HAW HAMBURG

BACHELOR THESIS
Paul Demski

Kamerabasierte Lokalisierung durch Merkmalserkennung in Straßenumgebungen für die Miniaturautonomie

FAKULTÄT TECHNIK UND INFORMATIK
Department Informatik

Faculty of Engineering and Computer Science
Department Computer Science

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
Hamburg University of Applied Sciences

Paul Demski

Kamerabasierte Lokalisierung durch Merkmalserkennung in Straßenumgebungen für die Miniaturautonomie

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science Informatik Technischer Systeme*
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Stephan Pareigis
Zweitgutachter: Prof. Dr. Tim Tiedemann

Eingereicht am: 5. Mai 2025

Paul Demski

Thema der Arbeit

Kamerabasierte Lokalisierung durch Merkmalserkennung in Straßenumgebungen für die Miniaturautonomie

Stichworte

Bildklassifizierung, Miniaturautonomie, kamerabasierte Lokalisierung

Kurzzusammenfassung

Die Selbstlokalisierung in Bereichen wie Augmented Reality, Robotik oder autonomes Fahren gewinnt zunehmend an Bedeutung. Oftmals ist eine Lokalisierung mittels Positionierungssystemen nicht möglich oder zu ungenau. In dieser Arbeit wird die kamerabasierte Lokalisierung mittels eines Klassifikationsmodells, spezifisch YOLOv8, im Bereich der Miniaturautonomie untersucht. Als Testumgebung wurde das Mikrowunderland der HAW Hamburg verwendet, da es eine Vielzahl von Straßenumgebungsmerkmalen abbildet. Es wurde untersucht, wie genau das Modell seine Position auf der Strecke vorhersagen kann und wie viele Bilder pro Klasse benötigt werden. Es wurde auch analysiert, was das Modell sieht und was die Gründe für eine falsche Vorhersage sind. Es stellte sich heraus, dass bereits eine geringe Anzahl von Bildern im Training eine Vorhersagegenauigkeit von 2 Zentimetern liefern kann. Falsche Klassifizierungen waren in den meisten Fällen auf eine fehlerhafte Aufnahme der Bilder zurückzuführen und nicht auf die Fähigkeit des Modells, Unterschiede richtig zu erkennen.

Paul Demski

Title of Thesis

Camera-based Localization through Feature Detection in Road Environments for Miniature Autonomy

Keywords

Image classification, miniature autonomy, image based localization

Abstract

Self-localization is becoming increasingly important in areas such as augmented reality, robotics and autonomous driving. Localization using positioning systems is often not possible or too imprecise. In this thesis, camera-based localization using a classification model, specifically YOLOv8, is investigated in the field of miniature autonomy. The HAW Hamburg's Mikrowunderland was used as a test environment, as it depicts a large number of road environment features. It was investigated how accurately the model can predict its position on the route and how many images are required per class. It was also analyzed what the model sees and what the reasons for an incorrect prediction are. It turned out that even a small number of images in training can provide a prediction accuracy of 2 centimeters. In most cases, incorrect classifications were due to incorrect image acquisition and not to the model's ability to correctly recognize differences.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Das autonome Fahren stellt eine der zentralen Herausforderungen moderner Verkehrssysteme dar und beschäftigt sich mit der Aufgabe, Fahrzeuge im Straßenverkehr ohne menschliche Hilfe zu steuern. Um dies zu ermöglichen, müssen autonome Systeme die Umgebung zuverlässig und präzise wahrnehmen können. Hierzu gehören die Erkennung von Straßenmarkierungen und anderen relevanten Umgebungsmerkmalen, da diese maßgeblich den Bewegungsraum eines Fahrzeugs definieren. Die genaue Klassifizierung solcher Merkmale ist essenziell, um Verhaltensregeln, wie das Befahren von Fahrspuren oder das Erkennen von Kreuzungen, korrekt umzusetzen.

In der Praxis erfordert die Entwicklung und das Testen solcher Systeme nicht nur komplexe Software, sondern auch leistungsfähige Hardware und sichere Testumgebungen. Der Aufbau entsprechender Testareale und der Einsatz echter Fahrzeuge sind jedoch mit enormen Kosten und Risiken verbunden. Daher ist die Nutzung von Simulationen weit verbreitet. Eine kostengünstigere und realitätsnähere Alternative zu Simulationen stellt der Einsatz von Miniaturmodellen dar. Diese erlauben es, autonome Systeme in physikalisch greifbaren Umgebungen zu erproben und dabei ähnliche Herausforderungen zu simulieren, wie sie im realen Straßenverkehr auftreten.

An der Hochschule für Angewandte Wissenschaften (HAW) Hamburg existiert eine speziell entwickelte Miniatur-Teststrecke im Maßstab 1:87, die für Experimente im Bereich der autonomen Systeme genutzt wird. Dieser Maßstab bietet die Möglichkeit, vielfältige Straßenszenarien abzubilden und autonome Systeme unter realen Bedingungen zu evaluieren. Gleichzeitig bringt die Skalierung erhebliche technische Herausforderungen mit sich, insbesondere im Hinblick auf die präzise Lokalisierung von Fahrzeugen.

Vor diesem Hintergrund untersucht die vorliegende Arbeit die kamerabasierte Lokalisierung in einer Miniaturumgebung mithilfe eines auf YOLOv8 basierenden Klassifikationsmodells. Ziel ist es, die Position eines Fahrzeugs durch die Erkennung und Klassifizierung von Straßenumgebungsmerkmalen mit einer Genauigkeit von wenigen Zentimetern

zu bestimmen. Dabei werden verschiedene Experimente durchgeführt, um die Leistungsfähigkeit des Modells zu evaluieren und die Einflüsse von Faktoren wie Datensatzgröße, Lichtverhältnissen und Bildvariabilität auf die Genauigkeit zu analysieren.

