Konzeption und Parametrierung von Algorithmen zur Abbildung von Veränderungen in unstrukturierten Bilddaten

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

M.Sc. Friedrich Rieken Münke

Tag der mündlichen Prüfung:	16.05.2025
Hauptreferent:	Prof. Dr.-Ing. Markus Reischl
Korreferenten:	Prof. Dr.-Ing. Christoph Stiller
	Prof. Dr. rer. nat. Stefan Elser

Zusammenfassung

Mit der Verfügbarkeit von mobilen Kameras ist es möglich, große Mengen von Bilddaten über die Zeit zu erheben. Bilder von Smartphones, Satelliten, Auto- und Überwachungskameras enthalten eine Vielzahl an Informationen, die manuell nicht mehr ausgewertet werden können. Bei in großem Umfang aufgezeichneten Bilddaten sind die Aufnahmebedingungen häufig unbekannt und erschweren die Auswertung und insbesondere die Vergleichbarkeit von Ergebnissen über die Zeit.

Die vorliegende Dissertation befasst sich mit der Konzeption und Parametrierung von Algorithmen zur Abbildung von Veränderungen in unstrukturierten Bilddaten. Hierfür wird ein allgemeines Konzept entwickelt. Anstatt Pixelvergleiche anzustellen, werden Objekte anhand von Zustandsvektoren verglichen. So können Störfaktoren, wie unterschiedliche Perspektiven und Umwelteinflüsse, kompensiert und komplexe Zustände erfasst werden. Das Konzept umfasst einen Algorithmus, der aus den drei Teilen Assoziation, Interpretation und Aggregation besteht. Die Assoziation wählt relevante Bilder oder Bildausschnitte aus und ordnet diese einzelnen Objekten zu, die Interpretation extrahiert Merkmale aus den zugeordneten Bildern, und die Aggregation fasst diese Informationen zu einem Gesamtzustand zusammen.

Um das Konzept für einen Anwendungsfall zu parametrieren, wird eine Bewertungsmethodik eingeführt, die ohne zusätzlichen Annotationsaufwand auskommt und die Robustheit und Deskriptivität eines Algorithmus bewertet. Die Bewertungsmethodik wird anhand eines Beispiels für die Erstellung eines Algorithmus zur Abbildung des Straßenzustands demonstriert. Unterschiedliche Algorithmen-Varianten werden parametrisiert und bewertet. Zusätzlich wird die Methodik auf

einen Datensatz aus dem Bereich *Remote Sensing* übertragen. In diesem Fall werden sowohl überwachte als auch unüberwachte Modelle konfiguriert und bewertet.

Es werden zwei Optionen für die Interpretation von Bilddaten vorgestellt: unüberwachte Methoden, wie *Bag of Visual Word*-Modelle und vortrainierte *Deep Convolutional Neural Networks*, die abstrakte Repräsentationen ohne annotierte Daten berechnen, und überwachte Modelle zur semantischen Segmentierung, wie *Conventional Image Processing Pipelines*, *Structured Encoder-Decoder* oder *U-Net*. Diese Arbeit vergleicht diese Methoden auf drei repräsentativen Datensätzen und bestimmt ihre Anwendbarkeit bei ungleichen Klassenverteilungen und in den Domänen *Fahrzeugbilddaten* und *Remote Sensing*.

Letztlich steht dem Anwender ein allgemeines Konzept für die Erstellung von Algorithmen zur Verfügung sowie eine Bewertungsmethodik, die deren effiziente Optimierung ermöglicht. Unterschiedliche Algorithmen wurden auf ihre Dateneffizienz untersucht, sodass basierend auf dieser Arbeit Algorithmen zur Abbildung von Veränderungen unter verschiedenen Bedingungen entwickelt werden können.

Vorwort

Die vorliegende Dissertation ist das Ergebnis meiner Forschungsarbeit der letzten fünfeinhalb Jahre, die ich im Rahmen meines Promotionsvorhabens durchführen durfte. Diese Zeit war geprägt von intensiven Studien, Herausforderungen und persönlichen Entwicklungen, die mich sowohl wissenschaftlich als auch persönlich bereichert haben.

Zu Beginn möchte ich meinem Betreuer Herrn Prof. Dr. Reischl meinen tiefsten Dank aussprechen. Durch seine ständige Unterstützung, sein Vertrauen und seine wertvollen Ratschläge wurde diese Arbeit in ihrer jetzigen Form möglich. Zudem möchte ich Herrn Prof. Dr. Stiller für die Übernahme des Korreferats danken.

Mein herzlicher Dank gilt auch meinen Kolleginnen und Kollegen, die mich durch ihr Fachwissen und ihre Hilfsbereitschaft unterstützt haben. Die Zusammenarbeit und der Austausch mit ihnen war stets bereichernd und hat die Entwicklung dieser Arbeit maßgeblich beeinflusst. Einen besonderen Dank möchte ich Jan Schützke aussprechen, der in dieser Zeit stets an meiner Seite war und mich immer unterstützt hat. Ebenso möchte ich Sandra Murr meinen aufrichtigen Dank aussprechen, die meine Dissertation mit großer Sorgfalt korrekturgelesen und damit wesentlich zur Qualität dieser Arbeit beigetragen hat.

Ich empfinde tiefe Dankbarkeit gegenüber meiner Familie und meinen Freunden, die mir über die Jahre hinweg stets Rückhalt und Motivation gegeben haben. Ihr Verständnis und ihre Geduld, insbesondere in den herausforderndsten Phasen meiner Forschung, waren für mich von unschätzbarem Wert. Besonders möchte ich meiner Frau Monique danken, deren unermüdliche Unterstützung es mir überhaupt erst ermöglicht hat, diese Doktorarbeit zu vollenden. Ebenso schätze

ich die Unterstützung meines großartigen Bruders Konrad, der immer ein offenes Ohr für mich hatte und mir stets mit Rat und Tat zur Seite stand.

Abschließend möchte ich mich auch bei allen weiteren Personen bedanken, die in irgendeiner Weise zum Gelingen dieser Arbeit beigetragen haben. Jeder noch so kleine Beitrag hat dazu geführt, dass diese Dissertation ihren jetzigen Abschluss fand.

Diese Arbeit ist der Höhepunkt einer intensiven und lehrreichen Reise, die mir gezeigt hat, wie wichtig Ausdauer, Neugierde und Leidenschaft für die Forschung sind. Ich hoffe, dass die Ergebnisse dieser Dissertation einen wertvollen Beitrag zur Abbildung von Veränderungen in unstrukturierten Bilddaten leisten und zukünftige Forschungen in diesem Bereich anregen können.

Inhaltsverzeichnis

Abkürzungen und Symbole

Abkürzungen

AE	Auto Encoder
ABV	Abbildung von Veränderungen
ANOVA	Analysis of Variance
AV	Algorithmus-Variante eines Algorithmus zur Abbildung von Veränderungen
BoVW	Bag of Visual Words
CD	Change Detection
CL	Contrastive-Learning
CIPP	Conventional Image Processing Pipeline
DL	Deep Learning
DNN	Deep Neural Network
DCNN	Deep Convolutional Neural Network
FSL	Few Shot Learning
GAN	Generative Adversarial Network
HOG	Histogram of oriented Gradients
HyBAR	Hypothesis-based Algorithm Rating

KBV	Konventionelle Bildverarbeitung
KMBV	klassische Merkmale der Bildverarbeitung
KIT	Karlsruher Institut für Technologie
LBP	Local Binary Patterns
MAE	Masked Auto Encoder
OCD	Object-based Change Detection
OSMC	Object-State-based Mapping of Changes
PX	Pixel Segmentor (strukturierter Klassifikator)
SAP	Segmentierungsaufgabenparameter
StED	Strukturierter Encoder-Decoder
STRUCT	Strukturierte Segmentierung
SVM	Support Vektor Maschine
vDCNN	vortrainiertes Deep Convolutional Neural Network

Lateinische Symbole und Variablen

a_K	Flächenanteil der Klasse K im Bild
b	Bild eines Datensatzes
F	Hypothesen-Quotient: Prozentualer Anteil an Objekten, die die Hypothese des konstanten Zustands erfüllen.
F_1	F1-Score oder Dice-Koeffizient
FP	False Positive
FN	False Negative

G	Durchschnittliche Segmentierungsgüte über alle Anzahlen an Bildern
q	Quelle eines Bildes
t	Zeit
TP	True Positive
TN	True Negative
M	Anzahl an Datensätzen
N_{img}	Anzahl an Bildern
N	Anzahl an Objekten
S_K	Art der Störfaktoren (λ_v, λ_g, λ_{v+g})
$\mathbf{X}_m$	m-ter Datensatz
x_n	Bildausschnitt des n-ten Objektes
x_n^*	Merkmale des n-ten Objektes
z_n	Zustand des n-ten Objektes
$\hat{z}_n$	geschätzter Zustand des n-ten Objektes

Griechische Symbole und Variablen

β	Streuungsverhältnis: Verhältnis der durchschnittlichen Intra-Objekt-Streuung zur durchschnittlichen Inter-Objekt-Streuung
λ	Störfaktoren (z.B. Belichtung, Schatten, wechselnde Perspektiven, etc.)
λ_v	Visuelle Störfaktoren (z.B. Belichtung, Schatten, Auflösung des Bildes, etc.)

λ_g	Geometrische Störfaktoren (z.B. wechselnde Perspektiven, verschiedene Abstände, etc.)
ϕ	Die Streuung der Zustände eines Objektes
Φ	Die durchschnittliche Intra-Objekt-Streuung
γ	Die Streuung eines Objektes relativ zu allen anderen Objekten
Γ	Die durchschnittliche Inter-Objekt-Streuung

Allgemeine Tiefindizes

n	Index des n-ten Objektes
m	Index des m-ten Datensatzes

1 Einleitung und Motivation

Mit der Hilfe von mobilen Kameras und dem Fortschritt von Computer-Vision-Technologien, wie tiefen neuronalen Netzwerken, ist es möglich, Bilddaten in bisher nie gekanntem Umfang zu erheben und zu analysieren. Jede Sekunde werden unzählige Bilder mit Smartphones, Satelliten, Auto- und Überwachungskameras aufgezeichnet. Jedes einzelne Bild eröffnet einen kurzen Einblick in die Welt und enthält Informationen, die nutzbar gemacht werden können. Die Vielzahl an Bildern und der darin enthaltenen Objekte macht eine manuelle Auswertung jedoch unmöglich und erschwert die Annotation der Bilddaten.

Bei der Erhebung von Daten durch verschiedene Sensorplattformen in diesem Umfang haben die Auswertenden kaum oder keinen Einfluss auf die Aufnahmemodalitäten. Solche Daten werden im Folgenden als unstrukturierte Daten bezeichnet. Die Aufnahmebedingungen solcher Bilder sind meist nicht dokumentiert und können stark voneinander abweichen: Zur Aufnahme der Bilder wird oft unterschiedliche Hardware verwendet, was zu unterschiedlichen Bildqualitäten und -auflösungen führt. Zudem haben wechselnde Aufnahmebedingungen, wie Wetter, Jahreszeit, Tageszeit, Position, Winkel und die Geschwindigkeit, der Kamera einen Einfluss auf die Aufnahmequalität. Ein weiteres Merkmal von unstrukturierten Bilddaten sind Ausreißer und unbekannte Daten, die unbeabsichtigt oder unwissentlich aufgenommen wurden. Eine einheitlich automatisierte Auswertung ist somit schwierig und muss von den verwendeten Algorithmen abgefangen werden.

Für eine Auswertung der Daten müssen die benötigten Informationen aus allen zur Verfügung stehenden Bildern herausgefiltert und zu einem kohärenten Gesamtbild zusammengefasst werden. Durch eine zeitliche Einordnung der einzelnen Bilder

ist es dann möglich, Veränderungen von Objekten nachzuvollziehen und zu analysieren und möglichst kommende Veränderungen zu extrapolieren. Die vorliegende Arbeit entwickelt deshalb ein Konzept zur Abbildung von Veränderungen in unstrukturierten Bilddaten: Der Zustand definierter Objekte wird über einen beliebigen Zeitraum beobachtet und entsprechende Zustandsänderungen werden quantifiziert. Ziel ist es dabei, sowohl individuelle Ereignisse zu detektieren als auch Trends in der Entwicklung von Zuständen zu bestimmen.

Nach dem hier vorgestellten Konzept lassen sich Algorithmen parametrieren, die Veränderungen über die Zeit abbilden können. Die Bewertung eines solchen Algorithmus stellt eine besondere Herausforderung dar, da nicht nur individuelle Zeitpunkte, sondern alle Bilder über die Zeit für eine Bewertung annotiert werden müssen. Um das Konzept effizient auf beliebige Problemstellungen anwenden zu können, wird eine Bewertungsmethodik eingeführt, die keine Annotationen benötigt und so eine effiziente Optimierung ermöglicht.

Außerdem sind die abzubildenden Veränderungen je nach Objekt oder Anwendungsfall unterschiedlich. Ein Konzept zur Abbildung von Veränderungen muss entsprechend mit unterschiedlichen Algorithmen kompatibel sein, die in der Lage sind, Bilddaten unter verschiedenen Gesichtspunkten auszuwerten. Diese Algorithmen müssen entsprechend leicht und effizient auf einen Anwendungsfall übertragbar sein. Daher wurden unterschiedliche Algorithmen auf ihre Dateneffizienz untersucht, sodass basierend auf dieser Arbeit Algorithmen zur Abbildung von Veränderungen unter verschiedenen Bedingungen entwickelt werden können.

Die Ergebnisse der Detektion von Veränderungen können anschließend in verschiedenen Bereichen eingesetzt werden, wie zum Beispiel in der Überwachung der Umwelt, der Planung zur Nutzung von Landflächen, dem Management von Infrastruktur und dem Katastrophenmanagement.

2 Stand der Forschung

2.1 Bilddaten

Bilder sind visuelle Repräsentationen von Szenen, Objekten oder Mustern, die durch Bildsensoren oder andere Erfassungseinrichtungen aufgenommen wurden. Dabei wird grundsätzlich zwischen Vektorgrafiken und Rastergrafiken unterschieden. Eine Vektorgrafik ist ein Grafikformat, das aus geometrischen Objekten, wie Linien, Kreisen und Polygonen zusammengesetzt ist. Dadurch ist es ermöglicht, Grafiken ohne Qualitätsverlust zu skalieren. Im Gegensatz dazu ist eine Rastergrafik ein Bildformat, das aus einer Rasterung von Pixeln besteht. Jeder Pixel repräsentiert eine diskrete Farbe. Zusammen bilden viele Pixel ein Bild. Rastergrafiken haben eine feste Anzahl von Pixeln und verlieren bei einer Skalierung an Qualität.

Die Farbe eines Pixels kann unterschiedlich codiert werden. Häufig wird das RGB (Rot, Grün, Blau) -Format verwendet. Das RGB-Format ist ein Farbraum, der Farben durch die Kombination dieser drei Grundfarben darstellt. Andere Beispiele für Farbräume sind: HSV [165], OPPONENT [28].

Bei der Aufnahme von Bilddaten spielen Störfaktoren eine maßgebliche Rolle. So kann dieselbe Szene oder dasselbe Objekt je nach Blickwinkel, Distanz und Umwelteinflüssen, wie Wetter oder Belichtung, unterschiedlich erscheinen. Störfaktoren sind immer Teil eines Bildes, jedoch kann eine strukturierte Datenerhebung Störfaktoren reduzieren, vereinheitlichen oder zumindest dokumentieren und somit eine spätere Auswertung vereinfachen. Unstrukturierte Bilddaten werden ohne die Berücksichtigung von Störfaktoren aufgenommen. So wird der Aufwand für die Erhebung der Daten minimiert.

2.2 Bildverarbeitung

Die Bildverarbeitung ist ein multidisziplinäres Gebiet, das sich mit der Verarbeitung und Analyse digitaler Bilder befasst. Mit der Verbreitung digitaler bildgebender Geräte und der zunehmenden Verfügbarkeit leistungsfähiger Computer ist die Bildverarbeitung zu einem wesentlichen Bestandteil in Bereichen, wie Medizin, Fernerkundung, Robotik und Unterhaltung geworden. Die Ziele der Bildverarbeitung sind vielfältig: Verbesserung der Bildqualität, Extraktion von Merkmalen, Klassifikation von Bildern, Detektion von Objekten und die Segmentierung von Bildern. Die Extraktion von Merkmalen transformiert ein Bild in einen repräsentativen Vektor. Die Bildklassifikation analysiert ein Bild und ordnet es einer bestimmten Klasse zu. Die Klassen sind dabei im Vorfeld festgelegt. Die Objektdetektion markiert und klassifiziert Regionen innerhalb eines Bildes. Diese Bildregionen werden als Bounding-Boxen bezeichnet und sind im Allgemeinen rechteckig. Die Bildsegmentierung ordnet jedem Pixel im Bild eine Klasse zu.

Konventionelle Bildverarbeitung (KBV) bezieht sich im Allgemeinen auf eine Reihe von manuell parametrisierten Operationen, die ein Bild deterministisch transformieren. Diese Methoden umfassen die Anwendung von Filteroperatoren, die Änderung der Bildgröße und morphologische Operationen. Diese Verfahren können beispielsweise zur Glättung von Bildern (Gauß-Filter, Mittelwertfilter) oder zur Kantendetektion (Prewitt-Filter [136], Sobel-Filter, Laplace-Filter) eingesetzt werden. Der Vorteil dieser Verfahren liegt in ihrer Geschwindigkeit, Effizienz und Nachvollziehbarkeit. Durch die vergleichsweise einfachen Rechenoperationen können Bilder schnell und effizient verarbeitet werden. Diese einfachen Operationen erlauben die manuelle Bestimmung von Parametern, die einen erklärbaren Einfluss auf die Algorithmen haben, sodass die Ergebnisse leicht interpretiert und validiert werden können.

Durch die manuelle Parametrisierung können Verfahren der KBV nur begrenzt an Störungen im Bild und unterschiedliche Aufnahmebedingungen angepasst werden. Selbstoptimierende Algorithmen sind in der Lage, Parameter anhand von Trainingsdaten zu erlernen und so den Aufwand der manuellen Parametrierung

zu reduzieren. Hier haben in den letzten Jahren Fortschritte im Bereich des maschinellen Lernens und des Deep-Learning die Bildverarbeitung revolutioniert und die Entwicklung hochpräziser und vielseitiger Algorithmen für Aufgaben wie Bildklassifikation, Segmentierung und generative Modellierung ermöglicht. Doch auch diese fortschrittlichen Algorithmen benötigen die manuelle Auswahl sogenannter Hyperparameter, die vor dem Training festgelegt werden und die Struktur und das Verhalten des Modells beeinflussen.

2.3 Bildsegmentierung

Bei der Bildsegmentierung wird jedem Pixel eine Klasse zugeordnet. Die einfachste Form der Segmentierung entspricht der Unterteilung in Vordergrund und Hintergrund (Binäre Segmentierung), während komplexe Problemstellungen zwischen mehreren verschiedenen Klassen und dem Hintergrund unterscheiden. Die Bildsegmentierung findet Anwendung in Bereichen, wie der medizinischen Bildanalyse, der Satellitenbildanalyse und der industriellen Qualitätskontrolle.

Es gibt unterschiedliche Ansätze zur Bildsegmentierung. Dazu gehören die Schwellwertbildung, die Kantendetektion, regionsbasierte Methoden, Clustering-Techniken, die graphbasierte Segmentierung oder die pixelweise Klassifikation. Bei der Schwellwertbildung werden die Pixel aufgrund ihrer Intensitätswerte relativ zu einem bestimmten Schwellenwert entweder dem Vordergrund oder dem Hintergrund zugeordnet [129]. Kantenerkennungsalgorithmen [37] identifizieren Grenzen zwischen verschiedenen Regionen in einem Bild, indem sie Änderungen in der Pixelintensität erkennen. Die Kanten werden anschließend als Kontur eines Objektes definiert und so segmentiert. Diese Prinzip wird von der Methode Active-Contour [88, 104, 166] aufgegriffen, die gezielt Splines an die detektierte Kanten anpasst.

Regionsbasierte Verfahren, wie Quickshift [67, 179] und SLIC [25, 53, 206] gruppieren Pixel zu sogenannten Super-Pixeln aufgrund von Ähnlichkeiten in Farbe,

Textur oder anderen Attributen in Regionen. Clustering-Techniken gruppieren Pixel auf der Basis der Ähnlichkeit von Merkmalen, wobei k-means-Clustering [18] und Mean-Shift-Clustering [50] aufgrund ihrer Effizienz und Flexibilität beliebte Methoden sind. Diese Super-Pixel können durch ihre Form, Farbe, Textur beschrieben und abschließend durch einen Klassifikator einer Klasse zugeordnet werden.

Bei graphbasierten Methoden, wie der Wasserscheidentransformation [23] oder der Felzenszwalb-Segmentierung [65], wird die Bildsegmentierung als ein Graphpartitionierungsproblem behandelt. Die Wasserscheidentransformation simuliert den Prozess des Wasserflusses entlang der Intensität eines Bildes und trennt Objekte effektiv auf der Basis von Intensitätsgradienten. Bei der Felzenszwalb-Segmentierung wird das Bild als Graph repräsentiert, bei dem Pixel als Knoten und Verbindungen zwischen Pixeln als Kanten dargestellt werden. Andere Verfahren segmentieren ein Bild durch eine pixelweise Klassifikation [55, 161, 183]. Hierfür werden für jeden Pixel Merkmale extrahiert, auf deren Basis ein Klassifikator trainiert wird.

2.4 Metriken für die Bildsegmentierung

Die Qualität einer Segmentierung wird auf Pixel-Ebene bewertet. Für eine Klasse A wird für jedes Pixel bestimmt, ob es richtig seiner Klasse A zugeordnet wurde (True Positive: TP), ob es falsch einer anderen Klasse B zugeordnet wurde (False Negative: FN), ob es richtig einer anderen Klasse als A zugeordnet wurde (True Negative: TN) oder ob es falsch der Klasse A zugeordnet wurde (False Positive: FP). Basierend auf den ermittelten TP, FN, TN und FP können Metriken für die Klassifikation berechnet werden.

Die Precision (Spezifität) misst den Anteil der korrekt einer Klasse zugeordneten Pixel an allen vom Segmentierungsalgorithmus als der Klasse zugehörig markierten Pixeln. Sie wird berechnet als das Verhältnis der korrekt einer Klasse

zugeordneten Pixel zur Summe der korrekt einer Klasse zugeordneten und falsch einer Klasse zugeordneten Pixel:

$$\text{Precision} = \frac{TP}{TP + FP}. \tag{2.1}$$

Recall (auch Sensitivität) misst den Anteil der korrekt einer Klasse zugeordneten Pixel unter allen einer Klasse zugeordneten Pixel Pixeln. Er wird berechnet als das Verhältnis von korrekt einer Klasse zugeordneten Pixeln zur Summe von korrekt einer Klasse zugeordneten und falsch einer Klasse nicht zugeordneten Pixeln.

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{2.2}$$

Der F1-Score (auch Dice-Coefficient) ist ein harmonisches Mittel aus Precision und Recall. So können auch Klassen mit geringem Anteil aussagekräftig bewertet werden und Klassen, die häufig vorkommen, dominieren die Metrik nicht.

$$\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}. \tag{2.3}$$

Die Accuracy (auch Intersection-over-Union) misst den Anteil der korrekt klassifizierten Pixel. Bei der Bildsegmentierung wird die Accuracy als das Verhältnis zwischen der Anzahl der korrekt segmentierten Pixel und der Gesamtzahl der Pixel im Bild berechnet. Im Gegensatz zum F1-Score berücksichtigt die Accuracy nicht die Häufigkeit der Klasse.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.4}$$

2.5 Extraktion von Merkmalen aus Bildern

Die Extraktion von Merkmalen aus Bildern hat zum Ziel, ein Bild durch einen repräsentativen, eindimensionalen Vektor zu beschreiben. Auf diese Weise werden Merkmale zweier Bilder durch die Anwendung von Metriken (Euklidische Distanz, Manhattan-Distanz, Cosinus-Ähnlichkeit, etc.) miteinander vergleichbar. Die Merkmale lassen sich nun durch Clustering-Verfahren gruppieren (k-means [18], Mean-Shift [50], ...), für die Auswahl ähnlicher Bilder aus einer Gruppe von Bildern verwenden (Image-Retrival) [105] oder in der Bildklassifikation nutzen. Solche Cluster können durch Metriken, wie den Silhouette-Score [149], bewertet oder durch Verfahren zur Dimensionsreduktion (PCA [115], t-SNE [110] oder UMAP [117]) im zweidimensionalen Raum visualisiert werden.

Methoden zur Extraktion von Merkmalen verwenden häufig Farb- und/oder Textur-Informationen, um Bilder repräsentativ zu beschreiben [89, 128, 190]. Grundlegende Merkmale eines Bildes sind beispielsweise die durchschnittliche Intensität (Mittelwert aller Pixel) oder auch Meta-Informationen, wie die Höhe und Breite des Bildes. Merkmale können sich auf interpretierbare Eigenschaften des Bildes beziehen. Komplexe Methoden zur Merkmalsextraktion berechnen abstrakte Merkmale, die Eigenschaften des Bildes widerspiegeln, aber nicht manuell interpretiert werden können.

Die Farb-Informationen eines Bild lassen sich als Histogramm der Pixelwerte repräsentiert und anschließend klassifiziert [40, 68, 121, 158]. Die Methode Local-Binary-Patterns (LBP) [127] beschreibt die Textur eines Bildes durch die binäre Codierung der Nachbarschaft aller Pixel. Für einen Pixel wird eine Nachbarschaft definiert und jeder Pixel im Nachbarschaftsfenster wird mit dem zentralen Pixel verglichen. Wenn der benachbarte Pixelwert größer oder gleich dem zentralen Pixelwert ist, wird dieser Pixel mit 1 markiert; andernfalls wird er mit 0 markiert. Aus dieser Codierung kann für jeden Pixel eine Zahl berechnet werden. Das Histogramm der Zahlen ergibt dann die Repräsentation des Bildes.

In der Veröffentlichung von Dala [51] wird die Methode Histgram-of-oriented-Gradients (HOG) verwendet, um die Textur verschiedener Bildbereiche zu repräsentieren und anschließend Personen zur lokalisieren. Hierfür werden die Kanten nach ihrer Orientierung in ein Histogramm zusammengefasst. Haar-Merkmale können verwendet werden, um Bildregionen zu beschreiben und anschließend Gesichter zu klassifizieren [182]. Die Methode Grey-Level-Covariance-Matrix (GLCM) [164] analysiert Texturen in Graustufenbildern, indem sie die Häufigkeit zählt, mit der bestimmte Paare von Grauwerten in einem festgelegten räumlichen Verhältnis zueinander auftreten. Aus der GLCM können statistische Merkmale, wie Kontrast, Homogenität, Energie und Entropie, berechnet werden, die die Textur des Bildes beschreiben.

2.6 Tiefe Neuronale Netzwerke

Tiefe Neuronale Netze (Deep Neural Network: DNN) werden dem Gebiet des Deep Learning (DL) zugeordnet. Sie sind in vielen Bereichen einsetzbar und haben die Bildverarbeitung seit 2012 mit dem Durchbruch von AlexNet [92] in der Bildklassifikation von ImageNet [151] revolutioniert. Ein DNN besteht aus Neuronen (siehe Abbildung 2.1). Ein Neuron ist seinem biologischen Vorbild nachempfunden und verarbeitet Signale. Dazu hat das Neuron mehrere Eingänge, denen jeweils ein Gewicht w zugeordnet ist. Mehrere Signale $x_1, x_2, ..., x_N$ werden so mit einem Gewicht $w_1, w_2, ..., w_N$ multipliziert und in einer nichtlinearen Funktion f (ReLu, Sigmoid, ...) zu einem Wert y zusammengefasst. Die einzelnen Neuronen können zu komplexen Netzwerken verknüpft werden, in denen die Daten mehrerer Neuronen nacheinander verarbeitet werden. Häufig sind diese Netzwerke so verschaltet, dass die Informationen nur in eine Richtung fließen (feed-forward). In diesem Fall können die Neuronen in Schichten organisiert sein. Die Art der Schichten wird dann durch die Art der Verbindungen zur vorhergehenden Schicht bestimmt. Wenn jedes Neuron einer Schicht mit allen Neuronen der folgenden Schicht verbunden ist, spricht man von einer Fully-Connected-Layer [94]. Die Aneinanderreihung solcher Schichten wird auch als

Multi-Layer-Perceptron (MLP) bezeichnet und ist beispielhaft in Abbildung 2.1 (rechts) dargestellt. Diese MLPs sind in der Lage komplexe Informationen zu verarbeiten und werden beispielsweise für die Klassifikation von tabellarischen Daten eingesetzt [54, 123, 140].

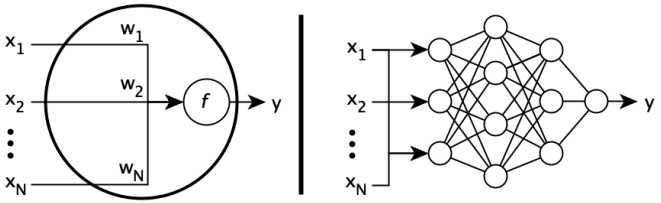


Abbildung 2.1: Der Aufbau eines Neurons (links) entspricht dem biologischen Vorbild. Unterschiedliche Signale $x_1, x_2, ..., x_N$ werden in dem Neuron durch die Gewichte $w_1, w_2, ..., w_N$ gewichtet und durch die nicht-lineare Funktion f zu einem Wert y zusammengefasst. Einzelne Neuronen lassen sich zu komplexen Netzwerken (rechts) kombinieren, die in der Lage sind Daten zu verarbeiten.

DNNs sind in der Lage, sich selbstständig und ausschließlich mit Hilfe von Trainingsdaten an unterschiedliche Aufgaben anzupassen. Dabei werden dem DNN wiederholt Beispieldaten (z.B. Bilder) mit der richtigen Lösung präsentiert und die Parameter des Netzes schrittweise angepasst, bis das DNN selbst die richtige Lösung berechnet. Dieser Prozess wird Training genannt. Während des Trainings iteriert das DNN über alle Trainingsdaten. Eine Iteration wird dabei als Epoche bezeichnet. Eine Epoche kann in einzelne Schritte zerlegt werden, wobei in jedem Schritt eine Gruppe von mehreren Trainingsdaten (Batch) für eine Anpassung verwendet wird. Die Anzahl der Trainingsdaten des Batchs ist ein wählbarer Hyperparameter. Für einen Batch berechnet die DNN eine Lösung, die mit der durch die Annotationen vorgegebenen Lösung verglichen wird. Die Abweichung wird als Loss bezeichnet. Je nach Größe des Loss wird dann eine Anpassung der Gewichte des Netzes vorgenommen. Die Veränderung wird von einem Optimierungsalgorithmus, wie Adam [91] berechnet.

Während des Trainings muss ein Kompromiss zwischen Überanpassung an die Trainingsdaten (Overfitting) und Unteranpassung (Underfitting) gefunden werden. Overfitting verhindert eine Generalisierung, da sich das DNN an zufälliges Rauschen in den Trainingsdaten anpasst, während Underfitting die Fähigkeit des Modells einschränkt, die zugrunde liegenden Muster in den Daten zu erfassen, was die Qualität der Ergebnisse reduziert. Early-Stopping wird verwendet, um das Training eines Modells zu beenden, bevor Overfitting eintritt. Für Early-Stopping wird ein Prozentsatz der Bilder in einen separaten Validierungsdatensatz überführt. Dieser Datensatz wird verwendet, um die Qualität des DNNs während des Trainings abzuschätzen. Das Training wird automatisch abgebrochen, wenn auf dem Validierungsdatensatz keine Verbesserung mehr erzielt wird.

In der Bildverarbeitung werden sogenannte Deep Convolutional Neural Networks (DCNN) und seit 2020 auch Vision Transformer (ViT) [57] in verschiedenen Bereichen der Bildverarbeitung eingesetzt. Dabei werden sie in verschiedenen Varianten für unterschiedliche Aufgaben genutzt, wie Merkmalsextraktion, Bildklassifikation [85, 135, 195, 201], Objektdetektion [145], Bildsegmentierung [31, 63, 119, 155, 192, 205] oder Objekt-Verfolgung [32, 159, 189]. Alle DCNNs verwenden einen Merkmalsextraktor (Backbone). Es gibt verschiedene vorkonfigurierte Backbones (ResNet [77], XceptionNet [48], MobileNet [79], MobileNetV2 [154], DenseNet [80], EfficientNet [170], ConvNext [107, 186]), die sich bewährt haben und für unterschiedliche Anwendungen optimiert sind. Die im Backbone extrahierten Merkmale können anschließend je nach Aufgabe weiterverwendet werden. In der Regel werden Backbones auf dem ImageNet [151] Datensatz trainiert, getestet und dann zusammen mit den trainierten Gewichten veröffentlicht. Diese Gewichte können aussagekräftige Merkmale extrahieren, auch wenn diese für den Einsatz auf ImageNet optimiert sind. Backbones, deren Gewicht ebenfalls veröffentlicht wurden, können entsprechend direkt (Transfer-Learning) oder als Startpunkt eines neuen Trainings (Fine-Tuning) wiederverwendet werden.

Das Backbone extrahiert Merkmale mit für die Bildverarbeitung spezifischen Schichten: Convolutional-Layer und Pooling-Layer. Ein Convolutional-Layer [95]

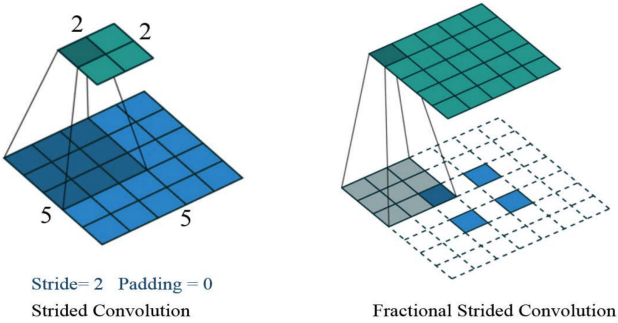


Abbildung 2.2: Visualisierung eines Convolutional-Layers: Bei einem Convolutional Layer bewegt sich der Filter (Kernel) mit der gewählten Schrittgröße (Stride) über das Bild und berechnet für jede Gruppe von Pixeln einen neuen Wert, wodurch die räumliche Auflösung in der Regel verringert wird. Eine Fractional Strided Convolution (auch als Transposed Convolution oder Deconvolution bezeichnet) wird verwendet, um die räumliche Auflösung des Bildes zu erhöhen, indem die Anzahl der Pixel in der Ausgabe erweitert wird. Diese Technik findet Anwendung in Aufgaben wie der Bildvergrößerung, der semantischen Segmentierung und der Generierung hochauflösender Bilder. [84]

dient dazu, Merkmale aus Eingabedaten, wie Bildern, zu extrahieren. Diese Merkmale sind räumlich angeordnet und werden Feature-Maps genannt. Die Extraktion geschieht durch die Anwendung von Filtern, die über die Eingabedaten verschoben werden. Ein Filter entspricht dabei einem Neuron, dessen Gewichte durch die Matrix des Filters strukturiert werden. Der Filter wird über das Eingabebild bewegt, wobei an jeder Position eine elementweise Multiplikation zwischen den Gewichten des Filters und den entsprechenden Eingabewerten durchgeführt wird. Die Ergebnisse dieser Multiplikation werden summiert und durch eine Funktion zu einem Wert y zusammengefasst. Durch die Bewegung des Filters über das Eingabebild wird für jede Position ein neuer Wert berechnet. Das Ergebnis entspricht der Feature-Map. Die Bewegung des Filters über das Eingabebild wird durch die Hyperparameter Schrittgröße (Stride) und Ergänzung von Werten am Rand (Padding) definiert. Der Stride gibt an, wie viele Pixel der Filter bei jedem Schritt verschoben wird. Padding fügt zusätzliche Pixel um das Eingabebild hinzu, um die Größe der Ausgabe zu kontrollieren und Informationen an den Rändern des Bildes zu erhalten. Ohne Padding wird die Ausgabe kleiner als das Eingabebild.

Mit Padding bleibt die Größe unverändert. Die resultierende Feature-Map enthält die extrahierten Merkmale des Eingabebildes und wird an die nächste Schicht im Netzwerk weitergeleitet. Ein Convolutional-Layer hilft somit dabei, relevante Muster und Merkmale aus den Eingabedaten zu extrahieren, wodurch das Netzwerk in der Lage ist, komplexe Bildstrukturen zu erkennen und zu verarbeiten.

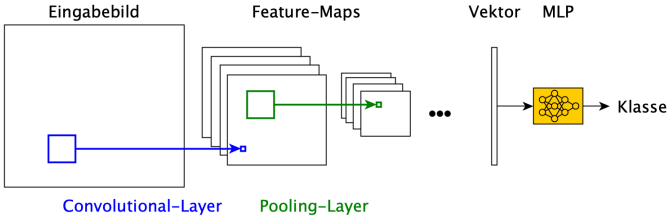


Abbildung 2.3: Aufbau eines DCNN zur Bildklassifikation: Ein Eingabebild wird in mehreren Iterationen durch Convolutional-Layer und Pooling-Layer (Backbone) verarbeitet, sodass sich die räumliche Auflösung des Bildes reduziert und eine größere Menge an Informationen extrahiert wird. Abschließend werden die Merkmale durch ein MLP klassifiziert.

Weitere Schichten, die typisch für den Bereich der Bildverarbeitung sind, sind Pooling-Layer. Diese reduzieren die räumliche Auflösung des Eingabebildes und fassen Merkmale zusammen. Pooling-Layer berechnen für ein zuvor definiertes Gitter je einen resultierenden Wert pro Gitterzelle. Das Max-Pooling-Layer wählt als resultierenden Wert das Maximum der Gitterzelle aus. Das Average-Pooling-Layer berechnet für jede Gitterzelle den Durchschnitt. Der Ablauf der Datenverarbeitung eines DCNN zur Klassifikation von Bildern ist beispielhaft in Abbildung 2.3 veranschaulicht.

Für eine Klassifizierungsaufgabe werden die Feature-Maps in einen Vektor transformiert. Dazu gibt es verschiedene Möglichkeiten: Flatten, Global-Average und Global-Maximum. Bei der Flatten-Operation werden alle verfügbaren Werte der Feature-Map aneinandergereiht, sodass die räumlichen Dimensionen der Merkmale erhalten bleiben. Die Operationen Global-Average und Global-Maximum berechnen den Mittelwert bzw. das Maximum über alle räumlichen Dimensionen.

So wird für das Beispiel ResNet50 aus einer Feature-Map 7 x 7 mit 2048 Merkmalen ein Vektor mit 2048 Merkmalen. Dadurch geht zwar der räumliche Bezug der Merkmale verloren, aber die Anzahl der Merkmale wird um ein Vielfaches reduziert. Der Vektor wird anschließend durch ein MLP klassifiziert.

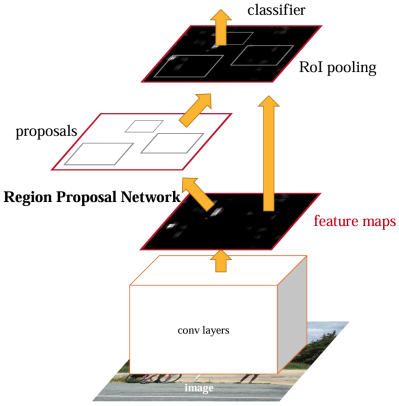


Abbildung 2.4: Visualisierung Faster-R-CNN: Das *Region Proposal Network* bestimmt im ersten Schritt alle potentiellen Bildbereiche in denen Objekte vorhanden sind ohne dabei eine Klassifizierung der Objekte vorzunehmen. Im zweiten Schritt werden diese vorgeschlagenen Bildregionen durch den *Classifier* einem Objekt zugewiesen Sowohl das *Region Proposal Network* als auch der *Classifier* greifen dabei auf dieselben Feature-Maps zurück und trainieren gemeinsam das Backbone für die Merkmalsextraktion. [145]

Die Merkmalsextraktion des Backbones kann an verschiedene Aufgaben angepasst werden, sodass das Backbone erweitert werden kann, um eine Objektdetektion durchzuführen. Grundsätzlich werden hier zwischen einstufigen (z. B. SSD [106], YOLO [143], RetinaNet [101]) und zweistufigen (Fast R-CNN [69], Faster R-CNN [145], Cascade R-CNN [36]) Verfahren zur DL-basierten Objektdetektion unterschieden. DL-basierte Objektdetektoren zerlegen ein Eingangsbild automatisch in eine Vielzahl zuvor definierter Bounding-Boxen (sogenannte Anchor-Boxen). Diese Anchor-Boxen werden anschließend in aktiv (Anchor-Box enthält eines der gesuchten Objekte) und inaktiv (Anchor-Box enthält keines der gesuchten Objekte) unterteilt. Durch die Verwendung der Anchor-Boxen wird

die Objektdetektion als Klassifizierungsproblem formuliert. Einstufige Verfahren bestimmen in einem Schritt direkt die Klasse des detektierten Objekts, während zweistufige Verfahren die Klassifikation des Objekts in einem zweiten separaten Schritt vornehmen. Das Faster-R-CNN ist in Abbildung 2.4 als Beispiel für zweistufige Objektdetektoren dargestellt.

