{1 - \delta(r)}{2} \quad (\text{A.7})$$

Für die Studie wurde jeweils der maximale Verstärkungsfaktor δ_{max} mit $\delta_{min} = \frac{1}{\delta_{max}}$ vorgegeben.

Bei der Untersuchung der Berechnungsmethoden der Variationsstärke für den Kontrastaugmentation (siehe Kapitel Unterabschnitt 4.2.6) wurden mit der Methode „stabiler Bildkanalmittelwert“ (gewählter Standard) ab einer Variationsstärke $\delta_{max} \geq 3$ die signifikant besten Ergebnisse erzielt (s. Abbildung A.1 rechts). Dabei war die maximal erreichte Performanz bei Berechnungsmethoden „stabil“ (Kanäle nicht getrennt) um $\Delta\text{AP} -1.0\%$ und bei der Methoden „korrigiert“ um $\Delta\text{AP} -2.2\%$ im direkten Vergleich zur Methode „stabiler Bildkanalmittelwert“ signifikant niedriger.

B. Anhang: Ergänzende Beispielbilder

Im Folgenden finden sich ergänzende Beispielbilder zu den untersuchten Augmentationsmethoden und zu den Kameraposen bzw. Sichtfeld in der Simulationsszene.

B. Anhang: Ergänzende Beispielbilder



(a) Originalbild



(b) Gauß. Rauschen (1024)



(c) Unschärfe Box-Filter (21)



(d) Unschärfe Gauß-Filter (21)



(e) Unschärfe und Rauschen



(f) Rauschen und Unschärfe

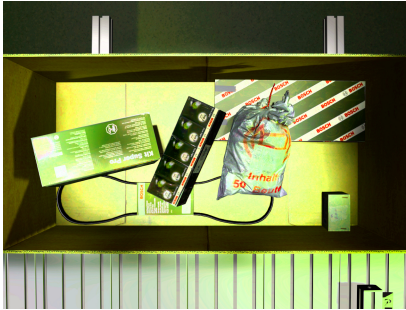
Abbildung B.1.: Beispielbilder (I) für verschiedene Augmentationstechniken mit ausgewählten Variationsstärken (in Klammern).



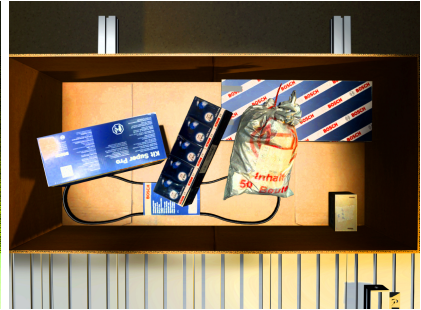
(a) Helligkeit (0.4)



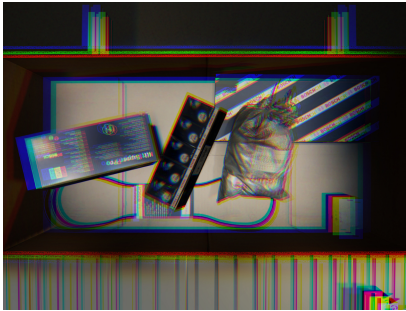
(b) Helligkeit (-0.4)



(c) HSV (1)



(d) SV (1)



(e) Chrom. Aberration (2)



(f) Kontrast (10)

Abbildung B.2.: Beispielbilder (II) für verschiedene Augmentationstechniken mit ausgewählten Variationsstärken (in Klammern).