Die Arbeit liefert damit nicht nur Erkenntnisse zur Machbarkeit kamerabasierter Lokalisierung im Maßstab 1:87, sondern bietet auch eine Grundlage für die Optimierung ressourcenschonender Verfahren, die potenziell auf reale Anwendungen übertragen werden können.

1.1 Zielsetzung

Das Ziel der vorliegenden Arbeit besteht in der Konzeption und Evaluierung eines kamerabasierten Lokalisierungssystems für eine Modellumgebung im Maßstab 1:87. Dabei liegt der Fokus der Arbeit auf der Anwendung eines YOLOv8-Klassifikationsmodells zur Detektion und Klassifikation von Straßenmerkmalen und damit vorhandener Objekte innerhalb vordefinierter Abschnitte der Teststrecke, um so die Position eines Fahrzeugs innerhalb der Teststrecke in genauer Weise festzulegen.

Zur Lokalisierung existieren verschiedene Ansätze, darunter Lidar-basierte Systeme, GPS-gestützte Verfahren sowie Kombinationen mehrerer Sensordaten. Diese Arbeit beschränkt sich gezielt auf den Ansatz der kamerabasierten Lokalisierung, um eine detaillierte Untersuchung der Leistungsfähigkeit nur eines Verfahrens innerhalb der vorgegebenen Rahmenbedingungen zu ermöglichen.

Der Schwerpunkt der Arbeit liegt in der Evaluierung der Performanz des Modells unter diversen Bedingungen. Dazu wird auf unterschiedliche Datensätze zurückgegriffen, die sich in verschiedenen Lichtverhältnissen und der Anwesenheit von weiteren Fahrzeugen auf der Fahrspur voneinander unterscheiden. Ein weiteres Ziel besteht in einer systematischen Analyse falscher Klassifikationen, um so die Ursachen hierfür präzise herausarbeiten zu können. Zur Erleichterung der Nachvollziehbarkeit des Klassifikationsvorgangs wird die Methode der Eigen-Class Activation Map (Eigen-CAM) eingesetzt.

Abschließend soll die Arbeit die Realisierbarkeit eines kamerabasierten Lokalisierungsverfahrens innerhalb der Modellumgebung belegen und eine Aussage zur erreichten Positionsgenauigkeit treffen. Es sollen auch die Limitationen und Herausforderungen des vorgeschlagenen Ansatzes dargestellt werden. Die durch die Arbeit erzielten Ergebnisse

sollen einen Beitrag zur Vergleichsbasis für künftige Anpassungen und ggf. Erweiterungen in größeren Maßstäben leisten.

1.2 Stand der Technik

Grundsätzlich lässt sich der Bereich der kamerabasierten Lokalisierung in zwei Kategorien untergliedern: Structure-based Localization (SBL, strukturbasierte Lokalisierung) und Retrieval-based Localization (RBL, abrufbasierte Lokalisierung). SBL umfasst Methoden wie Structure from Motion (SfM) [8, 15, 6] oder Simultaneous Localization and Mapping (SLAM) [4, 7]. Dies sind 2D-3D match-basierte Methoden, bei denen Merkmale eines 2D-Bildes mit einer 3D-Punktwolke abgeglichen werden.

Als Alternative zu SBL wird RBL zunehmend erforscht, da es eine effiziente und skalierbare Möglichkeit zur Lokalisierung bietet [20]. Anstatt eine 3D-Karte zu nutzen, wird ein aufgenommenes Bild mit einer Datenbank bereits georeferenzierter Bilder verglichen, um das Bild mit der größten Übereinstimmung zu finden [16, 9]. Hierbei werden globale Merkmale extrahiert und miteinander verglichen. Frühe Ansätze setzten auf PCA-Merkmalsextraktion [11, 10], Histogrammabgleich [3] oder globale Feature-Deskriptoren wie GIST [13]. Zusätzlich wurde mit WI-SURF [1] oder BRIEF-Gist [19] eine Aggregation lokaler Merkmale als globaler Bilddeskriptor genutzt. Eine weitere Möglichkeit, Bilder zu vergleichen, ist die Bag-of-Words (BoW)-Methode [17], die auf Feature-Clustering basiert und Bilder anhand von Histogrammen visueller Wörter vergleicht.

Den ersten Durchbruch im Bereich lernbasierter Methoden schafften Chen et al. [5], indem sie CNNs mit räumlichen und sequentiellen Filtern kombinierten. Weitere Arbeiten nutzten CNNs als Deskriptor-Extraktoren für Place Recognition. Sünderhauf et al. [18] setzten auf vortrainierte AlexNet-Features und zeigten, dass mid-level Merkmale robuster gegenüber Umgebungsvariationen sind. Zhang et al. [21] entwickelten einen graph-basierten Ansatz, bei dem visuelle Merkmale mit Informationen der sequentiellen Bildabfolge kombiniert wurden. Diese Methoden nutzten vortrainierte CNNs als Black-Box, um globale Features zu extrahieren.

Um dieses Problem zu lösen, entwickelten Relja et al. [2] mit NetVLAD einen End-to-End-Ansatz, der die Aggregation von CNN-Merkmalen verbessert. Dennoch haben all diese Lösungen einige Nachteile: Sie erfordern umfangreiche Datenbanken, hohe Rechenleistung und komplexe Feature-Aggregation, um robuste Ergebnisse zu erzielen.

Mein Ansatz hingegen verfolgt einen effizienteren, klassenbasierten Lokalisierungsansatz. Durch die direkte Klassifikation von Orten mittels YOLO wird eine schnelle und ressourcenschonende Alternative geboten, die mit geringerem Speicherbedarf und reduzierter Rechenlast auskommt. Dies macht sie besonders geeignet für Miniaturautonomie und echtzeitkritische Anwendungen.