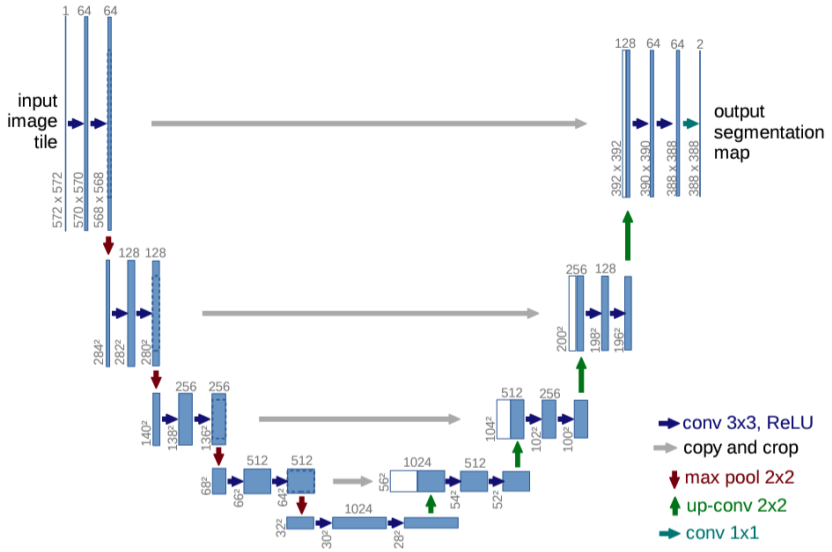


Abbildung 2.5: Aufbau DCNN zur Bildsegmentierung: Auf der linken Seit extrahiert das Backbone Merkmale während die räumliche Auflösung abnimmt. Anschließend werden Merkmale aus unterschiedlichen Auflösungen verwendet, um die ursprüngliche Größe des Bildes wieder herzustellen. [148]

Die vom Backbone extrahierten Merkmale können ebenfalls für die Bildsegmentierung eingesetzt werden. Da das Backbone die räumliche Auflösung reduziert, muss die räumliche Auflösung im Folgenden wieder hergestellt werden (vgl. Abbildung 2.5). Hierfür wird nach dem Backbone ein sogenannter Decodierer verwendet. Der Decodierer bezieht Informationen aus unterschiedlichen Teilen des Backbones und vergrößert schrittweise die Auflösung ohne Informationen zu verlieren. Typische Beispiele für solche Netzwerke sind das U-Net [148], das

SegNet [20] oder DeepLab [43, 45]. Häufig werden in diesem Kontext sogenannte Deconvolutional-Layer [126] eingesetzt (vgl. Abbildung 2.2). Nach dem Decodierer wird in einem letzten Schritt jedem Pixel von einem Convolutional-Layer eine eigene Klasse zugewiesen. Der Ansatz Mask R-CNN [76] kombiniert sogar die Detektion eines Objektes und eine anschließende Bildsegmentierung.

DCNNs benötigen aufgrund ihrer Komplexität umfangreiche Trainingsdatensätze, um sich an Problemstellungen anzupassen. Diese Komplexität führt ebenfalls zu einer mangelnden Erklärbarkeit, da die Entscheidungen innerhalb des DNNs nicht nachvollzogen werden können. Die Arbeiten [100, 102] können zwar Bildbereiche mit Entscheidungen des DCNNs verknüpfen, eignen sich jedoch nicht für den Einsatz bei der Bildsegmentierung oder Objektdetektion.

2.7 Augmentierung

Die Augmentierung [34, 162, 200] von Bilddaten ist eine weit verbreitete Technik in der Computer Vision und im maschinellen Lernen, die darauf abzielt, die Menge und Vielfalt der verfügbaren Trainingsdaten zu erhöhen, ohne zusätzliche neue Daten zu sammeln. Dies wird erreicht, indem die bestehenden Bilder durch verschiedene Transformationsmethoden modifiziert werden.

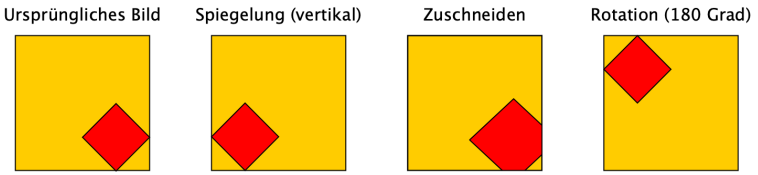


Abbildung 2.6: Augmentierung von Bilddaten: Durch verschiedene Methoden der Augmentierung können Bilder verfremdet werden, ohne dabei den Bildinhalt zu verändern. Einfach Beispiele, wie Spiegelung, Rotation und Zuschneiden, sind hier aufgeführt.

Zu diesen Transformationsmethoden gehören unter anderem Drehungen, Spiegelungen, Skalierungen, Verschiebungen, Helligkeits- und Kontraständerungen

oder Zuschneiden. Durch diese künstlich erzeugten Varianten der Originalbilder können Modelle robuster und besser generalisierbar werden. Insbesondere bei Anwendungen mit begrenzten Trainingsdaten kann die Bilddatenaugmentierung dazu beitragen, Overfitting zu reduzieren und die Leistung des Modells erheblich zu verbessern. Beispiele für einfache Methoden zur Augmentierung sind in Abbildung 2.6 visualisiert. Bei der Anwendung von Augmentierung ist zu beachten, dass die Verfremdung keinen Einfluss auf den relevanten Bildinhalt haben darf. Enthält beispielsweise die Orientierung eines Bildes wichtige Informationen, können die Methoden Spiegelung oder Rotation nicht verwendet werden.

2.8 Bag of Visual Words (BoVW)

Die Methode Bag-of-Visual-Words (BoVW) [105, 160, 193, 204] wird verwendet, um Merkmale aus Bildern zu extrahieren. Dabei wird sich an der Methode Bag-of-Words [86] aus der Sprachverarbeitung orientiert. Hier wird ein Text als Histogramm der vorkommenden Worte repräsentiert. Bilder hingegen werden durch ein Histogramm lokaler Deskriptoren beschrieben, wie SIFT [108], SURF [22], BRISK [97], ORB [150] oder KAZE [14]. Die lokalen Deskriptoren wurden ursprünglich für die Anwendung der Structure-from-Motion [157, 174, 185] entworfen.

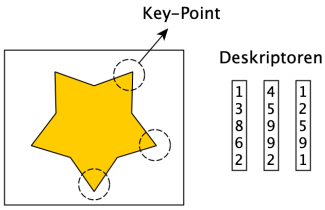


Abbildung 2.7: Berechnung lokale Deskriptoren: Zuerst werden Key-Points detektiert. Anschließend werden für jeden Key-Point Merkmale (Deskriptoren) extrahiert. Die Berechnungsmethoden unterscheiden sich, beruhen jedoch meist auf den Gradienten des Bildes.

Lokale Deskriptoren werden in zwei Schritten berechnet: Definition von sogenannten Key-Points und die Berechnung eines Deskriptors für jeden Key-Point (vgl. Abblidung 2.7). Key-Points entsprechen Bildregionen. Diese können unterschiedliche Größen, Positionen und sogar Orientierungen haben. Die Definition der Key-Points kann mit bereits existierenden Methoden von lokalen Deskriptoren oder mit einem vordefinierten Raster (Dense-Sampling [176]) erfolgen. Ein Deskriptor ist ein Vektor, der den Key-Point charakterisiert. Für ein Bild wird so ein Satz von lokalen Deskriptoren berechnet. Die Berechnung der lokalen Deskriptoren ist statisch und einstricht einer KBV. In einem zweiten Schritt kann der BoVW-Ansatz, vergleichbar mit einem DCNN, an einen Trainingsdatensatz angepasst werden. Der Ablauf des Trainings und der Anwendung des BoVW-Ansatzes ist in Abbildung 2.8 dargestellt.

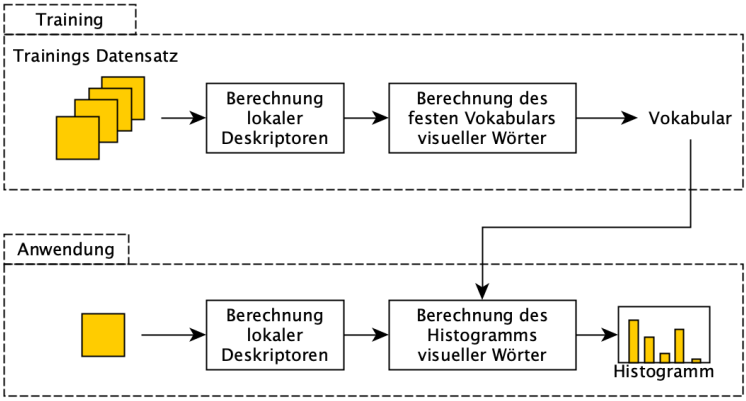


Abbildung 2.8: Konzept BoVW: Aus einem Trainingsdatensatz von Bildern werden Deskriptoren extrahiert. Aus diesen wird im Trainingsprozess ein festes Vokabular von visuellen Wörtern erstellt. In der Anwendung werden aus einem Bild erneut Deskriptoren extrahiert, die dann Vokabeln aus dem festen Vokabular zugeordnet werden. Ein Bild wird dann als Histogramm der vorkommenden visuellen Wörter repräsentiert.

Während des Trainings werden für alle Bilder des Trainingsdatensatzes lokale Deskriptoren berechnet. Das feste Vokabular visueller Wörter wird durch Clustering-Techniken, wie k-means [18], aus allen Deskriptoren des Trainingsdatensatzes gewonnen. Für die Berechnung des Vokabulars werden keine Annotationen zu den

Bildern benötigt, sodass diese Methode unüberwacht angewendet werden kann. In der Anwendung werden die lokalen Deskriptoren eines Bildes einem visuellen Wort des festen Vokabulars zugeordnet. Die visuellen Wörter werden dann als Histogramm dargestellt. Dieses Histogramm spiegelt die Verteilung der visuellen Muster im Bild wider und dient als repräsentativer Merkmalsvektor des Bildes.

2.9 Dateneffiziente Methoden der Bildverarbeitung

Die Dateneffizienz von Bildverarbeitungsalgorithmen bezieht sich auf das Verhältnis der benötigten Datenmenge zur erreichten Leistung eines Algorithmus. Je weniger Daten ein Algorithmus benötigt, desto höher ist dessen Dateneffizienz. Die Dateneffizienz lässt sich unter zwei Aspekten bewerten: Zum einen die absolute Anzahl der benötigten Bilder und zum anderen die Anzahl der annotierten Bilder, um ein Verfahren anzuwenden. So lassen sich Algorithmen zur Extraktion von klassischen Merkmalen der Bildverarbeitung (KMBV), wie Farb-Histogramme [158], LBP [127] oder HOG [51], GLCM [164], und vortrainierte DCNNs (vDCNN) [81, 151] häufig ohne im Vorfeld verfügbare Daten einsetzen.

In der nächsten Abstufung der Dateneffizienz gibt es Methoden, die Merkmale auf nicht-annotierten Bilddaten anpassen. In diesem Fall sind zwar keine aufwändigen Annotationen notwendig, jedoch werden größere Mengen von nicht-annotierten Daten vorausgesetzt. Nach der Anpassung können die angepassten Merkmale entweder direkt oder als Startpunkt für weitere Anpassung verwendet werden. Mit der Methode BoVW lassen sich unterschiedliche KMBV und lokale Deskriptoren durch die Berechnung eines visuellen Wörterbuchs anpassen. Ein Auto-Encoder (AE) [184] aus dem Bereich DL reduziert ein Bild auf eine begrenzte Anzahl von Merkmalen (Latente Repräsentation) und verwendet diese anschließend, um dasselbe Bild wieder zu rekonstruieren. In einer Anwendung wird die latente Repräsentation eines Bildes als Merkmalsvektor verwendet. Ein Masked-Auto-Encoder (MAE) [75] rekonstruiert ebenfalls ein Bild, verwendet als Eingabe aber nur einen Bruchteil des ursprünglichen Bildes. Auf diese Weise lernt das zugrundeliegende

Netz, ein Bild zu vervollständigen. Das Lernen von Unterschieden (Constrastive-Learning), wie SIMCLR [46], NNCLR [60] und Barlow-Twins [196], trainiert DCNNs auf Bilddatensätzen ohne Annotationen. Dazu werden alle Bilder eines Datensatzes durch Augmentierungen künstlich verfremdet. Im augmentierten Datensatz existieren nun verschiedene Versionen desselben Bildes. Mit diesen Daten können DNNs trainiert werden, die verschiedene Versionen eines Bildes gruppieren und diese gleichzeitig im Merkmalsraum von anderen Bildern unterscheiden. Die DL-basierten Verfahren des unüberwachten Lernens benötigen jedoch sehr große Datensätze und Rechenkapazitäten. Zudem sind Constrastive-Learning Verfahren komplex in der Anwendung, da sie stark von den gewählten Hyperparametern abhängig sind.

Im Bereich des überwachten Lernens werden keine Merkmale extrahiert, sondern Aufgaben, wie Bildklassifikation, Objektdetektion oder Bildsegmentierung, gelöst. Hier ist die Anzahl der benötigten annotierten Bilder entscheidend für eine Beurteilung der Dateneffizienz. Es gibt verschiedene Strategien im maschinellen Lernen mit limitierten Trainingsdaten umzugehen. Few-Shot-Learning (FSL) hat zum Ziel, Algorithmen mit einer limitierten Anzahl an Trainingsdaten pro Klasse zu trainieren. Hier wird konkret die Kombination aus Anzahl der Klassen und Anzahl der Trainingsbilder pro Klasse untersucht. Diese Methoden lassen sich im Besonderen dann einsetzten, wenn es viele verschiedene Klassen mit nur wenigen Beispielen gibt, sodass der Datensatz an sich immer noch viele Daten enthält. Eine der bekanntesten Methoden ist das Meta-Learning (MAML [66], REPTILE [124]), auch bekannt als „Learning to Learn“. Dabei wird das Modell auf eine Vielzahl an Aufgaben trainiert, um erlerntes Wissen aus dem Optimierungsprozess auf die neue Aufgabe zu übertragen. Transfer-Learning kann ebenfalls im Kontext von FSL eingesetzt werden und versucht im Gegensatz zu Meta-Learning, netzwerkinternes Wissen auf neue Aufgaben zu übertragen ohne den Optimierungsprozess anzupassen [78]. Prototype-Networks [56, 191] und Matching-Networks [111, 181] sind weitere FSL-Methoden, die ein Bild anhand zuvor abgespeicherter Prototypen oder einer gespeicherten Gruppe von Bildern klassifizieren. Ein weiterer Ansatz nutzt Generative Adversarial Networks (GAN) [35, 30, 70], um Bilddaten zusammen mit Annotationen zu synthetisieren,

und vervielfältigen damit die vorhandenen Daten. In diesem Fall müssen ebenfalls bereits nicht annotierte Daten zur Verfügung stehen. Die hier verwendeten Netzwerke sind komplex und aufwändig im Training und ebenfalls anfällig für Artefakte.

Aktuelle Methoden des unüberwachten Lernens benötigen große Datensätze und erhebliche Rechenressourcen, um die Hyperparameter der Methoden zu bestimmen und die Methoden zu trainieren. FSL-Methoden spezialisieren sich zwar auf kleinere Datensätze mit Annotationen, sind aber ebenfalls auf eine aufwändige Hyperparameter-Optimierung angewiesen. Auch Transfer-Learning ist nicht immer anwendbar, wenn keine relevanten Datensätze mit Domänenbezug vorliegen. Insbesondere wenn Algorithmen mit unbekannten und nicht verifizierten Daten trainiert werden sollen, sind die vorgestellten Methoden nicht einsetzbar. Dies wirft die Frage nach einfachen überwachten Modellen auf, die direkt mit möglichst wenigen Bildern trainiert werden können und somit recheneffizient und mit geringem Aufwand einsetzbar sind. Verschiedene wissenschaftliche Arbeiten [12, 16, 26, 29, 71, 120, 125, 180] vergleichen Segmentierungemethoden in Bezug auf die benötigten Trainingsdaten. Dabei werden jedoch stets die aktuellen Datensätze der Arbeit oder Methoden zur Auswahl der zu annotierenden Bilder bewertet. So wird kein allgemeiner Vergleich bezüglich der Dateneffizienz der Algorithmen vorgenommen. Es fehlt eine Übersichtsarbeit, die herkömmliche Segmentierungsmethoden entsprechend ihrer Dateneffizienz einordnet.

2.10 Abbildung von Veränderungen

Seit Daten großflächig erhoben und gespeichert werden, werden diese ebenfalls auf Veränderungen untersucht. Dies erlaubt die Erkennung von Trends und Mustern, die Überwachung und Frühwarnung von Prozessen, die Optimierung von Prozessen, die Verbesserung von Vorhersagemodellen und die Anpassung von Prozessen an dynamische Umgebungen. In dieser Arbeit wird sich auf die Abbildung von Veränderungen (ABV) in Bilddaten und im Besonderen von unstrukturierten Bilddaten fokussiert. Die ABV entspricht einer Quantifizierung

der Veränderung eines beliebigen Objektes zwischen zwei Zeitpunkten. Hierfür muss der Zustand von Objekten vergleichbar erfasst werden und ein Vergleich zwischen zwei Zeitpunkten durchgeführt werden. Der Zustand eines Objektes wird dabei als eine Sammlung relevanter Merkmale und Eigenschaften definiert, die dessen Erscheinungsbild und Beschaffenheit zu einem bestimmten Zeitpunkt beschreiben.

Veränderungen in Bilddaten sind vielfältig und lassen sich auf verschiedene Arten abbilden. Eine Möglichkeit besteht im direkten Vergleich zweier Bilder und wird als Change-Detection (CD) bezeichnet [33, 137]. Die CD vergleicht zwei Bilder und markiert veränderte Regionen mittels einer Maske. Für die Evaluierung von CD-Methoden gibt es verschiedene Datensätze im Bereich der Fernerkundung (Remote-Sensing) [42, 93, 194, 199] und der urbanen Szenen [153], bei denen Masken mit markierten Veränderungen für jedes Bildpaar manuell annotiert wurden. Der direkte Vergleich von zwei Bildern lässt sich ebenfalls auf Videos übertragen, in denen jeweils zwei Videoframes miteinander verglichen werden [112, 131]. Die Methoden der CD werden in [90] grundsätzlich in überwacht und unüberwacht unterteilt. Die überwachte CD verwendet einen Trainingsdatensatz mit mehreren annotierten Bildpaaren. Unüberwachte CD-Methoden sind häufig einfache pixelbasierte Verfahren, die die direkte Differenz zweier Bilder nutzen und so Veränderungen ohne annotierte Daten erkennen [49, 59]. Ähnliche Methoden bilden Differenzen aus zuvor extrahierten, pixelbasierten Merkmalen ab [163]. Überwachte Methoden können sich an Daten anpassen und sind häufig DL basiert [15, 39, 64, 72, 122]. Hier werden sog. Siamese-Networks verwendet. Diese verarbeiten parallel zwei Bilder mit demselben Backbone, vergleichen anschließend die extrahierten Merkmale und erzeugen eine Maske mit markierten Veränderungen zwischen den Bildern.

Die hier beschriebene CD benötigt keine zusätzlichen Informationen über die Art der Veränderung, kann jedoch spezifische Veränderungen nicht voneinander unterscheiden und ist damit anfällig für irrelevante Veränderungen, wie Belichtung und Schatten. Darüber hinaus ist die Methode CD in sich begrenzt, da bestimmte Arten von Veränderungen, wie z.B. der Alterungsprozess eines Menschen, nicht durch veränderte Bildbereiche abbildbar sind. Außerdem ist für die Anwendung

eine Übereinstimmung der Perspektive beider Bilder zwingend notwendig, da sonst kein Vergleich durchgeführt werden kann. In der Arbeit von Pollard [134] werden unterschiedliche Perspektiven durch eine Projektion der Pixel in den dreidimensionalen Raum und anschließende Registrierung gelöst. Dies ist jedoch nur möglich, wenn die Positionen und Orientierungen der Kameras genau bekannt sind. Bei einer Anwendung sind diese Informationen häufig nicht verfügbar oder ungenau. Daher ist CD für eine ABV von unstrukturierten Bilddaten nicht ausreichend.

In der Publikation von Chen [41] wird die Object-based-Change-Detection (OCD) eingeführt. Dabei werden gezielt Objekte erkannt und durch Merkmale beschrieben. Objekte sind hier abstrakte Super-Pixel [187, 198] (Eine Gruppe von Pixeln mit ähnlichen Eigenschaften, wie Farbgebung, Intensität oder Textur) oder bekannte und gezielt segmentierte Regionen [96]. Die Objekte lassen sich dann durch Form- oder Texturmerkmale beschreiben und statistisch auf Veränderungen untersuchen. Die OCD wird in den Arbeiten [96, 187, 198] ebenfalls nur zur Erstellung einer Maske der Veränderungen zwischen zwei Bildern analog zur CD verwendet. Grundsätzlich hat die OCD jedoch das Potential, unterschiedliche Betrachtungswinkel und Entfernungen auszugleichen und muss noch für die Anwendung auf unstrukturierte Bilddaten konzipiert und realisiert werden.

In der Veröffentlichung von O'Mahony [130] wird die Verwendung einer Merkmalsextraktion zur ABV diskutiert. Die bereits eingeführten Methoden zur Extraktion von Merkmalen, wie KMBV (Farb/Textur), BoVW, DL-basiertes Contrastive-Learning (CL), AE, MAE, berechnen miteinander vergleichbare Merkmale. Die Ähnlichkeit solcher Merkmale kann mittels Metriken quantifiziert und als Maß für eine Veränderung betrachtet werden. Hierfür sind im Allgemeinen die folgenden Metriken anwendbar: Euklidische Distanz, Manhattan Distanz oder Cosinus Ähnlichkeit. Analog lassen sich ebenfalls überwachte Bildverarbeitungsmethoden (Bildklassifizierung, Objektdetektion oder Bildsegmentierung) für eine ABV anwenden. So kann die Veränderung der zugeordneten Klasse, ein neu detektiertes Objekt oder eine Veränderung der Segmentierungsmaske als Veränderung interpretiert werden. Es fehlt eine Übertragung der unterschiedlichen Methoden auf die ABV in unstrukturierten Bilddaten.

2.11 Offene Probleme

In dieser Arbeit wird die Problemstellung der Abbildung von Veränderungen in unstrukturierten Bilddaten bearbeitet. Aus dem Stand der Forschung ergeben sich offene Probleme, die wie im Folgenden untersucht werden:

- Die Abbildung von Veränderungen in unstrukturierten Bilddaten stellt aufgrund der Störfaktoren (z.B. Aufnahmemodalitäten, Ausreißer und unbekannte Daten) sowie nicht eindeutig definierbarer Zustände und Veränderungen eine Herausforderung dar. Es gibt kein Konzept zur ABV, das sich auf unstrukturierte Daten übertragen lässt.
- Sind die sich verändernden Eigenschaften von Objekten bekannt, können Algorithmen zur Quantifizierung dieser Eigenschaften trainiert werden. Die Annotation von Daten ist zeitaufwändig und damit kostenintensiv. Gerade bei unstrukturierten Datensätzen gibt es viele unterschiedliche Objekte, die ausgewertet werden müssen. Es ist unklar, welche überwachten Methoden leicht, flexibel und dateneffizient für die Quantifizierung von Objekteigenschaften eingesetzt werden können.
- Aktuelle Ansätze für die Abbildung von Veränderungen basieren auf einem direkten Vergleich zwischen zwei Bildern. Dies limitiert ihre Anwendbarkeit, wenn Veränderungen sich nicht durch die Veränderung eines einzelnen Bildbereichs abbilden lässt, wie der Alterungsprozess einer Person.
- Datensätze im Bereich der Change-Detection bestehen im Allgemeinen aus zwei Bildpaaren und einer dazugehörigen Maske von veränderten Bereichen. Aspekte, wie unterschiedliche Blickwinkel, redundante / einander ergänzende Bilder und mehr als ein Zeitpunkt, werden nicht betrachtet. Es gibt keinen Referenzdatensatz, der alle Eigenschaften unstrukturierter Daten vereint und eine einheitliche Bewertung von Algorithmen zur Detektion von Veränderungen ermöglicht.

- Die Annotation von Datensätzen zur Evaluierung der Abbildung von Veränderungen sind besonders aufwändig, da Datenpunkte jeweils zu mehreren Zeitpunkten annotiert werden müssen. Es fehlen Ansätze, die eine Bewertung der Abbildung von Veränderungen mit minimalen Annotationsaufwand ermöglichen.

2.12 Zielsetzung

Diese Arbeit bearbeitet die folgenden wissenschaftlichen Aufgaben:

1. Es wird ein allgemeines Vorgehen entwickelt, das als Anleitung für den Entwurf eines Algorithmus zur Abbildung von Veränderungen in beliebigen Anwendungen fungiert.

2. Die Arbeit definiert ein Konzept „Object-State-based Mapping of Changes“ *OSMC* für die Abbildung von Veränderungen in unstrukturierten Bilddaten, welches als Blaupause für einen Algorithmus zur Abbildung von Veränderungen dient.

3. Die Methodik „Hypothesis-Based-Algorithm Rating“ *HyBAR* wird zur Bewertung verschiedener Varianten von Algorithmen zur Abbildung von Veränderungen eingeführt, um das vorgestellte Konzept ohne manuelle Annotationen optimal zu parametrieren.

4. Es werden verschiedene mit dem Konzept kompatible Algorithmen unter Berücksichtigung ihrer Dateneffizienz ausgewählt. Die überwachten Algorithmus-Komponenten werden zusätzlich hinsichtlich ihrer Dateneffizienz in verschiedenen Szenarien bewertet, sodass eine Übertragbarkeit auf neue Problemstellungen mit minimalem Annotationsaufwand gewährleistet ist.

5. Die ausgewählten Algorithmus-Komponenten des Konzepts *OSMC* werden modular als Programmpakete zur Verfügung gestellt, um die Übertragbarkeit auf neue Problemstellungen zu garantieren.

6. Für die Evaluation von Algorithmen zur Abbildung von Veränderungen in unstrukturierten Bilddaten wird der „Road State Change“ Datensatz als Referenzdatensatz eingeführt. Dieser Datensatz stellt alle Herausforderungen unstrukturierter Daten dar und ermöglicht einen Vergleich der Leistungsfähigkeit verschiedener Algorithmen.

7. Zur Veranschaulichung wird das allgemeine Vorgehen in dieser Arbeit auf zwei für die Abbildung von Veränderungen relevante Anwendungsfälle (*Fahrzeug-Bilddaten* / *Remote Sensing*) übertragen.

3 Konzept

3.1 Übersicht

In diesem Kapitel wird erstmals das allgemeine Konzept „Object-State-based Mapping of Changes“ *OSMC* zur ABV von unstrukturierten Bilddaten erarbeitet, das von Anwendern als Richtlinie für neue Anwendungen genutzt werden kann. Zusätzlich werden ausgewählte dateneffiziente Methoden und die neue Bewertungsmethodik „Hypothesis-Based-Algorithm Rating“ *HyBAR* bereitgestellt, die sich nahtlos in das Konzept *OSMC* integrieren lassen und so die Übertragung auf neue Anwendungen vereinfachen. Das Konzept *OSMC* greift die vorangegangenen Arbeiten von [1, 7] zum Thema der ABV in unstrukturierten Bilddaten auf. Die ABV, wie in Abschnitt 2.10 beschrieben, hat zum Ziel die Veränderung des Zustands eines Objektes über die Zeit zu quantifizieren.

Ziel ist es, die ABV zu standardisieren und eine Anwendung auf beliebige Problemstellungen zu unterstützen (siehe Abbildung 3.1). Zunächst wird die Problemstellung als Rahmen für das weitere Vorgehen definiert. Dabei werden die zu beobachtenden Objekte und die unstrukturierten Bilddaten beschrieben sowie der Zustand und die Veränderungen der Objekte modelliert. Anschließend wird das allgemeine Konzept *OSMC* vorgestellt und erläutert, das als Vorlage für den Entwurf eines Algorithmus zur ABV in unstrukturierten Bilddaten dient. Der Algorithmus zur ABV erfasst den Zustand von Objekten im Verlauf der Zeit und ermöglicht durch den Vergleich der Zustände eine ABV. Die Definition der Problemstellung und der Entwurf eines allgemeinen Algorithmus zur ABV dienen

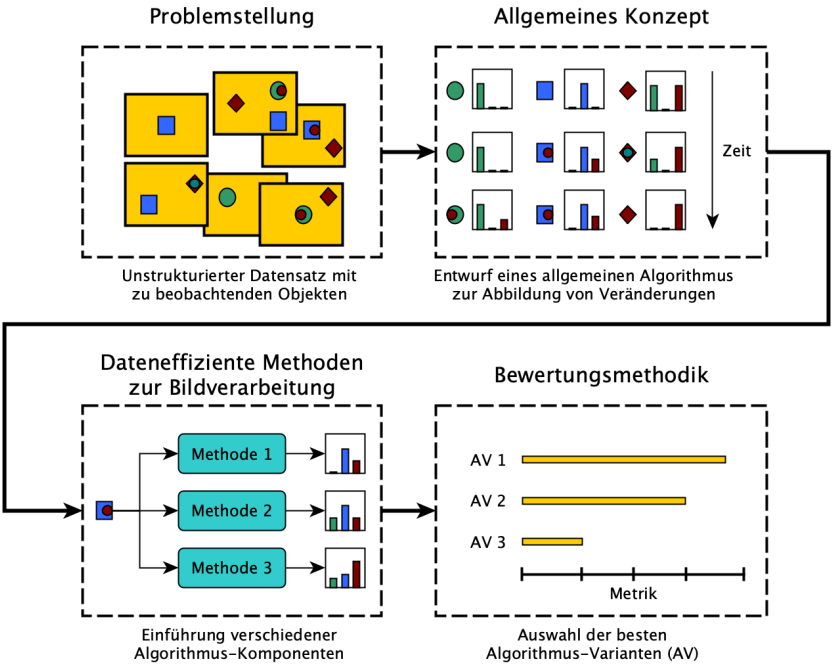


Abbildung 3.1: Vorgehen zur Abbildung von Veränderungen: In dieser Arbeit werden das allgemeine Konzept *OSMC* zur ABV und kompatible Algorithmus-Komponenten eingeführt, die es einem Anwender ermöglichen unterschiedliche Algorithmus-Varianten zu entwerfen und diese zu testen. Abschließend kann die optimale Variante mittels der vorgestellten Bewertungsmethodik *HyBAR* für den Einsatz ausgewählt werden.

als Rahmen für den Anwender bei der Übertragung auf eine neue Problemstellung. Dafür wird auch eine Datenstruktur implementiert, die eine Verarbeitung im Sinne des allgemeinen Konzepts unterstützt.

Zusätzlich werden verschiedene dateneffiziente Methoden zur Bildverarbeitung als Algorithmus-Komponenten vorgestellt, die als Bausteine im allgemeinen Konzept *OSMC* verwendet werden können. Dabei werden sowohl aktuelle Bildverarbeitungsmethoden bewertet als auch eigens entwickelte Algorithmen präsentiert. Die Implementierung der ausgewählten Algorithmen ist mit der Datenstruktur

kompatibel und lässt sich nahtlos in das allgemeine Konzept *OSMC* integrieren. Aus diesem allgemeinen Konzept und den Algorithmus-Komponenten können verschiedene konkrete Algorithmus-Varianten (AV) entworfen werden. Zudem kann der Anwender eigene, problembezogene Komponenten in das allgemeine Konzept *OSMC* integrieren und mit anderen Algorithmus-Komponenten kombinieren, was zu einer Vielzahl unterschiedlicher AV führt. Um die ideale AV für den Einsatz auszuwählen, wird abschließend die Bewertungsmethodik *HyBAR* vorgestellt. Die Bewertungsmethodik *HyBAR* ermöglicht einen effizienten Vergleich verschiedener AV für eine konkrete Anwendung.

3.2 Definitionen

Zuerst werden die grundlegenden Begrifflichkeiten definiert, auf deren Basis das vorgestellte Konzept *OSMC* aufbaut.

Objekt: Ein Objekt entspricht einem einzelnen Gegenstand oder Bereich. Mögliche Objekte aus dem Bereich der Infrastruktur, deren Veränderungen beobachtet werden kann, sind Verkehrszeichen, Straßenbepflanzung oder Straßenoberflächen. Es muss von den Bilddaten aufgezeichnet worden sein und zu unterschiedlichen Zeitpunkten wieder identifiziert werden können. Dies ermöglicht eine flexible Anwendung des vorgestellte Konzepts, solange ein Datensatz, der die entsprechenden Informationen enthält, vorliegt. Das Konzept *OSMC* zur ABV berücksichtigt auch Objekte, die durch ihre Größe nur partiell auf Bildern sichtbar sind.

Zustand und Veränderung: Der Zustand $z(t)$ eines Objektes zum Zeitpunkt t entspricht einer Ansammlung von Merkmalen oder Eigenschaften und wird in dieser Arbeit als Vektor repräsentiert. Der Zustandsvektor $z(t)$ markiert so eine Position des Objektes im Merkmalsraum. Auf diese Weise kann die Veränderung Δz eines Objektes zwischen zwei Zeitpunkten t_1 und t_2 als Abstand der Zustände definiert werden:

$$\Delta z = |z(t_2) - z(t_1)|. \tag{3.1}$$

Eine kontinuierliche Abbildung der Veränderung als Differential des Zustands über die Zeit wird nicht angestrebt, da durch die Verwendung von Bildern automatisch diskrete Zeitpunkte vorgegeben werden. Der Zustand kann je nach den aktuellen Anforderungen modelliert werden. So sind die Anzahl und Art der berücksichtigten Eigenschaften durch den Anwender frei wählbar. Eigenschaften können dabei binär sein (zum Beispiel ist das Verkehrszeichen beklebt/nicht beklebt, die Blume blüht/blüht nicht), diskret (zum Beispiel die Anzahl von Rissen in einem Straßenabschnitt) oder kontinuierlich (zum Beispiel die Länge der Risse im Straßenabschnitt). Der Zustand $z(t)$ kann auch abstrakt modelliert werden, sodass die einzelnen Aspekte des Vektors unbekannt sind, solange die einzelnen Aspekte zueinander vergleichbar sind. Dieses Vorgehen ist flexibel und kann mit Bildgruppen, unterschiedlichen Perspektiven und bekannten/unbekannten Zuständen angewendet werden. Bisherige Methoden aus dem Bereich der CD beschränken sich auf den direkten Vergleich von zwei Bildern [33, 41, 137].

Unstrukturierte Bilddaten: Unstrukturierte Bilddaten, wie in Abschnitt 2.1 beschrieben, zeichnen sich durch unstrukturierte Aufnahmeparameter aus. Dies führt zu inkonsistenten und unbekannten Störfaktoren λ. Im Gegenzug sind die Anforderungen an die Aufnahmeparameter gering, was die großflächige Datenerhebung vereinfacht. Kameras können ohne spezielle Kalibrierung flexibel eingesetzt werden und der Aufwand zur Vorbereitung der Datenerhebung ist minimal. Unstrukturierte Bilddaten sind aufgrund der inkonsistenten Störfaktoren für eine ABV eine besondere Herausforderung. Auf diese Weise wird ein Vergleich zwischen zwei Zeitpunkten erschwert, da sich sowohl das Objekt als auch die Störfaktoren unabhängig voneinander verändert haben. Dies wird in Abbildung 3.2 (links) veranschaulicht. Das Objekt (roter Kreis) erscheint im Bild je nach Belichtung (heller/dunkler) und Position der Kamera (groß/klein und elliptisch) anders. Die tatsächliche Veränderung (zweiter, kleiner Kreis) findet unabhängig statt. In dieser Arbeit werden zwei Arten von Störfaktoren unterschieden: visuelle Störfaktoren (Belichtung, Wetter, Jahreszeit, Schärfe) und geometrische Störfaktoren (Distanz, Blickwinkel). Vorangegangene Arbeiten aus dem Bereich der CD (siehe Abschnitt 2.10) können ausschließlich visuelle Störfaktoren kompensieren. Das Konzept *OSMC* zur ABV kann an alle Arten von Störfaktoren angepasst werden.

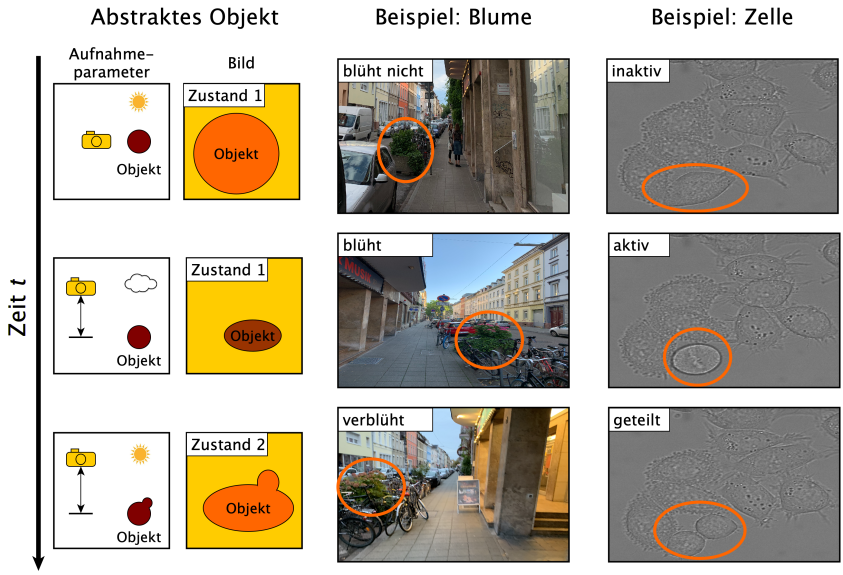


Abbildung 3.2: Unstrukturierte Bilddaten: Objekte in unstrukturierte Bilddaten werden konzeptionell in der linken Spalte dargestellt. Die Aufnahmeparameter von unstrukturierten Bilddaten sind inkonsistent. Unterschiede in der Belichtung führen zu Verfärbungen und unterschiedliche Kamerapositionen verformen und verkleinern das Objekt im Bild. Der Zustand des Objektes verändert sich unabhängig von den Störfaktoren. Als konkretes Beispiel wird hier eine Blume aus [7] aufgegriffen, die aus verschiedenen Entfernungen und Blickwinkeln betrachtet werden kann. Ein weiteres Beispiel sind Zellen [116], die sowohl ihren Zustand als auch ihre Form unabhängig voneinander ändern.

3.3 Allgemeines Konzept OSMC

Im Folgenden wird das allgemeine Konzept „Object-State-based Mapping of Changes“ *OSMC* zur ABV in unstrukturierten Bilddaten vorgestellt (Abbildung 3.3). Das Konzept dient als Vorlage für das Entwerfen von Algorithmen zur ABV. Bevor ein Algorithmus zur ABV entworfen werden kann, müssen Objekte und deren Veränderung definiert werden. Unstrukturierte Daten werden in M voneinander unabhängigen Serien erhoben. Eine Serie entspricht einem Datensatz $\mathbf{X}_m$ mit $m \in \{1, ..., M\}$ und besteht aus mehreren einzelnen Bildern $b(t)$. Jedes Bild ist

zu einem Zeitpunkt t aufgenommen worden und ist durch zeitabhängige Störfaktoren $\lambda(t)$ beeinträchtigt. Zu jedem Bild $b(t)$ können zusätzliche Metadaten (Position, Aufnahmeparameter, Zeitstempel) gespeichert werden.

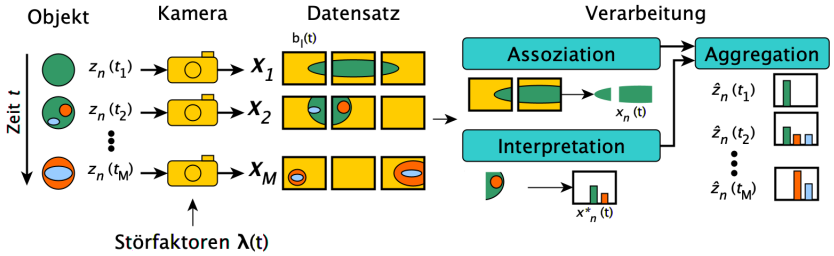


Abbildung 3.3: Konzept *OSMC* zur Abbildung von Veränderungen: Das n-te Objekt (farbiger Kreis) mit einem Zustand $z_n(t)$ (Färbung) wird zu unterschiedlichen Zeitpunkten von Kameras aufgezeichnet. Eine Kamera zeichnet einen durch zeitabhängige Störfaktoren $\lambda(t)$ beeinträchtigten Datensatz $\mathbf{X}$ auf, der aus mehreren Bildern $b(t)$ besteht. Für jedes Bild $b(t)$ müssen zunächst die relevanten Bildregionen $x_n(t, \lambda)$ dem n-ten Objekt zugeordnet werden (Assoziation). Anschließend müssen quantifizierbare Merkmale $x^*_n(t)$ aus der Region $x_n(t)$ extrahiert werden (Interpretation). Im letzten Schritt werden die einzelnen Informationen $x^*_n(t)$ zu einem Objektzustand $\hat{z}_n(t)$ zusammengefasst (Aggregation). Die Veränderung eines Objekts lässt sich nun anhand des direkten Vergleichs zweier Zustände $\hat{z}_n(t_1)$ und $\hat{z}_n(t_M)$ abbilden.

Es werden N Objekte definiert, deren Veränderung abgebildet werden soll. Ein Objekt hat einen Zustand $z_n(t)$ mit $n \in \{1, ..., N\}$, der sich in Abhängigkeit von der Zeit verändert. Die ABV des Objektes wird in drei Verarbeitungsschritten durchgeführt: Assoziation, Interpretation und Aggregation. Ein unstrukturierter Datensatz $\mathbf{X}_m$ enthält $N_m^{\mathbf{X}}$ Bilder, wobei jedes Bild $b_l(t)$ mit $l \in \{1, ..., N_m^{\mathbf{X}}\}$ das Objekt ganz, teilweise oder gar nicht zeigt. Die Assoziation hat zum Ziel, relevante Regionen in Bildern $x_n(t)$ dem entsprechenden Objekt zuzuordnen. Zuerst werden alle Bilder entfernt, die das Objekt nicht zeigen. Dabei werden zum Beispiel Bilder anhand von Randbedingungen aussortiert, die an einem anderen Ort aufgenommen wurden. Anschließend werden die Bilder ausgewertet. Mit Algorithmen können Regionen $x_n(t)$ in einem Bild markiert werden, die das Objekt zeigen. Dieser Verarbeitungsschritt ist essentiell für die Verarbeitung von unstrukturierten Daten, wird aber in bisherigen Arbeiten zum Thema CD entweder

ignoriert und als gegeben vorausgesetzt [15, 39, 64, 72, 122] oder durch bekannte Aufnahmeparameter ausgeglichen [134].