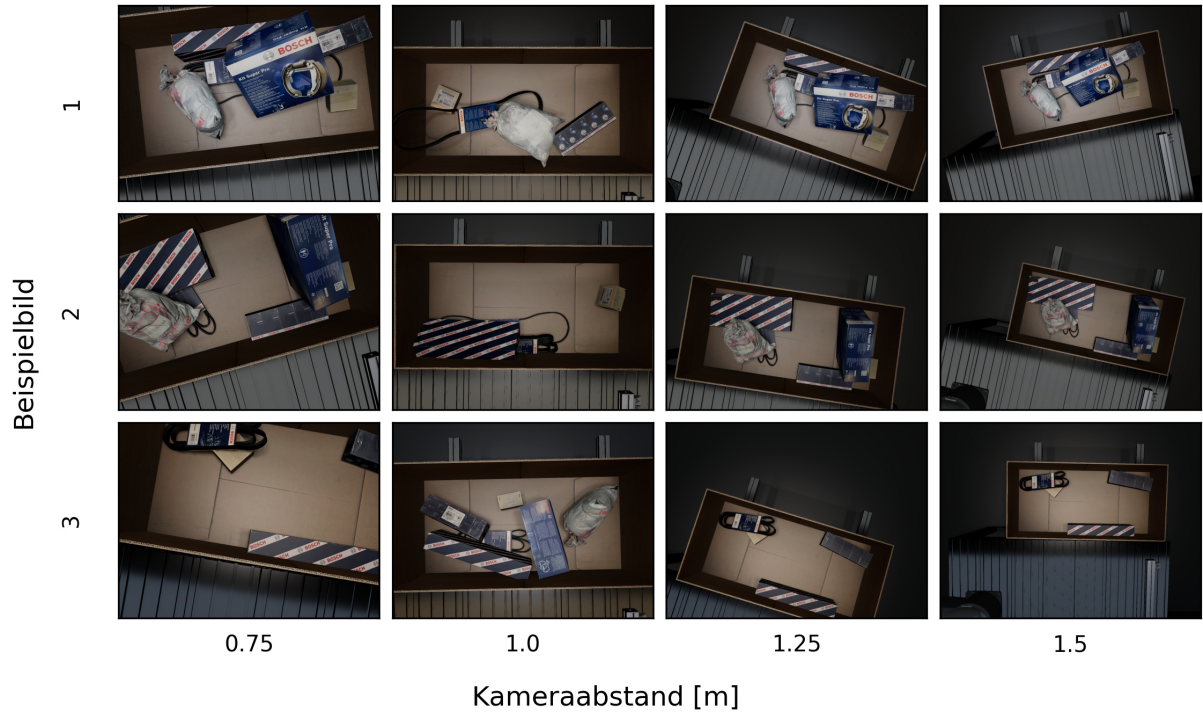


Abbildung B.3.: Beispielbilder (I) für verschiedene Kameraposen und sich ergebende Sichtfelder.

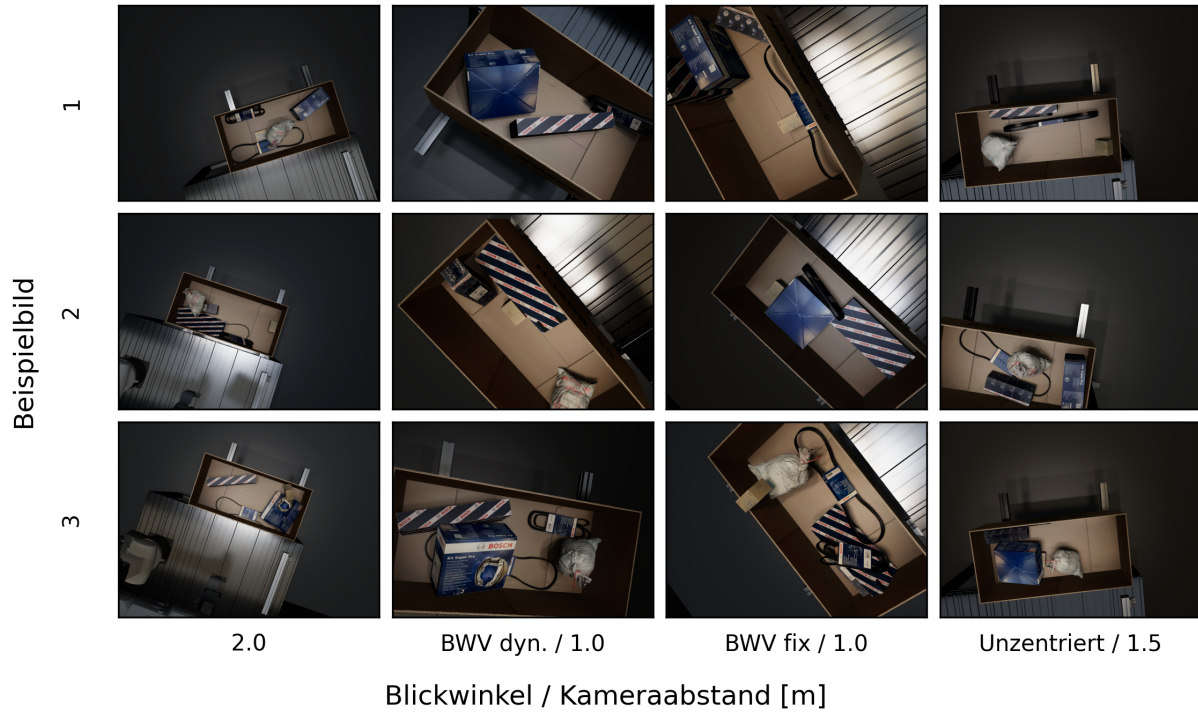
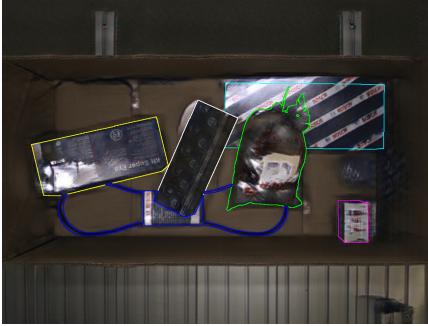


Abbildung B.4.: Beispielbilder (II) für verschiedene Kameraposen und sich ergebende Sichtfelder.

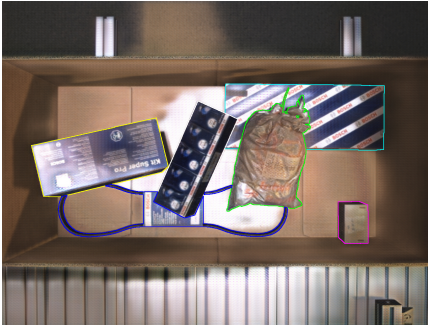
B. Anhang: Ergänzende Beispielbilder



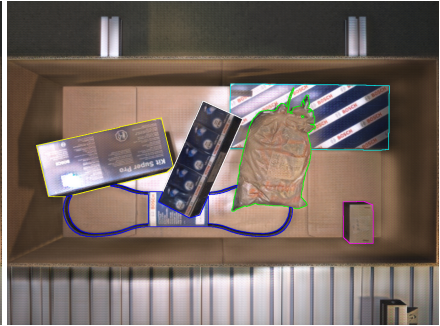
(a) Pre-Referenzdatensatz



(b) Voll-Referenzdatensatz



(c) Post-Referenzdatensatz



(d) Mini-Referenzdatensatz

Abbildung B.5.: Beispiele für transformierte Bilder bei unterschiedlichen realen Referenzdatensätzen für das GAN-Training.