2 Grundlagen

In diesem Kapitel werden die grundlegenden Konzepte und Methoden vorgestellt, die für diese Arbeit relevant sind. Der Schwerpunkt liegt auf der Bildklassifikation, die als zentrale Methode zur Zuordnung von Bildern zu vordefinierten Kategorien dient. Dazu wird zunächst auf die Funktionsweise von Convolutional Neural Networks (CNNs) eingegangen, die eine wichtige Grundlage für moderne Klassifikationsmodelle darstellen.

Anschließend wird das verwendete Modell YOLOv8 im Detail erläutert, das in dieser Arbeit zur Klassifikation der Straßenabschnitte eingesetzt wird. Ergänzend wird die Eigen-Class Activation Map (Eigen-CAM) vorgestellt, eine Technik, die zur Visualisierung der Entscheidungsprozesse des Modells genutzt wird.

2.1 Bildklassifikation und Abgrenzung zu verwandten Verfahren

Die Bildklassifikation ist eine zentrale Aufgabe der Computer Vision, bei der Bilder anhand ihrer visuellen Merkmale automatisch in vordefinierte Kategorien oder Labels eingeordnet werden. Es existieren verschiedene Arten: Die binäre Klassifikation teilt Bilder in zwei Kategorien ein, ideal für Ja/Nein-Entscheidungen. Die Mehrklassenklassifikation erweitert dies auf drei oder mehr Kategorien. Im Gegensatz dazu erlaubt die Mehrlabelklassifikation die gleichzeitige Zuordnung mehrerer Kategorien zu einem Bild. Schließlich ordnet die hierarchische Klassifikation Klassen in einer Baumstruktur an, mit allgemeinen Oberkategorien und spezifischeren Unterkategorien.

Im Rahmen dieser Arbeit wird eine Mehrklassenklassifikation eingesetzt, bei der jedem Bild genau ein Streckenabschnitt als Klasse zugewiesen wird.

Die Bildklassifikation ordnet einem gesamten Bild ein spezifisches Label zu, ohne die Positionen von Objekten zu berücksichtigen. Im Unterschied dazu identifiziert die Objektlokalisierung nicht nur Objekte, sondern bestimmt auch deren genaue Positionen im Bild mittels Bounding Boxes. Die Objektdetektion kombiniert beides, indem sie mehrere Objekte erkennt, labelt und ihre Positionen angibt. Diese Arbeit konzentriert sich ausschließlich auf die Bildklassifikation, bei der jedes Bild als Einheit betrachtet und einer Klasse zugeordnet wird, ohne einzelne Objekte zu lokalisieren.

2.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) sind eine spezialisierte Architektur künstlicher neuronaler Netze, die sich durch ihre besondere Eignung für die Verarbeitung visueller Daten auszeichnen. Sie bilden die Grundlage für viele moderne Bildverarbeitungs- und Klassifikationsmodelle, darunter auch das in dieser Arbeit verwendete YOLOv8-Modell. CNNs ermöglichen es, komplexe Muster und Merkmale in Bildern zu erkennen und diese effizient zu analysieren, was sie zu einem unverzichtbaren Werkzeug in der Computer Vision macht.

2.2.1 Convolutional Neural Networks in der Bildklassifikation

CNNs sind auf die Verarbeitung visueller Daten spezialisierte neuronale Netze. Sie nutzen Faltungsschichten zur Extraktion relevanter Merkmale wie Kanten oder Texturen, Pooling-Schichten zur Reduktion der Datenmenge sowie Aktivierungsfunktionen wie ReLU zur Einbringung von Nichtlinearitäten. Abschließend erfolgt die Klassifikation über vollständig verbundene Schichten.

Durch ihre Fähigkeit, sowohl lokale als auch globale Muster zu erkennen, sind CNNs besonders robust gegenüber Bildverzerrungen oder Beleuchtungsänderungen. In dieser Arbeit bildet ein CNN die Grundlage des verwendeten YOLOv8-Modells zur Klassifikation von Straßenabschnitten anhand visueller Merkmale.

2.3 YOLOv8

YOLOv8 ("You Only Look Once") ist eine moderne Architektur für Aufgaben der Bildverarbeitung, die sowohl zur Objektdetektion als auch zur Klassifikation eingesetzt werden kann. In dieser Arbeit wird die Klassifikationsvariante verwendet, bei der das Modell jeweils ein gesamtes Bild einer vordefinierten Klasse zuordnet. Diese Herangehensweise eignet sich besonders für rechenschwache Anwendungen wie die Miniaturautonomie, da auf die aufwändige Lokalisierung einzelner Objekte verzichtet wird.