Eine dem Objekt zugeordnete Region $x_n(t)$ bildet nicht ausschließlich den Zustand $z_n(t)$ des Objektes ab, sondern kann durch verschiedene Störfaktoren $\lambda(t)$ oder andere, jedoch nicht relevante, Objektzustände überlagert werden. Die Interpretation hat zum Ziel, den Einfluss der Störfaktoren λ zu reduzieren und für jede Region $x_n(t)$ quantifizierbare Merkmale $x_n^*(t)$ zu berechnen. Diese Merkmale $x_n^*(t)$ können bereits als Objektzustand $\hat{z}_n(t)$ interpretiert werden.

Die Interpretation kann von unüberwachten Merkmalsextraktoren (z.B. BoVW, vDCNNs, etc.) durchgeführt werden. Diese extrahieren aus einer Bildregion $x_n(t)$ Merkmale $x_n^*(t)$. Diese Merkmale lassen sich miteinander vergleichen und als Maßstab zur Abbildung von Veränderungen verwenden. Je nach Methode kann auf die abstrakten Merkmale Einfluss genommen werden. BoVW-Methoden können beispielsweise durch ihre Parametrierung angepasst werden und so Farben oder Texturen gezielt ignorieren oder fokussieren.

In Anwendungsfällen, in denen die Art der Veränderung bekannt ist und Trainingsdaten für Algorithmen zur Verfügung stehen, können überwachte Methoden zur Interpretation eingesetzt werden. Hierfür werden Algorithmen gezielt trainiert, um relevante Zustände in den Bildregionen $x_n(t)$ zu quantifizieren. Beispiele sind die Lokalisierung von Rissen und anderen Defekten auf einer Fahrbahn, die Klassifizierung der Entwicklungsstufe von Eizellen auf einer Reihe von Ultraschallbildern oder die Vermessung von Wäldern in Satellitenbildern. Mit diesen Informationen kann der aktuelle Zustand für einen Zeitpunkt t konkret erfasst und mit einem anderen Zeitpunkt verglichen werden. Die Ergebnisse der überwachten Interpretation lassen sich ebenso wie bei der unüberwachten Interpretation allgemein als Merkmale betrachten.

Das Vorgehen bei der Interpretation in der ABV unterscheidet sich von den etablierten Methoden der CD [15, 39, 64, 72, 122] (s. Abschnitt 2.10). Ein direkter Vergleich wird in Abbildung 3.4 veranschaulicht. Herkömmliche CD-Methoden vergleichen zwei Bilder miteinander und markieren veränderte Regionen durch

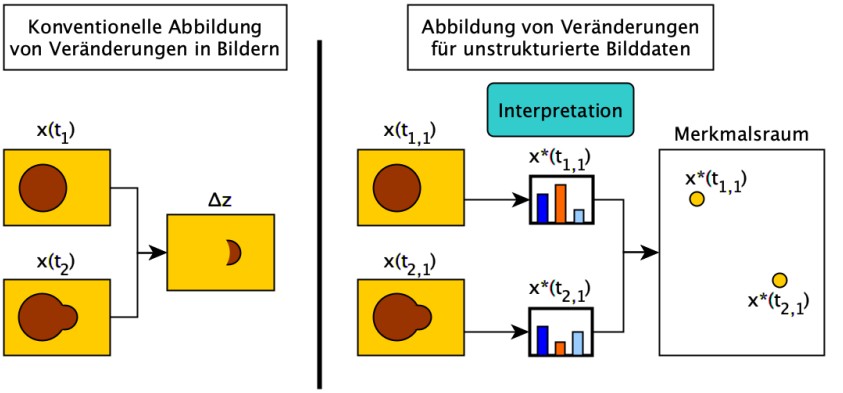


Abbildung 3.4: Direkter Vergleich der Abbildung von Veränderungen: Dargestellt sind die konventionelle CD links und das Konzept *OSMC* für unstrukturierte Bilddaten rechts. Beim konventionellen Ansatz werden zwei Bilder direkt verglichen und veränderte Bereiche werden markiert. Für unstrukturierte Bilddaten, bei denen der dynamische Blickwinkel einen direkten Vergleich erschwert, lassen sich robuste und vergleichbare Merkmale extrahieren.

eine Maske. Dieser direkte Vergleich ist aufgrund der unstrukturierten Bilddaten (z.B. dynamischen Blickwinkel und Abstände) nicht sinnvoll.

Das Konzept *OSMC* beschreibt jedes Bild einzeln, um anschließend die extrahierten Merkmale miteinander zu vergleichen. Dies ermöglicht im Falle von überwachten Methoden zur Interpretation einen Einblick in den geschätzten Ist-Zustand, ohne dass zwei Zeitpunkte erfasst sein müssen. Zusätzlich kann nach der Extraktion der Merkmale nicht nur der Vergleich von zwei Bildern erfolgen, sondern potenziell auch von Gruppen mehrerer Bilder. An dieser Stelle kommt die Aggregation zum Einsatz.

Im einfachsten Fall wird für jeden Zeitpunkt nur ein Satz an Merkmalen $x_n^*(t)$ extrahiert. Diese Merkmale entsprechen dem geschätzten Zustand $\hat{z}_n(t)$. Liegen mehrere Bildregionen zu einem Objekt für denselben Zeitpunkt vor, müssen die jeweiligen Merkmale durch die Aggregation zu einem geschätzten Zustand $\hat{z}_n(t)$ zusammengefasst werden.

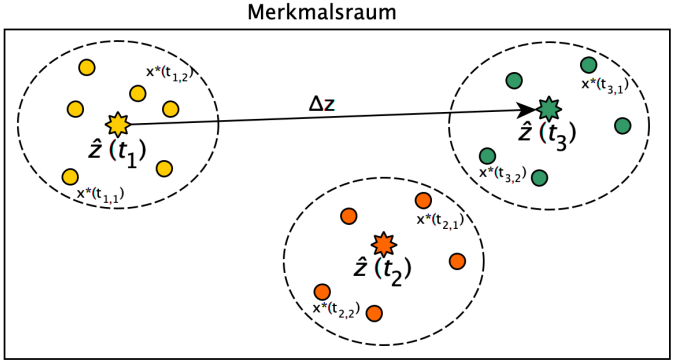


Abbildung 3.5: Visualisierung der Aggregation: Wenn für einen Zeitpunkt t mehrere Merkmale $x^*(t)$ durch die Interpretation bestimmt werden, müssen diese durch die Aggregation zu einem Zustand $\hat{z}(t)$ zusammengefasst werden. In der Abbildung werden aus drei Gruppen von Merkmalen zu den Zeitpunkten t_1 (gelb), t_2 (orange) und t_3 (grün) drei Zustände $\hat{z}(t_1)$, $\hat{z}(t_2)$ und $\hat{z}(t_3)$ berechnet. So werden letztlich die aggregierten Zustände miteinander verglichen und nicht einzelne Merkmale.

Dieses Vorgehen ist in Abbildung 3.5 veranschaulicht. Dabei müssen redundante Informationen entfernt und verbleibende Störungen oder unvollständige Informationen korrigiert werden. Die Aggregation schätzt den Zustand $\hat{z}_n(t)$ eines Objekts zum Zeitpunkt t. Dieser kann nun mit anderen Zuständen zu beliebigen Zeitpunkten verglichen werden, um Veränderungen abzubilden.

3.4 Dateneffiziente Methoden der Bildverarbeitung

3.4.1 Übersicht

Unstrukturierte Daten beinhalten eine Vielzahl an Informationen über unterschiedliche Arten und Veränderungen von Objekten, sodass Algorithmen entsprechend an verschiedene Problemstellungen angepasst werden müssen. Das zuvor eingeführte Konzept *OSMC* kann flexibel auf unterschiedliche Anwendungen

angepasst werden. Es ist jedoch notwendig, den Aufwand für eine Anwendung zu minimieren und die Qualität der Auswertung zu maximieren. In der Bildverarbeitung zur ABV lassen sich vier grundlegende Arten von Algorithmen identifizieren: Bildklassifikation, Objektdetektion, Bildsegmentierung und Merkmalsextraktion. Grundsätzlich ist jede dieser Methoden mit dem Konzept *OSMC* zur ABV kompatibel. Eine Merkmalsextraktion (unüberwacht) lässt sich mit dem geringsten Aufwand anwenden, da diese entweder direkt anwendbar ist (statische Merkmalsextraktion) oder sich automatisiert auf nicht-annotierte Daten anpasst (adaptive Merkmalsextraktion). Dafür sind die extrahierten Merkmale in der Regel nicht oder kaum interpretierbar. Die Annotation von Bilddaten für das Training einer Bildklassifikation ist minimal aufwändig, da pro Bild nur eine Klasse zugeordnet werden muss. Das Ergebnis der Bildklassifikation ist zwar interpretierbar, eröffnet aber nur begrenzte Einsichten. Die Objektdetektion erfordert einen moderaten Annotationsaufwand, da die Bildregion und Klasse annotiert werden müssen. Die Ergebnisse der Objektdetektion sind gut interpretierbar, da sie nicht nur die Anwesenheit eines Objekts, sondern auch dessen Position und Größe im Bild liefern. Die Bildsegmentierung hat den höchsten Annotationsaufwand, da jeder Pixel im Bild einer Klasse zugeordnet werden muss. Dies führt zu sehr präzisen und detaillierten Informationen über die Struktur und die Grenzen von Objekten im Bild. Die Ergebnisse der Bildsegmentierung sind daher am besten interpretierbar. Aus einer Segmentierung lassen sich sowohl präzise die Anteile verschiedener Bereiche im Bild als auch Merkmale über die Form und Anordnung von Objekten ableiten.

Bildverarbeitungsmethode	Annotationsaufwand	Interpretierbarkeit
Merkmalsextraktion	keiner	gering
Bildklassifikation	niedrig	mittel
Objektdetektion	mittel	hoch
Bildsegmentierung	hoch	sehr hoch

Tabelle 3.1: Bewertung von Bildverarbeitungsmethoden zur ABV

Die Bildverarbeitungsmethoden Merkmalsextraktion und Bildsegmentierung werden als ideal für die ABV bewertet. Diese beiden Methoden ermöglichen es einem Anwender passend je nach Anforderung einen Algorithmus zur ABV entweder mit minimalem Annotationsaufwand oder maximaler Interpretierbarkeit zu konfigurieren. Die Ergebnisse des Vergleichs von Algorithmen sind in Tabelle 3.1 zusammengefasst.

3.4.2 Merkmalsextraktion

Im Bereich der Merkmalsextraktion stehen verschiedene adaptive und statische Algorithmen zur Verfügung. Diese lassen sich nach den folgenden Kriterien bewerten: manuelle Anpassbarkeit der Methode (Flexibilität), Rechenaufwand für Training und Anwendung, Qualität der extrahierten Merkmale und automatisierte Anpassbarkeit (Adaptivität). Adaptive Merkmalsextraktion, wie CL (SIMCLR, NNCLR, Barlow-Twins), AE, MAE, und BoVW, erlernt Merkmale basierend auf nicht-annotierten Daten. Im Gegensatz dazu sind statische Methoden, wie KMBV (Farb-Histogramme, HOG, LBP und GLCM) und vDCNN, schon direkt ohne jeglichen Dateneinsatz anwendbar. Falls annotierte Daten zu Verfügung stehen, werden adaptive Methoden bevorzugt. In dieser Arbeit wird die Methode BoVW als adaptive Merkmalsextraktion ausgewählt, da diese durch die Auswahl von passenden lokalen Deskriptoren flexibel an Problemstellungen angepasst werden kann und keine aufwändigen Berechnungen durchgeführt werden müssen. Außerdem wurde die Funktionalität von BoVW-Methoden für eine ABV bereits in vorangegangenen Arbeiten [1, 7] bewiesen. Aufgrund dieser Ergebnisse wird in dieser Arbeit die BoVW-Methode mit SIFT-Deskriptoren [108] im Grau- und HSV-Farbraum verwendet. Im Gegensatz dazu benötigen CL und AE/MAE Methoden sehr aufwändige Berechnungen für eine Anwendung auf einen Datensatz.

Stehen keine Daten zur Verfügung, muss auf eine statische Methode zurückgegriffen werden. Hier wird vDCNN als effektivere Methode ausgewählt, da diese im direkten Vergleich zu KMBV bessere Merkmale extrahiert. Das vDCNN wird mit dem auf ImageNet [151] vortrainierten ResNet50 [77] Backbone verwendet.

Das ResNet50 Backbone ist als Standard etabliert und hat seine Vielseitigkeit in verschiedenen Anwendungen [13, 118, 142, 146] bewiesen. Die Auswertung wird in Tabelle 3.2 zusammengefasst.

Merkmalsextraktor	Flexibilität	Rechenaufwand	Adaptivität	Qualität
BoVW	hoch	mittel	adaptiv	hoch
CL	mittel	hoch	adaptiv	hoch
AE	niedrig	hoch	adaptiv	hoch
MAE	niedrig	hoch	adaptiv	hoch
KMBV	niedrig	niedrig	statisch	mittel
vDCNN	niedrig	niedrig	statisch	hoch

Tabelle 3.2: Auswahl von Methoden zur Merkmalsextraktion: Ausgewählte Methoden für den Einsatz zur ABV sind **hervorgehoben**.

3.4.3 Bildsegmentierung

In dieser Arbeit werden drei Arten von Bildsegmentierungs-Methoden unterschieden: Deep-Learning (DL), Konventionelle Bildverarbeitung (KBV) und strukturierte Segmentierung (STRUCT). DL-Methoden [44, 148, 203] haben sich in der Bildsegmentierung als besonders leistungsfähig erwiesen [26, 29, 87, 178] und werden häufig verwendet, ohne andere Methoden in Erwägung zu ziehen. Die Annotation von Daten für eine Bildsegmentierung ist aufwändig und DL-Methoden benötigen besonders viele annotierte Daten, sodass sich für eine Verwendung zur ABV die Frage nach dateneffizienten Alternativen stellt.

Aktuelle Methoden zur Minimierung des Annotationsaufwandes, wie CL [46, 60, 196], FSL [56, 111, 181, 191] oder Meta-Learning [66, 124] optimieren das Training von DL-Methoden. Sie benötigten große, nicht-annotierte Datensätze und/oder aufwändige Berechnungen, die eine Übertragung auf neue Problemstellungen erschweren. Zusätzlich sind die Methoden stark von den gewählten Hyperparametern abhängig. Die vorangegangene Arbeit [3] legt nahe, dass es dateneffiziente

Alternativen zu DL-Methoden gibt, die sich direkt (ohne zusätzlichen Aufwand) anwenden lassen. In dieser Arbeit werden DL-, STRUCT- und KBV-Methoden mit einander hinsichtlich ihrer Dateneffizienz verglichen[1]. Grundsätzlich ist jede der Methoden mit einer ABV kompatibel. Anhand dieses Vergleichs soll bestimmt werden, welche Methode zur Bildsegmentierung mit möglichst geringem Aufwand zur ABV eingesetzt werden kann.

Es gibt eine Vielzahl an DL-Methoden (U-Net [148], PSPNet [203], SegNet [20], DeeplabV3 [44], SegFormer [188], etc.) zur Bildsegmentierung. Das U-Net hat sich jedoch als Standardverfahren bewährt [10, 155] und wird daher als Referenz für die Dateneffizienz anderer Methoden verwendet.

Die KBV basiert auf einfachen Operationen, wie Schwellwertbildung, Kantenerkennung oder morphologischen Operationen, wobei jede Operation mit individuellen Parametern spezifiziert werden kann. Geschwindigkeit, Effizienz und Nachvollziehbarkeit sind die wichtigsten Argumente für die KBV. Insbesondere wenn mehrere Bildverarbeitungsoperationen nacheinander ausgeführt werden sollen, macht die manuelle Festlegung der Parameter diesen Ansatz jedoch zeitaufwändig und komplex. Aus diesem Grund wurde in der vorangegangenen Arbeit [3] die Conventional-Image-Processing-Pipeline (CIPP) entworfen. Eine CIPP besteht aus einer Sequenz von Operationen der KBV, die sich analog zu DL-Methoden automatisch anhand eines Trainingsdatensatzes parametrieren. Die Reihenfolge der Operationen wird dabei vom Anwender festgelegt und kann so direkt von dessen Wissen profitieren. CIPPs sind durch die Verwendung von einfachen KBV-Operationen robust gegen Overfitting und können bereits mit wenigen Bildern trainiert werden.

STRUCT- und DL-Methoden wurden bereits in der vorangegangenen Arbeit [8] hinsichtlich ihrer Qualität und Trainingszeit verglichen. Die STRUCT-Methoden segmentieren Bilder basierend auf einfachen pixelbasierten Merkmalen. Dafür werden die Bilder in Pixeldatensätze transformiert, die anschließend von einem

[1] KBV- und DL-Methoden wurden bereits in [3] auf synthetischen Bilddaten evaluiert. In dieser Arbeit werden diese Versuche mit der Berücksichtigung von STRUCT-Methoden fortgeführt und im Kontext der unstrukturierten Bilddaten getestet.

Klassifikator verarbeitet werden. In [8] werden zwei STRUCT-Methoden vorgestellt: Structured-Classifier (SC) und strukturierter Encoder-Decoder (StED). Es zeigt sich, dass der StED schnell anwendbar ist und vergleichbare Segmentierungsgüten erreicht wie DL-Methoden. Daher wird im Folgenden der StED für die Untersuchung zur Dateneffizienz herangezogen.

3.4.4 Experimente zur Bewertung der Dateneffizienz von Methoden zur Bildsegmentierung

Die vorgestellten Modelle CIPP, StED und U-Net verwenden dasselbe Datenformat und können leicht an vorhandene Trainingsdaten angepasst werden. Das Modell CIPP weist die geringste Komplexität[2] auf, gefolgt von dem StED-Modell. Das U-Net hat die höchste Komplexität. Nachdem Merkmalsextraktoren bereits bewertet und ausgewählt wurden, kann noch keine endgültige Aussage über Methoden zur Bildsegmentierung getroffen werden.

Von besonderem Interesse für den Einsatz bei der ABV ist die Dateneffizienz der Modelle. Die Dateneffizienz eines Modells entspricht der Güte der Segmentierung, die ein Modell bei der Verwendung weniger Trainingsdaten erreicht, und wird für jede Segmentierungsaufgabe einzeln bewertet. Eine Segmentierungsaufgabe besteht dabei aus der zu segmentierenden Klasse K und dem Trainingsdatensatz mit N_{img} Bildern. Die Güte der Segmentierung eines Modells für eine Segmentierungsaufgabe wird durch verschiedene Segmentierungsaufgabenparameter (SAP) beeinflusst:

- Flächenanteil a_K: Der Flächenanteil a_K der Klasse K eines Datensatzes entspricht dem Verhältnis der Pixel, die der Klasse K zugeordnet wurden, zu allen Pixeln im Datensatz. Je geringer der Flächenanteil a_K ist, desto weniger Fläche nimmt die Klasse ein. Dadurch wird die Segmentierung erschwert [98, 147].

[2] Komplexität wird hier als Anzahl der während des Trainings optimierten Parameter verstanden.

- Art der Störfaktoren S_λ: Dynamische[3] Störfaktoren λ erschweren die automatisierte Verarbeitung von Bilddaten. Die Art der Störfaktoren S_λ, wie in Abschnitt 3.2 eingeführt, unterscheiden sich in visuelle λ_v und geometrische λ_g). Visuelle Störfaktoren können häufig durch Methoden der Bildverarbeitung ausgeglichen werden, während geometrische Störfaktoren in der Regel mit visuellen Störfaktoren einhergehen und die Segmentierungsaufgabe deutlich erschweren.

Im Bereich der Bildverarbeitung gibt es viele unterschiedliche SAP[4]. In dieser Arbeit werden der Flächenanteil a_K und die Art der Störfaktoren S_K bewerten, da diese leicht und eindeutig zu bestimmen sind und so eine eindeutige und simple Anwendung ermöglichen. Im Rahmen der vorliegenden Doktorarbeit wird die Dateneffizienz der ausgewählten Modelle in Abhängigkeit der SAP bestimmt, wie in Abbildung 3.6 dargestellt. Zuerst werden Segmentierungsaufgaben ausgewählt und entsprechend ihrer SAP bewertet. Der Flächenanteil a_K wird in drei Intervalle geteilt $(0, 0.01]$, $(0.01, 0.1]$ und $(0.1, 0.99]$. Die drei Intervalle repräsentieren kleine, mittlere und große Flächenanteile a_K. Die Intervalle sind nicht gleichgroß, da bei höheren Flächeanteilen, der absolute Unterschied einen geringeren Unterschied macht. Die Art der dynamischen Störfaktoren S_λ wird in zwei Gruppen unterschieden: Visuelle Störfaktoren λ_v und geometrische/visuelle Störfaktoren λ_{v+g}. In realen Datensätzen bringen geometrische Störfaktoren ebenfalls visuelle Störfaktoren mit sich bspw. durch Veränderungen der Belichtung. Im direkten Vergleich ist die Art der Störfaktoren $S_\lambda = \lambda_{v+g}$ als schwieriger zu bewerten.

Um die Dateneffizienz der vorgestellten Modelle (U-Net, CIPP und StED) auf einer SAP zu vergleichen und entsprechende Anwendungsempfehlungen abzuleiten, werden die Experimente aus [3] fortgeführt. Ein Bildsegmentierer wird mit N_{img} zufällig ausgewählten Bildern aus dem Trainingsdatensatz trainiert. Anschließend

[3] Störfaktoren, die statisch bleiben, können ausgeglichen werden und erschweren die Segmentierungsaufgabe nicht nennenswert.

[4] Die Schwierigkeit einer Segmentierungsaufgabe kann anhand von extrahierten Merkmalen bewertet werden [103]. Durch den Einsatz von leistungsstarken DL-basierten Verfahren haben solche Methoden jedoch an Relevanz verloren.

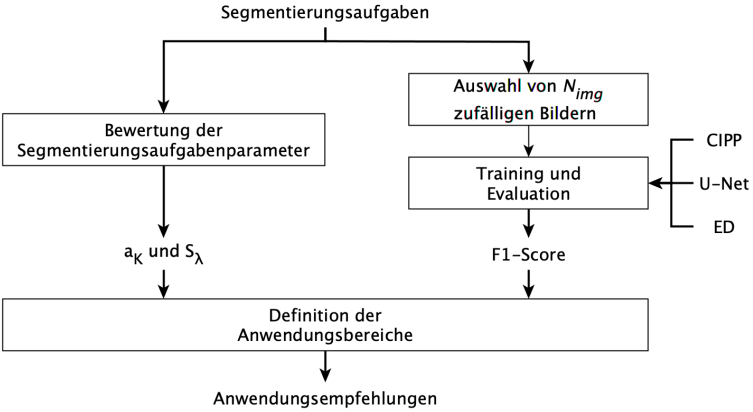


Abbildung 3.6: Ablauf zur Bestimmung der Dateneffizienz: Zuerst werden Segmentierungsaufgaben ausgewählt und bezüglich ihrer SAP bewertet. Die Dateneffizienz der Modelle für eine Segmentierungsaufgabe wird dann in mehreren Durchläufen durch das Training mit N_{img} zufällig ausgewählten Bildern und anschließender Evaluation bestimmt. Die Auswertung ermöglicht dann die Definition von Anwendungsempfehlungen in Abhängigkeit der SAP.

wird die Güte der Segmentierung (F1-Score) auf dem Testdatensatz berechnet. Der F1-Score kann je nach den ausgewählten Bildern schwanken. Aus diesem Grund wird der Prozess mehrfach wiederholt, sodass die Streuung des F1-Scores ebenfalls abgebildet wird. Aus diesen Ergebnissen können Zusammenhänge zwischen den SAP (a_K und S_λ), der verwendeten Anzahl an Trainingsbildern N_{img} und der Güte der Segmentierung (F1-Score) abgeleitet werden.

Das Ergebnis einer solchen Testreihe ist in Abbildung 3.7 zu sehen. Der Verlauf der Segmentierungsgüte über die Anzahl der Trainings-Bilder N_{img} gibt Aufschluss über die Dateneffizienz eines Algorithmus. Im direkten Vergleich mit anderen Algorithmen können so optimale Einsatzbereiche und Einsatzszenarien herausgearbeitet werden. Um Bildsegmentierer für ein Intervall $[N^{\min}, N^{\max}]$ von Trainingsbildern zu bewerten, wird die durchschnittliche Segmentierungsgüte $G_{N^{\min}}^{N^{\max}}$ als Mittelwert über alle Versuchsdurchläufe und Bildanzahlen N_{img}

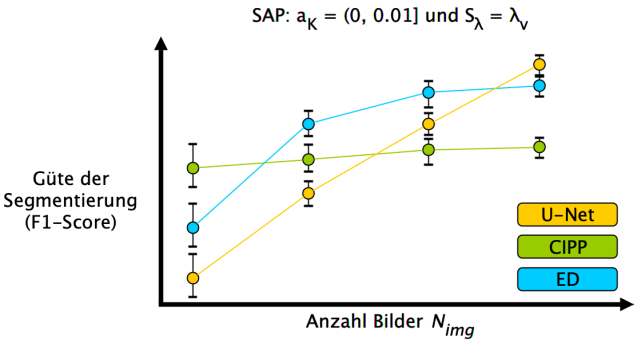


Abbildung 3.7: Die Dateneffizienz-Kurve eines Modells: Hier werden die Segmentierungsgüte der Algorithmen (U-net, CIPP, StED) über die benötigte Anzahl an Trainingsbildern N_{img} samt Unsicherheit abgebildet. Die SAP sind fest, sodass je nach SAP-Kombination die entsprechende Dateneffizienz-Kurve ausgewählt werden kann.

berechnet.[5] Die Metriken können anschließend unter Berücksichtigung der SAP ausgewertet werden.

3.5 Bewertungsmethodik HyBAR

In diesem Abschnitt wird die Bewertungsmethodik „Hypothesis-Based-Algorithm Rating“ *HyBAR* vorgestellt. Es wird eine einheitliche Methodik zur Bewertung von Algorithmen zur ABV benötigt, die es Anwendern erlaubt, Algorithmen zur ABV optimal zu konfigurieren und zu parametrieren. Komplexe Algorithmen und große Datensätze machen eine manuelle Bewertung unmöglich. Doch gerade bei unstrukturierten Bilddaten, die in großer Menge und nicht zielgerichtet erhoben werden, ist der tatsächliche Zustand $z_n(t)$ im Vorfeld nicht bekannt und die Bestimmung des Zustandes $z_n(t)$ nachträglich nicht möglich oder zu aufwändig. Zudem bestehen Algorithmen zur ABV aus einer Vielzahl an Komponenten, die sich gegenseitig beeinflussen. So kann die Aufnahme der Bilder deren Auswertung

[5] So lässt sich beispielsweise eine durchschnittliche Segmentierungsgüte G_4^{32} für das Intervall von $[4, 32]$ Trainingsbildern angeben.

beeinflussen. Dies macht eine indirekte Bewertungsmethodik notwendig, die ohne annotierte Daten anwendbar ist und die den vollständigen Algorithmus zur ABV betrachtetet.

Aus diesem Grund wird in dieser Arbeit die hypothesenbasierte Bewertungsmethodik *HyBAR* von Algorithmen zur ABV vorgestellt: Basierend auf dem domänenspezifischen Wissen lassen sich Hypothesen über die Veränderung des Zustandes $z_n(t)$ eines Objekts aufstellen. Als Beispiel lässt sich das Wachstum von Rissen in Straßen anführen: Hier kann von einer monoton steigenden Länge eines Risses ausgegangen werden. Folgen die geschätzten Zustände $\hat{z}_n(t)$ eines Algorithmus dieser Hypothese, bildet der Algorithmus die Veränderung gut ab.

In dieser Arbeit werden Datensätze betrachtet, die zeitlich nah beieinander liegen. Hier kann für einen hinreichend kleinen Zeitabstand die Hypothese formuliert werden, dass keine Veränderung stattgefunden hat. Ein Algorithmus, der Veränderungen robust abbildet, soll also ebenfalls keine Veränderungen erkennen, wobei jede Schwankung auf den Einfluss von Störfaktoren in den unstrukturierten Daten zurückgeführt werden muss. Es folgt für das i-te Objekt mit $i \in \{1, ..., N\}$ in einem Datensatz, bei dem aufgrund der geringen Zeitdifferenz Δt keine Veränderungen erwartet werden:

$$|\hat{z}_i(t_1) - \hat{z}_i(t_M)| = 0. \tag{3.2}$$

Auch wenn sich der Zustand $z_i(t)$ des i-ten Objektes nicht ändert, kommt es durch den Einfluss von Störfaktoren $\lambda(t)$ zu geringfügigen Schwankungen des geschätzten Zustands $\hat{z}_i(t)$. Um einen Algorithmus erfolgreich einzusetzen, dürfen diese Schwankungen jedoch nicht größer sein als der Unterschied zwischen unterschiedlichen Objekten. Die Streuung innerhalb eines Objektes muss also kleiner sein als die Streuung zwischen den Objekten, da sich sonst Veränderungen nicht zuverlässig erkannt werden können. Dieser Ansatz fußt auf der Diskriminanzanalyse [144] oder den Methoden des CL [46]. Es wird ein weiteres beliebiges j-tes Objekt mit $j \in \{1, ..., N\}$ und $i \neq j$ eingeführt. Es folgt für zwei beliebige Objekte:

$$|\hat{z}_i(t_M) - \hat{z}_i(t_1)| \leq |\hat{z}_i(t) - \hat{z}_j(t)|. \tag{3.3}$$

Basierend auf diesen Hypothesen (Gleichung 3.2 und 3.3) lassen sich Kennzahlen formulieren. Die Kennzahlen werden ausschließlich auf den geschätzten Zuständen $\hat{z}(t)$ eines Algorithmus berechnet und messen die Eignung eines Algorithmus für die ABV auf einem gegebenen Datensatz. Grundsätzlich können je nach Anwendung auch weitere Hypothesen aufgestellt werden. So kann ebenfalls eine monotone Veränderung oder eine oszillierende Zustandsveränderung angenommen werden. Der Fall eines gleichbleibenden Zustands kann jedoch im Allgemeinen erfüllt werden, wenn der Abstand zwischen den Zuständen klein genug gewählt wird.

Die Intra-Objekt-Streuung Φ bewertet die Robustheit eines Algorithmus gegenüber Störfaktoren $\lambda(t)$ und berechnet sich aus der durchschnittlichen Objekt-Streuung ϕ_n der geschätzten Zustände eines Objekts. Die Objekt-Streuung ϕ_n der Menge der Zustände $\mathbf{Z}_n = \{\hat{z}_n(t_1), ..., \hat{z}_n(t_M)\}$ eines Objekts entspricht dem durchschnittlichen paarweisen Abstand aller geschätzten Zustände $\hat{z}_n$ des Objekts zueinander[6]. Diese berechnet sich wie folgt:

$$\phi_n = \frac{2}{M(M-1)} \sum_{i=1}^{M} \sum_{j=i+1}^{M} |\hat{z}_n(t_i) - \hat{z}_n(t_j)|. \tag{3.4}$$

Die Intra-Objekt-Streuung Φ eines Algorithmus zur ABV berechnet sich aus der durchschnittlichen Streuung ϕ_n über alle Objekte im Datensatz. Ein Algorithmus, der robust gegenüber Störfaktoren $\lambda(t)$ ist, hat eine minimale Intra-Objekt-Streuung Φ (Unter der Annahme, dass keine Veränderung stattgefunden hat). Es gilt:

[6] Die Standardabweichung transportiert die gleichen Informationen, ist aber schwerer zu interpretieren. Der Abstand zwischen zwei Objektzuständen kann leicht berechnet und ausgewertet werden, während die Standardabweichung im direkten Vergleich nicht anwendbar ist. Als Maß für den Abstand zwischen zwei Darstellungen wird die Manhattan-Distanz verwendet.

$$\Phi = \frac{1}{N} \sum_{n=1}^{N} \phi_n. \tag{3.5}$$

Die Inter-Objekt-Streuung Γ bewertet die Deskriptivität eines Algorithmus und berechnet sich aus der durchschnittlichen Streuung γ_n zwischen den geschätzten Zuständen unterschiedlicher Objekte. Zuerst wird für jedes Objekt der geschätzte Zustand $\hat{z}_n$ unabhängig von der Zeit t folgendermaßen berechnet:

$$\hat{z}_n = \frac{1}{M} \sum_{m=1}^{M} \hat{z}_n(t_m). \tag{3.6}$$

Die Streuung eines Objektes relativ zu allen anderen Objekten γ_n berechnet sich als durchschnittlicher Abstand des Zustands $\hat{z}_n$ zu allen anderen Objektzuständen:

$$\gamma_n = \frac{1}{N-1} \sum_{i=1, i \neq n}^{N} |\hat{z}_i - \hat{z}_n|. \tag{3.7}$$

Anschließend ergibt sich die Inter-Objekt-Streuung Γ:

$$\Gamma = \frac{1}{N} \sum_{n=1}^{N} \gamma_n. \tag{3.8}$$

Die Inter-Objekt-Streuung Γ fungiert als Gegengewicht zur Intra-Objekt-Streuung Φ. Ohne die Inter-Objekt-Streuung Γ kann ein Algorithmus stets einen konstanten Zustand $\hat{z}(t)$ schätzen, um die Intra-Objekt-Streuung Φ zu minimieren. Γ soll entsprechend maximiert werden[7].

[7] Maximierung und Minimierung von Streuungen innerhalb und zwischen Objekten findet sich ebenfalls in der Methodik Analysis-of-Variance (ANOVA) [167].

Γ und Φ können verwendet werden, um einen Algorithmus zur ABV zu charakterisieren. Die Streuungen sind abhängig von den verwendeten Merkmalen und deren Skalierung, sodass für einen direkten Vergleich zwischen verschiedenen Algorithmen der Hypothesen-Quotient F eingeführt wird. F entspricht dem Anteil aller Objekte, für die $\phi_n \leq \gamma_n$ gilt, und damit die initiale Hypothese erfüllen. Der ideale Wert entspricht hier $F = 1$. Zusätzlich wird das Streuungsverhältnis β als Verhältnis zwischen Φ und Γ bestimmt:

$$\beta = \frac{\Gamma}{\Phi}. \tag{3.9}$$

Das Verhältnis β ist unabhängig von der Skalierung des zugrunde liegenden Merkmalsraums, sodass verschiedene Methoden direkt miteinander verglichen werden können. Für einen Algorithmus, der sich für die Abbildung von Veränderungen in unstrukturierten Bilddaten eignet, gilt $\beta > 1$. In diesem Fall ist die Inter-Objekt-Streuung Γ größer als die Intra-Objekt-Streuung Φ. Bei der Auswahl verschiedener Algorithmen soll β maximiert werden.

Für die Abbildung von Veränderungen lässt sich die Intra-Objekt-Streuung Φ als Referenzwert verwenden, um anschließend auf unbekannten Daten Veränderungen zu registrieren, die eine höhere Streuung aufweisen. Dies entspricht einer Kalibrierung, bei der das erwartete Grundrauschen, welches noch nicht einer Veränderung entspricht, als Grenzwert eingeführt wird.

Die Anwendung der Bewertungsmethodik *HyBAR* wird in Abbildung 3.8 veranschaulicht. Ein unstrukturierter Datensatz wird von einem Algorithmus zur ABV verarbeitet, wie in Abschnitt 3.3 beschrieben. In jedem Teildatensatz $\mathbf{X}_m$ mit $m \in M$ werden Objekte assoziiert, interpretiert und deren Zustände $\hat{z}(t_m)$ zum Zeitpunkt t_m geschätzt. Die geschätzten Zustände $\hat{z}(t_m)$ sind von der Parametrierung des Algorithmus zur ABV abhängig. Um die beste AV auszuwählen, werden die geschätzten Zustände über den gesamten Datensatz mit der beschriebenen Bewertungsmethodik verglichen. Die AV, die die Intra-Objekt-Streuung Φ minimiert und gleichzeitig die Inter-Objekt-Streuung Γ maximiert, kann abschließend für den weiteren Einsatz ausgewählt werden.

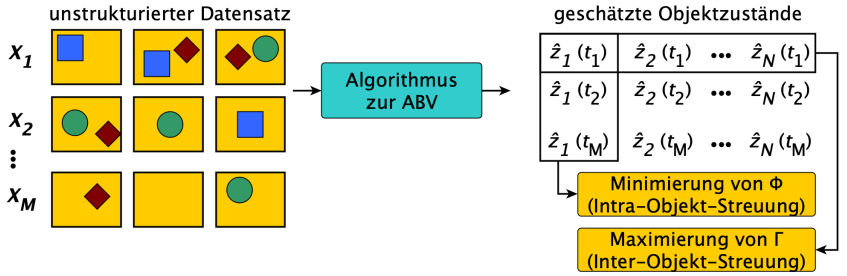


Abbildung 3.8: Anwendung der Bewertungsmethodik *HyBAR*: Der Zustand $\hat{z}(t_m)$ zum Zeitpunkt t_m eines jeden Objektes wird basierend auf dem entsprechenden Teildatensatz $\mathbf{X}_m$ geschätzt. Die Parametrierung des Algorithmus zur ABV bestimmt dabei den geschätzten Zustand $\hat{z}(t_m)$. Um die beste Parametrierung zu wählen, werden alle berechneten Zustände im unstrukturierten Datensatz mittels der Bewertungsmethodik *HyBAR* ausgewertet. Dabei soll die Intra-Objekt-Streuung Φ minimiert und gleichzeitig die Inter-Objekt-Streuung Γ maximiert werden.

Um die Bewertungsmethodik *HyBAR* anzuwenden, müssen die Bilddaten von mehreren Zeitpunkten vorliegen und die Zustände der Objekte dürfen sich nicht verändert haben. Diese Bedingungen erfüllen nicht alle Datensätze. Für den Transfer der Bewertungsmethodik *HyBAR* auf beliebige Datensätze kann die Augmentation verwendet werden, um verschiedene Aufzeichnungen konstanter Zustände eines zugehörigen Objekts zu erzeugen. Dieses Verfahren wird auch beim CL genutzt [46]. Die Augmentation simuliert den Einfluss von Störfaktoren $\lambda(t)$ bei der Aufzeichnung der Bilder.

4 Implementierung

4.1 Übersicht

Unstrukturierte Bilddaten benötigen eine Datenstruktur, die eine einfache und effektive Verarbeitung nach dem vorgestellten Konzept *OSMC* (Assoziation, Interpretation und Aggregation) ermöglicht. Neben der Datenstruktur werden in dieser Dissertation speziell ausgewählte und entwickelte Algorithmen zur Bildverarbeitung eingeführt: Merkmalsextraktoren, KBV und STRUCT.

Diese Datenstruktur wurde eigens für die ABV entworfen und ermöglicht es erstmals das Konzept *OSMC* zur ABV in unstrukturierten Bilddaten flexibel anzuwenden. Die Implementierung der Merkmalsextraktoren beinhaltet verschiedene Methoden zur Merkmalsextraktion und integriert sich direkt in die Datenstruktur. Dadurch können unterschiedlich parametrierte AV leicht miteinander verglichen und an eine Problemstellung angepasst werden.

Um eine dateneffiziente überwachte Interpretation von Bilddaten zu ermöglichen, wurden die Modelle CIPP basierend auf der KBV und StED basierend auf der STRUCT entwickelt. Während sich andere Implementierungen von KBV auf spezifische Problemstellungen oder Lösungsansätze konzentrieren [38, 113, 171, 172], wird in der Implementierung der CIPP erstmals eine allgemeine und dateneffiziente Alternative zu DL-Methoden für die Bildsegmentierung entwickelt.

Die Implementierung des StED-Modells ermöglicht erstmals eine Anwendung zur Bildsegmentierung. Hier liegt der Fokus auf Skalierbarkeit, Trainingsgeschwindigkeit und Dateneffizienz.

4.2 Datenstruktur

Die Verarbeitung unstrukturierter Bilddaten erfordert eine Reihe grundlegender Funktionalitäten:

- die Verarbeitung von räumlichen Zusammenhängen zur Unterstützung der Assoziation,
- das Laden von Bildern zur weiteren Verarbeitung (Assoziation und Interpretation),
- das Speichern und Verarbeiten (Aggregation) von berechneten Zuständen für einzelne Datenpunkte und Gruppen von Datenpunkten,
- die Verarbeitung von zeitlichen Zusammenhängen zur Abbildung von Veränderungen und
- die Bewertung der geschätzten Zustände nach der eingeführten Bewertungsmethodik *HyBAR*.

In dieser Arbeit wird eine flexible Datenstruktur implementiert, die für die Auswertung von unstrukturierten Bilddaten geeignet ist. Die Implementierung erfolgt in *Python 3*[1].

Unstrukturierte Bilddaten können auf der untersten Ebene in einzelne Bilder zerlegt werden, wobei jedes Bild eine Position, einen Zeitstempel und eine Quelle besitzt. Die Quelle bezieht sich dabei auf den Kontext der Datenerhebung, sodass zwei gleichzeitig aufgenommene Teildatensätze voneinander getrennt verarbeitet werden können. Auf diese Weise kann die Intra-Objekt-Streuung zwischen unterschiedlichen Aufnahmemodalitäten unabhängig von der Zeit berechnet werden. Dies wird programmatisch durch die Implementierung einer eigenen Klasse `UnstructuredImage` abgebildet. Der gesamte Datensatz, ein zu beobachtendes

[1] Die verwendete Implementierung wird in einem öffentlichen Git-Repository zur Verfügung gestellt: `https://github.com/FMuenke/structured_data`.

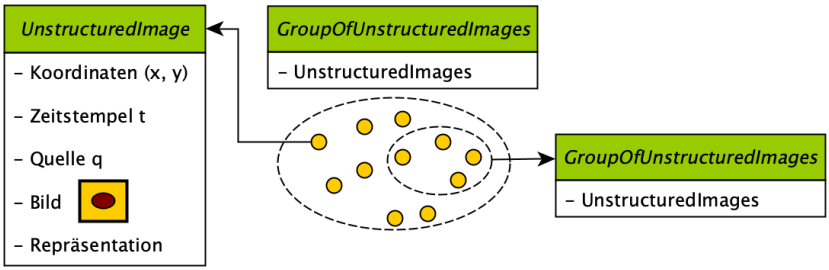


Abbildung 4.1: Datenstruktur: Ein unstrukturierter Datensatz besteht aus einzelnen Objekten (`UnstructuredImage`). Diese werden zu einem Objekt (`GroupOfUnstructuredImages`) zusammengefasst. Eine `GroupOfUnstructuredImages` kann auch in neue kleinere `GroupOfUnstructuredImages` zerlegt werden.