C. Anhang: Ergänzende Abbildungen und Tabellen

Im Folgenden finden sich ergänzende Abbildungen und Tabellen zu den im Rahmen dieser Arbeit durchgeführten Experimenten. Diese umfassen die Darstellung der Ergebnisse zum alternativen Post-Datensatz des Anwendungsfalls IP-A und der Ergebnisse des alternativen Anwendungsfalls IP-B sowie Darstellungen zu weiteren Datensatzeigenschaften.

C.1. Datengenerierung

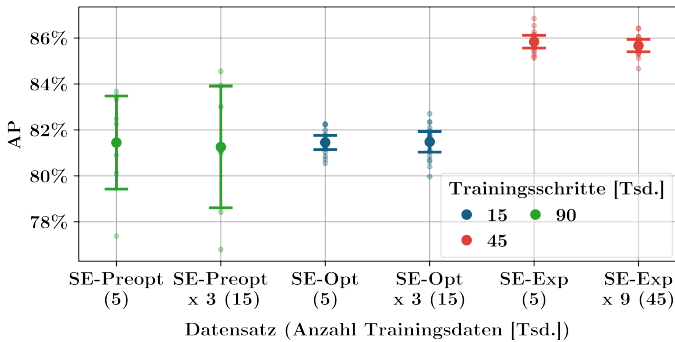


Abbildung C.1.: Ergebnisse der Studie zur Anzahl an Trainingsdaten für drei Scene Engineering-Datensätze mit einer Verdrei- oder Verneunfachung der Datensatzgröße mit zusätzlichen Bilddaten bei derselben Anzahl an Trainingsschritten.

C. Anhang: Ergänzende Abbildungen und Tabellen

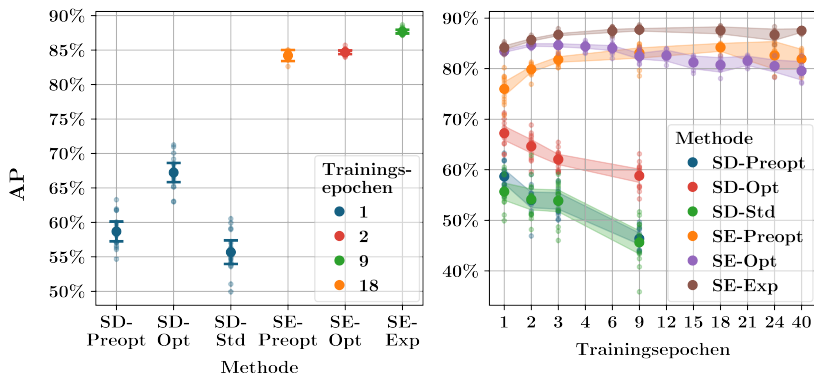


Abbildung C.2.: Ergebnisse der Studie zu den Datengenerierungsmethoden SynthDet (SD) und Scene Engineering (SE) in jeweils drei Varianten und bei Evaluierung mit dem Post-Datensatz.

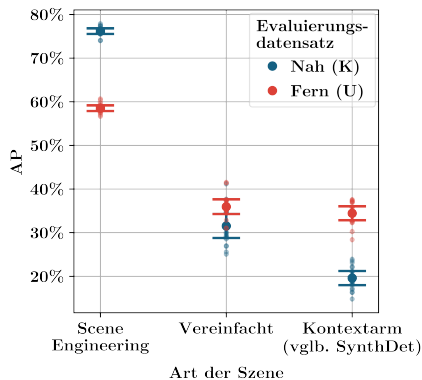


Abbildung C.3.: Ergebnisse der Studie zur Datengenerierung mit Scene Engineering sowie mit vereinfachter und mit kontextarmer Szene (ähnlich SynthDet beim IP-A Anwendungsfall) für den IP-B Anwendungsfall bei Evaluierung mit dem zum Trainingsdatensatz nahen und dem fernen Datensatz. (Modifiziert nach [119])