2.3.1 YOLOv8-Architektur

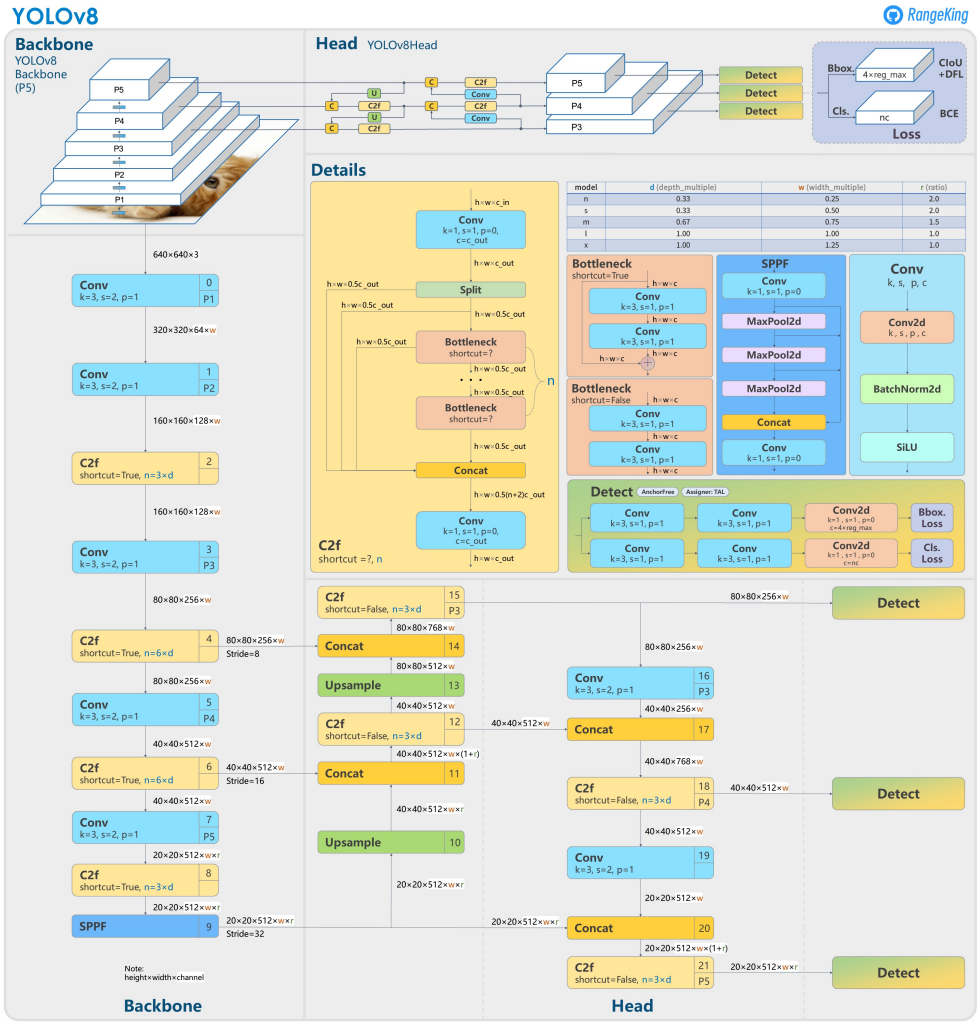
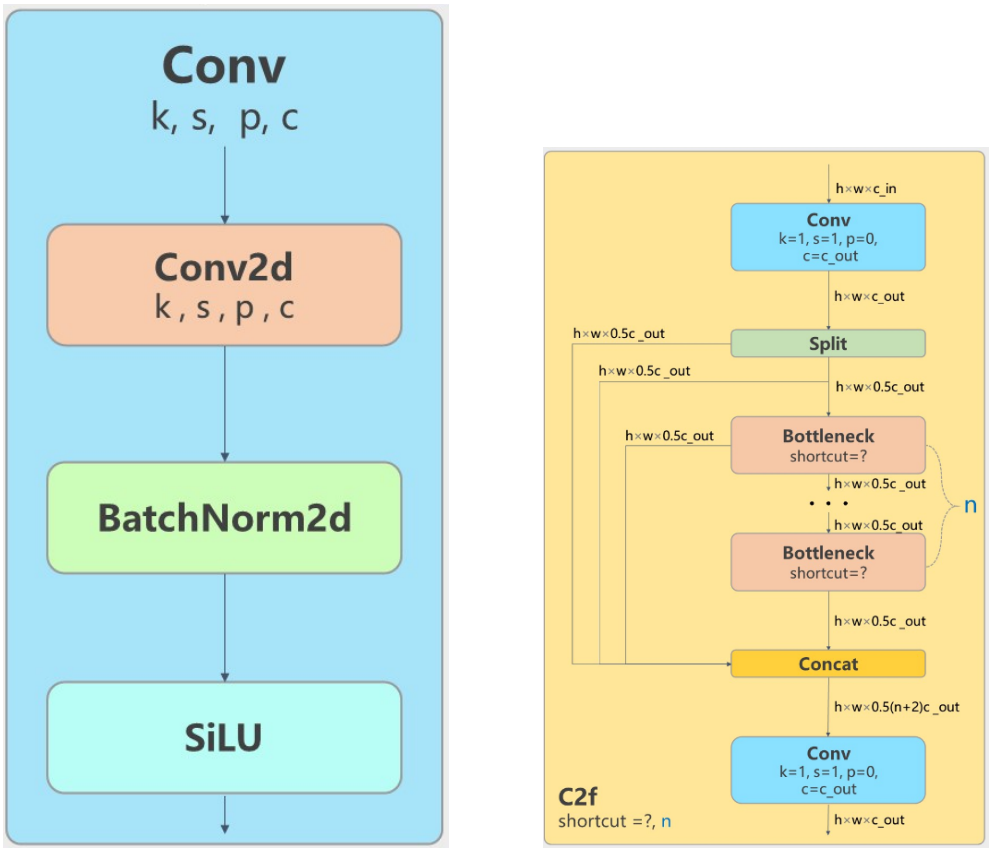


Abbildung 2.1: YOLOv8 Architektur
[14]

Die YOLOv8-Architektur besteht aus drei zentralen Komponenten: Backbone, Neck und Head. (siehe Abb. 2.1) Das Backbone dient als Grundlage des Modells und übernimmt die Aufgabe, wesentliche Merkmale aus dem Eingabebild zu extrahieren. Anschließend kombiniert der Neck die in den verschiedenen Schichten des Backbone gewonnenen Merkmale, um sie für die finale Verarbeitung vorzubereiten. Im letzten Schritt erfolgt die Ausgabe durch den Head, der sowohl die Klassen der Objekte als auch deren Bounding Boxes vorhersagt. Diese klar strukturierte Architektur ermöglicht eine effiziente und präzise Analyse von Bilddaten, wobei jede Komponente eine spezifische Rolle innerhalb des Modells erfüllt.