Objekt oder der Zustand $z(t)$ eines zu beobachtenden Objekts kann als Gruppe von Bildern definiert werden. Neben dem Einzelbild `UnstructuredImage` wird eine Klasse `GroupOfUnstructuredImages` definiert. Diese Datenstruktur ist in Abbildung 4.1 visualisiert.

Auf Basis dieser Objekte werden die Funktionalitäten implementiert. Die Verarbeitungsschritte sind in Abbildung 4.2 visualisiert. Die zusammenfassende Klasse `GroupOfUnstructredImages` ermöglicht die Sortierung/Aufteilung der Bilder anhand der vorhandenen Koordinaten, Zeitstempel und Quellen. Dabei werden verschiedene Clustering-Algorithmen aus der Scikit-learn-Bibliothek [133] verwendet (DBSCAN [62], MeanShift [50] und KMeans [18]). Zusätzlich erlaubt die Klasse die Gruppierung von Bildern nach Quellen.

Die Assoziation von Bilddaten und Objekten wird durch das Bereitstellen von ortsbasiertem Clustering vereinfacht. Die Klasse `UnstructuredImage` stellt zusätzlich eine Methode zum Laden des Bildes zur Verfügung. Weitere Methoden zur Assoziation im Bild können dann direkt auf das geladene Bild angewendet werden. Die Interpretation wird durch eine Klasse `RepresentationRepository` vereinfacht. Diese ist mit beliebigen Bildverarbeitungsmethoden zur Interpretation kombinierbar, speichert bereits berechnete Repräsentationen ab und verhindert

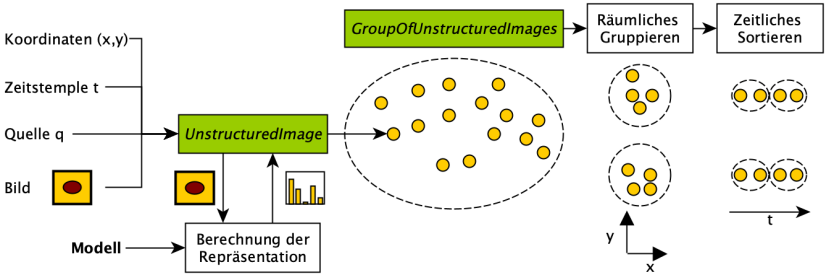


Abbildung 4.2: Darstellung der Datenverarbeitung: Die Bilder werden mit ihren Zusatzinformationen (Koordinaten, Zeitstempel, Quelle) in einem `UnstructuredImage` gespeichert. Für jedes `UnstructuredImage` wird eine Repräsentation mit einem bereitgestellten Modell berechnet und gespeichert. Das `UnstructuredImage` wird dann der `GroupOfUnstructuredImages` hinzugefügt. Anschließend kann die `GroupOfUnstructuredImages` räumlich gruppiert und zeitlich sortiert werden. Das Ergebnis der Teilungsoperation sind wieder mehrere `GroupOfUnstructuredImages`, die weiter geteilt werden können.

so redundante Berechnungen. Die Bewertungsmethodik *HyBAR* ist ebenfalls implementiert und direkt mit der Datenstruktur kompatibel.

Je nach Anwendungsfall ist die Verwendung von Augmentationen zur Simulation von Störfaktoren notwendig. Dazu wird ein `UnstructuredImage` mit der konfigurierbaren Klasse `Augmentations` multipliziert.

4.3 Bag of Visual Words

Für diese Arbeit wird ein installierbares *Python*-Paket entwickelt, das die einfache Berechnung von BoVW-Darstellungen für Bilder unterstützt[2]. Das Paket ermöglicht die schnelle Konfiguration eines BoVW-Modells und die effiziente Berechnung und Speicherung der Darstellungen.

[2] Die verwendete Implementierung ist in einem öffentlichen Git-Repository verfügbar: `https://github.com/FMuenke/handcrafted_image_representations`.

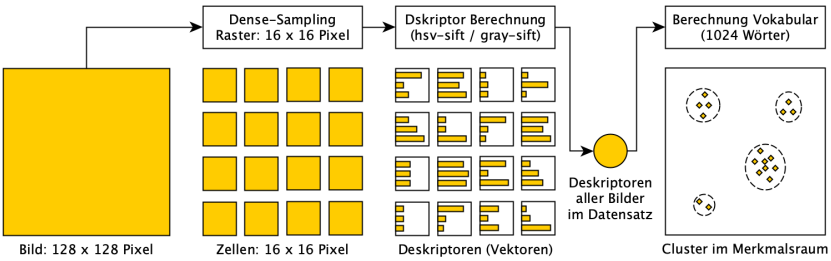


Abbildung 4.3: Umsetzung des BoVW-Modells: Jedes Bild wird auf 128 x 128 Pixel skaliert. Anschließend wird das Bild mittels Dense-Sampling [176] in ein Raster (16 x 16 Pixel) zerlegt. Für jede Zelle wird ein Deskriptor berechnet. Die Deskriptoren werden einmal auf dem Grauwertbild und einmal auf dem HSV-Bild mit dem SIFT-Verfahren berechnet [108]. Das Vokabular jedes Modells wird dann mit KMeans über alle verfügbaren Deskriptoren berechnet. In der Anwendung wird das Bild wie beschrieben verarbeitet, die berechneten Deskriptoren werden jeweils einem visuellen Wort zugeordnet und die Häufigkeit der visuellen Wörter wird als Repräsentation für das Bild verwendet.

Es werden zwei verschiedene BoVW-Modelle verwendet: *hsv-sift* und *gray-sift*. Dabei wird die Implementierung aus *OpenCV* [27] genutzt. Die Funktionsweise der Modelle ist in Abbildung 4.3 dargestellt. Beide Modelle verarbeiten Bilder auf ähnliche Weise. Die Bilder werden im Format 128 x 128 Pixel verarbeitet. Die Extraktion der Deskriptoren erfolgt in zwei Schritten. Zunächst werden mittels Dense-Sampling [176] Regionen im Bild definiert (Raster: 16 x 16 Pixel). Im nächsten Schritt wird mit dem SIFT-Algorithmus [108] für jede Bildregion ein Deskriptor berechnet. Das Modell *gray-sift* berechnet die Deskriptoren auf Basis des Graustufenbildes. Das Modell *hsv-sift* berechnet die Deskriptoren aus dem Bild im HSV-Farbraum. Dabei wird für jeden Farbkanal (H, S und V) ein SIFT-Deskriptor berechnet und anschließend zusammengeführt. Aus der Menge aller Deskriptoren über alle Bilder wird das Vokabular berechnet. Beide Modelle haben ein Vokabular von 1024 Wörtern und verwenden MiniBatchKMeans [133] zum Clustering. In der Anwendung werden die Deskriptoren, wie oben beschrieben, berechnet. Diese werden im Folgenden jeweils dem nächsten Bildwort zugeordnet.

Die Repräsentation eines Bildes besteht aus dem Histogramm aller vorkommenden visuellen Wörter innerhalb eines Bildes. Die Implementierung stellt zusätzlich

weitere Methoden zur Verfügung, um lokale Deskriptoren in einen Merkmalsvektor zu transformieren: VLAD [17], Fisher-Vektoren [168]. Diese Methoden berücksichtigen nicht nur die Anzahl der vorkommenden Deskriptoren sondern auch deren Verteilung.

4.4 Tiefe Neuronale Netzwerke

Eine weitere Methode, eine Repräsentation für ein Bild zu berechnen, wird durch DL ermöglicht. DNN dienen in erster Linie als Merkmalsextraktoren. Wenn neue Netze veröffentlicht werden, werden sie in der Regel mit bereits trainierten Gewichten veröffentlicht.

DNNs können für viele verschiedene Anwendungsfälle genutzt werden. Das Backbone fungiert als Merkmalsextraktor, dessen Feature-Maps für nachfolgende Aufgaben verwendet werden. Diese Merkmale werden während des Optimierungsprozesses aus dem Datensatz gelernt. Sie können auch für andere Aufgaben wiederverwendet werden. In dieser Arbeit werden bereits auf ImageNet trainierte Backbones als Merkmalsextraktoren verwendet. Die Implementierung des Backbones wird aus *tensorflow* [11] übernommen. Durch die Verwendung bereits trainierter Backbones wird die Menge der Hyperparameter deutlich reduziert. In dieser Arbeit wird das bereits trainierte ResNet50 als Backbone mit einer Eingabegröße von 128 x 128 Pixel und einer Global-Average-Operation zur Transformation der Feature-Maps verwendet.

4.5 Conventional Image Processing Pipelines

Die Conventional-Image-Processing-Pipeline (CIPP) [3] entspricht einer statischen Abfolge von konventionellen Bildverarbeitungsoperationen. In diesem Abschnitt wird detailliert auf ihre Implementierung eingegangen. Die Implementierung legt besonderes Augenmerk auf eine einfache und intuitive Anwendung nach demselben Konzept wie DL-Methoden und wird in einem *Python*-Paket

zur Verfügung gestellt[3]. Wie in Abbildung 4.4 gezeigt, bietet diese Implementierung eines CIPP-Modells einen Rahmen, in dem der Benutzer diese Operationen ohne manuelle Parametereinstellung anordnen kann. Jede Operation hat einen vordefinierten Satz von Parametern.

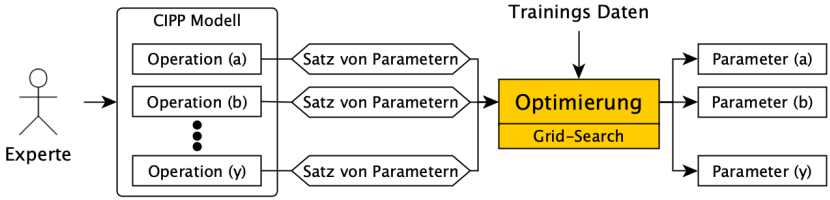


Abbildung 4.4: Optimierungsprozess eines CIPP: Die Reihenfolge der Operationen wird vom Anwender festgelegt und jeder Operation ist ein fester und vordefinierter Satz von Parametern zugeordnet. Während des Optimierungsprozesses werden die optimalen Parameter auf Basis der bereitgestellten Trainingsdaten durch Grid-Search ermittelt. [3]

Die Parameter mit der höchsten Segmentierungsgüte auf den Trainingsdaten werden dann automatisch ausgewählt, indem alle möglichen Kombinationen von Parametern (Grid-Search) auf dem Trainingsdatensatz getestet werden. Die Konfiguration mit der besten Bewertung wird als endgültige Parametrisierung ausgewählt und weiterverwendet. Neben Grid-Search bietet das Framework auch andere Optimierungsstrategien wie Random-Search oder genetische Algorithmen.

Die Parametrierung des CIPP-Modells zeigt Abbildung 4.5. Es wurde mit dem Ziel entwickelt, allgemein auf verschiedene Problemstellungen anwendbar zu sein. Wenn domänenspezifisches Wissen zur Verfügung steht, sind jedoch entsprechende Anpassungen empfohlen. Jedes Bild wird vor der Verarbeitung zunächst in ein Graustufenbild umgewandelt und auf eine Größe von 256 x 256 Pixel gebracht. Die Werte im Bild werden durch Division durch 255 auf einen Wertebereich zwischen 0 und 1 festgelegt. Diese Operationen sind statisch und werden bei der Optimierung ignoriert. Anschließend werden nacheinander die Operationen

[3] Die verwendete Implementierung wird in einem öffentlichen Git-Repository zur Verfügung gestellt: `https://github.com/FMuenke/cipp`

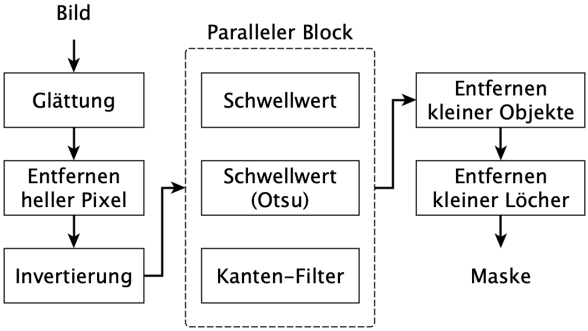


Abbildung 4.5: CIPP-Konfiguration: In dieser Arbeit wird eine CIPP mit insgesamt sechs Methoden verwendet. Die ersten drei Methoden *Glättung*, *Entfernen heller Pixel* und *Invertierung* bereiten die Segmentierung vor. Die vierte Methode wird aus drei Varianten (*Schwellwert*, *Schwellwert (Otsu)* [129] und *Kanten-Filter*) ausgewählt. Zum Schluss wird die Maske durch zwei Methoden nachbereitet: *Entfernen kleiner Objekte* und *Entfernen kleiner Löcher*.

Glätten, *Helle Pixel entfernen* und *Invertieren* angewendet. Diese Operationen dienen der Vorverarbeitung der Bilder, d.h. Rauschen wird reduziert, extreme Pixel werden entfernt und das Bild wird für die folgenden Operationen ausgerichtet. Die Segmentierung erfolgt dann durch die Operationen *Schwellwert*, *Schwellwert(Otsu)* [129] und *Kantenfilter*. Diese Operationen binarisieren das Bild und trennen so die Zielklasse vom Hintergrund eines Bildes. Abschließend werden in der Segmentierungsmaske kleine Restobjekte entfernt und kleine Restlücken gefüllt. Auf diese Weise kann die Segmentierungsmaske abschließend nachbearbeitet werden.

Die Implementierung der einzelnen Operationen mit den optimierbaren Parametern ist im Folgenden aufgelistet:

- Glätten [Filtergröße]: Dieser Vorverarbeitungsschritt entfernt potenzielles Rauschen und Störungen, indem das Bild mit einem Gauß-Filter geglättet wird. Während der Optimierung wird von der CIPP die Filtergröße 3, 5, 9, 17 bestimmt oder die Operation deaktiviert.

- Helle Pixel entfernen [Grenzwert]: Die Pixelwerte des Bildes werden an einem definierten Grenzwert abgeschnitten. Der Grenzwert wird hier durch die Perzentile der Pixelwerte im Bild adaptiv pro Bild gewählt. Zur Auswahl stehen die Perzentile 1, 5, 10, 25, 50.
- Invertierung [Aktivität]: Je nach Parametrierung wird das Bild invertiert. Dies kann für die Anwendung eines Schwellwertverfahrens erforderlich sein.
- Schwellwert [Schwellwert]: Die Eingabe wird durch einen Schwellwert binarisiert, d.h. alle Werte oberhalb des Schwellwertes entsprechen einer 1 und alle Werte unterhalb des Schwellwertes entsprechen einer 0. Es stehen die Schwellwerte 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 zur Verfügung.
- Schwellwert (Otsu) [Aktivität]: Die Eingabe wird durch einen Schwellwert binarisiert. Dieser Schwellwert wird nach dem Otsu-Verfahren [129] adaptiv für jedes Bild berechnet.
- Kantenfilter [Filtergröße]: Der Laplace-Operator wird auf das Bild angewendet. Das Ergebnis wird dann auf Werte zwischen 0 und 1 skaliert. Als Parameter wird hier die Filtergröße 3, 5, 9, 17 optimiert.
- Kleine Objekte entfernen [Objektgröße]: Die Operation entfernt alle Objekte im Bild, deren Fläche kleiner als ein vorgegebener Grenzwert ist[4]. Dieser Grenzwert entspricht dem Parameter 8, 32, 128, 256, 512 und wird optimiert oder die Operation wird für die Anwendung deaktiviert.
- Kleine Löcher entfernen [Lochgröße]: Die Operation entfernt alle Löcher im Bild, deren Fläche kleiner als ein vorgegebener Grenzwert ist[4]. Dieser Grenzwert entspricht dem Parameter 8, 32, 128, 256, 512 und wird optimiert oder die Operation wird für die Anwendung deaktiviert.

[4] Die Implementierung stammt aus dem Python-Paket Scikit-Image [177].

4.6 Strukturierte Segmentierung

Die strukturierte Segmentierung klassifiziert ein Bild Pixel für Pixel. In dieser Arbeit wurde ein Framework für die Anwendung strukturierter Segmentierungsalgorithmen entworfen[5]. Das entwickelte Framework integriert strukturierte Segmentierungsalgorithmen als modulare Komponenten, um die Erstellung effizienter und leistungsfähiger Mehrklassen-Segmentierungsmodelle zu ermöglichen.

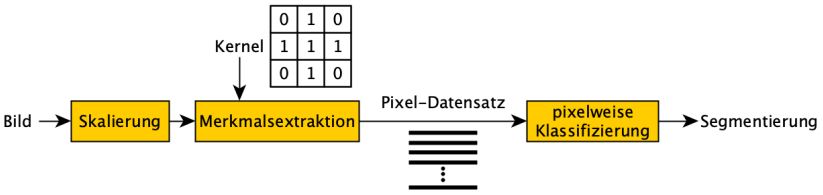


Abbildung 4.6: Konzept des strukturierten Klassifikators: Ein Bild wird Pixel für Pixel klassifiziert. Nach einer initialen Anpassung der Größe werden für alle Pixel Merkmale extrahiert und zu einen tabellarischen Datensatz transformiert. Dieser wird durch einen Klassifikator verarbeitet und in eine Segmentierung zurück transformiert.

Die einfachste Variante eines strukturierten Klassifikators (Pixel Segmentor: PX) zur strukturierten Segmentierung ist in Abbildung 4.6 dargestellt. Ein Bild wird zunächst auf die gewählte Größe skaliert. Anschließend wird das Bild mit den Dimensionen Höhe x Breite x Farbkanäle mit Hilfe des Kernels in einen Pixeldatensatz (ein Vektor pro Pixel) transformiert[6]. In dem neuen tabellarischen Datensatz hat jeder Pixel eine Zeile mit allen zugehörigen Merkmalen. Die Merkmale werden durch den definierten Kernel bestimmt. Der Kernel definiert eine Nachbarschaft von Pixeln, wobei jeder aktive Nachbar mit einer 1 gekennzeichnet ist. Für jedes Pixel werden die Werte der benachbarten und aktiven Pixel ausgewählt und als Merkmale für das betrachtete zentrale Pixel gespeichert. Anschließend wird

[5] Das beschriebene Framework ist eine Weiterführung von [8] und wird in einem öffentlichen Git-Repository zur Verfügung gestellt: `https://github.com/FMuenke/structured_segmentation`

[6] Die Implementierung verwendet hierfür Funktionen aus *NumPy* [74]

der Datensatz klassifiziert, sodass jedem Pixel eine Klasse zugeordnet wird. Diese Klassen werden dann wieder in die Form (Höhe x Breite) des Bildes transformiert. So entsteht eine Segmentierung für ein Eingabebild. Ein strukturierter Klassifikator ist in der Lage, Farb- und Graustufenbilder beliebiger Größe während des Trainings und der Anwendung zu verarbeiten. Als Klassifikatoren innerhalb des PX können beliebige Modelle (z.B. aus *Scikit-Learn* [133]) verwendet werden.

Während des Trainings werden aus allen Trainingsbildern separate Pixeldatensätze erzeugt und anschließend zusammengefasst. Der Pixeldatensatz ermöglicht eine gezielte Beeinflussung des Trainings durch die Auswahl von Pixeln. Pixel in einem Bild sind sich oft ähnlich und enthalten redundante Informationen. Um das Training zu beschleunigen, werden daher nicht alle Pixel eines Bildes für das Training ausgewählt. Nur ein zufälliger Prozentsatz der Pixel wird für das Training ausgewählt. Die Auswahl von Pixeln für das Training bietet weitere Vorteile. So unterstützt diese Implementierung die Verarbeitung von Datensätzen mit hohem Klassenungleichgewicht: Das Klassenungleichgewicht kann durch eine gezielte Auswahl der Pixel entsprechend ihrer Klasse reduziert werden, sodass prozentual mehr Pixel der Klassen mit wenigen Beispielen für das Training ausgewählt werden. Außerdem wird das Training auf nur teilweise annotierten Bildern ermöglicht[7].

Der Kernel schränkt einen einzelnen PX auf den Kontext innerhalb des Bildes ein. Jedes Pixel kann nur anhand seiner direkten Nachbarn klassifiziert werden. Um dieses Problem zu lösen, wurde das Encoder-Decoder-Modell (StED-Modell) [8] vorgestellt. Das Konzept dieses Modells ist in Abbildung 4.7 dargestellt. Es besteht aus mehreren einzelnen strukturierten Klassifikatoren, die das Bild nacheinander verarbeiten.

[7] Für das Training mit teilweise annotierten Bildern werden die nicht annotierten Pixel aus dem tabellarischen Pixeldatensatz herausgefiltert und während des Trainings ignoriert. Auf diese Weise können die annotierten Pixel trotzdem vollständig für das Training verwendet werden, da die Merkmale auf dem gesamten Bild generiert werden.

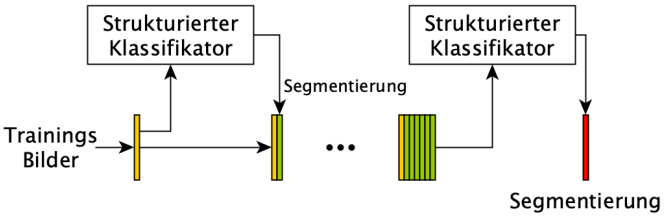


Abbildung 4.7: Das StED-Modell ist ein Ensemble von sechs Klassifikatoren, wobei jeder Klassifikator seine Vorhersagen für die nachfolgenden Klassifikatoren als Merkmale bereitstellt. Jeder Klassifikator hat seine eigene Skalierung und wird nur auf einem Drittel der Bilder trainiert [8].

Die Vorhersagen aller vorhergehenden Klassifikatoren[8] dienen als Eingang für den nächsten Klassifikator. Jeder Klassifikator verwendet den gleichen Kernel[9], lernt aber die Merkmale auf einer anderen Skala[10]. Die erste Hälfte der Klassifikatoren skaliert das Bild bei jedem Schritt um den Faktor 2 herunter, sodass diese Klassifikatoren bei jedem Schritt größere Zusammenhänge im Bild lernen. Die zweite Hälfte skaliert das Bild bei jedem Schritt um den Faktor 2 hoch, sodass die letzte Segmentierung wieder der Originalgröße entspricht[11]. Da mehrere Klassifikatoren im Einsatz sind, wird für jeden Klassifikator zufällig nur ein Bruchteil aller Bilder zum Training verwendet (Bootstrapping). Bootstrapping reduziert Overfitting und das Training wird effizienter, da die Trainingszeit reduziert wird[12].

Das Framework wurde so entwickelt, dass es dasselbe Datenformat wie DL-Methoden verwendet, wodurch die Kompatibilität mit verschiedenen Annotationsmethoden und Frameworks, die speziell für DL-Anwendungen entwickelt wurden, gewährleistet ist. Darüber hinaus können diese Modelle leicht angepasst werden.

8 Es wird der HistGradientBoostingClassifier verwendet.
9 Hier wird ein Ellipsen-förmiges Kernel mit einer Größe von 5 x 5 Pixeln genutzt.
10 Hier ist die ursprüngliche Eingabe ein Graustufenbild der Größe 256 x 256 Pixel.
11 Hier werden insgesamt 6 Klassifikatoren (3 herunterskalierend, 3 hochskalierend) verwendet.
12 Durch die Künstliche Verkleinerung der Datensätze für jeden Klassifikator ist das Training weniger aufwändig

4.7 Deep-Learning basierte Bildsegmentierung

Das U-Net [148] ist ein Standardverfahren zur semantischen Segmentierung von Bildern. Das verwendete U-Net ist in dem Python-Paket *segmentation models* [82] implementiert[13].

DNNs benötigen im Allgemeinen große Datenmengen für die Optimierung. Um das U-Net optimal auf die Dateneffizienz-Experimente vorzubereiten, wird das ResNet18 [77] Backbone als kleinste Version der ResNet-Reihe eingesetzt. Die relativ geringe Anzahl von Parametern entspricht der geringen Anzahl von Trainingsbildern[14], mit denen es in den folgenden Versuchen trainiert wird. Zusätzlich wird das Backbone mit bereits auf dem ImageNet trainierten Gewichten [151] als Startpunkt für jedes Training verwendet, sodass das U-Net von bereits gelernten Merkmalen profitieren kann[15]. Zuletzt werden Augmentationsmethoden (s. Anschnitt 2.7) angewendet, die das Bild verfremden, um ein möglichst robustes Training zu gewährleisten. Der Faktor der Augmentationen entspricht der Häufigkeit, mit denen eine Augmentation auf ein Bild angewendet wird. Diese Augmentationsmethoden umfassen:

- `horizontal_flip` (Faktor=0.17): Das Bild wird an der Horizontalen gespiegelt. Die Pixelwerte des Bildes werden dabei nicht verändert.
- `vertical_flip` (Faktor=0.17): Das Bild wird an der Vertikalen gespiegelt. Die Pixelwerte des Bildes werden dabei nicht verändert.
- `crop` (Faktor=0.17): Diese Funktion schneidet einen kleinen Bereich aus dem Eingabebild aus. Dabei werden auf jeder Seite zufällig zwischen 0 und 20 Prozent des Bildes abgeschnitten.

[13] Die verwendete Implementierung für das Training ist in einem öffentlichen Git-Repository verfügbar: `https://github.com/FMuenke/semantic_segmentation`

[14] Eingabe-Format: 256×256 [RGB-Farbraum]

[15] Als Algorithmus zur Optimierung wird Adam mit einer Lernrate von 10^{-4} und einer maximalen Batchgröße von 8 Bildern verwendet. Die Loss-Funktion ist die Binary-Crossentropy.

- `rotation` (Faktor=0.17): Das Bild wird zufällig um 0, 90, 180, 290 Grad gedreht. Die Pixelwerte des Bildes werden nicht verändert.

Um Overfitting während des Trainings zu vermeiden, wird Early-Stopping [21] verwendet[16]. Für den Validierungsdatensatz werden 20% der Trainingsdaten oder bei wenigen Bildern mindestens ein Bild verwendet. Dadurch kann sichergestellt werden, dass das U-Net auch auf unbekannten Daten eine gute Segmentierungsgüte erreicht. Sinkt die Segmentierungsgüte im Verlauf des Trainings auf den ausgewählten Daten zur Validierung, generalisiert das U-Net nicht mehr. Das U-Net mit dem niedrigsten Validierungs-Loss wird anschließend als trainiertes Modell in der Auswertung verwendet.

[16] Das Training wird nach 32 Epochen ohne Verbesserung des Validierungs-Loss gestoppt.

5 Experimente zur Bewertung der Dateneffizienz von Methoden zur Bildsegmentierung

5.1 Übersicht

In Abschnitt 3.4.3 wurden drei Modelle zur Bildsegmentierung (U-Net, CIPP und StED) als potentielle Algorithmus-Komponenten für die ABV ausgewählt. Die Annotation von Bilddaten für die semantische Segmentierung ist aufwendig und kostenintensiv, so ist die Dateneffizienz der Modelle von großer Wichtigkeit. In Abschnitt 3.4.4 wurde ein Vorgehen zur Bewertung der Dateneffizienz in Abhängigkeit der SAP vorgestellt. Jede Segmentierungsaufgabe einer Klasse K kann nach den SAP kategorisiert werden:

- Flächenanteil der Klasse a_K:
  - kleiner Flächenanteil $(0, 0.01]$
  - mittlerer Flächenanteil $(0.01, 0.1]$
  - großer Flächenanteil $(0.1, 0.99]$
- Art der Störfaktoren in den Bilddaten S_K:
  - visuelle Störfaktoren λ_v
  - geometrische/visuelle Störfaktoren λ_{v+g}.

In diesem Kapitel werden zuerst Datensätze für die Durchführung von Experimenten ausgewählt und die SAP der Segmentierungsaufgaben bewertet. Jede vorkommende Klasse in einem Datensatz entspricht einer eigenen Segmentierungsaufgabe. Die Art der Störfaktoren S_K wird für den Datensatz als Ganzes bestimmt und der Flächenanteil a_K wird einzeln für jede Klasse aus den annotierten Masken im Datensatz berechnet. Die Modelle werden dann auf den Segmentierungsaufgaben trainiert und evaluiert. Basierend auf den Ergebnissen der Experimente lassen sich Anwendungsempfehlungen in Abhängigkeit der SAP ableiten. Ziel ist es, einen Anwender zu befähigen, das ideale Modell (U-Net, CIPP und StED) als Algorithmus-Komponente für einen Algorithmus zur ABV auszuwählen.

5.2 Datensätze

5.2.1 Übersicht

Für die ABV sind im Besonderen die Domänen *Remote-Sensing* und *Fahrzeug-Bilddaten* von Interesse. Bilder aus diesen Domänen können leicht großflächig erhoben werden. Die Domäne *Remote-Sensing* ist dabei in der Regel durch visuelle Störfaktoren λ_v beeinträchtigt, während *Fahrzeug-Bilddaten* sowohl durch visuelle und geometrisch Störfaktoren λ_{v+g} beeinflusst werden. Zusätzlich werden Segmentierungsaufgaben aus allen Intervallen von Flächenanteilen a_K benötigt.

Als Basis für die Experimente wurden entsprechend drei reale Datensätze ausgewählt, die alle sechs definierten Kombinationen der SAP abdecken. Es handelt sich um die folgenden und unabhängig veröffentlichten Datensätze:

- PotholeMix [139]: Der Datensatz enthält die Segmentierung von Schäden auf asphaltierten Straßen. Der Datensatz enthält die Klassen *Riss* und *Schlagloch*. Dieser Datensatz besteht aus sechs Teildatensätzen, so lässt sich der Datensatz als Ganzes oder jeder Teildatensatz für sich auswerten. Zwei

der Datensätze stammen aus der Domäne *Fahrzeug-Bilddaten*. So ergeben sich viele unterschiedliche SAP-Kombinationen.

- Road Traversing Knowledge (RTK) [141]: Der Fokus des Datensatzes liegt auf Bildern, die aus einem Fahrzeug heraus aufgenommen wurden. Dieser Datensatz beinhaltet eine Vielzahl der verschiedenen Klassen mit unterschiedlichen Texturen, sodass die Domäne *Fahrzeug-Bilddaten* vollständig abgedeckt werden kann.
- FloodNet [138]: Dieser Datensatz, bestehend aus Drohnen-Aufnahmen (Unmanned Aerial Vehicle: UAV), repräsentiert die Domäne *Remote Sensing*. UAVs sind besonders relevant für die ABV, da sie deutlich leichter für die großflächige Datenerhebung eingesetzt werden können als herkömmliche Satelliten. Es steht eine Auswahl von mehreren Klassen mit unterschiedlichen Texturen zur Verfügung.

Die ausgewählten Datensätze bilden in ihrer Gesamtheit sowohl zwei wichtige Domänen (*Remote-Sensing* und *Fahrzeug-Bilddaten*), als auch eine Vielzahl von Segmentierungsaufgaben mit unterschiedlichen Flächenanteilen a_K und Störfaktoren S_K ab. Im Folgenden werden nun jeweils die ausgewählten Datensätze im Detail vorgestellt.

5.2.2 PotholeMix

Der erste Datensatz ist der PotholeMix Datensatz [139]. Dieser enthält Segmentierungsmasken von Rissen und Schlaglöchern. Die Segmentierung von Rissen ist komplex, da sie nur einen kleine Flächenanteil a_K haben. Im Gegensatz dazu sind Schlaglöcher kompakt und zeichnen sich durch ihre Textur aus. Der Datensatz besteht aus sechs Teildatensätzen mit unterschiedlichen Perspektiven, Abständen, Auflösungen und Hintergründen. Beispielbilder sind zur Veranschaulichung in Abbildung 5.1 dargestellt. Die Zusammensetzung aus unterschiedlichen Datensätzen erhöht die Diversität und stellt eine große Herausforderung für Algorithmen dar. Im Folgenden werden die Teildatensätze vorgestellt:

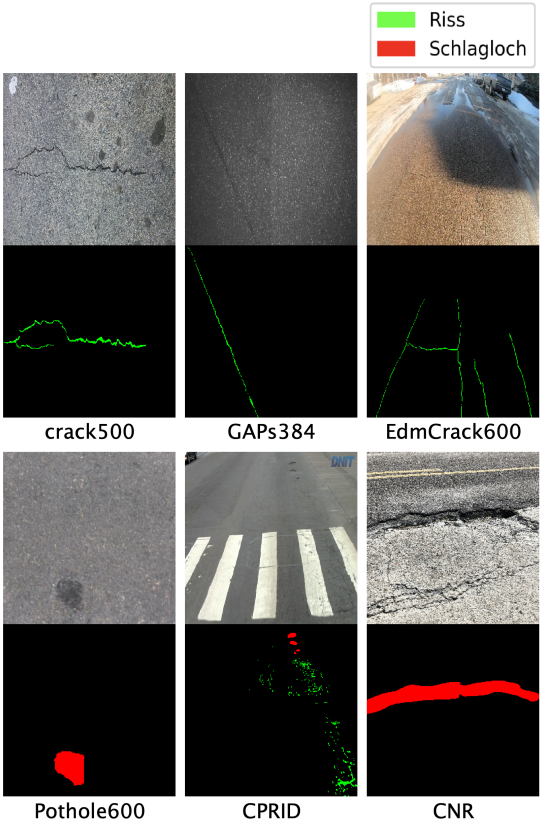


Abbildung 5.1: Beispielbilder aus den Teildatensätzen des PotholeMix Datensatzes und der dazugehörigen Annotationen. Alle Datensätze thematisieren dieselben Klassen im Bereich von Straßenschäden (*Riss* und *Schlagloch*) unterscheiden sich jedoch durch Auflösung, Winkel und Abstand.

- *Crack500* [201]: Hochauflösende Bilder zeigen unterschiedliche Risse, die aus der Vogel-Perspektive aufgenommen wurden. Dabei unterscheidet sich das Material des Hintergrunds, die Beleuchtung, die Breite und die Verzweigung des Risses.

- *GAPs384* [61]: Risse werden aus der Vogel-Perspektive durch hochauflösende Bilder abgebildet. Die Risse sind schmaler und kürzer als im Crack500 Datensatz.
- *EdmCrack600* [119]: Die hochauflösenden Bilder wurden aus einem fahrenden Fahrzeug heraus aufgenommen und zeigen Risse. Zur Aufnahme wurde eine kommerziell verfügbare Kamera (GoPro Hero 7) am Heck mit einem Winkel von 45 Grad befestigt. Die Bilder zeigen einen vielfältigen Hintergrund (Schnee, Bordstein, ...) und werden durch weitere Störungen, wie Schatten, direkte Sonneneinstrahlung und nasse Fahrbahn, beeinträchtigt.
- *Pothole600* [63]: Die Bilder zeigen Schlaglöcher aus der Vogel-Perspektive. Der Hintergrund der Bilder unterscheidet sich visuell nicht und besteht stets aus glattem und unbeschädigtem Asphalt. Die Schlaglöcher sind im Fokus des Bildes.
- *CPRID* [132]: Es werden sowohl Risse als auch Schlaglöcher in den Bildern abgebildet. Die Bilder wurden aus einem speziellen Fahrzeug zur Zustandserfassung von Autobahnen mit einer hochauflösenden Videokamera heraus aufgenommen. Schatten, Dreck und Fahrzeuge stören die Bilder und der Abstand zur Straße ist größer als bei den anderen Datensätzen.
- *CNR* [139]: Die Bilder haben eine variable Auflösung und zeigen verschiedene Schlaglöcher. Die Bilder haben unterschiedliche Perspektiven und zeigen das Schlagloch im Fokus.

Bei den Teildatensätzen, die konstant aus der Vogel-Perspektive aufgenommen wurden, sind ausschließlich visuelle Störfaktoren λ_v vorhanden. Im Gegensatz dazu sind die Teildatensätze, die während der Fahrt aufgezeichnet wurden durch visuelle und geometrische Störfaktoren λ_{v+g} beeinflusst[1].

[1] Durch die Bewegung des Fahrzeugs werden automatisch verschiedene Winkel und Abstände zu den Objekten angenommen.

Die Flächenanteile a_K der Klassen *Riss* und *Schlagloch* unterscheiden sich je nach Datensatz erheblich. Aus dem PotholeMix Datensatz und dessen Teildatensätzen lassen sich insgesamt acht Segmentierungsaufgaben[2] mit konkreten SAP ableiten. Die Auflistung der Segmentierungsaufgaben und deren SAP sind in Tabelle 5.1 aufgeführt.

Segmentierungsaufgabe	Flächenanteil a_K	Störfaktoren S_K
Riss (PotholeMix)	0.011	λ_{v+g}
Schlagloch (PotholeMix)	0.011	λ_{v+g}
Riss (Crack500)	0.030	λ_v
Riss (GAPs384)	0.003	λ_v
Riss (EdmCrack600)	0.007	λ_{v+g}
Schlagloch (Pothole600)	0.079	λ_v
Riss (CPRID)	0.009	λ_{v+g}
Schlagloch (CPRID)	0.003	λ_{v+g}

Tabelle 5.1: Segmentierungsaufgaben PotholeMix Datensatz: Durch die Kombination von Klasse und Datensatz/Teildatensatz lassen sich aus dem PotholeMix Datensatz insgesamt acht Segmentierungsaufgaben definieren. Jede Segmentierungsaufgaben hat eigene SAP.

5.2.3 Road Traversing Knowledge

Der RTK-Datensatz [141] repräsentiert die Domäne *Fahrzeug-Bilddaten* und enthält detaillierte Informationen über die Fahrbahnbeschaffenheit. Der Datensatz besteht aus insgesamt 701 Bildern (Training: 601, Test: 100) mit einer Auflösung von 352 x 288 Pixel, die mit einer kostengünstigen Kamera (HP Webcam HD-4110) aufgenommen wurden. Diese wurden während der Fahrt aus einem Fahrzeug in Brasilien aufgezeichnet. Die Abbildung 5.2 zeigt Beispiele, die den RTK-Datensatz repräsentieren.

[2] Da der CNR Datensatz nur aus 19 Bildern besteht, wird er hier nicht als eigene Segmentierungsaufgabe betrachtet, sondern nur als Teil des PotholeMix Datensatzes.

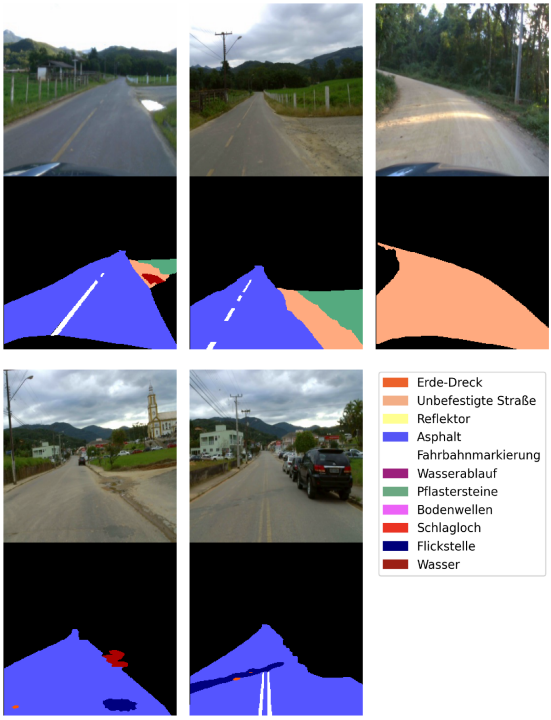


Abbildung 5.2: Beispielbilder aus dem RTK-Datensatz und den dazugehörigen Annotationen. Der Datensatz repräsentiert die Domäne *Fahrzeug-Bilddaten* und eine Vielzahl von SAP-Kombinationen.

Insgesamt werden elf Klassen unterschieden: *Erde/Dreck, Asphalt, Pflastersteine, unbefestigte Straße, Fahrbahnmarkierungen, Bodenwellen, Reflektoren, Wasserabläufe, Flickstellen, Wasser/Schlamm* und *Schlaglöcher*. In Tabelle 5.2 sind die Segmentierungsaufgaben und ihre SAP aufgelistet. Die Klassen sind sehr selten, sodass einige Segmentierungsaufgaben eine Flächenanteil $a_K < 0.001$ aufweisen.

Segmentierungsaufgabe	Flächenanteil a_K	Störfaktoren S_K
Erde/Dreck (RTK)	0.007	λ_{v+g}
Unbefestigte Straße (RTK)	0.042	λ_{v+g}
Reflektor (RTK)	< 0.001	λ_{v+g}
Asphalt (RTK)	0.105	λ_{v+g}
Fahrbahnmarkierung (RTK)	0.004	λ_{v+g}
Wasserablauf (RTK)	< 0.001	λ_{v+g}
Pflastersteine (RTK)	0.202	λ_{v+g}
Bodenwellen (RTK)	0.004	λ_{v+g}
Schlaglöcher (RTK)	< 0.001	λ_{v+g}
Flickstelle (RTK)	< 0.001	λ_{v+g}
Wasser/Schlamm (RTK)	< 0.001	λ_{v+g}

Tabelle 5.2: Segmentierungsaufgaben RTK-Datensatz: Für den RTK-Datensatz lassen sich 11 Segmentierungsaufgaben mit individuellen SAP definieren. Da die Segmentierungsaufgaben alle auf denselben Bilddaten basieren und sich die Perspektive und Abstand während der Aufnahme ändert, haben alle dieselbe Art von Störfaktoren S_K (λ_{v+g}).

5.2.4 FloodNet

Der FloodNet Datensatz [138] zeigt Bilder aus überschwemmten Gebieten und wurde mit einer Drohne (DJI Mavic Pro Quadcopter) nach dem Hurrikan Harvey aufgenommen. Beispielbilder sind in Abbildung 5.3 zu sehen. Mit dem Datensatz sollen Computer Vision Modelle trainiert werden, die Schäden nach Naturkatastrophen abschätzen können. Der Datensatz besteht aus insgesamt 2343 Bildern (Training: 1995, Test: 448) mit einer Auflösung von 4000 x 3000 Pixel.