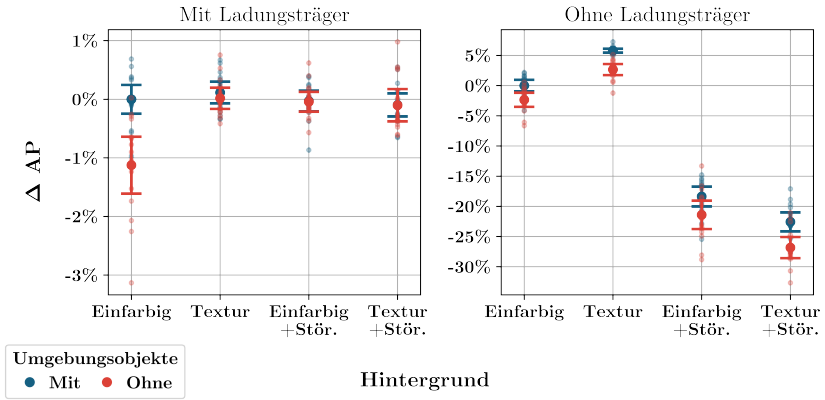


Abbildung C.4.: Ergebnisse der Studie zur Szenenstruktur für unterschiedliche Hintergründe (links) mit und (rechts) ohne Ladungsträger bei Evaluierung mit dem Post-Datensatz. Es wird jeweils die Veränderung der Performanz gegenüber der Basis mit einfarbigem Hintergrund und sichtbaren Umgebungsobjekten angegeben. (Modifiziert nach [119])

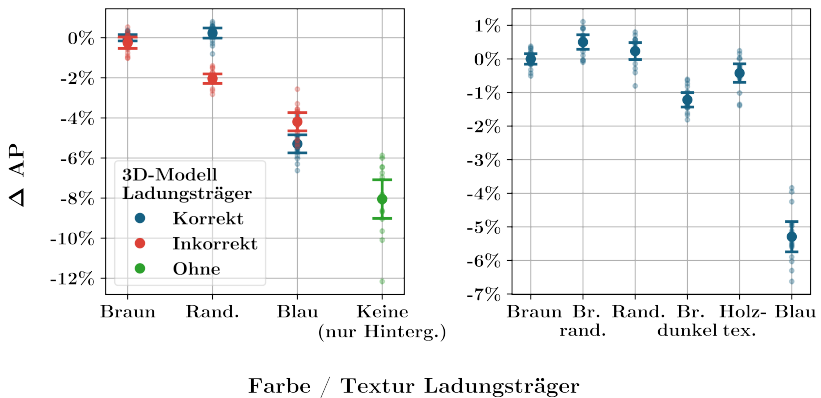


Abbildung C.5.: Ergebnisse der Studie zum Ladungsträger mit unterschiedlichen Farben und 3D-Modellen (links) sowie mit unterschiedlichen Farben und Texturen (rechts) bei Evaluierung mit dem Post-Datensatz.

C. Anhang: Ergänzende Abbildungen und Tabellen

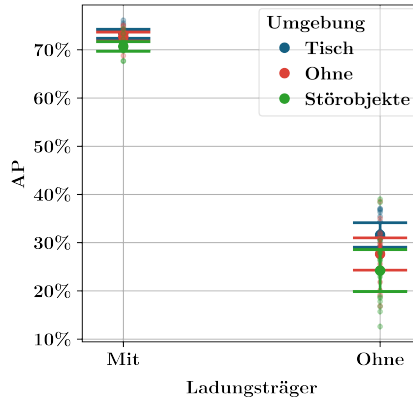


Abbildung C.6.: Ergebnisse der Studie zur Szenenstruktur mit und ohne Ladungsträger sowie mit unterschiedlicher Umgebungsobjekten für den Anwendungsfall IP-B. (Modifiziert nach [119])

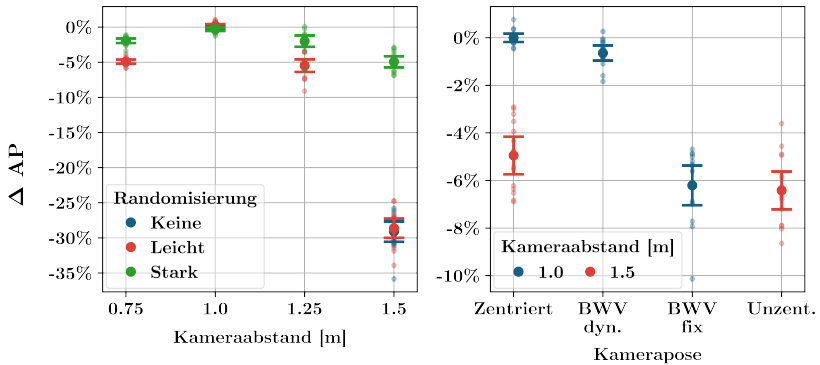


Abbildung C.7.: Ergebnisse der Studie zum Sichtfeld für unterschiedliche Kameraabstände und drei Randomisierungsgrade (links) sowie für 6D-Blickwinkelveränderungen (BWV) und die (Un-)Zentriertheit des Ladungsträgers im Bild (rechts) bei Evaluierung mit dem Post-Datensatz.