(a) (b)

Abbildung 2.2: Conv-Block und C2f-Block im Detail

Die interne Struktur des Backbones basiert auf typischen CNN-Bausteinen wie dem ConvBlock, dem Bottleneck und dem C2f-Block. Der ConvBlock besteht aus einer Kombination von Faltung, Batch-Normalisierung und Aktivierungsfunktion (hier SiLU). Er

ist dafür zuständig, grundlegende Merkmale wie Kanten oder Texturen zu erkennen. Die Bottleneck-Blöcke ermöglichen eine effiziente Merkmalsextraktion mit optionalen Shortcut-Verbindungen, die das Training beschleunigen und stabilisieren. Der C2f-Block kombiniert mehrere Feature-Maps und führt sie durch einen Concat-Vorgang zusammen, was eine stärkere Repräsentation wichtiger Merkmale ermöglicht (vgl. Abb.2.2).

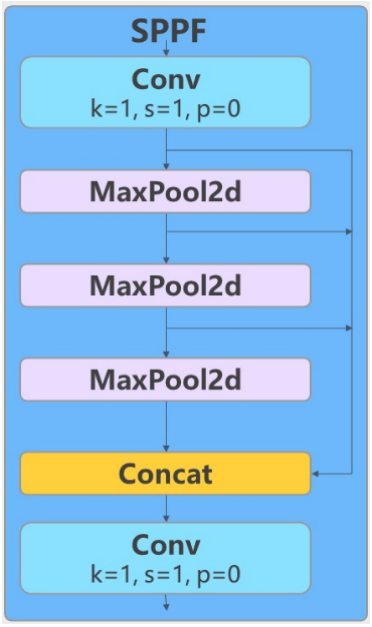


Abbildung 2.3: SPPF-Block im Detail

Ein weiterer zentraler Bestandteil der Architektur ist der SPPF-Block (Spatial Pyramid Pooling Fast, vgl. Abb. 2.3), der Feature-Maps auf unterschiedlichen Skalen analysiert und zusammenführt. Dieser Block ermöglicht die Extraktion relevanter Informationen unabhängig von der Größe der dargestellten Objekte. Für Aufgaben wie die Objektdetektion ist dies von hoher Bedeutung. In der hier verwendeten Klassifikationsvariante ist der SPPF-Block weiterhin Bestandteil des Backbones und trägt zur robusten Merkmalsextraktion bei.

Die Detektionskomponenten (Detect-Block, Bounding Box Prediction, etc.) entfallen in der Klassifikationsvariante vollständig. An ihrer Stelle wird ein einfacher Klassifikations-Head verwendet, der auf den durch den Backbone extrahierten Features basiert. Dadurch ergibt sich ein besonders kompaktes Modell, das sich gut für embedded Anwendungen eignet.

2.3.2 Modellanpassung durch Skalierungsparameter

Die Parameter *depth_ multiple* (d), *width_ multiple* (w) und *max_ channels* (mc) spielen eine zentrale Rolle bei der Anpassung der Modelle (vgl. Abb.2.4). Der Parameter *depth_ multiple* bestimmt die Anzahl der Bottleneck-Blöcke in den C2f-Blöcken und skaliert somit die Tiefe des Netzwerks. Ein Wert kleiner als 1 reduziert die Tiefe, was das Modell kompakter und schneller macht, jedoch potenziell zulasten der Genauigkeit. Ein Wert größer als 1 erhöht die Tiefe und damit die Anzahl der Schichten, wodurch das Modell genauer, aber auch langsamer wird. Der Parameter *width_ multiple* skaliert die Anzahl der Kanäle in den Convolutional Layers. Ein Wert kleiner als 1 verringert die Anzahl der Kanäle, wodurch das Netzwerk schlanker und schneller wird, jedoch möglicherweise an Präzision verliert. Werte größer als 1 erhöhen die Kanalanzahl, was die Genauigkeit steigern kann, jedoch gleichzeitig die Rechenanforderungen erhöht. *Max_ channels* legt eine Obergrenze für die Anzahl der Kanäle im Netzwerk fest, um das Modell bei hohen Werten von *width_ multiple* vor einer übermäßigen Verbreiterung zu schützen. Dies hilft, die Modellgröße zu kontrollieren und Überanpassungen zu vermeiden.

Model Variant	d(depth_multiple)	w(width_multiple)	mc(max_channels)
n	0.33	0.25	1024
s	0.33	0.50	1024
m	0.67	0.75	768
l	1.00	1.00	512
xl	1.00	1.25	512

Abbildung 2.4: Übersicht YOLOv8-Modelle

Die Detektionskomponenten (Detect-Block, Bounding Box Prediction, etc.) entfallen in der Klassifikationsvariante vollständig. An ihrer Stelle wird ein einfacher Klassifikations-Head verwendet, der auf den durch den Backbone extrahierten Features basiert. Dadurch ergibt sich ein besonders kompaktes Modell, das sich gut für embedded Anwendungen eignet.

In dieser Arbeit wurde eine vortrainierte YOLOv8-Klassifikationsarchitektur verwendet und für den spezifischen Anwendungsfall der Lokalisierung in einer Miniaturumgebung feingetunt. Die Bildauflösung wurde auf 320x320 Pixel gesetzt, um eine möglichst schnelle Inferenzzeit zu erreichen.

Die Architektur wurde mithilfe der Parameter *depth_multiple = 0.33* und *width_multiple = 0.25* konfiguriert. Dadurch ergab sich ein schlankes Modell mit einer geringen Anzahl von Kanälen, das dennoch eine ausreichend hohe Genauigkeit für die Klassifikation der Bildabschnitte ermöglichte. Die Kombination aus Effizienz, Flexibilität und guter Generalisierungsfähigkeit macht YOLOv8 zu einer geeigneten Wahl für die in dieser Arbeit verfolgte Aufgabe der bildbasierten Positionsbestimmung.