Die Bilder wurden pixelgenau für die semantische Segmentierung annotiert. Dabei werden insgesamt neun Klassen unterschieden: *Überflutetes Gebäude*, *Gebäude*, *Überflutete Straße*, *Straße*, *Wasser*, *Baum*, *Fahrzeug*, *Pool* und *Gras*. Die SAP-Kombinationen der Segmentierungsaufgaben sind in Tabelle 5.3 aufgelistet.

Abbildung 5.3: Beispielbilder aus dem FloodNet-Datensatz und den dazugehörigen Annotationen. Der Datensatz repräsentiert die Domäne *Remote-Sensing* und eine Vielzahl an SAP-Kombinationen.

5.3 Ergebnisse

Aus den vorgestellten Datensätzen lassen sich insgesamt 28 Segmentierungsaufgaben bilden. Die Segmentierungsaufgaben lassen sich nun nach den definierten SAP gruppieren. Durch die Auswahl der Datensätze ist sichergestellt, dass für jede Kombination von SAP mindestens zwei Segmentierungsaufgaben vorhanden sind. Die Zusammenfassung der Segmentierungsaufgaben und deren Verteilung der SAP sind in Tabelle 5.4 zu sehen. Die meisten Segmentierungsaufgaben befinden sich im Bereich $a_K = (0, 0.01]$ und $S_k = \lambda_{v+g}$. Diese Häufung ergibt

Segmentierungsaufgabe	Flächenanteil a_K	Störfaktoren S_K
Überflutetes Gebäude (FloodNet)	0.008	λ_v
Gebäude (FloodNet)	0.020	λ_v
Überflutete Straße (FloodNet)	0.010	λ_v
Straße (FloodNet)	0.033	λ_v
Wasser (FloodNet)	0.062	λ_v
Baum (FloodNet)	0.102	λ_v
Fahrzeug (FloodNet)	0.001	λ_v
Pool (FloodNet)	0.001	λ_v
Gras (FloodNet)	0.320	λ_v

Tabelle 5.3: Segmentierungsaufgaben FloodNet Datensatz: Für den FloodNet-Datensatz lassen sich 9 Segmentierungsaufgaben mit individuellen SAP definieren. Da die Segmentierungsaufgaben alle auf denselben Bilddaten basieren und sich die Perspektive und Abstand während der Aufnahme nicht ändert, haben alle dieselbe Art von Störfaktoren S_K (λ_v).

sich durch seltene Klassen in Datensätzen aus der Domäne *Fahrzeug-Bilddaten*. Seltene Klassen tauchen nur selten auf Bildern auf und beziehen sich auf kleine Objekte (z.B. Reflektoren im RTK-Datensatz).

Für jede der 28 Segmentierungsaufgaben werden nun die drei ausgewählten Modelle trainiert und evaluiert. Zur Bewertung der Dateneffizienz werden, wie in Abschnitt 3.4.4 beschrieben, verschiedene Anzahlen an Trainingsbildern N_{img} (4, 8, 16, 32, 64) getestet. So kann für jede Kombination aus SAP eine Dateneffizienz-Kurve berechnet werden. Um statistische Schwankungen des Optimierungsprozesses und den Einfluss der zufälligen Auswahl der Trainingsbilder auszugleichen, wird jeder Algorithmus zehnmal trainiert und ausgewertet. Alle Modelle verarbeiten Bilder derselben Größe (256×256 Pixel) und werden für jedes Experiment auf denselben Bildern trainiert.

Zuerst wird der grundsätzliche Einfluss der SAP-Flächenanteil a_K und Art der Störfaktoren S_K diskutiert. Hierfür wird die durchschnittliche Güte G_4^{64} über alle Experimente und Anzahlen der Bilder ermittelt. Die Güte G_4^{64} ist damit unabhängig von der Anzahl der Trainingsbilder N_{img}. Die Güte G_4^{64} repräsentiert die Dateneffizienz, da das Intervall von $N_{\text{img}} = [4, 64]$ nur einen Bruchteil der zur

	$S_k = \lambda_v$	$S_k = \lambda_{v+g}$
$a_K = (0, 0.01]$	4	11
$a_K = (0.01, 0.1]$	6	3
$a_K = (0.1, 0.99]$	2	2

Tabelle 5.4: SAP der Segmentierungsaufgaben: Die Verteilung der SAP über die 28 vorgestellten Segmentierungsaufgaben.

Verfügung stehenden Trainingsdaten beinhaltet. Die Ergebnisse sind in Tabelle 5.5 zusammengefasst.

Die Güte G_4^{64} steigt deutlich für jedes Modell mit wachsendem Flächenanteil a_K der Segmentierungsaufgabe. Im Durchschnitt steigert sich die Güte G_4^{64} im Vergleich zum nächst größeren Flächenanteil a_K für das Modell CIPP um $+0.17$, für das Modell StED um $+0.20$ und für das Modell U-Net um $+0.13$. Das Modell StED ist damit am stärksten und das U-Net am wenigsten stark von einem kleinen Flächenanteil a_K betroffen. Dieses Verhalten ist zu erwarten, da ein kleiner Flächenanteil a_K die Segmentierung erschwert. Die Art der Störfaktoren haben ebenfalls einen Einfluss auf die die Güte G_4^{64}.

Störfaktor S_K	Flächenanteil a_K	CIPP*	StED*	U-Net
λ_v	$(0, 0.01]$	**0.044**	0.038	0.015
λ_v	$(0.01, 0.1]$	0.261	**0.266**	0.156
λ_v	$(0.1, 0.99]$	0.477	**0.486**	0.249
λ_{v+g}	$(0, 0.01]$	0.018	**0.031**	0.026
λ_{v+g}	$(0.01, 0.1]$	**0.105**	0.082	0.064
λ_{v+g}	$(0.1, 0.99]$	0.265	**0.398**	0.308
Insgesamt		0.134	**0.147**	0.092

Tabelle 5.5: Durchschnittliche Güte G_4^{64} nach SAP: Die Güte G_4^{64} fasst alle F1-Scores eines Modells für eine Kombination von SAP zusammen. Dadurch kann der Einfluss der SAP unabhängig von der Anzahl der Trainingsbilder N_{img} betrachtet werden. Die besten Ergebnisses sind **markiert**. Die eigens entworfenen Modelle sind mit einen * markiert.

Im Vergleich der Störfaktoren $S_K = \lambda_v$ und $S_K = \lambda_{v+g}$ für jeden Flächenanteil a_K sinkt die Güte G_4^{64} für das Modell CIPP um -0.132, für das Modell StED um -0.093 und für das Modell U-Net um -0.007. Die reduzierte Güte ist zu erwarten, da durch zusätzliche geometrische Störfaktoren die Komplexität erhöht wird. Das Modell CIPP ist dabei am meisten von zusätzlichen geometrischen Störfaktoren betroffen, da die einfachen Bildverarbeitungsmethoden der KBV solche geometrischen Veränderungen nicht ausgleichen können. Das U-Net hingegen ist kaum durch zusätzliche Störfaktoren beeinträchtigt. Es wird vermutet, dass die Verwendung des vortrainierten ResNet18 als Backbone das U-Net bei der Verarbeitung von geometrischen Störfaktoren unterstützt.

Insgesamt ist die Güte G_4^{64} für alle Modelle bei einem kleinen Flächenanteil a_K mit $G_4^{64} < 0.05$ sehr niedrig und ähnlich für alle Modelle. Daher wird der Einsatz der Modelle bei einem Flächenanteil $a_K < 0.01$ und $N_{\text{img}} \leq 64$ nicht empfohlen. Stattdessen sollten andere Methoden oder mehr Trainingsdaten in Betracht gezogen werden. Bei Segmentierungsaufgaben mit einem mittleren Flächenanteil $a_K = (0.01, 0.1]$ erreichen die Modelle StED und CIPP eine ähnliche Güte $G_4^{64} = 0.26$, während das U-Net mit $G_4^{64} = 0.16$ (-0.1) zurückliegt. Der Unterschied bei Störfaktoren $S_K = \lambda_{v+g}$ ist kleiner. Somit liegt die Güte der Modelle im Bereich $[0.064, 0.105]$. Damit sind die Modelle CIPP und StED unabhängig für die Art der Störfaktoren eine gute Option für einen Flächenanteil $a_K = (0.01, 0.1]$.

Für einen hohen Flächenanteil $a_K = (0.1, 0.99]$ und visuelle Störfaktoren $S_K = \lambda_v$ erreichen die Modelle CIPP ($G_4^{64} = 0.48$) und StED ($G_4^{64} = 0.49$) die höchste Güte, während das U-Net ($G_4^{64} = 0.25$) schlechter abschneidet. Dies ändert sich für Segmentierungsaufgaben mit visuellen und geometrischen Störfaktoren. In diesem Fall erreicht das StED-Modell die höchste Güte $G_4^{64} = 0.4$ und das U-Net ist erstmals auf dem zweiten Platz mit einer Güte $G_4^{64} = 0.31$.

Über alle Versuche erreicht das Modell StED in vier von sechs SAP-Kombinationen die höchste Güte G_4^{64}, gefolgt von der CIPP mit zwei von sechs Kombinationen. So zeigt sich, dass trotz des Einflusses von SAP das StED-Modell in den meisten

Fällen für das Intervall von $N_{\text{img}} = [4, 64]$ Trainingsbildern die besten Ergebnisse liefert. Das U-Net liegt in jeder Kategorie hinter den CIPP und StED zurück.

Im Folgenden wird der Einfluss von der Anzahl der Trainingsdaten N_{img} auf die Kombinationen der SAP diskutiert. Die Ergebnisse zu den Dateneffizienz-Kurven der Segmentierungsaufgaben für visuelle Störfaktoren $S_k = \lambda_v$ sind in Abbildung 5.4 zusammengefasst. Der F1-Score steigt für einen kleinen Flächenanteil $a_K = (0, 0.01]$ für die Modelle StED und CIPP leicht, während das U-Net konstant bleibt. Im Bereich von einem mittleren Flächenanteil $a_K = (0.01, 0.1]$ steigern sich im Besonderen die Modelle StED und U-Net mit steigender Anzahl an Bildern. Die Streuung der F1-Scores ist jedoch für jedes Modell hoch. Bei einer Segmentierungsaufgabe mit hohem Flächenanteil $a_K = (0.1, 0.99]$ erreichen die Modelle StED und CIPP für alle Anzahlen an Trainingsbildern N_{img} einen höheren F1-Score als das U-Net, wobei das U-Net mit steigender Anzahl an Bildern aufholt.

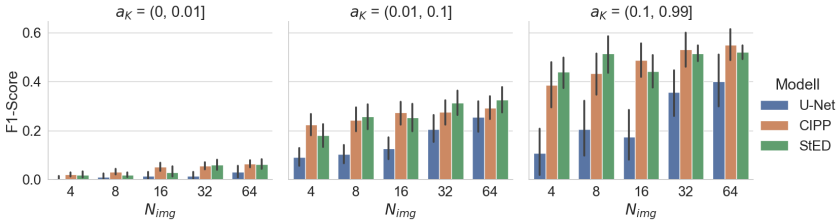


Abbildung 5.4: Dateneffizienz-Kurve für visuelle Störfaktoren: Verlauf des F1-Scores für verschiedene Anzahlen an Trainingsbildern N_{img} für jedes Flächenanteil a_K Intervall.

Die Ergebnisse für eine Kombination aus visuellen und geometrischen Störfaktoren $S_K = \lambda_{v+g}$ werden in Abbildung 5.5 dargestellt. Auch für $S_K = \lambda_{v+g}$ steigert sich der F1-Score aller Modelle mit zunehmender Anzahl an Trainingsbildern N_{img}. Das U-Net überholt die Modelle StED und CIPP für einen Flächenanteil $a_K = (0.01, 0.1]$ bei 64 und für einen Flächenanteil von $a_K = (0.1, 0.99]$ sogar bei 32 Trainingsbildern. Der F1-Score bei visuellen und geometrischen Störfaktoren $S_K = \lambda_{v+g}$ ist niedriger als bei ausschließlich visuellen Störfaktoren λ_v,

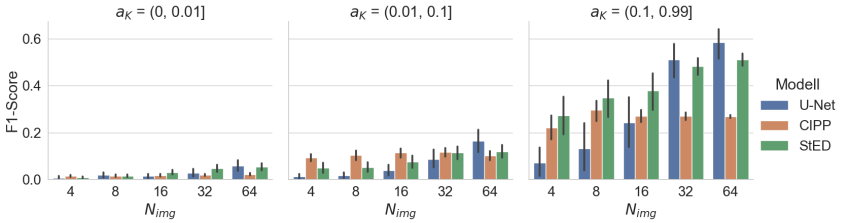


Abbildung 5.5: Dateneffizienz-Kurve für visuelle und geometrische Störfaktoren: Verlauf des F1-Scores für verschiedene Anzahlen an Trainingsbildern N_{img} für jedes Flächenanteil a_K Intervall.

sodass das U-Net mit zunehmender Zahl an Trainingsbildern auf- und überholen kann. Für Segmentierungsaufgaben mit hohem Flächenanteil $a_K = (0.1, 0.99]$ zeigt sich, dass der F1-Score der CIPP ab $N_{\text{img}} = 16$ stagniert, während StED und U-Net weiter steigen. Durch die Bildverarbeitungsmethoden der KBV, kann sich die CIPP nicht an geometrischen Störfaktoren anpassen.

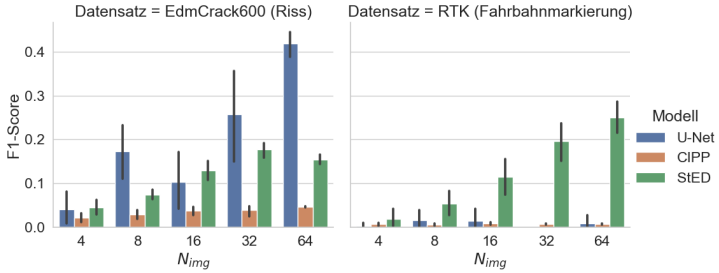


Abbildung 5.6: Dateneffizienz-Kurven der Ausreißer: Riss (EdmCrack600) und Fahrbahnmarkierung (RTK)

Es gibt zwei Segmentierungsaufgaben, die als Ausreißer hervorstechen: *Riss (EdmCrack600)* und *Fahrbahnmarkierung (RTK)*. Die Segmentierungsaufgabe *Riss (EdmCrack600)* hat einen kleinen Flächenanteil $a_K = 0.007$ und Störfaktoren $S_K = \lambda_{v+g}$ und wird als einzige Segmentierungsaufgabe mit diesen SAP mit einem F1-Score > 0.4 für $N_{\text{img}} = 64$ von dem Modell U-Net gelöst. Die

Abbildung 5.6 zeigt die Dateneffizienz-Kurve der Segmentierungsaufgabe *Riss (EdmCrack600)*. Es zeigt sich, dass das U-Net fähig ist, auch Aufgaben mit einem kleinen Flächenanteil dateneffizient zu segmentieren. Besonders bemerkenswert ist die Segmentierungsaufgabe *Riss (EdmCrack600)*, bei der sich auf jedem Bild ein zusammenhängender Riss über einen Großteil des Bildes erstreckt. Im Gegensatz dazu weisen ähnliche Datensätze wie CRIPD oder GAPs384 entweder Risse auf, die auf einen kleinen Bereich des Bildes beschränkt sind, oder es ist nicht auf jedem Bild ein Riss zu sehen.

Die Segmentierungsaufgabe *Fahrbahnmarkierung (RTK)* hat ebenfalls einen kleinen Flächenanteil $a_K = 0.004$ und Störfaktoren $S_K = \lambda_{v+g}$. Die Dateneffizienz-Kurve ist in Abbildung 5.6 zu sehen. Hier erreicht das StED-Modell für $N_{\text{img}} = 64$ einen F1-Score > 0.24 und liegt damit weit vor den anderen Modellen. Die Segmentierungsaufgabe *Fahrbahnmarkierung (RTK)* ist aufgrund der einheitlichen und deutlich weißen Färbung der Markierungen relativ einfach. Jedoch sind Fahrbahnmarkierungen nicht auf jedem Bild zu sehen. Durch das Ausbalancieren (s. Abschnitt 4.6) der verwendeten Pixel während des Trainingsprozesses gelingt es dem StED-Modell, diese Segmentierungsaufgabe besser zu lösen als den Modellen CIPP und U-Net.

Eine vollständige Aufschlüsselung nach betrachtetem Datensatz findet sich im Anhang A.1.2, A.1.3 und A.1.4.

5.4 Fazit

In diesem Abschnitt werden die Ergebnisse bewertet und abschließend Anwendungsempfehlungen formuliert.

Die Ergebnisse der Experimente zeigen, dass eine Segmentierungsaufgabe mit einem niedrigen Flächenanteil $a_K < 0.01$ eine große Herausforderung darstellt und kein Modell einen F1-Score von mehr als 0.2 erreichen kann. Daher wird für

das Training eines Modells zur Bildsegmentierung bei einem niedrigen Flächenanteil ein großer Datensatz oder ein spezialisiertes Vorgehen aus Abschnitt 3.4 empfohlen.

Für Segmentierungsaufgaben mit visuellen Störfaktoren und einem Flächenanteil $a_K > 0.01$ wird das Modell StED empfohlen. Dieses lässt sich direkt anwenden und erreicht konstant hohe F1-Scores. Ist eine Folge von KBV-Methoden offensichtlich, kann in diesem Fall ebenfalls die CIPP-Methode eingesetzt werden. Das U-Net eignet sich in diesem Fall nicht.

Treten neben visuellen Störfaktoren auch geometrische Störfaktoren auf, sind die Modelle StED und U-Net zu empfehlen. Das CIPP-Modell kann sich aufgrund der limitierenden KBV-Methoden nicht an die geometrischen Störfaktoren anpassen. Die Modelle StED und U-Net verhalten sich bei Segmentierungsaufgaben mit einem Flächenanteil von $a_K > 0.01$ ähnlich. Stehen mehr als 32 Bilder für das Training zur Verfügung, ist das U-Net die bessere Wahl, bei weniger Bildern ($N_{\text{img}} \leq 32$) sollte das Modell StED verwendet werden.

6 Anwendung zur Abbildung von Veränderungen des Straßenzustands

6.1 Übersicht

In diesem Kapitel werden das allgemeine Konzept und die Bewertungsmethodik *HyBAR* aus Kapitel 3 angewendet, um einen Algorithmus zur ABV des Straßenzustands zu entwickeln. Anhand dieses Beispiels werden die folgenden Punkte erläutert:

- Übertragbarkeit des allgemeinen Konzeptes auf das konkrete Beispiel des Straßenzustands,
- Vergleich von überwachten und unüberwachten Methoden zur Interpretation des Straßenzustands,
- Vergleich unterschiedlicher Methoden zur Aggregation,
- Verwendung der Bewertungsmethodik *HyBAR* zur Optimierung eines Algorithmus zur ABV und
- Aufbau eines Datensatzes zur Anwendung der Bewertungsmethodik *HyBAR*.

Zuerst werden für eine ABV die Objekte und deren Zustand definiert. Der Zustand $z(t)$ der Straße bezieht sich hier auf die Qualität der Straße, die beispielsweise

durch Risse, Schlaglöcher oder Reparaturen beeinflusst wird. Um den Straßenzustand großflächig zu erfassen, werden Straßenabschnitte als Objekte verwendet.

Der gegenwärtige Zustand der Straßen und dessen Veränderungen sind von maßgeblicher Bedeutung für die Erhaltung sowie die strategische Planung von Instandsetzungsmaßnahmen. Dies wird in mehreren wissenschaftlichen Arbeiten aus den Bereichen der Bildverarbeitung zur Lokalisierung von Straßenschäden [47, 52] und Signalverarbeitung [58, 202, 114, 24, 83, 152] zur Bestimmung der Ebenheit der Fahrbahn behandelt. Diese Arbeiten konzentrieren sich jedoch ausschließlich auf die Bestimmung des aktuellen Zustands und vernachlässigen dabei die ABV.

Unstrukturierte Daten eignen sich für die automatisierte Darstellung des aktuellen Straßenzustands und seiner Veränderung, da eine schnelle und einfache Datenerfassung möglich ist. So lassen sich mobile Kameras (bspw. in Smartphones) leicht und ohne aufwändige Kalibrierung in beliebigen Fahrzeugen der Abfallbeseitigung, Post oder des Bauhofs installieren. Diese mobilen Kameras können entsprechend leicht mitgeführt werden und auch während der Erledigung alltäglicher Aufgaben (z.B. Streckenkontrolle) Daten aufzeichnen. Durch die Montage der mobilen Kameras hinter der Windschutzscheibe wird die Fahrbahn automatisch während der Fahrt aufgezeichnet.

Das Ziel ist, einen robusten Algorithmus zur ABV des Straßenzustands zu entwerfen. Dieser Algorithmus orientiert sich am in Kapitel 3 vorgestellten allgemeinen Konzept und beinhaltet entsprechend Methoden zur Assoziation, Interpretation und Aggregation. Um den Algorithmus und dessen Verarbeitungsschritte samt Parametrierung zu optimieren und zu kalibrieren, wird die Bewertungsmethodik *HyBAR* (s. Abschnitt 3.5) angewendet.

6.2 Road State Change Datensatz

In diesem Abschnitt wird der Road-State-Change (RSC) Datensatz vorgestellt. Der RSC-Datensatz dient als Grundlage zur Anwendung der Bewertungsmethodik *HyBAR* aus Abschnitt 3.5 und wurde hierfür im Rahmen der vorliegenden

Doktorarbeit konzipiert und aufgezeichnet, da andere Datensätze [175, 19] zur Abbildung des Straßenzustands ausschließlich Momentaufnahmen mit annotierten Einzelbildern enthalten. Der RSC-Datensatz muss daher die folgenden Anforderungen erfüllen:

- Erfassung von Objekten: Der Datensatz muss mehrere Objekte von Interesse (in diesem Fall Straßenabschnitte) visuell erfassen. Die Zustände der erfassten Objekte sollen sich unterscheiden.
- Erfassung der Zeit: Die Bildaufnahmen müssen einen zeitlichen Kontext haben, sodass Veränderungen über die Zeit untersucht werden können.
- GPS-Koordinaten zur Assoziation: Die Assoziation von unstrukturierten Bilddaten ist komplex, sodass hier GPS-Daten erfasst werden, um die Zuordnung der Bilddaten zu Objekten (Straßenabschnitten) zu vereinfachen.
- Konstante Objektzustände z: Die Bewertungsmethodik *HyBAR* setzt einen unveränderten Zustand $z(t)$ über die Zeit für jedes Objekt im Datensatz voraus. Gleichzeitig sollen sich die Zustände unterschiedlicher Objekte unterscheiden.
- Variable Störfaktoren $\lambda(t)$: Die unstrukturierten Bildaufnahmen müssen verschiedene Störfaktoren $\lambda(t)$ aufweisen, um die Entwicklung robuster Algorithmen zu ermöglichen, die diesen Störfaktoren standhalten.

Der RSC-Datensatz besteht aus insgesamt 276.139 unstrukturierten Bilddaten, die alle 4m während der Fahrt aus einem Fahrzeug heraus mit einem Smartphone aufgenommen wurden.[1] Durch die Positionierung des Smartphones hinter der Windschutzscheibe ist gewährleistet, dass jedes Bild die Fahrbahnoberfläche vor dem Fahrzeug aufzeichnet und damit relevante Informationen bezüglich des Straßenzustands enthält. In Abbildung 6.1 sind die Route, Smartphone-Position und die einige resultierende Bilddaten dokumentiert. Es wurde eine Strecke von

[1] Es wird hier kein Video verwendet, um die Menge an Daten zu reduzieren. Stattdessen wird GPS-basiert alle 4m ein Bild aufgezeichnet.

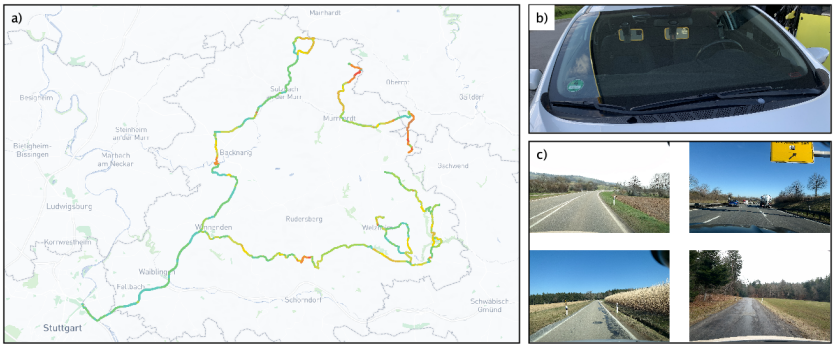


Abbildung 6.1: Übersicht des RSC-Datensatzes: a) Karte des befahrenen Gebietes b) Zur Aufnahme der Bilder wurden verschiedene Smartphone-Modelle hinter der Windschutzscheibe von Fahrzeugen montiert. Um die Anzahl redundanter Befahrungen zu maximieren, wurden Bilddaten mit mehreren Smartphones gleichzeitig aufgenommen. c) Vier Beispielbilder, die im Rahmen der Befahrungen aufgezeichnet wurden.

1.717km im Rems-Murr-Kreis in Baden-Württemberg aufgenommen, sodass mit Sicherheit angenommen werden kann, dass unterschiedliche Straßenzustände im Datensatz vorliegen.

Id	Fahrzeug	Smartphone (iPhone)
1	Cupra Formentor (SUV)	11 / 13
2	Opel Vivaro (Transporter)	11
3	VW Caddy (Transporter)	11 / 13
4	Mercedes Benz Vito 116L (Transporter)	11 / 13 / XR
5	Nissan Micra (Kleinwagen)	11 / 13

Tabelle 6.1: Übersicht über die verschiedenen Fahrzeuge und iPhone-Versionen für den RSC-Datensatz zur Evaluation der ABV.

Neben den Bilddaten werden zusätzlich für jedes Bild Zeitstempel und GPS-Koordinaten aufgezeichnet, sodass später eine Abbildung über die Zeit und die Assoziation damit möglich ist. Die Daten wurden im Zeitraum von 03.03.2022 bis 21.03.2022 erhoben. Da sich der Straßenzustand im Rahmen von Jahren

verändert, kann angenommen werden, dass keine signifikante Veränderung des Straßenzustands innerhalb dieses Zeitraums stattgefunden hat.

Es wurden insgesamt zehn Teildatensätze erhoben, in denen dieselbe Strecke aufgezeichnet wurde. Jeder der Teildatensätze zeigt daher dieselben Straßenabschnitte mit unverändertem Straßenzustand. Die Störfaktoren λ sind für jeden Teildatensatz unterschiedlich. Eine Übersicht über die Teildatensätze findet sich in Tabelle 6.1. Durch die unterschiedlichen Zeitpunkte unterscheiden sich Belichtungsverhältnisse, Wetter und Schatten. Die Teildatensätze wurden zusätzlich mit verschiedenen Smartphones und Fahrzeugen aufgenommen, sodass sich die Positionen/Perspektiven der Bilder und die Kamera-Systeme unterscheiden.

6.3 Entwurf des Algorithmus

Mit dem Ziel die Veränderung der Qualität von Straßenabschnitten abzubilden, wird nun ein Algorithmus zur ABV nach dem Konzept aus Abschnitt 3.3 entworfen. In diesem Beispiel entspricht ein Objekt einem Straßenabschnitt. Diese Straßenabschnitte werden während der Auswertung dynamisch durch Clustering gebildet. Dieses Vorgehen zeigt die Flexibilität des Konzeptes, das nicht auf statische und eindeutige Objekte angewiesen ist. Anschließend schätzt der Algorithmus die Zustände eines jeden Clusters zu verschiedenen Zeitpunkten und vergleicht diese. Das Ergebnis lässt sich durch die Bewertungsmethodik *HyBAR* evaluieren, sodass unterschiedliche Algorithmus-Varianten (AV) miteinander verglichen werden können.

Der Ablauf ist in Abbildung 6.2 dargestellt. Der RSC-Datensatz besteht aus zehn Teildatensätzen $\mathbf{X}_m$ mit $m \in M$ ($M = 10$), wobei jedes Bild einen Zeitstempel und GPS-Koordinaten hat. Alle verfügbaren Bilder werden durch Clustering der GPS-Koordinaten gruppiert. Jedes der N Cluster entspricht einem Straßenabschnitt (Objekt). Dieser Schritt entspricht einer GPS-basierten Assoziation, die gleichzeitig die vorhandenen Bilddaten konkreten Straßenabschnitt zuordnet

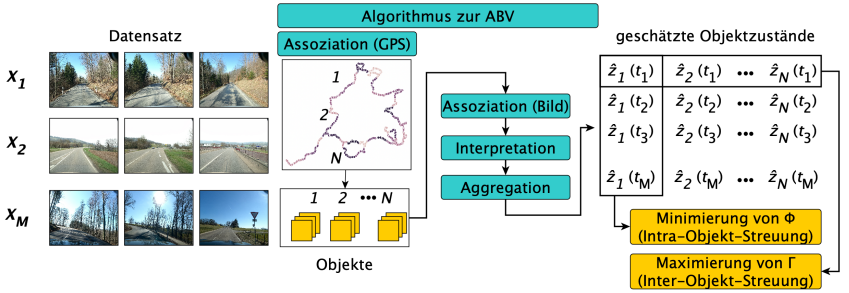


Abbildung 6.2: Entwurf eines Algorithmus zur ABV des Straßenzustands: In M Datensätzen $\mathbf{X}$ werden Bilddaten in einem Gebiet gesammelt. Diese Bilddaten werden anhand ihrer GPS-Koordinaten gruppiert und so N Straßensegmenten (Objekten) zugeordnet. Jedes Straßensegment wird durch mehrere Bilder beschrieben, die durch bildbezogene Assoziation und Interpretation ausgewertet werden. Die Ergebnisse werden anschließend zu Zuständen $\hat{z}_n(t)$ zusammengefasst und auf Robustheit und Deskriptivität untersucht.

und diese Straßenabschnitte definiert. So entstehen insgesamt $N = 1987$ Straßenabschnitte[2]. Die zugeordneten Bilddaten werden im Folgenden durch weitere Methoden der bildbasierten Assoziation, Interpretation und Aggregation weiterverarbeitet, um den Zustand $\hat{z}_n(t)$ und dessen Veränderung abzubilden. Die Kombination der bildbasierten Assoziation, Interpretation und Aggregation entspricht einer AV[3]. Um die optimale AV auszuwählen, wird die Bewertungsmethodik *HyBAR* angewendet. Jede zur Auswahl stehende AV schätzt die Zustände und deren Veränderung der Straßenabschnitte über die Zeit. Die geschätzten Zustände werden nun bewertet. Bei einem idealen Modell ist die Intra-Objekt-Streuung Φ minimal und die Inter-Objekt-Streuung Γ maximal.

[2] Ursprünglich werden $N = 4290$ Straßenabschnitte gebildet. Es werden jedoch alle Straßenabschnitte entfernt, die Daten aus weniger als neun Zeitpunkte haben. Auf diese Weise wird gewährleistet, dass eine aussagekräftige Intra-Objekt-Streuung berechnet werden kann.

[3] Die GPS-basierte Assoziation wird in diesem Fall von der Optimierung ausgeklammert, da hier keine Alternativen zur Verfügung stehen.

Nach der GPS-basierten Assoziation stehen verschiedene Methoden zur weiteren bildbasierten Assoziation, Interpretation und Aggregation zur Auswahl. Aus diesen können verschiedene AV konfiguriert werden, die Veränderungen des Straßenzustands abbilden und die in den folgenden Abschnitten bewertet werden können. Jede AV berechnet dabei aus mehreren Bildern einen geschätzten Zustand $\hat{z}(t)$.

Im Folgenden werden ein unüberwachter und ein überwachter Ansatz zur ABV des Straßenzustands beschrieben. Der unüberwachte Ansatz kann in beliebigen Szenarien eingesetzt werden, da keine Annotationen benötigt werden. Die Optimierung der Interpretation zur Extraktion der passenden Merkmale ist jedoch rechenintensiv. Aufgrund der Vielzahl bereits existierender Datensätze und Modelle ist es jedoch wahrscheinlich, eine überwachte Methode zur Interpretation zu finden, die auf einem anderen Datensatz trainiert wurde und bereits relevante Merkmale extrahieren kann. In diesem Fall kann die Bewertungsmethodik *HyBAR* eingesetzt werden, um die Aggregation der extrahierten Merkmale zu optimieren.

Unüberwachter Ansatz

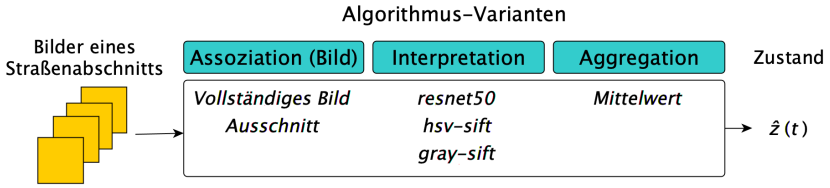


Abbildung 6.3: Unüberwachte AV zur Abbildung des Straßenzustands: Jede AV verarbeitet die Bilder eines Straßenabschnitts zu einem Zeitpunkt t in einem Vektor, der dessen Zustand $\hat{z}(t)$ beschreibt. Hierfür verwendet das Modell dem allgemeinen Konzept entsprechend Methoden zur Assoziation, Interpretation und Aggregation. Es stehen verschiedene Parametrierungen zur Verfügung.

Die unterschiedlichen AV des unüberwachten Ansatzes sind in Abbildung 6.3 veranschaulicht. Jede AV kombiniert unterschiedliche Methoden zur bildbasierten Assoziation, Interpretation und Aggregation, um aus einer Gruppe Bilder einen Zustand $\hat{z}(t)$ für ein Objekt zum Zeitpunkt t zu schätzen. Die Optionen für

eine bildbasierte Assoziation sind in Abbildung 6.4 dargestellt. Entweder wird keine bildbasierte Assoziation durchgeführt (*vollständiges Bild*), sodass das Bild vollständig durch die Interpretation ausgewertet wird oder es wird ein *Ausschnitt* 4m von dem Ende der Motorhaube nach vorne definiert. Dadurch wird der Anteil des Hintergrunds und der Umgebung reduziert und der Anteil der Straße im Bild erhöht.

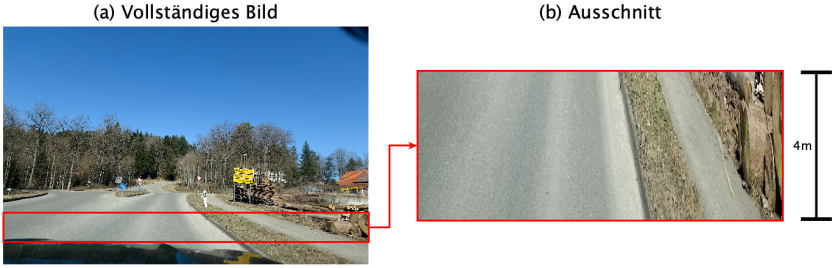


Abbildung 6.4: Bildbasierte Assoziation für die Abbildung des Straßenzustands: a) Das vollständige Bild wird ohne weitere Verarbeitungsschritte ausgewertet (*vollständiges Bild*). b) Ein Ausschnitt des Bildes, der 4m von der Motorhaube nach vorne reicht, wird für die weitere Verarbeitung ausgewählt (*Ausschnitt*).

Im Abschnitt 3.4 wurden dateneffiziente Methoden für die Interpretation vorgestellt. Es werden hier die folgenden unüberwachten Interpretationsmethoden angewendet: *hsv-sift*, *gray-sift* und *resnet50*. Die Modelle *hsv-sift* und *gray-sift* sind BoVW-Modelle, bei denen eins den Farbraum *hsv* berücksichtigt, während das Modell *gray-sift* gezielt Farben ignoriert. Als Ergänzung wird das *resnet50* Modell als Vertreter der DL-basierten Methoden (vDCNN) untersucht. Die Implementierung kann in Abschnitt 4.3 und 4.4 nachvollzogen werden. Die verwendete Interpretationsmethode berechnet aus jedem Bild $x_n(t)$ eines Straßenabschnitts Merkmale $x_n^*(t)$. Da zu jedem Zeitpunkt mehrere Bilder des Straßenabschnitts zur Verfügung stehen, müssen die einzelnen Merkmale durch die Aggregation zu einem Zustand $\hat{z}_n(t)$ zusammengefasst werden. Die Aggregation wird in diesem Fall vereinfacht als Mittelwert über alle Merkmale $x_n^*(t)$ eines Objektes berechnet.

Überwachter Ansatz

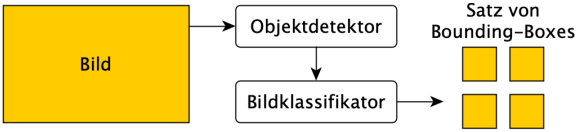


Abbildung 6.5: Überwachte Interpretation des Straßenzustands: Für die Bewertung des Straßenzustands wird eine Kombination von Objektdetektor und Bildklassifikator verwendet. Der Objektdetektor markiert beschädigte Regionen durch Bounding-Boxen im Bild. Handelt es sich bei der Region um einen *Riss* wird dieser im folgenden durch den Bildklassifikator spezifiziert.

Ein überwachter Ansatz benötigt Trainingsdaten und ist nicht immer als Option verfügbar. In diesem Fall wird ein externer Datensatz für das Training bereitgestellt. Mit diesem Datensatz kann das Modell *faster-rcnn* zur überwachten Interpretation des Straßenzustands trainiert werden. Das Modell *faster-rcnn* kombiniert einen Objektdetektor (Faster R-CNN [145]) mit einem Bildklassifikator (ResNet50 [77]). Der Objektdetektor markiert Beschädigungen der Fahrbahn mit Bounding-Boxen im Bild. Dieses Anwendungsbeispiel zeigt, die Flexibilität des allgemeinen Konzepts, da es sich mit einer Vielzahl von unterschiedlichen und komplexen Methoden zur Interpretation kombinieren lässt.

Es werden die folgenden Schadensklassen unterschieden: *Ausbruch*, *Flickstelle*, *Aufgelegte Flickstelle*, *Riss* und *Gefüllter Riss*. Bounding-Boxen der Klasse *Riss* werden anschließend vom Bildklassifikator weiter in die Klassen *Einzelriss*, *Rissanhäufung* und *Netzriss* unterteilt. Insgesamt ergeben sich so sieben unterschiedliche Schadensklassen. Beispiele für die unterschiedlichen Schadensklassen finden sich in Abbildung 6.6. Die Schadensklassen haben unterschiedliche Auswirkungen auf die Straßenoberfläche. Ziel ist es, einen Zustandsvektor zu schätzen, der für jede Schadensklasse einen repräsentativen Wert enthält. So können Veränderungen einzelner Schadensklassen erfasst werden.

Das Modell *faster-rcnn* wird ausschließlich auf dem *Ausschnitt* angewendet. Sodass keine Methoden zur bildbezogenen Assoziation verglichen werden müssen.

Abbildung 6.6: Schadensklassen Straßenzustand: Das überwachte Modell zur Interpretation des Straßenzustands unterscheidet die aufgeführten Schadensklassen [156].

Um aus den detektierten Bounding-Boxen einen Zustandsvektor zu berechnen, können unterschiedlichen Methoden zur Aggregation genutzt werden:

- *Anzahl-Schäden*: Die Anzahl von Bounding-Boxen für jede der sieben Schadensklassen.
- *Fläche-Schäden*: Die Fläche der Bounding-Boxen[4] relativ zur Gesamtfläche des Bildes für jede der sieben Schadensklassen.
- *Fläche-Schäden (A)*: Die Fläche der Bounding-Boxen[4] relativ zur Fläche des *Ausschnitt* des Bildes für jede der sieben Schadensklassen.

Die vorgestellten Methoden zur Aggregation berechnen zuerst einen Zustandsvektor pro Bild und verwenden anschließend deren Durchschnitt als Zustandsvektor für den Straßenabschnitt. Die Bewertungsmethodik *HyBAR* wird in diesem Fall verwendet, um die ideale Methode zur Aggregation auszuwählen.

[4] Überlappende Bounding-Boxen werden nicht mehrfach berücksichtigt.

Zusätzlich wird der Einfluss des Bildklassifikators getestet. Die AV *Fläche-Schäden-R (A)* fasst ausschließlich die Fläche der Bounding-Boxen[4] relativ zur Fläche des *Ausschnitts* des Bildes für die fünf Schadensklassen des Objektdetektors zusammen, ohne die Klasse *Riss* weiter zu spezifizieren.

6.4 Plausibilitätsprüfung der Bewertungsmethodik

Vor der Anwendung und Auswertung der vorgestellten AV werden zusätzliche theoretische Experimente durchgeführt. Diese Experimente erlauben eine Plausibilitätsprüfung der Bewertungsmethodik *HyBAR* und setzen die berechneten Metriken in Relation. In diesem Abschnitt werden Ergebnisse der theoretischen Experimente vorgestellt.

Es werden drei theoretische AV eingeführt und auf dem RSC-Datensatz bewertet: *Zufall*, *Ideal*, *Invers*.
Das theoretische Modell *Zufall* generiert für jeden Straßenabschnitt einen zufälligen Zustand. Dieser berechnet sich aus dem Mittelwert von pro Bild zufällig generierten Vektoren.
Das theoretische Modell *Ideal* verwendet die GPS-Koordinaten des Bildes als zweidimensionalen Zustand. Dadurch ist gewährleistet, dass die Intra-Objekt-Streuung Φ im Vergleich zur Intra-Objekt-Streuung Γ sehr klein ist und damit das optimale Ergebnis imitiert.
Im Gegensatz dazu berechnet das theoretische Modell *Invers* den Zustand eines Straßenabschnitts basierend auf den Identifikationsnummern der vorhandenen Befahrungen (siehe Tabelle 6.1). Da allen Straßenabschnitten Bilder aus denselben zehn Befahrungen zugeordnet werden, führt dies automatisch zu einer hohen Intra-Objekt-Streuung Φ und einer geringen Inter-Objekt-Streuung Γ.