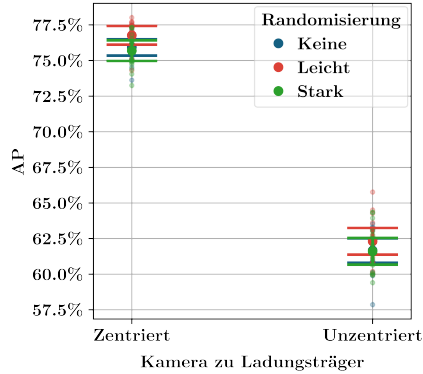


Abbildung C.8.: Ergebnisse der Studie zum Sichtfeld mit zentrierter und unzentrierter Kamerapose sowie mit drei Randomisierungsgraden der Kameraposen für den Anwendungsfall IP-B. (Modifiziert nach [119])

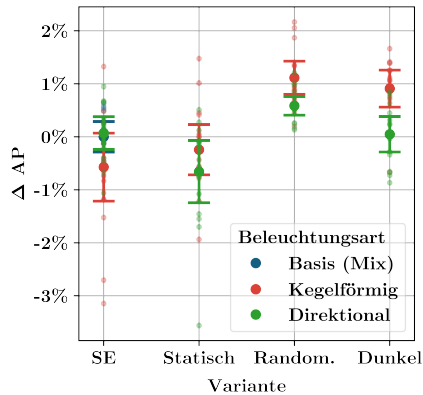


Abbildung C.9.: Ergebnisse der Studie zur Beleuchtung mit drei Beleuchtungsarten sowie für unterschiedliche Randomisierungsgrade und Helligkeiten bei Evaluierung mit dem Post-Datensatz.

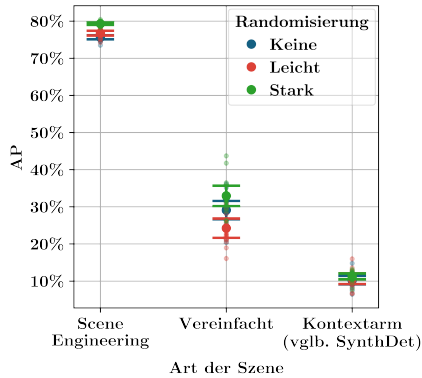


Abbildung C.10.: Ergebnisse der Studie zur Beleuchtung mit drei Szenenarten sowie mit drei Randomisierungsgraden der Beleuchtung für den Anwendungsfall IP-B. (Modifiziert nach [119])

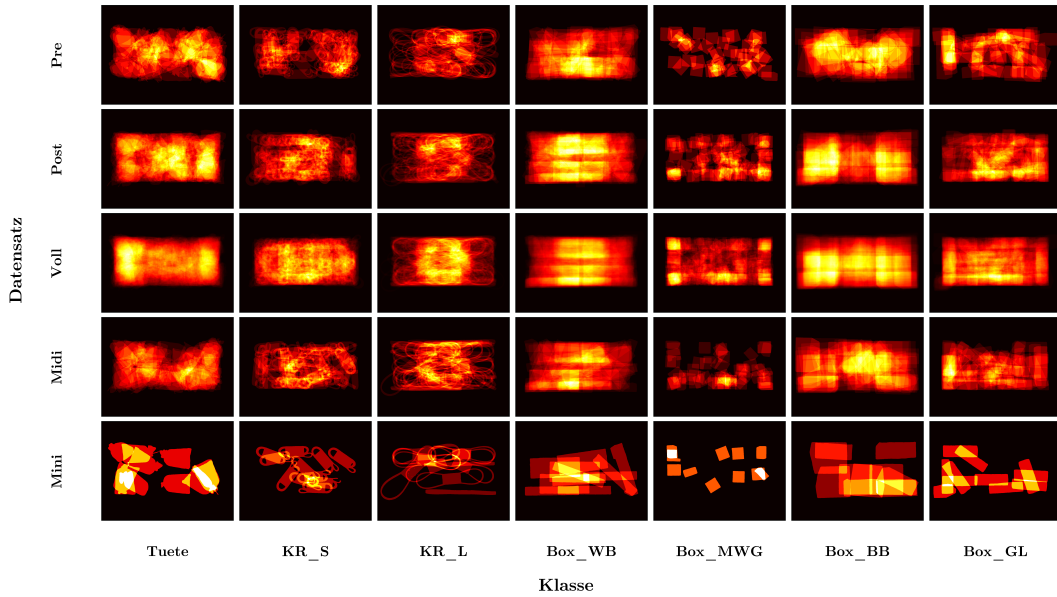


Abbildung C.11.: Auftreten der Objekte im Bild (normierte Überlagerung der Masken) je Klasse für die realen Datensätze des IP-A-Anwendungsfalls.