2.4 Eigen-CAM

Um die Entscheidungsprozesse eines neuronalen Netzwerks besser verstehen zu können, wurde die Eigen-Class Activation Map(Eigen-CAM) entwickelt.[12] Dies ist eine Visualisierungstechnik, welche die Möglichkeit bietet, die regionsspezifische Relevanz innerhalb eines Bildes zu analysieren, die zur Klassenzuweisung geführt hat.

2.4.1 Theoretische Grundlagen der Eigen-CAM

Die Eigen-CAM-Methode nutzt die Aktivierungen der Feature-Maps, die in den letzten Faltungsschichten eines neuronalen Netzwerks erzeugt werden. Diese Feature-Maps stellen die extrahierten Merkmale des Eingabebildes dar, die für die Klassifikation oder die Objektidentifikation genutzt werden. Die Besonderheit der Eigen-CAM liegt darin, dass sie die Eigenvektoren der Kovarianzmatrix der Feature-Maps berechnet, um die für die Entscheidungsfindung relevanten Regionen im Bild zu visualisieren.

Im Gegensatz zu Methoden wie Grad-CAM, die auf der Berechnung von Gradienten basieren, verwendet Eigen-CAM keine Rückpropagierung von Fehlern, sondern setzt auf die Analyse der Hauptkomponenten der Aktivierungen. Dabei wird eine Kovarianzmatrix erstellt, die die Beziehungen zwischen den Aktivierungen der verschiedenen Filter in der letzten Convolutional Layer darstellt. Die Eigenvektoren dieser Matrix geben die Hauptmerkmale an, die für die Klassifikation eines Bildes ausschlaggebend sind. Diese Eigenvektoren repräsentieren die relevanten Merkmale und Regionen, die das Netzwerk für seine Entscheidung verwendet hat.

Durch diese Vorgehensweise wird eine Heatmap erzeugt, die über das Originalbild gelegt wird. Diese Heatmap hebt die Bildbereiche hervor, die am wichtigsten für die Klassifikation des Netzwerks sind. Die Methode ist dabei besonders robust, da keine Gradientenin-

formationen benötigt werden, was sie weniger anfällig gegenüber adversarialen Eingaben oder Fehlern macht.

2.4.2 Funktionsweise der Eigen-CAM

Die Funktionsweise der Eigen-CAM basiert auf der Analyse der Aktivierungen in den letzten Convolutional Layers eines neuronalen Netzwerks. Zunächst werden die Feature-Maps extrahiert, die die Merkmale des Eingabebildes repräsentieren. Diese Feature-Maps dienen als Grundlage für die weiteren Berechnungen.

Um die Beziehungen zwischen den verschiedenen Dimensionen der Feature-Maps zu modellieren, wird eine Kovarianzmatrix erstellt. Diese Matrix enthält Informationen darüber, wie stark die Aktivierungen der einzelnen Merkmale miteinander korrelieren. Anschließend werden die Eigenvektoren dieser Kovarianzmatrix berechnet. Die dominantesten Eigenvektoren, die die bedeutendsten Merkmalskomponenten repräsentieren, werden ausgewählt, um regionsspezifische Informationen zu extrahieren.

Basierend auf diesen Eigenvektoren wird eine gewichtete Summe der Feature-Maps erstellt, die anschließend in eine Heatmap umgewandelt wird. Diese Heatmap zeigt an, welche Regionen des Bildes für die Entscheidung des Netzwerks besonders relevant waren. Durch die Überlagerung dieser Heatmap mit dem Originalbild entsteht eine anschauliche Visualisierung, die die Entscheidungsgrundlagen des Netzwerks deutlich macht.

2.4.3 Vorteile und Anwendungsbereiche

Eigen-CAM bietet mehrere Vorteile gegenüber herkömmlichen Visualisierungsmethoden wie Grad-CAM. [12] Da keine Gradientenberechnung erforderlich ist, ist die Methode robuster gegenüber adversarialen Eingaben und liefert auch bei fehlerhaften Klassifikationen konsistente Ergebnisse. Zudem ist sie effizienter, da zusätzliche Berechnungen wie die Rückverfolgung von Gradienten entfallen. Ein weiterer Vorteil ist die Flexibilität der Methode, da sie nicht klassenabhängig ist und somit vielseitig einsetzbar.

Die Anwendungsbereiche der Eigen-CAM sind breit gefächert. Sie wird häufig zur Visualisierung der Entscheidungsprozesse neuronaler Netzwerke eingesetzt, insbesondere um zu verstehen, welche Bereiche eines Bildes für die Klassifikation ausschlaggebend waren. Darüber hinaus eignet sich die Methode für das Debugging, da sie unerwartete oder fehlerhafte Modellverhalten sichtbar machen kann.

3 Bildbasierte Datenerhebung und Labeling entlang definierter Streckenabschnitte

Alle Fotos wurden mit der Kamera eines iPhone 15 Pro's aufgezeichnet. Die Bilder wurden so aufgenommen, dass sie der Perspektive der Onboard-Kamera möglichst entsprechen. Daher wird im weiteren Verlauf der Arbeit auch von dem Auto gesprochen. Die Hauptkamera verfügt über 48 Megapixel und eine variable Blende von f/1.78. Um einen Arbeitsschritt in der Datenverarbeitung zu sparen, sind die Fotos bereits in einem quadratischen Format aufgezeichnet worden. Die Dimension der Bilder wurde nicht mehr verkleinert, da YOLO beim Einlesen der Bilder durch einen Parameter die Größe bestimmen lässt. Diese wurde auf 320x320 gesetzt.