Die Kennzahlen aller theoretischen AV sind in Tabelle 6.2 aufgelistet. Die Ergebnisse der AV *Ideal* und *Invers* entsprechen den Erwartungen. Das AV *Ideal*

Metrik	Φ	Γ	F	β
Zufall	116.3181	41.8442	0.0	0.36
Ideal	1.3024	5118.5514	1.0	3914.77
Invers	3.2124	0.2645	0.0	0.11

Tabelle 6.2: Übersicht der Ergebnisse für die theoretischen AV: *Zufall*, *Ideal* und *Invers*.

hat eine geringe Intra-Objekt-Streuung $\Phi < 1.4$ und eine deutlich höhere Inter-Objekt-Streuung $\Gamma > 5000$. Das Verhältnis der Streuungen β ist entsprechend hoch ($\beta > 3900$) und die AV erreicht den optimalen Wert von $F = 1.0$, dh. es gilt für alle Straßenabschnitte im RSC-Datensatz $\phi < \gamma$. Im Gegensatz dazu ist die Intra-Objekt-Streuung $\Phi > 3$ der theoretischen AV *Invers* deutlich höher als die Intra-Objekt-Streuung $\Gamma < 0.3$. Dies entspricht ebenfalls den Erwartungen. Mit einem Verhältnis von $F = 0.0$ erfüllt kein einziger Straßenabschnitt die Ausgangshypothese ($\phi < \gamma$).

Für die theoretische AV *Zufall* ergibt sich eine hohe Intra-Objekt-Streuung $\Phi > 116$. Im Vergleich dazu ist die Inter-Objekt-Streuung $\Gamma < 42$ deutlich geringer. Die Inter-Objekt-Streuung Γ errechnet sich aus der Streuung aller Objekte zueinander, wodurch sich Glättungseffekte durch die große Anzahl der gemittelten Merkmale ergeben. Aus diesem Grund ist die Streuung Γ kleiner als die Streuung Φ. Die theoretische AV *Zufall* dient als Maßstab für alle anderen AV. Eine AV muss entsprechend mindestens ein Streuungsverhältnis $\beta > 0.36$ erreichen, um die AV *Zufall* zu überbieten. Gleichzeitig zeigt die AV *Zufall* die Schwierigkeit der Aufgabe, da bei zufälliger Erzeugung der Zustände der Hypothesen-Quotient $F = 0$ ist.

Die Verteilungen der Streuungen ϕ_n und γ_n für jeden einzelnen Straßenabschnitt sind in Abbildung 6.7 dargestellt. Auffällig sind die schmalen Verteilungen für ϕ_n (*Ideal*) und γ_n (*Invers*). Dies zeigt, dass das theoretische Modell *Ideal* für jeden Straßenabschnitt eine gleichförmige Intra-Objekt-Streuung ϕ aufweist und somit die Größe der gebildeten Straßenabschnitte sehr ähnlich ist. Das Modell *Invers* zeigt in der Verteilung Inter-Objekt-Streuung γ zwei Peaks. Diese beiden

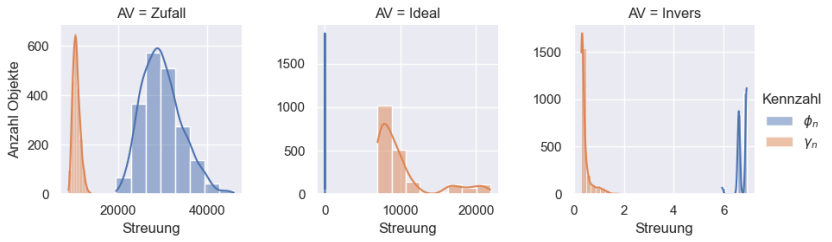


Abbildung 6.7: Darstellung der Verteilung der Streuungen ϕ_n und γ_n über alle N Objekte des RSC-Datensatzes in Abhängigkeit von der verwendeten Methode. Die AV sind theoretisch und sollen die Funktionalität der Bewertungsmethodik *HyBAR* verifizieren. Die AV liefern die erwarteten Ergebnisse. Die AV *Zufall* hat eine hohe Streuung innerhalb eines Objektes und eine geringere Streuung im Mittel über alle anderen Objekte. *Ideal* hat eine geringe Streuung innerhalb eines Objekts und eine große Streuung zwischen den Objekten. Die letzte AV *Invers* hat eine hohe Streuung innerhalb eines Objekts und eine geringe Streuung zwischen den Objekten.

Peaks lassen sich auf Straßenabschnitte, die jeweils Daten aus neun bzw. zehn Befahrungen aufweisen, zurückführen.

6.5 Ergebnisse der unüberwachten Interpretation

In diesem Abschnitt werden die Ergebnisse der zuvor eingeführten unüberwachten AV zur Abbildung des Straßenzustands auf dem RSC-Datensatz vorgestellt. Die Möglichkeiten für das Erstellen von AV wurden in Abschnitt 6.3 erörtert. Der Fokus bei unüberwachten AV liegt entsprechend auf der Interpretationsmethode und der Verwendung bildbezogener Assoziation. AV, die die bildbezogene Assoziation *Ausschnitt* verwenden, sind mit einem *(A)* gekennzeichnet.

Die Ergebnisse sind in Tabelle 6.3 zusammengefasst. Das beste Ergebnis mit einem Hypothesen-Quotient $F = 0.96$ und Streuungsverhältnis $\beta = 1.15$ erzielt die AV *gray-sift (A)*. Durch die Fokussierung auf die Straße und das bewusste Ignorieren von Farbinformationen ist die AV in der Lage, Störungen $\lambda(t)$ durch Aufnahmeparameter erfolgreich zu reduzieren.

Metrik	Φ	Γ	F	β
resnet50	0.3458	0.3213	0.28	0.93
resnet50 (A)	0.2289	0.2345	0.56	1.02
hsv-sift	0.0526	0.0577	0.63	1.09
hsv-sift (A)	0.0495	0.0563	0.93	1.13
gray-sift (A)	0.0481	0.0553	0.96	1.15

Tabelle 6.3: Ergebnisse des RSC-Datensatzes (unüberwacht): Auflistung der Ergebnisse der unüberwachten AV. (*(A)*: Die Assoziationsmethode *Ausschnitt* wird verwendet.)

Im direkten Vergleich erreichen AV, die den *Ausschnitt* verwenden, höhere Hypothesen-Quotienten F und Streuungsverhältnisse β. Für die Interpretationsmethode *resnet50* verbessert sich der Hypothesen-Quotient F um 0.28 und für die Interpretationsmethode *hsv-sift* verbessert sich der Hypothesen-Quotient F um 0.30. Die AV *resnet50* erreicht den niedrigsten Hypothesen-Quotient $F = 0.28$. Durch die verschiedenen Blickwinkel kommen viele Informationen aus dem Hintergrund in das Bild und es wird vermutet, dass die Interpretationsmethode *resnet50* besonders anfällig für Informationen aus dem Hintergrund ist. Im direkten Vergleich dazu schneidet die AV *hsv-sift* (beide AV verwenden nicht den *Ausschnitt*) mit einem Hypothesen-Quotient $F = 0.63$ deutlich besser ab $(+0.35)$.

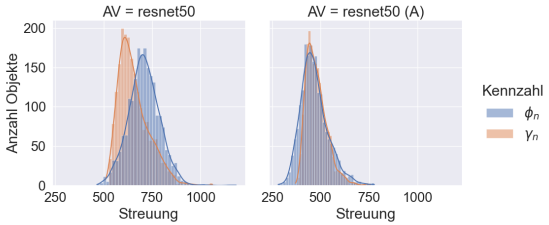


Abbildung 6.8: Visualisierung der Ergebnisse RSC (DCNN): Darstellung der Verteilung der Streuungen ϕ_n und γ_n über alle N Straßenabschnitte im RSC-Datensatz. Links: Die Verteilung für die AV *resnet50* auf dem vollständigen Bild. Rechts: Die Verteilung für das *resnet50 (A)* Modell auf dem *Ausschnitt*.

Abbildung 6.8 zeigt die Verteilung der Streuungen ϕ_n und γ_n für jeden Straßenabschnitt im RSC-Datensatz für die AV *resnet50* und *resnet50 (A)*. Die Verteilung der Streuung für die AV *resnet50* zeigt, dass die Inter-Objekt-Streuung ϕ_n häufig kleiner als die Intra-Objekt-Streuungen sind. Wird der Zustand mit der AV *resnet50 (A)* auf dem *Ausschnitt* geschätzt, verschieben sich die Verteilungen und überlagern sich. Dies zeigt deutlich die Verbesserung durch die Verwendung des *Ausschnitt*. Obwohl die Ergebnisse im Vergleich zu anderen AV, wie *gray-sift (A)* und *hsv-sift (A)* schlechter sind, ist das Ergebnis deutliche besser als das Ergebnis eines zufällig erzeugten Zustands (vgl. AV *Zufall*).

Die entsprechenden Verteilungen der Streuungen über alle Straßenabschnitte für die AV mit BoVW-basierten Interpretationsmethoden sind in Abbildung 6.9 dargestellt. Während sich die Verteilungen für die AV *hsv-sift* deutlich überlagern, sind Unterschiede zwischen den Verteilungen für die AV *hsv-sift (A)* und *gray-sift (A)* erkennbar und die Streuung γ_n und in den meisten Fällen deutlich größer als die Streuung ϕ_n.

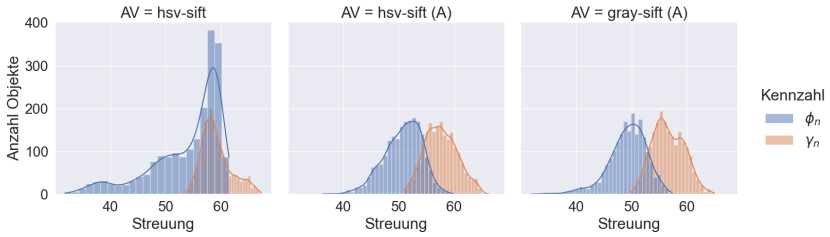


Abbildung 6.9: Visualisierung der Ergebnisse RSC (BoVW): Darstellung der Verteilung der Streuungen ϕ_n und γ_n über alle N Straßenabschnitte im RSC-Datensatz mit BoVW-Merkmalen. Links: Die Verteilung für die AV *hsv-sift*. Mitte: Die Verteilung für die AV *hsv-sift (A)*. Rechts: Die Verteilung für die AV *gray-sift (A)*.

Die Ergebnisse zeigen, dass es für jede AV einzelne Straßenabschnitte gibt, auf denen die Streuung ϕ_n größer als γ_n ist und damit den Hypothesen-Quotienten F reduziert. Diese Ausreißer werden im Folgenden für die AV *gray-sift (A)* untersucht.

(a) Befahrung 4 / iPhone 11

(b) Befahrung 4 / iPhone 13

(c) Befahrung 4 / iPhone XR

Abbildung 6.10: Ausreißer: Die Verwendung des Scheibenwischers in drei der insgesamt zehn Bildreihen erhöht die Streuung ϕ_n der Merkmale erheblich. Der Einsatz der Scheibenwischer tritt hier in drei Bildreihen auf, da diese parallel am selben Fahrzeug erhoben wurden.

Im Datensatz werden alle 19 Straßenabschnitte ausgewertet, die die Grundbedingung $\phi_n \leq \gamma_n$ nicht erfüllen. Für diese Beispiele lassen sich seltene Sonderfälle identifizieren, wie z.B. die Aktivierung des Scheibenwischers in drei der insgesamt zehn Befahrungen (vgl. 6.10).

(a) Befahrung 1 / iPhone 13

(b) Befahrung 1 / iPhone 13

(c) Befahrung 1 / iPhone 13

Abbildung 6.11: Ausreißer: Verschwommene Aufnahmen führen zu einer erhöhten Streuung ϕ_n. Im Gegensatz dazu sind die anderen Bildreihen nicht verschwommen.

Bei den verbleibenden 18 Ausreißern ist die Befahrung 1/ iPhone 13 unscharf. Dieser Unterschied erhöht die Streuung ϕ_n. Dies zeigt, dass der Datensatz viele verschiedene Herausforderungen im Bereich der Abbildung von Veränderungen in unstrukturierten Bilddaten enthält, die auch mit den hier vorgestellten Methoden nicht vollständig gelöst werden können. Um solchen Ausreißern gezielt zu

begegnen, können Methoden eingesetzt werden, die die Validität der Bilder vorab bewerten und Bilder mit geringer Validität aussortieren und deren Ergebnisse ignorieren.

6.6 Ergebnisse der überwachten Interpretation

In diesem Abschnitt werden die Ergebnisse des überwachten Ansatzes vorgestellt. Der Objektdetektor markiert in jedem Bild Bounding-Boxen mit verschiedenen Schadensklassen. Beispiele für die Ausgabe des Objektdetektors ist in Abbildung 6.12 visualisiert.

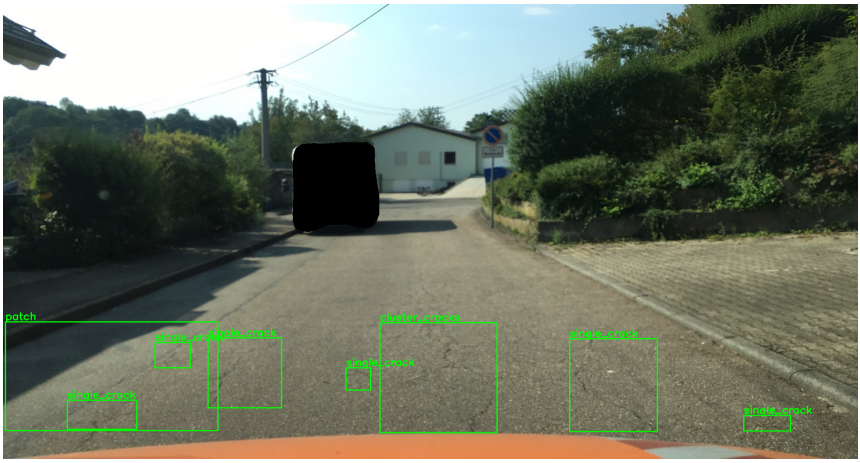


Abbildung 6.12: Beispiel Detektionen des Objektdetektors: Vor der Verarbeitung der Bilder werden persönliche Informationen entfernt. Die Bounding-Boxen werden nur im *Ausschnitt* detektiert.

Im Gegensatz zu den unüberwachten AV ist die Interpretationsmethode hier vorgegeben und es ändert sich ausschließlich die Methode zur Aggregation der detektierten Bounding-Boxen. Daher werden die AV nach der verwendeten Methode zur Aggregation benannt. Die Ergebnisse sind in Tabelle 6.4 zusammengefasst.

Metrik	Φ	Γ	F	β
Anzahl-Schäden	0.0659	0.1235	0.96	1.87
Fläche-Schäden	0.1215	0.2215	0.96	1.82
Fläche-Schäden (A)	0.5540	1.1365	0.98	2.05
Flächen-Schäden-R (A)	0.6904	1.5586	0.99	2.26

Tabelle 6.4: Ergebnisse des RSC-Datensatzes (überwacht): Auflistung der Ergebnisse der überwachten AV. Die AV unterscheiden sich durch die Methode zur Aggregation der Ergebnisse.

Die Aggregationsmethoden führen zu unterschiedlichen Ergebnissen, wobei die AV *Fläche-Schäden-R (A)* das höchste Streuungsverhältnis von $\beta = 2.26$ erreicht, gefolgt von der AV *Fläche-Schäden (A)* ($\beta = 2.05$). Beide AV betrachten die Fläche der beschädigten Fahrbahn relativ zum *Ausschnitt*. Im direkten Vergleich zeigt sich jedoch, dass die Spezifizierung der Klasse *Riss* die Vergleichbarkeit der Zustände verschlechtert. Dies lässt sich darauf zurückführen, dass Risse bei feuchter Fahrbahn ausgeprägter erscheinen als bei trockener. Wobei hier Fahrten bei trockener und feuchter Fahrbahn durchgeführt wurden. Die beiden AV *Fläche-Schäden-R (A)* und *Fläche-Schäden (A)* verwenden beide den *Ausschnitt* bei der Aggregation. Der *Ausschnitt* misst jeweils 4 Meter von der Motorhaube des Fahrzeugs nach vorne, wodurch unterschiedliche Perspektiven der Kamera und damit unterschiedlich große Bounding-Boxen aufgrund des Blickwinkels ausgeglichen werden. Gleichzeitig wird die Überlappung zwischen den Bildern minimiert, und das mehrfache Zählen von Beschädigungen vermieden. Aus diesen Gründen schneidet die AV *Fläche-Schäden* mit einem Streuungsverhältnis von $\beta = 1.82$ am schlechtesten ab.

Die AV *Anzahl-Schäden* wird nicht durch die Verzerrung der Fläche der Bounding-Boxen infolge eines Perspektivwechsels beeinträchtigt. Allerdings ist sie ebenfalls anfällig dafür, einen Schaden mehrfach zu detektieren und ihn trotz überlappender Bounding-Boxen mehrfach zu zählen. Die Bewertungsmethodik *HyBAR* erweist sich auch bei einem bereits konfigurierten Modell zur Interpretation als nützlich und verbessert die Vergleichbarkeit der geschätzten Zustände maßgeblich.

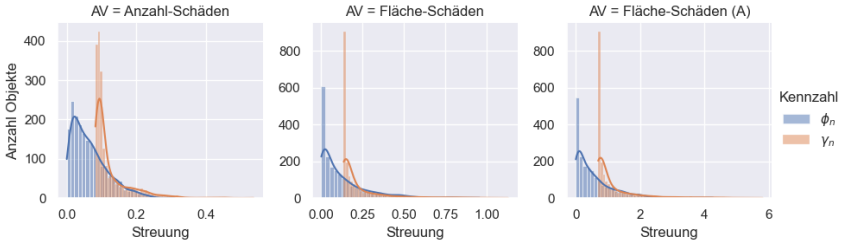
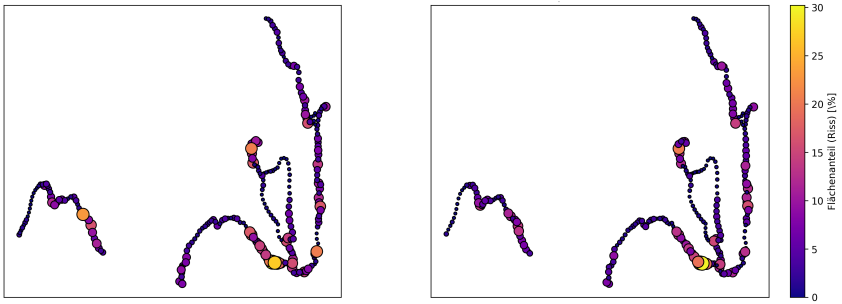


Abbildung 6.13: Visualisierung der Ergebnisse RSC (überwacht): Darstellung der Verteilung der Streuungen ϕ_n und γ_n über alle N Straßenabschnitte im RSC-Datensatz.

Insgesamt kann jede der AV als Algorithmus zur ABV verwendet werden, da jede ein Streuungsverhältnis von $\beta > 1$ aufweist. Im Vergleich zu den unüberwachten AV zeigt sich, dass die überwachten AV höhere Streuungsverhältnisse erreichen. Da überwachte Interpretationsmethoden speziell für einen Anwendungsfall entworfen werden, ist dies zu erwarten.

In Abbildung 6.13 sind die Verteilungen der Streuungen dargestellt. Im Gegensatz zu den Verteilungen des unüberwachten Ansatzes tritt kein oder nur geringes Rauschen der geschätzten Zustände auf, wodurch die Mehrzahl der Objekte eine Streuung $\phi_n = 0$ hat. Dies gilt auch für die Streuung γ_n zwischen den geschätzten Zuständen verschiedener Objekte, die leicht nach rechts verschoben ist und eine relativ höhere Streuung aufweist. In sämtlichen Fällen besteht eine geringe Überlagerung der Streuung, wenn durch Ausreißer falsche Detektionen entstehen.

Die Implementierung der Datenstruktur erlaubt eine Übertragung der Ergebnisse auf eine Karte und ermöglicht so den direkten Vergleich zwischen zwei Zeitpunkten. In Abbildung 6.14 sind zwei unterschiedliche Zeitpunkte derselben Strecke des RSC-Datensatzes zu sehen. Jeder Punkt repräsentiert einen Straßenabschnitt und enthält Informationen aus mehreren Bildern. Die Farbe entspricht dabei der Fläche der Klasse *Riss* zusammengefasst nach der Aggregationsmethode *Fläche-Schäden (A)*. Im visuellen Vergleich zeigt sich, dass die AV *Fläche-Schäden (A)* an denselben Stellen Risse erkennt.



(a) Befahrung 1 / iPhone 13 **(b)** Befahrung 4 / iPhone 11

Abbildung 6.14: Visualisierung eines Ausschnitts des RSC-Datensatzes: Hier wird die Fläche der detektierten Risse zu zwei unterschiedlichen Zeitpunkten dargestellt. Es ist keine Veränderung zwischen den Zeitpunkten zu erwarten.

6.7 Fazit

In diesem Kapitel wird erstmals ein Algorithmus zur ABV des Straßenzustands entworfen und optimiert. Die ABV des Straßenzustands ist aufgrund der unstrukturierten Daten besonders herausfordernd. Hierfür wurde der RSC-Datensatz konzeptioniert und erhoben. Der RSC-Datensatz besteht aus unstrukturierten Bilddaten, die aus einem Fahrzeug heraus mit Smartphones aufgezeichnet wurden. So wird die Erhebung der Daten vereinfacht und eine spätere großflächige Anwendung des Ansatzes ermöglicht. Der Datensatz bildet dabei explizit keine Veränderungen ab, um die direkte Anwendung der Bewertungsmethodik *HyBAR* zu ermöglichen. Der RSC-Datensatz ist damit als Benchmark Datensatz zur robusten Abbildung des Straßenzustand einzigartig.

Anschließend wird das allgemeine Konzept zur ABV an den Anwendungsfall angepasst und passende Algorithmus-Komponenten ausgewählt. Hier werden zwei Ansätze (überwacht/unüberwacht) miteinander verglichen. Stehen keine annotierten Bilddaten zur Verfügung, kann eine unüberwachte Interpretationsmethoden für die Auswertung zum Einsatz kommen. Hier werden zwei verschiedenen Assoziationsmethoden (*vollständiges Bild* / *Ausschnitt*) und drei Methoden zur unüberwachten Interpretation (*gray-sift*, *hsv-sift* und *resnet50*) miteinander verglichen.

Existiert ein Datensatz oder ein bereits trainiertes Modell zur Interpretation, kann dieses direkt eingesetzt werden, um Veränderungen abzubilden. Herkömmliche Modelle (wie in diesem Fall ein trainierter Objektdetektor) extrahieren Informationen für jedes einzelne Bild. Um die einzelnen Informationen eines Zustands für ein Objekt zusammenzufassen, können unterschiedliche Methoden zur Aggregation genutzt werden. Die unterschiedlich parametrierten AV (überwacht/unüberwacht) schätzen die Veränderungen des Straßenzustands über die Zeit und können so mit der in Abschnitt 3.5 eingeführten Bewertungsmethodik *HyBAR* verglichen werden.

Vor der eigentlichen Evaluierung werden theoretische AV als Referenz evaluiert. Diese AV zeigen einerseits die Funktionalität der Bewertungsmethodik *HyBAR* und entsprechen andererseits den Erwartungen. Die AV *Ideal* erzielt entsprechend gute und die AV *Invers* entsprechend schlechte Ergebnisse. Zusätzlich dient die AV *Zufall* als Maßstab.

Für den unüberwachten Ansatz zeigt sich, dass bereits unüberwachte Interpretationsmethoden in der Lage sind, konstante Zustände robust gegenüber Störfaktoren abzubilden. Die höchste Bewertung ($\beta = 1.15$ und $F = 0.96$) erreicht die AV *gray-sift (A)*. Die AV *gray-sift (A)* verwendet die Assoziationsmethode *Ausschnitt* und die Interpretationsmethode *gray-sift*. Auf diese Weise ist die AV geeignet, den Straßenzustand robust darzustellen, ohne durch Hintergrund- oder Farbinformationen gestört zu werden. Es verbleiben einige Ausreißer, die als Veränderung registriert werden, obwohl keine tatsächliche Veränderung stattgefunden hat. Dies ist auf die Verwendung unüberwachter Methoden zurückzuführen, die anfälliger für Störungen sind.

Die Analyse des überwachten Ansatzes legt nahe, dass überwachte Interpretationsmethoden selbst geringfügige Veränderungen ohne oder nur mit minimalem Rauschen darzustellen vermögen. Dies resultiert in einer deutlich höheren Bewertung (Streuungsverhältnis $\beta > 1.8$) im Vergleich zum unüberwachten Verfahren (Streuungsverhältnis $\beta < 1.16$). In diesem Zusammenhang vermag die Bewertungsmethodik *HyBAR* durch die Auswahl der optimalen Aggregationsmethode (*Fläche-Schäden (A)*) das Streuungsverhältnis β um +0.18 zu verbessern. Die

Aggregationsmethode *Fläche-Schäden (A)* gleicht dabei Verzerrungen durch die unterschiedlichen Perspektiven aus, sodass die Fläche der Bounding-Boxen normiert wird.

Die Ergebnisse dieses Kapitels zeigen deutlich die Flexibilität des allgemeinen Konzepts und die Vorteile der eingeführten Bewertungsmethodik *HyBAR*. Komplexe Algorithmen zur ABV können einfach an Anwendungen angepasst und ohne Annotationsaufwand parametrisiert und gegeneinander abgewogen werden. Dabei lassen sich sowohl überwachte als auch unüberwachte Methoden einsetzen, miteinander vergleichen und optimieren. Auch die Integration von bereits vorhandenen Modellen, wie Objektdetektoren, ist möglich und kann durch die Bewertungsmethodik *HyBAR* optimal in den Algorithmus zur ABV eingebunden werden.

7 Anwendung zur Abbildung von Veränderungen der Nutzung von Landflächen

7.1 Übersicht

In diesem Kapitel wird ein Algorithmus zur ABV der Nutzung von Landflächen entworfen und mit der in Abschnitt 3.5 eingeführten Bewertungsmethodik *HyBAR* evaluiert und optimiert. Anhand dieses Beispiels werden die folgenden Punkte erläutert:

- Übertragen des allgemeinen Konzeptes auf das konkrete Beispiel der Nutzung von Landflächen,
- Veranschaulichung der Flexibilität des Konzeptes durch den Einsatz in einem Anwendungsfall (*Remote Sensing*) der herkömmlichen ABV und
- Transfer der Bewertungsmethodik *HyBAR* zur Optimierung eines Algorithmus zur ABV auf eine beliebigen Datensatz.

Als Grundlage für die ABV der Landflächennutzung werden Bilddaten aus der Domäne *Remote Sensing* ausgewertet. Satelliten, Drohnen oder Flugzeuge erheben große Datenmengen in regelmäßigen Abständen und eignen sich gut für die Abbildung von Veränderungen. Ein Objekt entspricht dabei einem zuvor definierten Stück Land, dessen Nutzung (Zustand) durch dessen Bebauung festgelegt wird. Eine Veränderung der Bebauung wird dann im Folgenden als Veränderung der Nutzung und damit des Zustands definiert. Die Beobachtung der Veränderung

der Nutzung von Landflächen über die Zeit ist aus mehreren Gründen wichtig: Sie ermöglicht den Schutz der Umwelt und Biodiversität und unterstützt die Stadtplanung und Infrastrukturentwicklung. Zudem hilft sie bei der Analyse von Bevölkerungsdynamik und wirtschaftlicher Entwicklung, verbessert das Katastrophenmanagement und die Risikoabschätzung.

Die Domäne *Remote Sensing* wird häufig in der herkömmlichen ABV ausgewertet, da hier aufgrund der statischen Vogelperspektive ausschließlich visuelle Störfaktoren vorliegen und ein direkter Vergleich zwischen zwei Bildern möglich ist. Das in Kapitel 3 vorgestellte allgemeine Konzept ist nicht auf den direkten Bildvergleich angewiesen, kann jedoch trotzdem in diesem Kontext eingesetzt werden.

Die Bewertungsmethodik *HyBAR* aus Abschnitt 3.5 setzt einen Datensatz voraus, der Objekte von konstantem Zustand und mehrere redundante Zeitpunkte enthält. Der hier verwendete Datensatz enthält jedoch pro Objekt nur zwei Zeitpunkte, bei denen eine Veränderung stattgefunden hat. In Abschnitt 3.5 wird für einen solchen Fall Augmentation zur Simulation verschiedener Aufnahmen mit konstantem Zustand empfohlen.

7.2 SECOND-Datensatz

SECOND [194] ist ein Datensatz zur semantischen Segmentierung von Veränderungen der Nutzung von Landflächen. Der SECOND-Datensatz stammt aus der herkömmlichen ABV, die auf einen direkten Bildvergleich ausgelegt ist. Daher besteht der Datensatz aus Bildpaaren, die jeweils dasselbe Objekt/Landfläche zu unterschiedlichen Zeitpunkten zeigen. Um die Datenvielfalt zu gewährleisten, wurden Luftbildpaare von verschiedenen Plattformen und Sensoren gesammelt. Diese Bildpaare sind über Städte wie Hangzhou, Chengdu und Shanghai verteilt. Da dieser Datensatz für die herkömmliche ABV aufgezeichnet wurde, gibt es zu

jedem Bildpaar Masken, die die Veränderung zwischen zwei Bildpaaren markieren. Insgesamt enthält der Datensatz 4662 Bildpaare mit Masken (Training: 2968, Test: 1694).

Jedes Bild hat eine Größe von 512 x 512 Pixel und ist auf Pixel-Ebene annotiert. Annotiert wird nicht nur die veränderte Fläche, sondern auch die Art der Veränderung. Beispielhafte Bildpaare und deren annotierte Masken aus dem SECOND-Datensatz sind in Abbildung 7.1 zu sehen. Sechs Landbedeckungsklassen werden eingeführt: *unbewachsene Bodenoberfläche*[1], *Bäume*, *niedrige Vegetation*, *Wasser*, *Gebäude* und *Spielplätze*. Unveränderte Flächen werden nicht annotiert, bei Änderungen wird die neue Landbedeckungsklasse angegeben (z.B. Es wurden Häuser gebaut.). Aus den sechs Landbedeckungsklassen ergeben sich so insgesamt 30 Arten der Veränderung (einschließlich Nicht-Änderung). Durch die zufällige Auswahl von Bildpaaren spiegelt SECOND die tatsächlichen Verteilungen der Landbedeckungsklassen wider.

Durch die Annotation der Art der Veränderung kann der Datensatz ebenfalls für das Training eines überwachten Modells zur Bildsegmentierung verwendet werden. Jedes Bild für sich genommen hat eine Maske mit Pixel-genauen Annotationen, auf denen die veränderten Bereiche mit der entsprechenden Landbedeckungsklasse markiert wurde. Bereiche, in denen keine Veränderung stattgefunden hat, sind durchweg als nicht-verändert markiert (ohne spezifische Landbedeckungsklasse). So entspricht die Maske der Veränderung eines Bilds einer teilweise-annotierten Segmentierungsmaske der Landbedeckungsklassen.

7.3 Entwurf des Algorithmus

Der SECOND-Datensatz kann verwendet werden, um einen Algorithmus zur ABV der Landflächennutzung zu optimieren. Das Vorgehen wird in Abbildung 7.2 veranschaulicht. Der SECOND-Datensatz besteht aus zwei Zeitpunkten t_1 und

[1] Im Datensatz entspricht unbewachsene Bodenoberfläche hauptsächlich unversiegelten Flächen und Brachland.

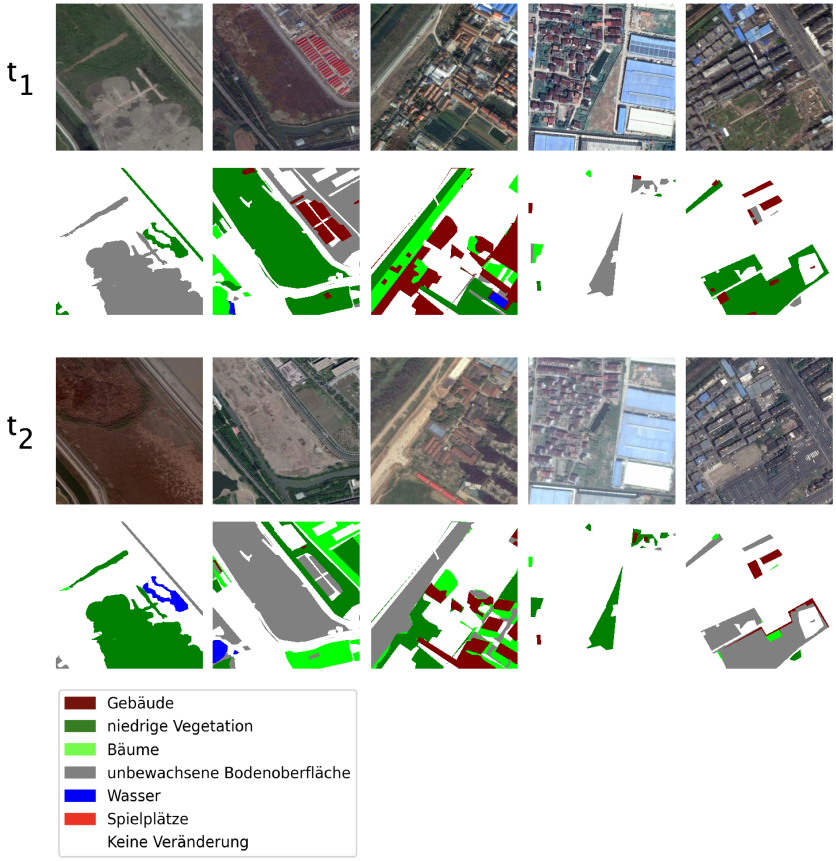


Abbildung 7.1: Beispiel Bildpaare aus dem SECOND-Datensatz [194]. Zu jedem Bildpaar gibt es jeweils zwei Bilder und Masken zu den Zeitpunkten t1 und t2, die die Veränderungen und die Art der Veränderung zwischen den Bildern segmentieren. Ist in einem Bild ein Haus erbaut worden, ist die entsprechende Fläche in der Maske mit der Klasse *Gebäude* markiert.

t_2, wobei zwischen beiden Zeitpunkten eine Veränderung stattgefunden hat. Da keine Bilddaten mit konstanten Zuständen zur Verfügung stehen, werden mittels Augmentation (vgl. Abschnitt 2.7) Bilddaten mit konstanten Zuständen simuliert.

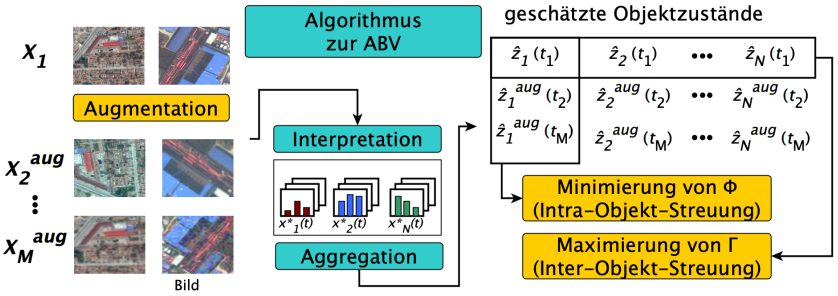


Abbildung 7.2: Entwurf eines Algorithmus zur ABV der Landflächennutzung: Durch Augmentation werden mehrfache Aufnahmen eines Objektes mit konstantem Zustand simuliert. Eine Zuordnung durch Assoziation ist nicht notwendig, sodass dieser Schritt hier vernachlässigt werden kann. Die Methoden zur Interpretation und Aggregation schätzen Zustände basierend auf den augmentierten Bilddaten. Diese geschätzten Zustände können dann mit der Bewertungsmethodik *HyBAR* evaluiert werden.

$\mathbf{X}_1$ besteht aus den ersten Bildern aller Bildpaare aus dem Trainingsdatensatz. Die Datensätze $\mathbf{X}_2^{aug}$, ..., $\mathbf{X}_M^{aug}$ werden durch Augmentation erzeugt, wobei jedes Bild in $\mathbf{X}_1$ zufällig einmal augmentiert wird. Die Wirkung der Augmentationen kann mit einem Faktor zwischen 0 und 1 moduliert werden. Dieser Faktor beschreibt die Wahrscheinlichkeit, mit der beim Laden des Bildes eine Augmentation angewendet wird. D.h. bei einem Faktor von 0.1 wird etwa jedes zehnte Bild mit der Funktion augmentiert. Die Augmentationen wurden so gewählt, dass sie potenzielle Veränderungen zwischen verschiedenen Zeitpunkten abbilden. In dieser Arbeit werden die folgenden Augmentationsmethoden verwendet:

- `noise`: Die Methode `noise` fügt dem Bild *salt-n-pepper* Rauschen hinzu. Die Höhe der Differenz des Rauschens wird dabei bei jeder Anwendung

aus [4, 8, 16] zufällig ausgewählt und die Größe der Kernel wird ebenfalls dynamisch zufällig aus [1, 2, 4, 8, 16] gewählt.

- `rotation` (Faktor=0.3): Mit dieser Methode wird das Eingabebild zufällig um 0, 90, 180, 270 Grad gedreht.
- `flip` (Faktor=0.3): Das Eingabebild wird zufällig entweder horizontal oder vertikal gespiegelt.
- `brightness` (Faktor=0.3): Hier wird die Helligkeit des Bildes zufällig um [1, 2, 4, 8, 16] gleichzeitig erhöht oder reduziert.
- `tiny_rotation` (Faktor=0.3): Im Gegensatz zu `rotation` wird das Bild nur um einen kleinen Winkel zwischen [-10, 10] Grad gedreht.
- `crop` (Faktor=0.3): Diese Funktion schneidet eine kleine Region aus dem Eingabebild heraus. Dabei werden an von jeder Seite zufällig zwischen 0 und 20 Prozent des Bildes abgeschnitten.

Durch die Verwendung von Augmentationen ist eine Assoziation der Bilddaten nicht notwendig, da das ursprüngliche Bild bekannt ist. Bilddaten aus der Domäne *Remote Sensing* benötigen selten Methoden zur Assoziation, da Bilddaten direkt durch Position und Höhe der Kamera zugeordnet werden können. So ist eine Zuordnung der Bildpaare im SECOND-Datensatz ebenfalls nicht notwendig und wird durch die Struktur des Datensatzes vorgegeben.

Daher werden im Folgenden in verschiedenen Methoden zur Interpretation miteinander verglichen. Der SECOND-Datensatz stellt teilweise annotierte Bilddaten zur Verfügung, sodass hier eine überwachte Methode zur Interpretation eingesetzt werden kann. In diesem Fall wird ein *StED*-Modell ausgewählt, da dieses im Fall von visuellen Störfaktoren λ_v gute Ergebnisse erzielt hat (vgl. Kapitel 3.4). Außerdem ermöglicht die Implementierung aus Abschnitt 4.6 ein Training mit teilweise annotierten Bilddaten. Ein Bildsegmentierer berechnet für ein Bild eine Segmentierungsmaske, die nicht direkt als Merkmalsvektor x^* für ein Bild verwendet werden kann. Die Merkmale x^* berechnen sich in einem zweiten Schritt aus der

Segmentierungsmaske eines Bildes. In dieser Anwendung wird die prozentuale Fläche der sechs Klassen in einem Vektor zusammengefasst.

Zusätzlich zu überwachten Interpretationsmethoden werden zum Vergleich unüberwachte Methoden analog zu Kapitel 6 getestet: *resnet50*, *gray-sift* und *hsv-sift*. Der Zustand $\hat{z}$ berechnet sich abschließend durch die Aggregation aller Merkmale x^* wie in Kapitel 6 durch den Mittelwert. So stehen insgesamt vier verschiedene AV mit unterschiedlichen Interpretationsmethoden zur Auswahl: *StED* (überwacht), *resnet50*, *gray-sift* und *hsv-sift*. Jede AV kann abschließend basierend auf ihren geschätzten Zuständen $\hat{z}$ nach der in Abschnitt 3.5 beschriebenen Bewertungsmethode *HyBAR* evaluiert werden.

7.4 Bildsegmentierung SECOND-Datensatz

Der SECOND-Datensatz kann als teilweise annotierter Segmentierungs-Datensatz betrachtet werden. Die Aufstellung der Segmentierungsaufgaben im SECOND-Datensatz ist in Tabelle 7.1 zusammengefasst. Die Segmentierungsaufgaben Fallen in den Bereich eines mittleren Flächenanteils $a_K = (0.01, 0.1]$ und haben nur visuelle Störfaktoren λ_v. Die Ausnahme bilden *Wasser* und *Spielplätze*, die einen kleinen Flächenanteil $a_K < 0.01$ aufweisen. Für diese Anforderungen hat sich das StED-Modell als leistungsstark erwiesen.

Da in diesem Fall der vollständige Datensatz zur Verfügung steht wird das StED-Modell auf allen 2968 Bilderpaaren aus dem Trainingsdatensatz trainiert und auf den 1694 Bildpaaren des Testdatensatzes evaluiert.

Die Ergebnisse des StED-Modells sind in Tabelle 7.2 zusammengefasst. Der F1-Score liegt für die Klassen mit mittlerem Flächenanteil deutlich höher als für die mit kleinem Flächenanteil. Ein Segmentierungsbeispiel ist in Abbildung 7.3 zu sehen. Während der SECOND-Datensatz nur teilweise annotierte Masken für das Training und die Auswertung zur Verfügung stellt, berechnet das StED-Modell eine vollständige Maske für das Eingangsbild.

Segmentierungsaufgabe	Flächenanteil a_K	Störfaktor S_K
Nicht annotiert	0.801	λ_v
niedrige Vegetation	0.065	λ_v
unbewachsene Bodenoberfläche	0.086	λ_v
Gebäude	0.031	λ_v
Bäume	0.014	λ_v
Wasser	0.003	λ_v
Spielplätze	< 0.001	λ_v

Tabelle 7.1: Segmentierungsaufgaben SECOND-Datensatz: Für den SECOND-Datensatz lassen sich 6 Segmentierungsaufgaben mit individuellen SAP definieren. Da die Segmentierungsaufgaben alle auf denselben Bilddaten basieren und sich die Perspektive und Abstand während der Aufnahme nicht ändert, haben alle dieselbe Art von Störfaktoren S_K (λ_v). Die Klasse *nicht annotiert* wird nicht für das Training verwendet.