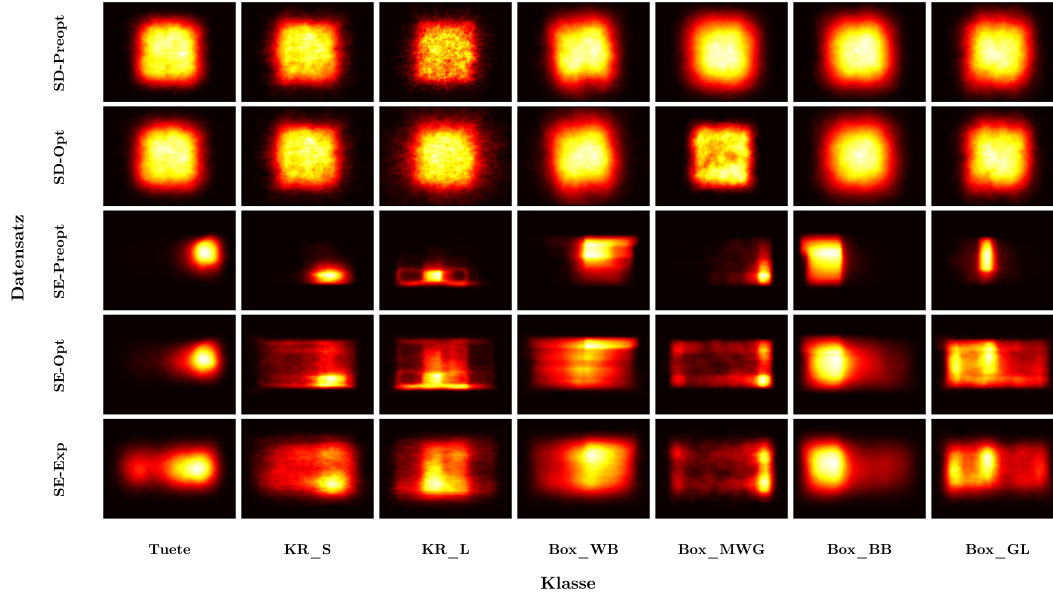


Abbildung C.12.: Auftreten der Objekte im Bild (normierte Überlagerung der Masken) je Klasse für ausgewählte synthetische Datensätze des IP-A-Anwendungsfalls.

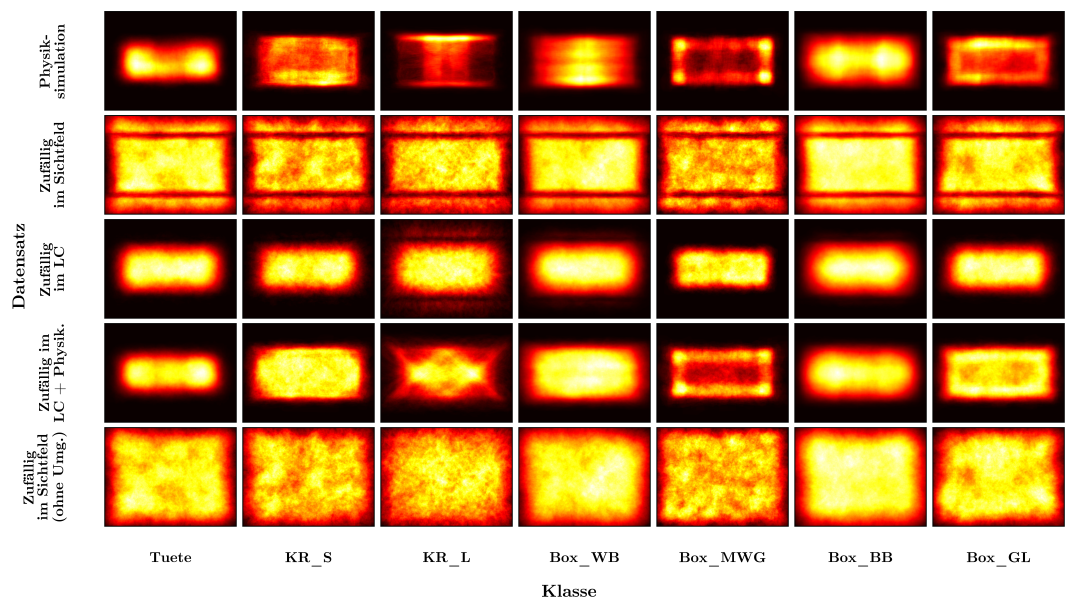


Abbildung C.13.: Auftreten der Objekte im Bild (normierte Überlagerung der Masken) je Klasse für synthetische Datensätze der Studie zur Objektpositionierung.

C. Anhang: Ergänzende Abbildungen und Tabellen

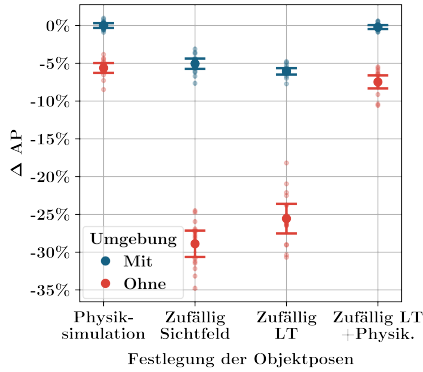


Abbildung C.14.: Ergebnisse der Studie zur Objektpositionierung mit physikalischer Kollisionssimulation und zufallsbasierter Posenbestimmung ohne Kollisionsprüfung sowie mit und ohne Umgebung bei Evaluierung mit dem Post-Datensatz.

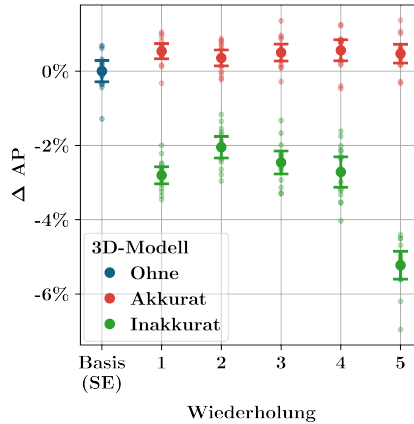


Abbildung C.15.: Ergebnisse der Studie zur Scene Adaptation mit jeweils fünf Wiederholungen sowie akkuratem und inakkuraten 3D-Modell bei Evaluierung mit dem Post-Datensatz.