3.1 Analyse der Trainingsdaten

Da eine Aufzeichnung der kompletten Teststrecke inkl. Kennzeichnung der Position für diese Arbeit einen zu großen Arbeitsumfang bedeuten würde, sind Teile der Strecke in fünf sogenannte Sektionen unterteilt.

Abbildung 3.1: Teststrecke von oben

Diese Sektionen sind zwölf bis 16 cm lange Abschnitte, in denen auf der Fahrbahn ca. 2 cm lange Striche quer zur Fahrbahn eingezeichnet worden sind. Diese Markierungen sind für die Kamera jedoch nicht identifizierbar und sorgen somit nicht für verfälschte Ergebnisse. Die jeweiligen Sektionen werden durch die Pfeile in Abb. 3.1 dargestellt. Die Pfeilrichtung zeigt die simulierte Fahrtrichtung des Fahrzeugs. Die Sektionen repräsentieren verschiedene Merkmale der Teststrecke und gewährleisten so eine umfassende Abbildung der Straßenumgebung. Zu den erfassten Unterschieden zählen Variationen in der Straßenoberfläche (heller oder dunkler Belag), Fahrbahnmarkierungen (vorhanden oder nicht vorhanden), Umgebungsmerkmale wie Bäume, Häuser und Kirchen sowie infrastrukturelle Elemente wie Straßenlaternen und Kreuzungen. Diese Vielfalt deckt das

Spektrum möglicher Straßenumgebungen realistisch ab und ermöglicht eine aussagekräftige Bewertung des Modells.

(a) Sektion 1

(b) Sektion 2

(c) Sektion 3

Abbildung 3.2: Jeweils ein Bild aus dem Trainingsdatensatz für Sektionen 1-3

Die ursprüngliche Auflösung der Bilder in Abb. 3.2 beträgt 1772x1772 Pixel, entsprechend der vom iPhone auf den PC übertragenen Dateien. Im Training kommt eine reduzierte Auflösung von 320x320 Pixeln zum Einsatz, um die Inferenzzeit zu minimieren.

Sektion eins beschreibt einen dunklen Fahrbahnabschnitt, auf dem sich das Fahrzeug einer Kreuzung ohne Fahrbahnmarkierungen nähert. Die Umgebung weist einen ländlichen Charakter auf, geprägt von einer Kirche, die von zahlreichen Bäumen umgeben ist. Straßenlaternen oder weitere Bauwerke sind nicht vorhanden.

Sektion zwei umfasst ebenfalls einen Fahrbahnabschnitt auf dunklem Untergrund, zeichnet sich jedoch durch klar erkennbare Fahrbahnmarkierungen aus. Charakteristische infrastrukturelle Merkmale sind ein Mehrfamilienhaus, mehrere Einfamilienhäuser sowie eine Hecke. Zwei Tannen und eine Straßenlaterne ergänzen die Umgebungsmerkmale.

Sektion drei bildet erneut eine ländliche Umgebung ab und umfasst einen kurvigen Fahrbahnabschnitt auf dunklem Untergrund. Das Fahrzeug bewegt sich dabei auf einen Bereich mit hellem Fahrbahnbelag und vorhandenen Fahrbahnmarkierungen zu. Verglichen mit Sektion eins befinden sich hier weniger Bäume am Fahrbahnrand. Auf der rechten Straßenseite steht ein Haus, das teilweise von einer Tanne verdeckt ist. Gegenüber liegt ein Brunnen vor einem Hügel.

(a) Sektion 4 (b) Sektion 5

Abbildung 3.3: Jeweils ein Bild aus dem Trainingsdatensatz für Sektionen 4-5

In Abbildung 3.3 sind exemplarisch je ein Trainingsbild der Sektionen vier und fünf dargestellt. Sektion vier bildet einen weiteren kurvigen Fahrbahnabschnitt mit vorhandenen Fahrbahnmarkierungen ab (vgl. Abb. 3.3a). Rechts neben der Fahrbahn steht ein Haus, zudem verteilen sich drei Bäume entlang der Strecke. Zusätzlich befinden sich hier Straßenlaternen einer zweiten Bauart, die sich optisch deutlich von den zuvor gezeigten Laternen unterscheiden.

Sektion fünf grenzt unmittelbar an Sektion zwei an und liegt in Fahrtrichtung gesehen davor (vgl. Abb. 3.3b). Aufgrund der räumlichen Nähe ähneln sich die Merkmale beider Sektionen stark. Sektion fünf zeigt jedoch zusätzlich Teile einer Kreuzung auf der rechten Fahrbahnseite, einschließlich der zugehörigen Fahrbahnmarkierungen.

Wie bereits beschrieben, ist jede Sektion in kleinere Teilabschnitte von je einem Zentimeter Länge unterteilt. Zur Sicherstellung reproduzierbarer Bedingungen und einer genauen Modellbewertung dienen kleine Markierungen auf der Fahrbahn als Referenzpunkte. Diese Markierungen ermöglichen die Bildaufnahme stets von derselben Position aus. Die Aufnahme der Bilder erfolgt jeweils zu Beginn jedes Teilabschnitts.

(a) Hell (b) Auto (c) Dunkel

Abbildung 3.4: Übersicht der Datensätze einer Sektion (hier Sektion zwei)

Die Datensätze sind nach Sektionen gegliedert, wobei jede Sektion für Testzwecke in drei Kategorien unterteilt ist: Hell, Auto und Dunkel (vergl. Abb. 3.4). Um die Robustheit des Modells zu erhöhen, wurden Bilder sowohl unter guten als auch schlechten Lichtverhältnissen aufgenommen sowie mit und ohne Fahrzeuge auf der Fahrbahn. Einige Datensätze enthalten daher Aufnahmen bei schlechten Lichtverhältnissen inklusive Autos, andere dagegen nicht. Diese Variation sorgt für eine größere Diversität im Training.