Segmentierungsaufgabe	F1-Score
niedrige Vegetation	0.54
unbewachsene Bodenoberfläche	0.67
Gebäude	0.76
Bäume	0.41
Wasser	0.11
Spielplätze	0.32
Insgesamt	0.64

Tabelle 7.2: Segmentierungsergebnisse SECOND-Datensatz: Die Auswertung des StED-Modells auf dem SECOND-Datensatz.

7.5 Ergebnisse

In diesem Abschnitt werden die Ergebnisse der beschriebenen AV auf dem SECOND-Datensatz aufgeführt: *resnet50*, *hsv-sift*, *gray-sift* und *StED* (überwacht). Die AV unterscheiden sich durch die verwendete Interpretationsmethode.

Abbildung 7.3: Beispiel Segmentierung SECOND-Datensatz: Eingangsbild in das StED-Modell (links). Manuell annotierte Maske aus dem Datensatz, wobei die weiße Fläche nicht annotiert wurde (Mitte). Segmentierungsmaske des StED-Modells (rechts).

Die Ergebnisse aller AV beziehen sich auf den Testdatensatz des SECOND-Datensatzes. So ist das Training des verwendeten StED-Modells von der Evaluation getrennt (s. Abschnitt 7.4).

Metrik	Φ	Γ	F	β
resnet50	0.223	0.464	1.0	2.08
hsv-sift	0.037	0.061	1.0	1.65
gray-sift	0.041	0.066	1.0	1.61
StED (ü)	0.016	0.131	1.0	8.08

Tabelle 7.3: Ergebnisse des SECOND-Datensatzes: AV, die eine überwachte Interpretationsmethode verwenden, sind mit einem (ü) markiert.

Die Ergebnisse der AV auf dem SECOND Test-Datensatz sind in Tabelle 7.3 aufgeführt. Für alle AV ist die durchschnittliche Intra-Objekt-Streuung Φ kleiner als die durchschnittliche Inter-Objekt-Streuung Γ ($\beta > 1$). Entsprechend erreichen alle AV einen Hypothesen-Quotient $F = 1.0$. Daraus lässt sich schließen, dass die ausgewählten AV robust gegenüber den durch die Augmentation simulierten Störungen sind. Im direkten Vergleich schneidet die überwachte AV *StED* am besten ab (Streuungsverhältnis $\beta = 8.08$). Von den AV, die unüberwachte Interpretationsmethoden verwenden, erreicht die AV *gray-sift* den höchsten Wert $\beta = 2.08$ und liegt damit weit hinter dem überwachten Ansatz.

Die Streuung der AV *resnet50* sind in Abbildung 7.4 visualisiert. Hier zeigt sich deutlich die Trennung von der Intra-Objekt-Streuungen γ_n und Inter-Objekt-Streuungen ϕ_n über alle Objekte. Sowohl die Inter-Objekt-Streuung als auch die Intra-Objekt-Streuung sind einer Normalverteilung ähnlich. Die Augmentation hat einen gleichmäßigen Einfluss auf die Merkmalsextraktion der AV *resnet50*.

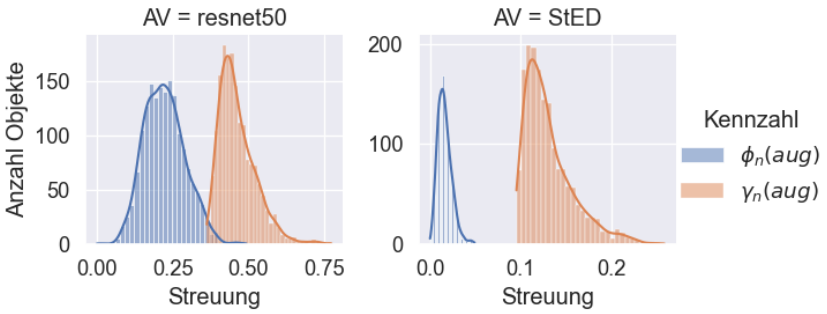
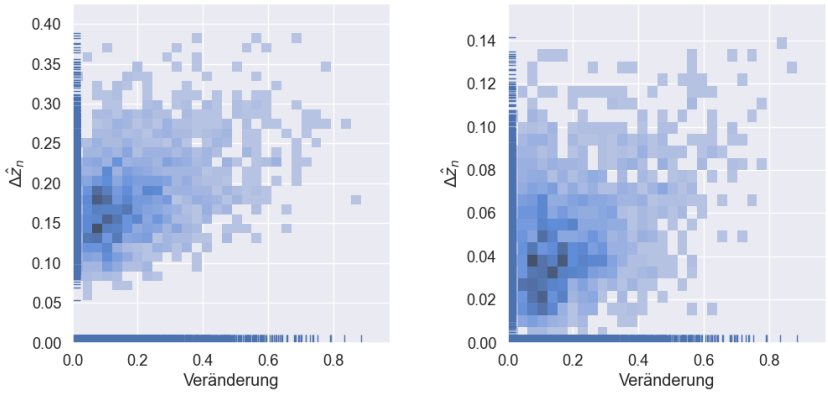


Abbildung 7.4: Visualisierung Ergebnisse SECOND-Datensatz der AV *resnet50* und *StED*: Darstellung der Verteilung der Streuungen ϕ_n und γ_n über alle N Landflächen im SECOND-Datensatz.

Die Verteilung der Intra-Objekt-Streuungen γ_n und Inter-Objekt-Streuungen ϕ_n der AV *StED* über alle Objekte ist in Abbildung 7.4 zu sehen. Hier zeigt sich, dass die Verteilung der Inter-Objekt-Streuungen ϕ_n deutlich schmaler ist als die Verteilung der Intra-Objekt-Streuungen γ_n. Daraus lässt sich schließen, dass die angewendeten Augmentationen einen geringen Einfluss auf die AV *StED* haben.

Der SECOND-Datensatz verfügt über Bildpaare zu einem zweiten Zeitpunkt bei denen Veränderungen annotiert wurden. Dies erlaubt es, einen Zustand $\hat{z}$ für die Zeitpunkte t_1 und t_2 zu schätzen und anschließend die geschätzte Veränderung $\Delta\hat{z}$ zwischen den Zeitpunkten zu berechnen. Diese Veränderung $\Delta\hat{z}$ kann anschließend mit der Größe der tatsächlich veränderten Fläche aus den Annotationen verglichen werden.

In Abbildung 7.5 wird die geschätzte Veränderung $\Delta\hat{z}$ zwischen den Zeitpunkten t_1 und t_2 mit der tatsächlichen Veränderung aus den annotierten Masken für jedes



(a) AV: *resnet50*

(b) AV: *StED*

Abbildung 7.5: Visualisierung der geschätzten Veränderung auf dem SECOND-Datensatz: Die geschätzte Veränderung $\Delta\hat{z}$ durch die AV *resnet50* und *StED* werden hier mit dem Ausmaß der tatsächlichen Veränderung aus den Annotationen verglichen. Die Veränderung entspricht dabei dem Anteil der veränderten Fläche in der annotierten Maske eines jeden Bildpaares.

Bildpaar im Testdatensatz verglichen. Sowohl die AV *resnet50* als auch die AV *StED* bilden die Veränderung ab, sodass die geschätzte Veränderung in beiden Modellen mit der tatsächlichen Veränderung ansteigt. Allerdings zeigt die AV *resnet50* bereits bei Bildpaaren mit einer geringen prozentualen Veränderung einen Unterschied zwischen den Zuständen an. Die visuellen Störfaktoren zwischen den Aufnahmen erzeugen einen Unterschied in den extrahierten Merkmalen, der jedoch kleiner ist als bei einer tatsächlichen Veränderung. Im Gegensatz dazu kann die AV *StED* auch niedrige Veränderungen gut abbilden. Das überwachte *StED*-Modell reduziert durch die Berechnung der Segmentierungsmaske den Einfluss von Störfaktoren. Beide AVs zeigen jedoch eine gewisse Streuung und bilden die Veränderung nicht fehlerfrei ab.

7.6 Fazit

In diesem Kapitel wird ein Algorithmus zur ABV der Landflächennutzung entwickelt. Die verschiedenen AVs werden auf dem SECOND-Datensatz bewertet. Hierfür werden Augmentationen verwendet, um konstante Zustände zu simulieren und die Anwendung der Bewertungsmethodik *HyBAR* zu ermöglichen. Verglichen werden eine überwachte AV (*StED*) und drei nicht-überwachte AVs (*resnet50*, *hsv-sift*, *gray-sift*). Von den nicht-überwachten AVs hat die AV *resnet50* das höchste Streuungsverhältnis von $\beta = 2.08$ erreicht. Im Vergleich dazu erreicht die überwachte AV *StED* ein Streuungsverhältnis von $\beta = 8.08$. Dies zeigt, dass die überwachte AV um ein Vielfaches robuster gegenüber den durch die Augmentationen simulierten Störfaktoren ist. Der Vergleich zwischen der tatsächlichen und der geschätzten Veränderung zeigt ebenfalls, dass die überwachte AV robuster gegenüber den tatsächlichen visuellen Störfaktoren ist.

8 Zusammenfassung und Ausblick

Die vorliegende Dissertation beschäftigt sich mit der Abbildung von Veränderungen in unstrukturierten Bilddaten. Bei der Abbildung von Veränderungen von unstrukturierten Daten lassen sich herkömmliche Methoden der Change-Detection nicht anwenden, da diese auf einem direkten Vergleich zweier Bilder beruhen. Zusätzlich sind Algorithmen zur Abbildung von Veränderungen komplex mit vielen Verarbeitungsschritten und die auszuwertenden Datensätze umfangreich, sodass Bewertungsmethoden alle Verarbeitungsschritte benötigt werden, die mit minimalen Annotationsaufwand eingesetzt werden können. In Kapitel 3 werden die grundlegenden Konzepte, Methoden und verwendeten Komponenten der Arbeit erörtert. Die Implementierung und Umsetzung der einzelnen Komponenten wird anschließend in Kapitel 4 dargelegt. Um die Anwendung der Komponenten unter den richtigen Bedingungen zu gewährleisten, werden Experimente in verschiedenen Szenarien durchgeführt und in Kapitel 5 ausgewertet. In Kapitel 6 und 7 werden zwei repräsentative Anwendungen für eine Abbildung von Veränderungen mit den vorgestellten Methoden und Komponenten durchgeführt. Die Arbeit bearbeitet die folgenden Punkte:

1. Erarbeiten eines allgemeinen Vorgehens für den Entwurf eines Algorithmus zur Abbildung von Veränderungen,
2. Erstellen des allgemeinen Konzeptes „Object-State-based Mapping of Changes“ *OSMC* zur Abbildung von Veränderungen in unstrukturierten Bilddaten,
3. Einführung die Bewertungsmethodik „Hypothesis-Based-Algorithm Rating“ *HyBAR* zur Parametrierung des allgemeinen Konzepts *OSMC* entsprechend der gegebenen Anwendung,

4. Auswahl von Algorithmus-Komponenten unter dem Gesichtspunkt der Dateneffizienz und Bewertung der überwachten Algorithmus-Komponenten in Abhängigkeit der Segmentierungsaufgabenparameter (Flächenanteil / Störfaktoren),

5. Implementierung von Algorithmus-Komponenten als modulare Programmpakete,

6. Entwicklung des Road State Change Datensatzes als Referenzdatensatz für die Bewertung der Leistungsfähigkeit von Algorithmen zur Abbildung von Veränderungen in unstrukturierten Bilddaten und

7. Demonstration des Vorgehens für den Entwurf eines Algorithmus zur Abbildung von Veränderungen anhand zweier Beispiele (*Fahrzeug-Bilddaten* / *Remote Sensing*).

Das erarbeitete allgemeine Konzept *OSMC* zur Abbildung von Veränderungen geht über den bisher üblichen direkten Vergleich zweier Bilder hinaus. Anstatt zwei Bilder pixelweise zu vergleichen, werden Objekte auf Basis von Zustandsvektoren miteinander verglichen. Dadurch können Störfaktoren, wie unterschiedliche Perspektiven und Umwelteinflüsse, kompensiert und Zustände abgebildet werden, die nicht aus der Zusammensetzung von Komponenten bestehen. Das Konzept *OSMC* beschreibt einen Algorithmus, der aus drei Teilen besteht: Assoziation, Interpretation und Aggregation. Die Assoziation ordnet Informationen zuvor definierten Objekten zu. Die Interpretation extrahiert aus jedem Teilstück Informationen. Die Aggregation fasst diese Informationen zu einem Zustand zusammen.

Das Konzept *OSMC* besteht aus vielen Schritten und die Optimierung ist komplex. In dieser Dissertation wird die Bewertungsmethodik *HyBAR* vorgestellt, die ohne zusätzlichen Annotationsaufwand angewendet werden kann. So können verschiedene Algorithmus-Varianten gegeneinander abgewogen und letztlich die ideale Version ausgewählt werden. Die Bewertungsmethodik *HyBAR* verwendet die Hypothese eines konstanten Zustands über die Zeit, um die Robustheit und Deskriptivität einer Algorithmus-Variante zur Abbildung von Veränderungen zu

bewerten. Eine beliebige Algorithmus-Variante, die den Zustand von Objekten über die Zeit abbildet, muss dann eine Änderung des Zustands eines Objekts aufweisen, die kleiner ist als die Zustandsdifferenz zwischen verschiedenen Objekten. Die Algorithmus-Variante, die diese Bedingung erfüllt, ist positiv zu bewerten.

Unstrukturierte Daten lassen sich auf unterschiedliche Arten von Veränderungen hin untersuchen. Um eine effiziente und schnelle Auswertung zu ermöglichen, werden Algorithmen benötigt, die dateneffizient arbeiten. So können Modelle effizient an unterschiedliche Aspekte angepasst werden. In dieser Arbeit werden zwei Optionen für die Interpretation von Bilddaten vorgestellt. Die erste Option umfasst unüberwachte Methoden zur Interpretation, wie BoVW-Modelle und vortrainierte DCNNs, die abstrakte Repräsentationen für ein Bild berechnen, ohne annotierte Daten zu benötigen. Diese Repräsentationen können nicht direkt interpretiert, jedoch miteinander verglichen werden. Die zweite Option besteht aus überwachten Modellen zur semantischen Segmentierung, wie CIPP, StED oder U-Net-Modelle. Diese Modelle können ein Bild interpretieren, indem das Bild in definierte Bestandteile zerlegt wird. Ein Modell segmentiert hierfür in jedem Bild die definierten Klassen. Der Zustand ergibt sich anschließend aus den prozentualen Flächenanteilen der Klassen. Dieser Zustand ist für sich genommen bereits interpretierbar, jedoch benötigen die Modelle zuvor Trainingsdaten, die aufwändig annotiert werden müssen. In dieser Arbeit wird untersucht, welche dieser Modelle am dateneffizientesten arbeiten, d.h. möglichst wenig annotierte Daten benötigen, sodass die Modelle leicht und effizient auf verschiedene Problemstellungen angewendet werden können. Hierfür werden Segmentierungsaufgaben entsprechend ihrer Segmentierungsaufgabenparameter (Flächenanteil und Störfaktoren) bewertet. Anschließend lassen sich für neue Segmentierungsaufgaben allgemeingültige Anwendungsempfehlungen je nach vorhandenen Segmentierungsaufgabenparametern formulieren.

Die Untersuchung ergibt, dass die Auswertung von Segmentierungsaufgaben mit einem kleinen Flächenanteil und wenigen annotierten Bilddaten im Allgemeinen nicht möglich ist. Insgesamt erreicht das StED-Modell die durchschnittlich beste Bewertung in den durchgeführten Versuchen. Dabei zeichnet sich das StED-Modell durch die Fähigkeit aus, mit steigender Anzahl von Trainingsbildern seine

Segmentierungsgüte besonders zu verbessern. Das StED-Modell erreicht eine hohe Güte bei visuellen Störfaktoren und ist gegenüber geometrischen Störfaktoren robust. Das CIPP-Modell liefert konstante Ergebnisse und verbessert sich nicht mit steigender Anzahl an Trainingsbildern, ist jedoch bereits bei wenigen Bildern effektiv. Allerdings ist das CIPP-Modell anfällig für geometrische Störfaktoren und erreicht selbst bei einem hohen Flächenanteil nur eine niedrige Segmentierungsgüte. Das U-Net schneidet besonders bei visuellen Störfaktoren schlechter ab, kann diese Lücke jedoch bei geometrischen Störfaktoren schließen. Am deutlichsten steigert sich das U-Net mit der Anzahl der Trainingsbilder.

Grundsätzlich gilt: Wenn der Flächenanteil der Segmentierungsaufgabe niedrig ist, muss mit einem hohen Annotationsaufwand gerechnet werden. Das StED-Modell ist bei ausschließlich visuellen Störfaktoren und einem mittleren bis hohen Flächenanteil zu verwenden. Das CIPP-Modell kann eingesetzt werden, wenn es für die Problemstellung eine eindeutigen Lösung durch konventionelle Bildverarbeitungsmethode gibt oder nur wenige Rechenressourcen zur Verfügung stehen. Treten geometrische und visuelle Störfaktoren in Kombination mit einem hohen Flächenteil auf, kann entweder das StED-Modell oder bei mehr als 64 Trainingsbildern das U-Net verwendet werden. Für Segmentierungsaufgaben mit einem mittleren Flächenanteil bei geometrischen und visuellen Störfaktoren ist ein hoher Annotationsaufwand zu erwarten.

Die Bewertungsmethodik wird in der vorliegenden Dissertation anhand eines Beispiels zur Erstellung eines Algorithmus zur Abbildung der Veränderung des Straßenzustands demonstriert. Verschiedene Realisierungen des Algorithmus werden anhand dieses Beispiels evaluiert und mithilfe der Bewertungsmethodik parametriert. Zuerst wird ein unüberwachter Ansatz verwendet. Der unüberwachte Algorithmus besteht aus mehreren Teilschritten, und es zeigt sich, dass die Kombination aus einer Bild-basierten Assoziation *Ausschnitt* (Zuschneiden des Bildes auf einen 4 Meter Abschnitt) und der Interpretationsmethode *gray-sift* die besten Ergebnisse liefert. Das Modell *gray-sift* reduziert den Einfluss von Farbinformationen, die für eine Bewertung des Straßenzustands nicht relevant sind und kann so Störfaktoren eleminieren. Gleichzeitig verbessert die Methode *Ausschnitt*

durch die Fokussierung auf die Straße und das Entfernen des Hintergrunds die Leistungsfähigkeit der untersuchten Algorithmus-Varianten.

Um die Flexibilität des Konzepts *OSMC* zu veranschaulichen, wird zusätzlich ein weiterer überwachter Ansatz untersucht. In diesem Fall wird ein bereits trainiertes Modell (eine Kombination aus Faster R-CNN und Bildklassifikator) zur Interpretation des Straßenzustands genutzt. Die Bewertungsmethodik wird erfolgreich eingesetzt, um die optimale Aggregationsmethode für die Berechnung des Zustandsvektors zu ermitteln. Die am besten bewertete Methode zur Aggregation ist *Fläche-Schäden (A)*. *Fläche-Schäden (A)* zählt sich überlappende Bounding-Boxen in einem Bild nicht mehrfach und gleicht durch die Berücksichtigung des 4-Meter-*Ausschnitts* unterschiedliche Perspektiven der Kamera aus. Das Konzept *OSMC* kann leicht mit verschiedenen Algorithmen kombiniert und durch die Bewertungsmethode in seiner Gesamtheit bewertet werden, sodass das Zusammenspiel der unterschiedlichen Komponenten verglichen werden kann.

Zusätzlich wird die Bewertungsmethodik auf einen Datensatz aus dem Bereich *Remote Sensing* angewendet. Der entwickelte Algorithmus kann anschließend Veränderungen in der Nutzung von Landflächen abbilden. Die Bewertungsmethodik ist jedoch nicht direkt auf den vorhandenen Datensatz anwendbar, da nicht genügend Daten über verschiedene Zeitpunkte hinweg vorhanden sind und zudem zwischen diesen Zeitpunkten Veränderungen stattgefunden haben können. Daher wird das Verfahren zur Anwendung der Bewertungsmethodik durch Augmentation erweitert. Diese Augmentation ermöglicht die Simulation von konstanten Zuständen, um anschließend die verschiedenen Algorithmus-Varianten zu bewerten. Hier erzielt das Modell *resnet50* die besten Ergebnisse unter den unüberwachten Methoden. Auch hier wird das Modell StED als überwachte Methode zur Interpretation verwendet und schneidet im Vergleich deutlich besser ab als die unüberwachte Methode. Da in diesem Fall Daten mit einer tatsächlichen Veränderung vorliegen, wird die geschätzte Veränderung mit der tatsächlichen verglichen. Beide AV erfassen diese Veränderung, wobei die unüberwachte Algorithmus-Variante (*resnet50*) durch visuelle Störfaktoren stets einen geringen Unterschied anzeigt, der unabhängig von der Größe der tatsächlichen Veränderung ist. Die

überwachte Algorithmus-Variante (*StED*) hingegen ist robuster gegenüber diesen visuellen Störfaktoren.

Dem Anwender steht nun das allgemeines Konzept *OSMC* für die Erstellung von Algorithmen zur Abbildung von Veränderungen in unstrukturierten Bilddaten zur Verfügung und zudem eine Bewertungsmethodik, die die effiziente Optimierung eines solchen Algorithmus ermöglicht. Zusätzlich werden verschiedene Algorithmen als Komponenten für die Verwendung innerhalb des Algorithmus erprobt und auf ihre Dateneffizienz hin untersucht, sodass unter verschiedenen Bedingungen anhand dieser Arbeit Algorithmen zur Abbildung von Veränderungen entworfen werden können.

Um die vorliegende Dissertation weiterzuführen, können die einzelnen Teilschritte des Konzepts *OSMC* weiter ausgearbeitet und bereits vorgestellte Methoden optimiert werden. Dadurch würde die Auswahl an Komponenten für die Anwendung auf unterschiedliche Problemstellungen vervollständigt werden. Während die Methoden der Interpretation ausführlich untersucht werden, stehen für Assoziation und Aggregation noch zahlreiche weitere Methoden zur Verfügung. Eine systematische Analyse verschiedener Anwendungsbereiche und Herausforderungen der ABV soll in zukünftigen Arbeiten dazu beitragen, eine Auswahl standardisierter Assoziationsmethoden zu treffen. Durch die Implementierung und Integration in das bestehende Softwarepaket können diese Methoden anschließend den Anwendern zur Verfügung gestellt werden. Im Bereich der Assoziation können weitere GPS-basierte und insbesondere bildbasierte Methoden, wie Structure-from-Motion (SfM) [157, 174], SLAM [109, 169], Disparity-Maps [73, 99] oder Optical-Flow [197], zur direkten Zuordnung von Bildausschnitten und Objekten im Raum untersucht werden. Durch bildbasierte Assoziation können Ungenauigkeiten im GPS [173] ausgeglichen werden. In dieser Arbeit wird entweder der Durchschnitt über alle Repräsentationen oder spezialisierte Methoden zur Aggregation von Bounding-Boxen verwendet. Weitere Methoden könnten auch den Median, das Maximum oder das Minimum umfassen. Zudem könnten komplexere Aggregationsmethoden die Repräsentationen gewichten, sodass beispielsweise verschwommene Bilder ignoriert oder Bilder aus großer Distanz geringer gewichtet werden.

Die vorgestellten Methoden zur Interpretation lassen sich ebenfalls weiter optimieren. Die Implementierung des Modells CIPP unterstützt keine Multiklassen-Segmentierung, sodass für jede Klasse ein separates Modell trainiert werden muss. Ein Modell, das direkt unterschiedliche Klassen unterstützt, könnte sowohl eine bessere Segmentierungsgüte liefern als auch die Anwendung vereinfachen. Das Modell StED kann ebenfalls optimiert werden, indem die Auswahl der Pixel für das Training in jeder Stufe möglichst divers erfolgt.

Die Bewertungsmethodik eröffnet dem Anwender die Möglichkeit, Algorithmus-Varianten auszuwählen und effektive Algorithmen zur Abbildung von Veränderungen zu testen. Durch die Anwendung von Deep-Learning könnten neuronale Netzwerke für unterschiedliche Einsatzbereiche mit der Bewertungsmethodik als Metrik aktiv trainiert werden, um gezielt Störfaktoren zu ignorieren. Dies imitiert das Vorgehen von Contrastive-Learning-Verfahren, verwendet jedoch statt Augmentationen natürliche Aufnahmen desselben Objekts. So kann ein Anwender ohne manuellen Aufwand eine Interpretationsmethode gezielt für den Anwendungsfall optimieren. Da die Bewertungsmethodik den vollständigen Algorithmus zur Abbildung von Veränderungen bewertet, kann sogar der vollständige Algorithmus automatisiert an einen Anwendungsfall angepasst werden, solange die Assoziation und Aggregation implementiert werden. Auf diese Weise würde eine vollständig automatisierte Abbildung von Veränderungen auf beliebigen Daten realisiert werden.

A Anhang

A.1 Experimente zur Dateneffizienz

A.1.1 Ergänzende Informationen zum PotholeMix Datensatz

Zusätzliche Informationen zum PotholeMix Datensatz werden im Folgenden aufgeführt. Die Teildatensätze haben unterschiedliche Anteile im Datensatz. Die ungleiche Verteilung hat einen Einfluss auf die Bewertung der Segmentierungen und auf die Auswahl der Bilder für das Training. So wird die Segmentierungsgüte eines Modells maßgeblich durch die Segmentierungsgüte auf dem CPRID Datensatz bestimmt, der den Hauptteil des PotholeMix Datensatzes ausmacht. Eine Zusammenfassung der Größe der Teildatensätze und der Bilder in den Teildatensätzen ist in Tab. A.1 zu sehen.

A.1.2 Ergebnisse des PotholeMix Datensatz

In diesem Abschnitt werden die Ergebnisse des PotholeMix Datensatz vorgestellt. Der PotholeMix Datensatz als Ganzes zeichnet sich durch seine hohe Diversität und ungleiche Klassenverteilung aus. Um die ungleiche Klassenverteilung gesondert zu untersuchen, werden die Teildatensätze einzeln ebenfalls ausgewertet. Auf diese Weise wird die Diversität reduziert und es lassen sich unterschiedlich stark ungleiche Klassenverteilungen miteinander vergleichen. Die durchschnittliche Segmentierungsgüte G_4^{64} über alle Testläufe und Bildanzahlen N_{img} ist in Tab. A.2 angegeben.

Name	Training	Test	Auflösung [Pixel]
Crack500	250	50	2560×1440
GAPs384	353	4	1920×1080
EdmCrack600	480	60	1920×1080
Pothole600	240	180	400×400
CPRID	2000	200	1024×640
CNR	17	2	variabel
Insgesamt	3340	496	variabel

Tabelle A.1: Zusammensetzung des *PotholeMix* Datensatzes: Der *PotholeMix* Datensatz besteht aus sechs Teildatensätzen. Hier wird das Format der Bilder und die Anzahl der Bilder im Datensatz aufgelistet.

Klasse (Datensatz)	ED	CIPP	U-Net
Riss (*PotholeMix*)	0.05	0.08	0.04
Schlagloch (*PotholeMix*)	0.04	0.08	0.04
Riss (*Crack500*)	0.50	0.47	0.33
Riss (*GAPs384*)	0.06	0.06	0.06
Riss (*EdmCrack600*)	0.12	0.03	0.20
Schlagloch (*Pothole600*)	0.44	0.38	0.35
Riss (*CPRID*)	0.06	0.08	0.06
Schlagloch (*CPRID*)	0.02	0.02	0.02
Schlagloch (*CNR*)	0.73	0.71	0.55
Durchschnitt	0.22	0.21	0.18

Tabelle A.2: Ergebnisse *PotholeMix* Datensatz: Hier wird durchschnittliche Segmentierungsgüte G_4^{64} für alle Klassen und Teildatensätze des *PotholeMix* Datensatzes aufgeführt.

Die untersuchten Modelle erreichen auf dem PotholeMix Datensatz eine niedrige durchschnittliche Segmentierungsgüte $G_4^{64} < 0.9$ für beide Klassen *Riss* und *Schlagloch*. Das Modell CIPP erreicht die höchste durchschnittliche Segmentierungsgüte $G_4^{64} = 0.08$ für beide Klassen. Im Vergleich erreichen andere Modelle auf den Teildatensätzen höhere durchschnittliche Segmentierungsgüten G_4^{64}. Ausschließlich auf dem CPRID Datensatz erreicht das CIPP Modell für die Klassen

Riss ebenfalls die höchste durchschnittliche Segmentierungsgüte $G_4^{64} = 0.08$. Der CPRID Datensatz entspricht fast zwei Drittel der Daten des PotholeMix Datensatzes und bestimmt daher maßgeblich dessen Segmentierungsgüte G_4^{64} mit. Diese Beobachtung legt nahe, dass die CIPP sich im Besonderen für die dateneffiziente Auswertung von Datensätzen mit einer hohen Diversität eignet.

Die weitere Auswertung der Teildatensätze gibt Aufschluss über die Fähigkeit der Modelle, ungleiche Klassenverteilungen zu bewältigen. Die durchschnittliche Segmentierungsgüte G_4^{64} der Modelle variiert stark je nach betrachtetem Teildatensatz. Die drei Datensätze mit dem größten Ungleichheit der Klassenverteilung sind GAPs384, EdmCrack600 und CPRID. Hier wird ein Flächenanteil der Klasse (Riss/Schlagloch) im Bild kleiner als 0.01 erreicht (vgl. Tab. 5.1). Diese Datensätze erreichen entsprechend eine niedrige durchschnittliche Segmentierungsgüte G_4^{64} (< 0.20) für alle ausgewerteten Modelle. Da die verwendeten Modelle eine Eingabegröße von 256×256 Pixel haben, ist die Auflösung der Modelle zu klein, um effektiv zu segmentieren. Auf dem EdmCrack600 erreicht das U-Net die höchste Segmentierungsgüte $G_4^{64} = 0.20$, gefolgt von dem ED-Modell mit einer durchschnittlichen Segmentierungsgüte $G_4^{64} = 0.13$. Auf dem CPRID Datensatz wird die höchste Segmentierungsgüte $G_4^{64} = 0.08$ ($+0.02$) von der CIPP erreicht. Der CPRID Datensatz zeigt die Klasse *Riss* in kompakten Ausprägungen während der EdmCrack600 Datensatz die Klasse *Riss* als lange und dünne Linien zeigt. Daraus wird abgeleitet, dass die CIPP eher mit kompakten Klassen umgehen kann als lange und filigrane Klassen.

Im Gegensatz dazu erreichen die Modelle auf den Datensätzen Crack500, Pothole600 und CNR eine deutlich höhere mittlere Segmentierungsgüte $G_4^{64} > 0.43$. Diese Datensätze haben einen höheren Anteil an Klassen in den Annotationen > 0.03 (vgl. Tab. 5.1) und können daher leichter verarbeitet werden. In diesem Datensätzen mit einer eher ausgewogenen Klassenverteilung erreicht das ED-Modell die höchste durchschnittliche Segmentierungsgüte G_4^{64}. Insgesamt erreicht das ED-Modell im Schnitt über alle Datensätze die höchste durchschnittliche Segmentierungsgüte $G_4^{64} = 0.22$. Liegen entsprechend keine Informationen über die Klassenverteilung vor, ist das ED-Modell die beste Option.

A.1.3 Ergebnisse des RTK Datensatz

In diesem Abschnitt werden die Ergebnisse auf dem RTK Datensatz erörtert. Der Datensatz dient hier als Repräsentant für die Domäne *Fahrzeug-Bilddaten* und zeigt die Funktionalität der Modelle im Kontext der unstrukturierten Bilddaten. Die Ergebnisse sind in Tab. A.3 zusammengefasst. Der Fokus der Auswertung liegt auf den Klassen *Unbefestigte Fahrbahn*, *Asphalt* und *Pflastersteine*, die eine hohe Klassenverteilung (Flächenanteil > 0.01) haben. Zusätzlich wird die Klasse *Fahrbahnmarkierung* (Flächenanteil $= 0.004$) untersucht. Trotz der ungleichen Klassenveteilung ist diese Klasse gut sichtbar und unterscheidet sich deutlich vom Hintergrund und das ED Modell erreicht eine hohe durchschnittliche Segmentierungsgüte G_4^{64}. Die verbliebenen Klassen werden nicht ausgewertet, da alle Modelle eine niedrige Segmentierungsgüte $G_4^{64} < 0.03$ erreicht haben.

Klasse	ED	CIPP	U-Net
Unbefestigte Straße	0.16	0.15	0.11
Asphalt	0.40	0.23	0.26
Fahrbahnmarkierung	0.13	0.01	0.01
Pflastersteine	0.40	0.30	0.35
Durchschnitt	0.27	0.18	0.18

Tabelle A.3: Ergebnisse RTK Datensatz: Hier wird durchschnittliche Segmentierungsgüte G_4^{64} für die ausgewerteten Klassen des RTK Datensatzes aufgeführt.

Das ED Modell erreicht für alle betrachteten Klassen die höchsten durchschnittliche Segmentierungsgüte G_4^{64}. Besonders groß ist der Abstand für die Klassen *Asphalt* und *Fahrbahnmarkierung* mit einer Differenz der Segmentierungsgüte von > 0.1.

Die betrachteten Klassen haben eine im Vergleich zum PotholeMix gleiche Klassenverteilung. Dies legt nahe, dass das ED Modell im besonderen bei Datensätzen einsetzbar ist, die eine verhältnismäßig ausgewogene Klassenverteilung von > 0.01 haben. Die Klassen *Fahrbahnmarkierung* bildet dabei eine Ausnahme. Es

kann angenommen werden, das kompakte Klassen, die sich in ihrer Helligkeit von der Umgebung absetzten ebenfalls gut durch das ED Modell segmentieren lassen.

A.1.4 Ergebnisse des FloodNet Datensatz

In diesem Abschnitt werden die Ergebnisse für die vier Klassen *Straße*, *Wasser*, *Baum* und *Gras* dargestellt. Die Ergebnisse geben Aufschluss über die Leistungsfähigkeit der Modelle ED, CIPP, U-Net im Bereich *Remote Sensing*. Für die verbleibenden Klassen mit hoher Ungleichverteilung erreichen alle Modelle eine mittlere Segmentierungsgüte G_4^{64} von maximal 0.08. Diese Klassen werden im Folgenden nicht weiter betrachtet, da ungleiche Klassenverteilungen bereits im Datensatz *PotholeMix* diskutiert wurden.

Datensatz	ED	CIPP	U-Net
Gebäude (überflutet)	0.06	0.08	0.00
Gebäude	0.06	0.06	0.02
Straße (überflutet)	0.03	0.06	0.00
Straße	0.35	0.45	0.14
Wasser	0.22	0.15	0.10
Baum	0.43	0.35	0.16
Fahrzeug	0.02	0.03	0.00
Pool	0.01	0.01	0.00
Gras	0.54	0.61	0.33
Durchschnitt	0.39	0.39	0.18

Tabelle A.4: Ergebnisse FloodNet Datensatz: Hier wird durchschnittliche Segmentierungsgüte G_4^{64} für die ausgewerteten Klassen des FloodNet Datensatzes aufgeführt.

Die Ergebnisse der Modelle sind in Tab. A.4 zusammengefasst. Insgesamt liegen die Modelle ED und CIPP im Mittel über alle Klassen gleichauf ($G_4^{64} = 0.39$). Für die Klassen *Straße* ($G_4^{64} = 0.45$) und *Gras* ($G_4^{64} = 0.61$) erreicht das Modell CIPP die höchste mittlere Segmentierungsgüte G_4^{64}. Da die Klasse *Straße* hell und

die Klasse *Gras* dunkel ist, kann CIPP eine einfache und effektive Lösung finden. Es zeigt sich, dass gerade im Bereich *Remote Sensing* solche einfachen Bildverarbeitungsmethoden erfolgreich eingesetzt werden können. Das ED Modell segmentiert am besten die Klassen *Wasser* ($G_4^{64} = 0.22$) und *Baum* ($G_4^{64} = 0.43$). Diese Klassen unterscheiden sich in Textur, Farbe und Helligkeit von anderen Klassen. Solche komplexen Zusammenhänge können von CIPP nicht erfasst werden, während U-Net nicht genügend Trainingsdaten zur Verfügung stehen. Das Modell U-Net kann sich in keiner Klasse gegen die anderen Modelle durchsetzen und ist für einen dateneffizienten Einsatz in der Domäne *Remote Sensing* nicht geeignet. Grundsätzlich können die Modelle ED und CIPP im Bereich *Remote Sensing* eingesetzt werden, da sie im Durchschnitt gute Ergebnisse erzielen.

A.2 Experimente zu verschiedenen Netzwerk Architekturen auf dem RSC Datensatz

Das ResNet50 wurde als Repräsentant für vDNNs bei dem Entwurf eines Algorithmus zur ABV des Straßenzustands auf dem RSC Datensatz verwendet. Es gibt eine Vielzahl von weiteren vDNNs die in diesem Kontext ausgewertet werden können. In diesem Abschnitt werden weiterführende Experimente aufgeführt, die die vDNNs XceptionNet [48], MobileNetV2 [154] und EfficientNet (B0) [170] als Komponente im Algorithmus zur ABV untersuchen. Alle untersuchten AVs verwenden die vDNNs mit denselben Parametern[1] und nutzen die Methode *Ausschnitt* zur Assoziation. Die Gewichte werden durch *tensorflow* bereitgestellt. Es werden zusätzlich zum bereits verwendeten ResNet50 die Backbones getestet.

Die Ergebnisse auf dem RSC Datensatz sind in Tab. A.5 aufgelistet. Es zeigt sich, dass die AV *efficientnetB0 (A)* ($\beta = 1.05$) und *xception (A)* ($\beta = 1.17$) ein besseres Streuungsverhältnis erreicht als die AV *resnet50 (A)* ($\beta = 1.02$). Nur die AV *mobilenet (A)* ($\beta = 0.76$) eignet sich nicht für den Einsatz zur ABV.

[1] Vortrainiert auf ImageNet, Eingabegröße von 128 x 128 und ein GlobelAverage-Layer zur Transformation der Feature-Maps.

Metrik	Φ	Γ	F	β
efficientnetB0 (A)	0.1114	0.1168	0.59	1.05
mobilenet (A)	0.1212	0.0919	0.19	0.76
xception (A)	1.4947	1.7406	0.66	1.17

Tabelle A.5: Übersicht zusätzliche Ergebnisse RSC Datensatz (unüberwacht): Auflistung der Ergebnisse weiterer AV mit unterschiedlichen vDNNs als Methode zur Interpretation. Alle AV verwenden die Methode *Ausschnitt* zur Assoziation.

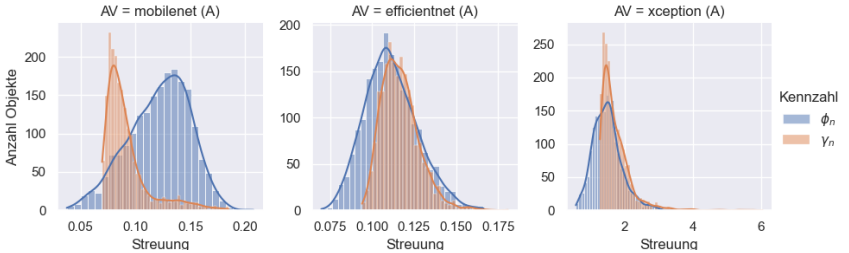


Abbildung A.1: Visualisierung zusätzlicher Ergebnisse RSC Datensatz: Darstellung der Verteilung der Streuungen ϕ_n und γ_n über alle N Straßenabschnitte im RSC Datensatz für die zusätzlich ausgewerteten vDNNs. Alle AV verwenden die Methode *Ausschnitt* zur Assoziation.

Mit einem Streuungsverhältnis von $\beta = 1.17$ liegt die AV *xception (A)* noch vor der AV *gray-sift (A)*. Hier zeigt sich, dass der Hypothesen-Quotient F ebenfalls berücksichtigt werden muss. Im Vergleich ist der Hypothesen-Quotient von *xception (A)* ($F = 0.66$) deutlich kleiner als von *gray-sift (A)* ($F = 0.96$). Bei der Visualisierung der Streuungen in Abbildung A.1 wird deutlich, dass die AV *xception (A)* eine Vielzahl (33%) von Ausreißern hat, bei denen die Intra-Objekt-Streuung ϕ_n größer als die Inter-Objekt-Streuung γ_n ist. Bei den Verbliebenen Straßenabschnitten hingegen wird die Hypothese eindeutig erfüllt, sodass das Streuungsverhältnis im Mittel gut ist.

Diese Ergebnisse zeigen, dass verschiedenen vDNNs in Betracht gezogen werden können, da sich die extrahierten Merkmale voneinander unterscheiden. Gleichzeitig bleibt das Modell *gray-sift* in Kombination mit der Methode *Ausschnitt* die beste AV für die unüberwachte Bewertung des Straßenzustands.

Abbildungsverzeichnis

Tabellenverzeichnis

Eigene Veröffentlichungen

Zeitschriftenartikel

[1] Münke, F., Marcel, S., Mikut, R., und Reischl, M. (2021). Evaluierung von Merkmalen zur Abbildung von Veränderungen in ungeordneten Bilddaten. *at - Automatisierungstechnik*, 69(10):892–902.

[2] Münke, F., Schenk, M., Murr, S., und Reischl, M. (2024a). Adaptable Accelerometer Signal Processing Pipelines for Smartphone based Evenness Estimation. *Journal of Infrastructure, Policy and Development*, 96:617–626.

[3] Münke, F., Schützke, J., Berens, F., und Reischl, M. (2024b). A Review of adaptable Conventional Image Processing Pipelines and Deep Learning on limited Datasets. *Machine Vision and Applications*, 35(2):1–17.