C.2. Domain Randomization

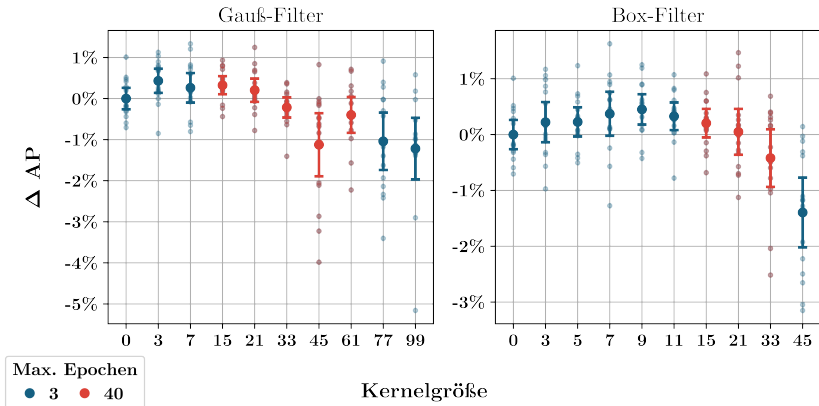


Abbildung C.16.: Ergebnisse der Augmentation mit Unschärfe auf Basis des Gauß- (links) und des Box-Filters (rechts) jeweils für unterschiedliche Variationsstärken bei Evaluierung mit dem Post-Datensatz.

C. Anhang: Ergänzende Abbildungen und Tabellen

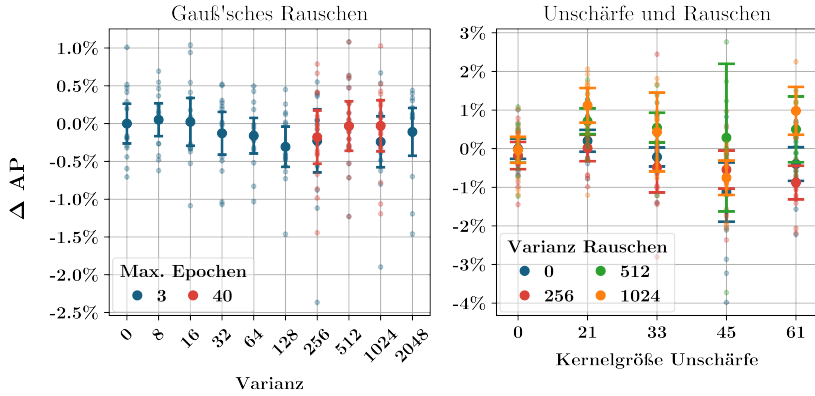


Abbildung C.17.: Ergebnisse der Augmentation mit Gauß'schem Rauschen für unterschiedliche Variationsstärken (links) sowie der kombinierten Augmentation mit Unschärfe (Gauß-Filter) und Gauß'schem Rauschen (rechts) bei Evaluierung mit dem Post-Datensatz.

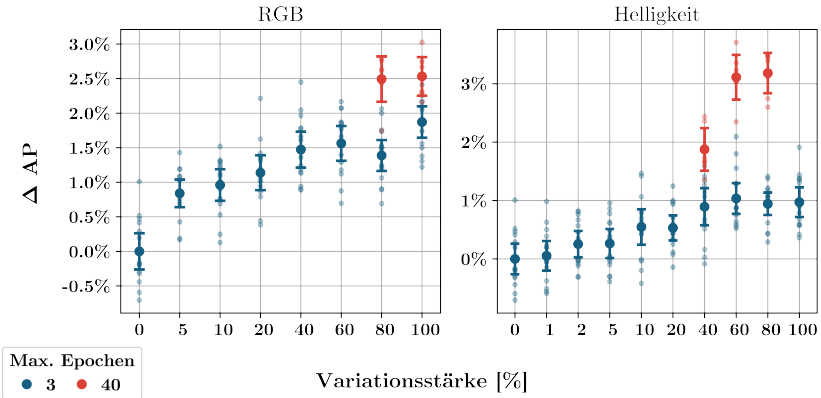


Abbildung C.18.: Ergebnisse der Augmentation mit RGB-Farb- (links) und mit Helligkeitsvariation (rechts) für unterschiedliche Variationsstärken bei Evaluierung mit dem Post-Datensatz.

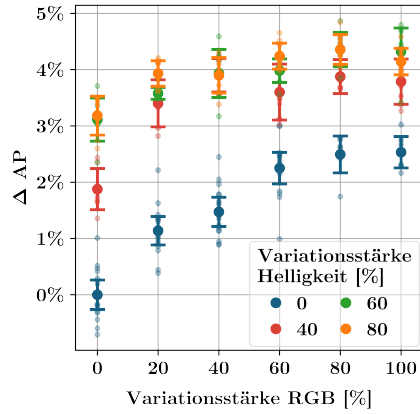


Abbildung C.19.: Ergebnisse der kombinierten Augmentationen mit RGB-Farb- und Helligkeitsvariation für unterschiedliche Variationsstärken und bis zu 40 Trainingsepochen bei Evaluierung mit dem Post-Datensatz.