[4] Schilling, M., Scherr, T., Münke, F., Neumann, O., Schutera, M., Mikut, R., und Reischl, M. (2022). Automated Annotator Variability Inspection for Biomedical Image Segmentation. *IEEE Access*, 10:2753–2765.

[5] Schützke, J., Schweidler, S., Münke, F., Orth, A., Khandelwal, A., Breitung, B., Aghassi-Hagmann, J., und Reischl, M. (2023). Accelerating Materials Discovery: Automated Identification of Prospects from X-Ray Diffraction Data in Fast Screening Experiments. *Advanced Intelligent Systems*, 2300501:1–9.

Konferenzbeiträge

[6] Albers, A., Stürmlinger, T., Wantzen, K., Bartosz, G., und Münke, F. (2017). Prediction of the Product Quality of Turned Parts by Real-time Acoustic Emission Indicators. In *Procedia CIRP*, Band 63, Seiten 348–353, Gulf of Naples, Italy.

[7] Münke, F., Marcel, S., Mikut, R., und Reischl, M. (2019). Evaluation of Features for Change Detection in Unstructured Image Data. In *Conference: 29. Workshop Computational Intelligence*, Seiten 1–23, Dortmund, Germany. KIT Scientific Publishing.

[8] Münke, F., Rettenberger, L., Popova, A., und Reischl, M. (2023). A Lightweight Framework for Semantic Segmentation of Biomedical Images. In *Current Directions in Biomedical Engineering*, Band 9, Seiten 190–193, Duisburg, Germany.

[9] Rettenberger, L., Münke, F., Bruch, R., und Reischl, M. (2023). Mask R-CNN Outperforms U-Net in Instance Segmentation for Overlapping Cells. In *Current Directions in Biomedical Engineering*, Band 9, Seiten 335–338, Duisburg, Germany.

[10] Schilling, M., Rettenberger, L., Münke, F., Cui, H., Popova, A., Levkin, P., Mikut, R., und Reischl, M. (2021). Label Assistant: A Workflow for Assisted Data Annotation in Image Segmentation Tasks. In *Conference: 31. Workshop Computational Intelligence*, Seiten 211–234, Berlin, Germany. KIT Scientific Publishing.

Literaturverzeichnis

[11] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G., Davis, A., Dean, J., Devin, M., und et al., S. G. (2015). TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. arXiv:1603.04467. zuletzt abgerufen 2024-03-25.

[12] Abdalla, A., Cen, H., Wan, L., Rashid, R., Weng, H., Zhou, W., und He, Y. (2019). Fine-tuning Convolutional Neural Network with Transfer Learning for Semantic Segmentation of Ground-level Oilseed Rape Images in a Field with high Weed Pressure. *Computers and Electronics in Agriculture*, 167:105091.

[13] Al-Haija, Q. und Adebanjo, A. (2020). Breast Cancer Diagnosis in Histopathological Images Using ResNet-50 Convolutional Neural Network. In *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, Seiten 1–7, Vancouver, BC, Canada.

[14] Alcantarilla, P., Bartoli, A., und Davison, A. (2012). KAZE Features. In *Computer Vision – ECCV 2012*, Seiten 214–227, Florence, Italy.

[15] Alcantarilla, P., Stent, S., Ros, G., Arroyo, R., und Gherardi, R. (2018). Street-View Change Detection with Deconvolutional Networks. *Autonomous Robots*, 42:1301–1322.

[16] Alom, M., Yakopcic, C., Hasan, M., Taha, T., und Asari, V. (2019). Recurrent residual U-Net for medical Image Segmentation. *Journal of Medical Imaging*, 6.

[17] Arandjelovic, R. und Zisserman, A. (2013). All About VLAD. In *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seiten 1578–1585, Portland, OR, USA.

[18] Arthur, D. und Vassilvitskii, S. (2007). k-means++: The Advantages of Careful Seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, Seiten 1027–1035, New Orleans, LA, USA.

[19] Arya, D., Maeda, H., Ghosh, S., Toshniwal, D., und Sekimoto, Y. (2021). RDD2020: An Annotated Image Dataset for Automatic Road Damage Detection using Deep Learning. *Data in Brief*, 36:107133.

[20] Badrinarayanan, V., Kendall, A., und Cipolla, R. (2016). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. arXiv:1511.00561. zuletzt abgerufen 2024-03-25.

[21] Bai, Y., Yang, E., Han, B., Yang, Y., Li, J., Mao, Y., Niu, G., und Liu, T. (2021). Understanding and Improving Early Stopping for Learning with Noisy Labels. arXiv:2106.15853. zuletzt abgerufen 2024-03-07.

[22] Bay, H., Tuytelaars, T., und Van Gool, L. (2006). SURF: Speeded Up Robust Features. In *Computer Vision – ECCV 2006*, Seiten 404–417, Graz, Austria.

[23] Beucher, S. (2000). The Watershed Transformation Applied to Image Segmentation. *Scanning. Microsc.*, 6.

[24] Bisconsini, D., Nicoletti, R., Nuñez, J. Y., und Fernandes Jr, J. (2018). Pavement Roughness Evaluation with Smartphones. *International Journal of Science and Engineering Investigations*.

[25] Boemer, F., Ratner, E., und Lendasse, A. (2018). Parameter-free Image Segmentation with SLIC. *Neurocomputing*, 277:228–236.

[26] Boumaraf, S., Liu, X., Wan, Y., Zheng, Z., Ferkous, C., Ma, X., Li, Z., und Bardou, D. (2021). Conventional Machine Learning versus Deep Learning for Magnification Dependent Histopathological Breast Cancer Image Classification: A Comparative Study with Visual Explanation. *Diagnostics*, 11(3).

[27] Bradski, G. (2000). The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

[28] Bratkova, M., Boulos, S., und Shirley, P. (2009). oRGB: A Practical Opponent Color Space for Computer Graphics. *IEEE Computer Graphics and Applications*, 29(1):42–55.

[29] Brehar, R., Mitrea, D., Vancea, F., Marita, T., Nedevschi, S., Lupsor-Platon, M., Rotaru, M., und Badea, R. (2020). Comparison of Deep-Learning and Conventional Machine-Learning Methods for the Automatic Recognition of the Hepatocellular Carcinoma Areas from Ultrasound Images. *Sensors*, 20(11).

[30] Bruch, R., Keller, F., Böhland, M., Vitacolonna, M., Klinger, L., Rudolf, R., und Reischl, M. (2023). Synthesis of Large Scale 3D Microscopic Images of 3D Cell Cultures for Training and Benchmarking. *PLOS ONE*, 18(3):1–18.

[31] Bruch, R., Rudolf, R., Mikut, R., und Reischl, M. (2020). Evaluation of Semi-Supervised Learning using Sparse Labeling to Segment Cell Nuclei. *Current Directions in Biomedical Engineering*, 6(3):398–401.

[32] Bruch, R. and Scheikl, P. and Mikut, R. and Loosli, F. and Reischl, M. (2021). epiTracker: A Framework for Highly Reliable Particle Tracking for the Quantitative Analysis of Fish Movements in Tanks. *SLAS Technology*, 26(4):367–376.

[33] Bruzzone, L. und Bolovo, F. (2013). A Novel Frameworkfor the Design of Change-Detection Systems for Very-High-ResolutionRemote Sensing Images. *Proceedings of the IEEE*, 101(3):609–630.

[34] Buslaev, A., Iglovikov, V., Khvedchenya, E., Parinov, A., Druzhinin, M., und Kalinin, A. (2020). Albumentations: Fast and Flexible Image Augmentations. *Information*, 11(2):125.

[35] Böhland, M., Bruch, R., Löffler, K., und Reischl, M. (2023). Unsupervised GAN Epoch Selection for Biomedical Data Synthesis. *Current Directions in Biomedical Engineering*, 9:467–470.

[36] Cai, Z. und Vasconcelos, N. (2018). Cascade R-CNN: Delving Into High Quality Object Detection. In *Proceedings of the IEEE Conference on Computer*

Vision and Pattern Recognition (CVPR), Seiten 6154–6162, Salt Lake City, UT, USA.

[37] Canny, J. (1986). A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698.

[38] Carpenter, A., Jones, T., Lamprecht, M., Clarke, C., Kang, I., Friman, O., Guertin, D., Chang, J., Lindquist, R., Moffat, J., Golland, P., und Sabatini, D. (2006). CellProfiler: Image Analysis Software for Identifying and Quantifying Cell Phenotypes. *Genome Biology*, 7(10):R100.

[39] Caye Daudt, R., Le Saux, B., und Boulch, A. (2018). Fully Convolutional Siamese Networks for Change Detection. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, Seiten 4063–4067, Athens, Greece.

[40] Chang, Y. und Mukai, N. (2022). Color Feature Based Dominant Color Extraction. *IEEE Access*, Seiten 1–8.

[41] Chen, G., Hay, G., Carvalho, L., und Wulder, M. (2012). Object-based Change Detection. *International Journal of Remote Sensing*, 33(14):4434–4457.

[42] Chen, H. und Shi, Z. (2020). A Spatial-Temporal Attention-Based Method and a New Dataset for Remote Sensing Image Change Detection. *Remote Sensing*, 12(10):1662.

[43] Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., und Yuille, A. (2016). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40:834–848.

[44] Chen, L., Papandreou, G., Schroff, F., und Adam, H. (2017). Rethinking Atrous Convolution for Semantic Image Segmentation. arXiv:1706.05587. zuletzt abgerufen 2024-08-08.

[45] Chen, L., Zhu, Y., Papandreou, G., Schroff, F., und Adam, H. (2018). Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8–14, 2018, Proceedings, Part VII*, Seite 833–851, Munich, Germany.

[46] Chen, T., Kornblith, S., Norouzi, M., und Hinton, G. (2020). A Simple Framework for Contrastive Learning of Visual Representations. arXiv:2002.05709. zuletzt abgerufen 2024-03-07.

[47] Choi, W. und Cha, Y. (2020). SDDNet: Real-Time Crack Segmentation. *IEEE Transactions on Industrial Electronics*, 67(9):8016–8025.

[48] Chollet, F. (2017). Xception: Deep Learning With Depthwise Separable Convolutions. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seiten 1251–1258, Honolulu, HI, USA.

[49] Choppin, P. und Bauer, M. (1996). Digital Change Detection in Forest Ecosystems with Remote Sensing Imagery. *Remote Sensing Reviews*, 13(3-4):207–234.

[50] Comaniciu, D. und Meer, P. (2002). Mean Shift: A Robust Approach toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619.

[51] Dalal, N. und Triggs, B. (2005). Histograms of oriented Gradients for Human Detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Seiten 886–893, San Diego, CA, USA.

[52] Dhiman, A. und Klette, R. (2020). Pothole Detection Using Computer Vision and Learning. *IEEE Transactions on Intelligent Transportation Systems*, 21(8):3536–3550.

[53] Ding, S., Wang, L., und Cong, L. (2021). Super-Pixel Image Segmentation Algorithm based on Adaptive Equalisation Feature Parameters. *IET Image Processing*, 14(17):4461–4467.

[54] Dino, H. und Abdulrazzaq, M. (2019). Facial Expression Classification Based on SVM, KNN and MLP Classifiers. In *2019 International Conference on Advanced Science and Engineering (ICOASE)*, Seiten 70–75, Zakho - Duhok, Iraq.

[55] Dollar, P. und Zitnick, C. (2013). Structured Forests for Fast Edge Detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Seiten 1841–1848, Sydney, Australia.

[56] Dong, N. und Xing, E. (2018). Few-Shot Semantic Segmentation with Prototype Learning. In *British Machine Vision Conference*, Seite 4, Newcastle, UK.

[57] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., und Houlsby, N. (2020). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. arXiv:2010.11929. zuletzt abgerufen 2024-03-07.

[58] Douangphachanh, V. und Oneyama, H. (2013). A Study on the Use of Smartphones for Road Roughness Condition Estimation. *Journal of the Eastern Asia Society for Transportation Studies*, 10:1551–1564.

[59] Du, P., Liu, S., Gamba, P., Tan, K., und Xia, J. (2012). Fusion of Difference Images for Change Detection Over Urban Areas. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(4):1076–1086.

[60] Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., und Zisserman, A. (2021). With a Little Help from My Friends: Nearest-Neighbor Contrastive Learning of Visual Representations. arXiv:2104.14548. zuletzt abgerufen 2024-02-12.

[61] Eisenbach, M., Stricker, R., Seichter, D., Amende, K., Debes, K., Sesselmann, M., Ebersbach, D., Stoeckert, U., und Gross, H. (2017). How to Get Pavement Distress Detection Ready for Deep Learning? A Systematic Approach. In *2017 International Joint Conference on Neural Networks (IJCNN)*, Seiten 2039–2047, Anchorage, AK, USA.

[62] Ester, M., Kriegel, H., Sander, J., und Xu, X. (1996). A Density-based Algorithm for Discovering Clusters in large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, Seite 226–231, Portland, OR, USA.

[63] Fan, R., Wang, H., Bocus, M., und Liu, M. (2020). We Learn Better Road Pothole Detection: From Attention Aggregation to Adversarial Domain Adaptation. In *Computer Vision – ECCV 2020*, Seiten 285–300, Glasgow, UK.

[64] Fang, S., Li, K., und Li, Z. (2023). Changer: Feature Interaction is What You Need for Change Detection. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–11.

[65] Felzenszwalb, P. und Huttenlocher, D. (2004). Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2):167–181.

[66] Finn, C., Abbeel, P., und Levine, S. (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. arXiv:1703.03400. zuletzt abgerufen 2023-12-20.

[67] Fulkerson, B. und Soatto, S. (2012). Really Quick Shift: Image Segmentation on a GPU. In *Trends and Topics in Computer Vision*, Seiten 350–358, Heraklion, Crete, Greece.

[68] García-Lamont, F., Cervantes, J., Lopez-Chau, A., und Rodríguez, L. (2018). Segmentation of Images by Color Features: A Survey. *Neurocomputing*, 292.

[69] Girshick, R. (2015). Fast R-CNN. In *2015 IEEE International Conference on Computer Vision (ICCV)*, Seiten 1440–1448, Santiago, Chile.

[70] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., und Bengio, Y. (2020). Generative Adversarial Networks. *Commun. ACM*, 63(11):139–144.

[71] Gu, Y., Jin, Z., und Chiu, S. (2015). Active Learning Combining Uncertainty and Diversity for Multi-class Image Classification. *IET Computer Vision*, 9(3):400–407.

[72] Guo, E., Fu, X., Zhu, J., Deng, M., Liu, Y., Zhu, Q., und Li, H. (2018). Learning to Measure Changes: Fully Convolutional Siamese Metric Networks for Scene Change Detection. arXiv: 1810.09111v3. zuletzt abgerufen 2023-12-20.

[73] Hamzah, R. und Ibrahim, H. (2015). Literature Survey on Stereo Vision Disparity Map Algorithms. *Journal of Sensors*, 2016:1–23.

[74] Harris, C., Millman, J., van der Walt, S., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N., Kern, R., und et al. (2020). Array Programming with NumPy. *Nature*, 585(7825):357–362.

[75] He, K., Chen, X., Xie, S., Li, Y., Dollár, P., und Girshick, R. (2021). Masked Autoencoders Are Scalable Vision Learners. arXiv:2111.06377. zuletzt abgerufen 2023-12-20.

[76] He, K., Gkioxari, G., Dollár, P., und Girshick, R. (2017). Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*, Seiten 2980–2988, Venice, Italy.

[77] He, K., Zhang, X., Ren, S., und Sun, J. (2015). Deep Residual Learning for Image Recognition. arXiv:1512.03385. zuletzt abgerufen 2023-12-20.

[78] Hosna, A., Merry, E., Gyalmo, J., Alom, Z., Aung, Z., und Azim, M. (2022). Transfer Learning: A friendly Introduction. *Journal of Big Data*, 9(1):102.

[79] Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., und Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. arXiv:1704.04861. zuletzt abgerufen 2024-07-13.

[80] Huang, G., Liu, Z., Maaten, L., und Weinberger, K. (2018). Densely Connected Convolutional Networks. arXiv:1608.06993. zuletzt abgerufen 2024-07-13.

[81] Huh, M., Agrawal, P., und Efros, A. (2016). What makes ImageNet good for Transfer Learning? arXiv:1608.08614. zuletzt abgerufen 2024-06-12.

[82] Iakubovskii, P. (2020). Segmentation Models. GitHub. zuletzt abgerufen 2024-03-25.

[83] Jeong, J., Jo, H., und Ditzler, G. (2020). Convolutional Neural Cetworks for Pavement Roughness Assessment using Calibration-free Vehicle Dynamics. *Computer-Aided Civil and Infrastructure Engineering*, 35(11):1209–1229.

[84] Jiao, L. und Zhao, J. (2019). A Survey on the New Generation of Deep Learning in Image Processing. *IEEE Access*, 7:172231–172263.

[85] Jmour, N., Zayen, S., und Abdelkrim, A. (2018). Convolutional Neural Networks for Image Classification. In *2018 International Conference on Advanced Systems and Electric Technologies (IC ASET)*, Seiten 397–402, Hammamet, Tunisia.

[86] Juluru, K., Shih, H., Keshava Murthy, K., und Elnajjar, P. (2021). Bag-of-Words Technique in Natural Language Processing: A Primer for Radiologists. *Radiographics : a review publication of the Radiological Society of North America, Inc*, 41:1420–1426.

[87] Karabağ, C., Jones, M., Peddie, C., Weston, A., Collinson, L., und Reyes-Aldasoro, C. (2020). Semantic Segmentation of HeLa Cells: An objective Comparison between one traditional Algorithm and four Deep-Learning Architectures. *PLOS ONE*, 15(10):1–21.

[88] Kass, M., Witkin, A., und Terzopoulos, D. (1988). Snakes: Active Contour Models. *International Journal of Computer Vision*, 1(4):321–331.

[89] Kavitha, J. und Suruliandi, A. (2016). Texture and color feature extraction for classification of melanoma using SVM. In *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE'16)*, Seiten 1–6, Kovilpatti, India.

[90] Khelifi, L. und Mignotte, M. (2020). Deep Learning for Change Detection in RemoteSensing Images: Comprehensive Reviewand Meta-Analysis. *IEEE Access*, 8:126385–126400.

[91] Kingma, D. und Ba, J. (2017). Adam: A Method for Stochastic Optimization. arXiv:1412.6980. zuletzt abgerufen 2024-05-02.

[92] Krizhevsky, A., Sutskever, I., und Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, Seite 84–90, Lake Tahoe, NV, USA.

[93] Lebedev, M., Vizilter, Y., Vygolov, O., Knyaz, V., und Rubis, A. (2018). Change Detection In Remote Sensing Images Using Conditional Adversarial Networks. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2:565–571.

[94] LeCun, Y., Bengio, Y., und Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.

[95] LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., und Jackel, L. (1989). Handwritten Digit Recognition with a Back-Propagation Network. In *Neural Information Processing Systems*, Seiten 396–404, Denver, CO, USA.

[96] Leichtle, T., Geiß, C., Wurm, M., Lakes, T., und Taubenböck, H. (2017). Unsupervised Change Detection in VHR Remote Sensing Imagery: An Object-based Clustering Approach in a Dynamic Urban Environment. *International Journal of Applied Earth Observation and Geoinformation*, 54:15–27.

[97] Leutenegger, S. und Chli, M.and Siegwart, R. (2011). BRISK: Binary Robust Invariant Scalable Keypoints. In *2011 International Conference on Computer Vision*, Seiten 2548–2555, Barcelona, Spain.

[98] Li, Z., Kamnitsas, K., und Glocker, B. (2021). Analyzing Overfitting Under Class Imbalance in Neural Networks for Image Segmentation. *IEEE Transactions on Medical Imaging*, 40(3):1065–1077.

[99] Lie, W., Chiu, H., und Chiang, J. (2020). Disparity Map Estimation From Stereo Image Pair Using Deep Convolutional Network. In *2020 International Computer Symposium (ICS)*, Seiten 365–369, Tainan, Taiwan.

[100] Lin, D., Li, Y., Prasad, S., Nwe, T., Dong, S., und Oo, Z. (2020). CAM-UNET: Class Activation MAP Guided UNET with Feedback Refinement for Defect Segmentation. In *2020 IEEE International Conference on Image Processing (ICIP)*, Seiten 2131–2135, Abu Dhabi, United Arab Emirates.

[101] Lin, T., Goyal, P., Girshick, R., He, K., und Dollár, P. (2017). Focal Loss for Dense Object Detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, Seiten 2999–3007, Venice, Italy.

[102] Linardatos, P., Papastefanopoulos, V., und Kotsiantis, S. (2021). Explainable AI: A Review of Machine LearningInterpretability Methods. *Entropy*, 23(18).

[103] Liu, D., Xiong, Y., Pulli, K., und Shapiro, L. (2011). Estimating Image Segmentation Difficulty. In *Machine Learning and Data Mining in Pattern Recognition*, Seiten 484–495, New York, NY, USA.

[104] Liu, H., Fang, J., Zhang, Z., und Lin, Y. (2021). Localised Edge-Region-based Active Contour for Medical Image Segmentation. *IET Image Processing*, 15(7):1567–1582.

[105] Liu, J. (2013). Image Retrieval based on Bag-of-Words model. arXiv:1304.5168. zuletzt abgerufen 2023-12-20.

[106] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., und Berg, A. (2016). SSD: Single Shot MultiBox Detector. In *Computer Vision – ECCV 2016*, Seiten 21–37, Amsterdam, The Netherlands.

[107] Liu, Z., Mao, H., Wu, C., Feichtenhofer, C., Darrell, T., und Xie, S. (2022). A ConvNet for the 2020s. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seiten 11966–11976, New Orleans, LA, USA.

[108] Lowe, D. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110.

[109] Lu, F. und Milios, E. (1997). Globally Consistent Range Scan Alignment for Environment Mapping. *Autonomous Robots*, 4(4):333–349.

[110] Maaten, L. und Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605.

[111] Mai, S., Hu, H., und Xu, J. (2019). Attentive Matching Network for Few-Shot Learning. *Computer Vision and Image Understanding*, 187:102781.

[112] Mandal, M., Dhar, V., Mishra, A., Vipparthi, S., und Abdel-Mottaleb, M. (2021). 3DCD: Scene Independent End-to-End Spatiotemporal Feature Learning Framework for Change Detection in Unseen Videos. *IEEE Transactions on Image Processing*, 30:546–558.

[113] Martin, V. und Thonnat, M. (2008). A Cognitive Vision Approach to Image Segmentation. *Tools in Artificial Intelligence*, Seiten 265–294.

[114] Masino, J., Thumm, J., Levasseur, G., Frey, M., Gauterin, F., Mikut, R., und Reischl, M. (2018). Characterization of Road Condition with Data Mining Based on Measured Kinematic Vehicle Parameters. *Journal of Advanced Transportation*, 2018:1–10.

[115] Maćkiewicz, A. und Ratajczak, W. (1993). Principal Components Analysis (PCA). *Computers and Geosciences*, 19(3):303–342.

[116] Maška, M., Ulman, V., Delgado-Rodriguez, P., Gómez-de Mariscal, E., Nečasová, T., Guerrero Peña, F., Ren, T., Meyerowitz, E., Scherr, T., Löffler, K., und Mikut, R. e. a. (2023). The Cell Tracking Challenge: 10 Years of Objective Benchmarking. *Nature Methods*, 20(7):1010–1020.

[117] McInnes, L., Healy, J., und Melville, J. (2018). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *The Journal of Open Source Software*, 3(29):861.

[118] Mehra, A.and Bhati, A., Kumar, A., und Malhotra, R. (2021). Skin Cancer Classification Through Transfer Learning Using ResNet-50. In *Emerging Technologies in Data Mining and Information Security*, Seiten 55–62, Singapore, Signapore.

[119] Mei, Q., Gül, M., und Azim, M. (2020). Densely Connected Deep Neural Network considering Connectivity of Pixels for Automatic Crack Detection. *Automation in Construction*, 110:103018.

[120] Mejbri, S., Franchet, C., Ismat-Ara, R., Mothe, J., Brousset, P., und Faure, E. (2019). Deep Analysis of CNN Settings for New Cancer Whole-slide Histological Images Segmentation: The Case of Small Training Sets. In *Proceedings of the 12th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2019) - BIOIMAGING*, Seiten 120–128, Prague, Czech.

[121] Mohidul Islam, S., Jahan, T., und Das, B. (2014). Color Feature based Video Content Extraction and its Application for Poster Generation with Relevance Feedback. In *16th Int'l Conf. Computer and Information Technology*, Seiten 197–202, Khulna, Bangladesh.

[122] Mou, L., Bruzzone, L., und Zhou, X. (2018). Learning Spectral-Spatial-Temporal Features via a Recurrent Convolutional Neural Network for Change Detection in Multispectral Imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 57(2):924–935.

[123] Murtagh, F. (1991). Multilayer Perceptrons for Classification and Regression. *Neurocomputing*, 2(5):183–197.

[124] Nichol, A., Achiam, J., und Schulman, J. (2018). On First-Order Meta-Learning Algorithms. arXiv:1803.02999. zuletzt abgerufen 2023-12-20.

[125] Ning, H., Li, Z., Wang, C., und Yang, L. (2020). Choosing an appropriate Training Set Size when using Existing Data to train Neural Networks for Land Cover Segmentation. *Annals of GIS*, 26(4):329–342.

[126] Noh, H., Hong, S., und Han, B. (2015). Learning Deconvolution Network for Semantic Segmentation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, Seiten 1520–1528, Santiago, Chile.

[127] Ojala, T., Pietikäinen, M., und Mäenpää, T. (2000). Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns. In *Computer Vision – ECCV 2000*, Seiten 404–420, Dublin, Ireland.

[128] Ooi, W. und Lim, C. (2006). Fuzzy Clustering of Color and Texture Features for Image Segmentation: A Study on Satellite Image Retrieval. *Journal of Intelligent and Fuzzy Systems*, 17(3):297–311.

[129] Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66.

[130] O'Mahony, N., Campbell, S., Krpalkova, L., Carvalho, A., Walsh, J., und Riordan, D. (2021). Representation Learning for Fine-Grained Change Detection. *Sensors*, 21(13):4486.

[131] Park, J., Jang, J., Yoo, S., Lee, S., Kim, U., und Kim, J. (2021). ChangeSim: Towards End-to-End Online Scene Change Detection in Industrial Indoor Environments. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Seiten 8578–8585, Prague, Czech Republic.

[132] Passos, B. T., Cassaniga, M., Fernandes, A., Medeiro, K., und Comunello, E. (2020). Cracks and Potholes in Road Images Dataset. Mendeley Data. zuletzt abgerufen 2024-02-07.

[133] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., und Vanderplas, J. e. a. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[134] Pollard, T. und Mundy, J. (2007). Change Detection in a 3-d World. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, Seiten 1–6, Minneapolis, MN, USA.

[135] Polomoshnov, M., Reichert, K., Rettenberger, L.and Ungerer, M., Hernandez-Sosa, G., Gengenbach, U., und Reischl, M. (2024). Image-based Identification of Optical Quality and Functional Properties in Inkjet-printed Electronics using Machine Learning. *Journal of Intelligent Manufacturing*.

[136] Prewitt, J. (1970). Object Enhancement and Extraction. In *Picture Processing and Psychopictorics*, Seiten 75–149, New York, NY, USA.

[137] Radke, R., Andra, S., AlKofahi, O., und Roysam, B. (2005). Image Change Detection Algorithms: A Systematic Survey. *IEEE Transactions on Image Processing*, 14(3):294–307.

[138] Rahnemoonfar, M., Chowdhury, T., Sarkar, A., Varshney, D., Yari, M., und Murphy, R. (2020). FloodNet: A High Resolution Aerial Imagery Dataset for Post Flood Scene Understanding. arXiv:2012.02951. zuletzt abgerufen 2024-03-26.

[139] Ranieri, A., Thompson, E. M., und Biasotti, S. (2022). Pothole Mix. Mendeley Data. zuletzt abgerufen 2024-02-07.

[140] Rashedi, K., Ismail, M., Al Wadi, S., Serroukh, A., Alshammari, T., und Jaber, J. (2024). Multi-Layer Perceptron-Based Classification with Application to Outlier Detection in Saudi Arabia Stock Returns. *Journal of Risk and Financial Management*, 17(2).

[141] Rateke, T., Justen, K., und Wangenheim, . (2019). Road Surface Classification with Images Captured From Low-cost Cameras – Road Traversing Knowledge (RTK) Dataset. *Revista de Informática Teórica e Aplicada*, 26(3):50–64.

[142] Reddy, A. und Juliet, D. (2019). Transfer Learning with ResNet-50 for Malaria Cell-Image Classification. In *2019 International Conference on Communication and Signal Processing (ICCSP)*, Seiten 0945–0949, Chennai, India.

[143] Redmon, J., Divvala, S., Girshick, R., und Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seiten 779–788, Las Vegas, NV, USA.

[144] Reischl, M. (2006). *Ein Verfahren zum automatischen Entwurf von Mensch-Maschine-Schnittstellen am Beispiel myoelektrischer Handprothesen*. Schriftenreihe des Instituts für Angewandte Informatik / Automatisierungstechnik ; 13. KIT Scientific Publishing, Karlsruhe, Germany.

[145] Ren, S., He, K., Girshick, R., und Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems*, Band 28, Montreal, Canada.

[146] Rezende, E., Ruppert, G., Carvalho, T., Ramos, F., und de Geus, P. (2017). Malicious Software Classification Using Transfer Learning of ResNet-50 Deep Neural Network. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Seiten 10111–014, Cancun, Mexico.

[147] Rodríguez Outeiral, R., Bos, P., van der Hulst, H., Al-Mamgani, A., Jasperse, B., Simões, R., und van der Heide, U. (2022). Strategies for Tackling the Class Imbalance Problem of Oropharyngeal Primary Tumor Segmentation on magnetic Resonance Imaging. *Physics and Imaging in Radiation Oncology*, 23:144–149.

[148] Ronneberger, O., Fischer, P., und Brox, T. (2015). U-Net: Convolutional Networks for BiomedicalImage Segmentation. arXiv:1505.04597. zuletzt abgerufen 2023-12-20.

[149] Rousseeuw, P. (1987). Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis . *Journal of Computational and Applied Mathematics*, 20:53–65.

[150] Rublee, E., Rabaud, V., Konolige, K., und Bradski, G. (2011). ORB: An efficient Alternative to SIFT or SURF. In *2011 International Conference on Computer Vision*, Seiten 2564–2571, Barcelona, Spain.

[151] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., und et al. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):1573–1405.

[152] Sabapathy, A. und Biswas, A. (2023). Road surface classification using accelerometer and speed data: evaluation of a convolutional neural network model. *Neural Computing and Applications*, Seiten 1–12.

[153] Sakurada, K., Shibuya, M., und Wang, W. (2022). Weakly Supervised Silhouette-based Semantic Scene Change Detection. arXiv:1811.11985. zuletzt abgerufen 2023-12-20.

[154] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., und Chen, L. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seiten 4510–4520, Salt Lake City, UT, USA.

[155] Scherr, T., Löffler, K., Böhland, M., und Mikut, R. (2020). Cell Segmentation and Tracking using CNN-based Distance Predictions and a Graph-based Matching Strategy. *PLOS ONE*, 15(12):1–22.

[156] Schwarz, T. und Santana, F. (2021). *Road Damage Reference Guide*. vialyics, Stuttgart, Germany.

[157] Schönberger, J. und Frahm, J. (2016). Structure-from-Motion Revisited. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seiten 4104–4113, Las Vegas, NV, USA.

[158] Sergyan, S. (2007). Color Content-based Image Classification. In *5th Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence and Informatics*, Seiten 427–434, Poprad, Slovakia.

[159] Shagdar, Z., Ullah, M.and Ullah, H., und Cheikh, F. (2021). Geometric Deep Learning for Multi-Object Tracking: A Brief Review. In *2021 9th European Workshop on Visual Information Processing (EUVIP)*, Seiten 1–6, Paris, France.

[160] Shekhar, R. und Jawahar, C. (2012). Word Image Retrieval Using Bag of Visual Words. In *2012 10th IAPR International Workshop on Document Analysis Systems*, Seiten 297–301, Gold Coast, Australia.

[161] Shi, Y., Cui, L., Qi, Z., Meng, F., und Chen, Z. (2016). Automatic Road Crack Detection Using Random Structured Forests. *IEEE Transactions on Intelligent Transportation Systems*, 17(12):3434–3445.

[162] Shorten, C. und Khoshgoftaar, T. (2019). A Survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1):60.

[163] Singh, A. und Singh, K. (2018). Unsupervised Change Detection in Remote Sensing Images using Fusion of Spectral and Statistical Indices. *The Egyptian Journal of Remote Sensing and Space Science*, 21(3):345–351.

[164] Singh, S., Srivastava, D., und Agarwal, S. (2017). GLCM and its Application in Pattern Recognition. In *2017 5th International Symposium on Computational and Business Intelligence (ISCBI)*, Seiten 20–25, Dubai, United Arab Emirates.

[165] Smith, A. (1978). Color Gamut Transform Pairs. In *Proceedings of the 5th Annual Conference on Computer Graphics and Interactive Techniques*, Seite 12–19, New York, NY, USA.

[166] Srikham, M. (2012). Active Contours Segmentation with Edge based and Local Region based. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, Seiten 1989–1992, Tsukuba, Japan.

[167] Stahle, L. und Wold, S. (1989). Analysis of Variance (ANOVA). *Chemometrics and Intelligent Laboratory Systems*, 6(4):259–272.

[168] Sánchez, J., Mensink, T., und Verbeek, J. (2013). Image Classification with the Fisher Vector: Theory and Practice. *International Journal of Computer Vision*, 105.

[169] Taheri, H. und Xia, Z. (2021). SLAM; Definition and Evolution. *Engineering Applications of Artificial Intelligence*, 97:104032.

[170] Tan, M. und Le, Q. (2020). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. arXiv:1905.11946. zuletzt abgerufen 2024-07-13.

[171] Taveira, L., Kurc, T., Melo, A., Kong, J., Bremer, E., Saltz, J., und Teodoro, G. (2018). Multi-objective Parameter Auto-tuning for Tissue Image Segmentation Workflows. *Journal of Digital Imaging*, 32:521–533.

[172] Teodoro, G., Kurç, T., Taveira, L., Melo, A., Gao, Y., Kong, J., und Saltz, J. (2017). Algorithm Sensitivity Analysis and Parameter Tuning for Tissue Image Segmentation Pipelines. *Bioinformatics*, 33(7):1064–1072.

[173] Thin, L., Ting, L., Husna, N., und Husin, M. (2016). GPS Systems Literature: Inaccuracy Factors and Effective Solutions. *The International Journal of Computer Networks and Communications (IJCNC)*, 8(2):123–131.

[174] Tomasi, C. und Kanade, T. (1992). Shape and Motion from Image Streams under Orthography: AFfactorization Method. *International Journal of Computer Vision*, 9(2):137–154.

[175] Tong, Z., Ma, T., Huyan, J., und Zhang, W. (2022). Pavementscapes: a large-scale hierarchical image dataset for asphalt pavement damage segmentation. arXiv:2208.00775. zuletzt abgerufen 2024-03-07.

[176] Tuytelaars, T. (2013). Dense Interest Points. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Seiten 2281–2288, San Francisco, CA, USA.

[177] Van der Walt, S., Schönberger, J., Nunez-Iglesias, J., Boulogne, F., Warner, J., Yager, N., Gouillart, E., und Yu, T. (2014). scikit-image: Image Processing in Python. *PeerJ*, 2:e453.

[178] Van Valen, D., Kudo, T., Lane, K., Macklin, D., Quach, N., DeFelice, M., Maayan, I., Tanouchi, Y., Ashley, E., und Covert, M. (2016). Deep Learning Automates the Quantitative Analysis of Individual Cells in Live-Cell Imaging Experiments. *PLOS Computational Biology*, 12(11):1–24.

[179] Vedaldi, A. und Soatto, S. (2008). Quick Shift and Kernel Methods for Mode Seeking. In *Computer Vision – ECCV 2008*, Seiten 705–718, Marseille, France.

[180] Vezhnevets, A., Buhmann, J., und Ferrari, V. (2012). Active Learning for Semantic Segmentation with expected Change. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Seiten 3162–3169, Providence, RI, USA.

[181] Vinyals, O., Blundell, C., Lillicrap, T., kavukcuoglu, k., und Wierstra, D. (2016). Matching Networks for One Shot Learning. In *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, Seiten 3637–3645, Barcelona, Spain.

[182] Viola, P. und Jones, M. (2001). Rapid Object Detection using a Boosted Cascade of Simple Features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Band 1, Kauai, HI, USA.

[183] Vizcaíno, A., Sánchez-Cruz, H., Sossa, H., und Quintanar, J. (2021). Pixel-Wise Classification in Hippocampus Histological Images. *Computational and mathematical methods in medicine*, 2021:6663977.

[184] Wang, N. und Yeung, D. (2013). Learning a Deep Compact Image Representation for Visual Tracking. In *Advances in Neural Information Processing Systems*, Seite 809–817, Lake Tahoe, NV, USA.

[185] Wang, Y., Yuan, Y., und Lei, Z. (2020). Fast SIFT Feature Matching Algorithm Based on Geometric Transformation. *IEEE Access*, 8:88133–88140.

[186] Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I., und Xie, S. (2023). ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seiten 16133–16142, Vancouver, BC, Canada.

[187] Wu, J., Li, B., Qin, Y., Ni, W., und Zhang, H. (2021). An Object-based Graph Model for Unsupervised Change Detection in High Resolution Remote Sensing Images. *International Journal of Remote Sensing*, 42(16):6209–6227.

[188] Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J., und Luo, P. (2021). SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. arXiv:2105.15203. zuletzt abgerufen 2023-12-20.

[189] Xu, Y., Zhou, X., Chen, S., und Li, F. (2019). Deep Learning for Multiple Object Tracking: A Survey. *IET Computer Vision*, 13(4):355–368.

[190] Yan, Y., Shen, Y., und Li, S. (2009). Unsupervised Color-Texture Image Segmentation Based on A New Clustering Method. In *2009 International Conference on New Trends in Information and Service Science*, Seiten 784–787, Beijing, China.

[191] Yang, B., Liu, C., Li, B., Jiao, J., und Ye, Q. (2020). Prototype Mixture Models for Few-Shot Semantic Segmentation. In *Computer Vision – ECCV 2020*, Seiten 763–778, Glasgow, UK.

[192] Yang, F., Zhang, L., Yu, S., Prokhorov, D., Mei, X., und Ling, H. (2019). Feature Pyramid and Hierarchical Boosting Network for Pavement Crack Detection. *IEEE Transactions on Intelligent Transportation Systems*, 99(1):1–11.

[193] Yang, J., Jiang, Y., Hauptmann, A., und Ngo, C. (2007). Evaluating Bag-of-Visual-Words Representations in Scene Classification. In *MIR '07: Proceedings of the International Workshop on Multimedia Information Retrieval*, Seiten 197–206, Augsburg, Germany.

[194] Yang, K., Xia, G., Liu, Z., Du, B., Yang, W., Pelillo, M., und Zhang, L. (2022). Asymmetric Siamese Networks for Semantic Change Detection in Aerial Images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–18.

[195] Yim, J., Ju, J., Jung, H., und Kim, J. (2015). Image Classification Using Convolutional Neural Networks With Multi-stage Feature. In Kim, J., Yang, W., Jo, J., Sincak, P., und Myung, H., Herausgeber, *Robot Intelligence Technology and Applications 3*, Band 345 in *Advances in Intelligent Systems and Computing*, Seiten 587–594, Cham, Schweiz. Springer.

[196] Zbontar, J., Jing, L., Misra, I., LeCun, Y., und Deny, S. (2021). Barlow Twins: Self-Supervised Learning via Redundancy Reduction. arXiv:2103.03230. zuletzt abgerufen 2024-02-12.

[197] Zhai, M., Xiang, X., Lv, N., und Kong, X. (2021). Optical Flow and Scene Flow Estimation: A Survey. *Pattern Recognition*, 114:107861.

[198] Zhan, T., Gong, M., Jiang, X., und Zhang, M. (2020). Unsupervised Scale-Driven Change Detection With Deep Spatial–Spectral Features for VHR Images. *IEEE Transactions on Geoscience and Remote Sensing*, 58(8):5653–5665.

[199] Zhang, C., Yue, P., Tapete, D., Jiang, L., Shangguan, B., Huang, L., und Liu, G. (2020). A Deeply Supervised Image Fusion Network for Change Detection in High Resolution bi-temporal Remote Sensing Images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 166:183–200.

[200] Zhang, H., Cissé, M., Dauphin, Y., und Lopez-Paz, D. (2017). mixup: Beyond Empirical Risk Minimization. arXiv:1710.09412. zuletzt abgerufen 2023-12-20.

[201] Zhang, L., Yang, F., Zhang, Y. D., und Zhu, Y. (2016). Road Crack Detection using Deep Convolutional Neural Network. In *2016 IEEE International Conference on Image Processing (ICIP)*, Seiten 3708–3712, Phoenix, AZ, USA.

[202] Zhao, B. und Nagayama, T. (2017). IRI Estimation by the Frequency Domain Analysis of Vehicle Dynamic Responses. *Procedia Engineering*, 188:9–16.

[203] Zhao, H., Shi, J., Qi, X., Wang, X., und Jia, J. (2016). Pyramid Scene Parsing Network. arXiv:1612.01105. zuletzt abgerufen 2024-08-08.

[204] Zhao, L., Tang, P., und Huo, L. (2014). Land-Use Scene Classification Using a Concentric Circle-Structured Multiscale Bag-of-Visual-Words Model. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(12):4620–4631.

[205] Zheng, Z., Zhong, Y., Wang, J., Ma, A., und Zhang, L. (2023). FarSeg++: Foreground-Aware Relation Network for Geospatial Object Segmentation in High Spatial Resolution Remote Sensing Imagery. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13715–13729.

[206] Zhou, N., Liu, Y., und Hong, L. (2019). An Improved SLIC Super-pixel Extraction Algorithm Based on MMTD. In *2019 Tenth International Conference on Intelligent Control and Information Processing (ICICIP)*, Seiten 233–238, Marrakesh, Morocco.