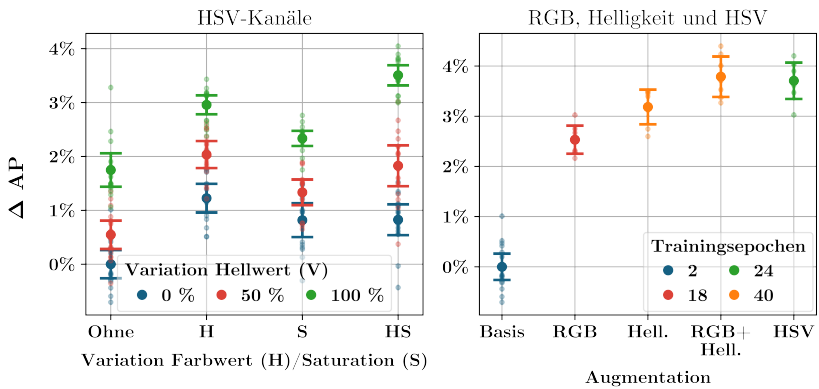


Abbildung C.20.: Ergebnisse der kombinierten Augmentation der Kanäle im HSV-Farbraum (links) sowie der Vergleich der RGB- und Helligkeits- mit der HSV-Augmentationen beim Training mit bis zu 40 Epochen (rechts) bei Evaluierung mit dem Post-Datensatz.

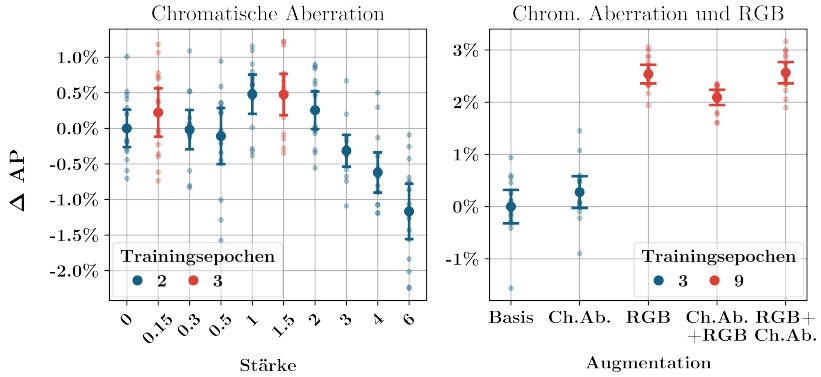


Abbildung C.21.: Ergebnisse der Augmentation mit chromatischer Aberration für unterschiedliche Variationsstärken (links) sowie der kombinierten Augmentation von RGB-Farbvariation und chromatischer Aberration (rechts) bei Evaluierung mit dem Post-Datensatz.

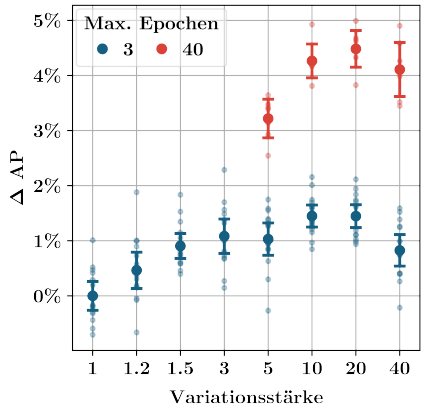


Abbildung C.22.: Ergebnisse der Augmentation mit Kontrastvariation für unterschiedliche Variationsstärken bei Evaluierung mit dem Post-Datensatz.

Tabelle C.1.: Maximale Verbesserungen der Performanz (ΔAP [%]) für alle einzelnen Augmentationen und ausgewählte Kombinationen bei Evaluierung mit dem Post-Datensatz. Die signifikant höchsten Kennwerte je Kategorie sind hervorgehoben.

Augmentation	Max. Perf. $\Delta\text{AP} \pm \text{Std.}$	Direkter Vergl. ΔAP (Aug.)	Epochen max. Perf.
Basis (SE-Opt)	0.0 ± 0.5		2
Gauß. Rauschen (N)	0.0 ± 0.6		6
Unschärfe Box (X)	0.0 ± 0.7		2
Unschärfe Gauß (U)	0.2 ± 0.5		2
Chrom. Aberration (C)	0.5 ± 0.7		6
Helligkeitsvariation (L)	1.7 ± 0.5		18
SV-Variation (SV)	2.3 ± 0.3		6
RGB-Variation (RGB)	2.5 ± 0.3		18
HSV-Variation (HSV)	3.7 ± 0.4		24
Kontrastvariation (K)	4.5 ± 0.4		40
N + U	0.7 ± 0.6	0.5 (U)	9
RGB + C	2.5 ± 0.4	0.0 (RGB)	9
N + U + RGB + L	2.5 ± 0.3	-1.3 (RGB + L)	18
N + U + K	2.9 ± 0.6	-1.6 (K)	40
N + U + HSV	2.9 ± 0.7	-0.8 (HSV)	40
N + U + SV	3.3 ± 0.5	1.0 (SV)	40
RGB + L	3.8 ± 0.4	1.3 (RGB)	40
RGB + L + K	4.4 ± 0.6	-0.1 (K)	40
SV + K	4.8 ± 0.4	0.3 (K)	40
HSV + K	4.9 ± 0.5	0.4 (K)	40