Zur leichteren Auswertung der Testergebnisse folgt die Benennung der Trainings- und Validierungsbilder einem festgelegten Schema. Der Bildname beginnt mit einem „S“, gefolgt von der jeweiligen Sektion. Anschließend kennzeichnet eine Ziffer (0, 1 oder 2) nach einem Unterstrich die jeweilige Kategorie: 0 steht für Hell, 1 für Auto und 2 für Dunkel. Nach einem weiteren Unterstrich folgt der Index des Bildes, wobei die Bilder von Anfang bis Ende jedes Abschnitts fortlaufend nummeriert sind. Die Nummerierung erfolgt dabei stets beginnend in Fahrtrichtung. Beispielsweise bezeichnet `S2_2_10.JPEG` das Bild in Abbildung 3.4c, welches zur Sektion zwei gehört, bei zehn Zentimetern aufgenommen wurde und der Kategorie „Dunkel“ entspricht.

Die Testbilder folgen einem leicht abweichenden Schema: Auch hier beginnt der Name mit „S“, gefolgt von der Sektion und dem Index des Bildes nach einem Unterstrich. Ein Testbild der zweiten Sektion, aufgenommen bei Zentimeter fünf, heißt demzufolge `S2_5.JPEG`.

3.2 Vorbereitung der Trainingsdaten

Da drei Bilder pro Teilabschnitt für das Trainieren eines Convolutional Neural Networks(CNN) sehr wenig sind, wurden die Bilder jeweils vierfach augmentiert.

Es wurden hierbei verschiedene Bildaugmentationseffekte angewendet, von denen für jedes Bild zufällig zwei bis vier Effekte ausgewählt und kombiniert wurden. Die Helligkeit der Bilder wurde zufällig um bis zu ±20 angepasst, während die Farbsättigung und der Kontrast innerhalb der Bereiche 0,70 bis 1,30 bzw. 0,8 bis 1,2 skaliert wurden. Zusätzlich wurde ein geringer Graustufenanteil (bis zu 15%) hinzugefügt, um Szenarien mit weniger Farbintensität zu simulieren. Zufälliges Gaußsches Rauschen mit einer Intensität von bis zu 1% und Verschiebungen im Farbton um ±10 wurden eingesetzt, um das Modell robuster gegen verrauschte und farblich variierende Eingaben zu machen. Zur Erhöhung der Schärfevielfalt wurde die Schärfe zwischen 0% und 100% mit leicht variierender Lichtstärke (0,8 bis 1,2) angepasst. Diese Augmentationen wurden gezielt kombiniert, um die Trainingsdaten zu diversifizieren und das Risiko von Overfitting zu minimieren.

Für optimale Trainingsergebnisse wurden die augmentierten Bilder dem Trainingsdatensatz und die Originalbilder dem Validierungsdatensatz zugewiesen. Daraus ergibt sich ein Verhältnis von 80% Trainings- zu 20% Validierungsbildern. Separate Testbilder wurden zusätzlich bei guten Lichtverhältnissen aufgenommen.

Sektion	**Training**	**Validierung**	**Test**
Eins	192	48	16
Zwei	192	48	16
Drei	144	36	12
Vier	192	48	16
Fünf	144	36	12
Gesamt	**864**	**216**	**72**

Tabelle 3.1: Anzahl der Bilder pro Sektion in Training, Validierung und Test

Wie in Tabelle 3.1 zu erkennen ist, verfügt der gesamte Datensatz über 1152 Fotos. Dieser setzt sich zusammen aus jeweils 192 Fotos in Sektion eins, zwei und vier und 144 Fotos in Sektion drei und fünf im Trainingsdatensatz. Sektion eins, zwei und vier verfügen jeweils über 48 Fotos im Validierungsdatensatz und 16 für Testzwecke. Im Gegensatz dazu verfügen Sektion drei und fünf über jeweils 36 Fotos im Validierungs- und 12 im Testdatensatz.

4 Modelltraining und Evaluation

Das Training konzentriert sich zunächst auf einzelne Sektionen der Strecke. Ziel ist es, die Genauigkeit der Positionsvorhersage des Fahrzeugs zentimetergenau zu bestimmen. Das Training beginnt in Sektion eins mit zwei Klassen, welche jeweils einen Teilabschnitt der Sektion repräsentieren. Da Sektion eins eine Gesamtlänge von 16 cm aufweist, umfasst der erste Teilabschnitt die Zentimeter eins bis einschließlich acht, der zweite Teilabschnitt entsprechend die Zentimeter neun bis 16.

4.1 Vergleich Sektion 1

Die ersten Trainingsdurchläufe erfolgten ausschließlich mit Bildern des Datensatzes Hell, wobei jeweils 20, 50 und 100 Epochen trainiert wurden. Anschließend erfolgte zunächst die Erweiterung des Trainingsdatensatzes um die Kategorie Auto und danach zusätzlich um die Kategorie Dunkel.

Testfall	**Genauigkeit**
S1_8cm_20Epochen	81.25%
S1_8cm_50Epochen	100.00%
S1_8cm_100Epochen	100.00%
S1_8cm_20Epochen_Autos	93.75%
S1_8cm_50Epochen_Autos	100.00%
S1_8cm_100Epochen_Autos	100.00%
S1_8cm_20Epochen_Autos_Dunkel	100.00%
S1_8cm_50Epochen_Autos_Dunkel	100.00%
S1_8cm_100Epochen_Autos_Dunkel	100.00%

Tabelle 4.1: Vorhersagegenauigkeit pro Testfall in Sektion eins

Tabelle 4.1 zeigt, dass bereits mit einer geringen Anzahl an Bildern und wenigen Trainingsepochen überzeugende Ergebnisse erzielt wurden.

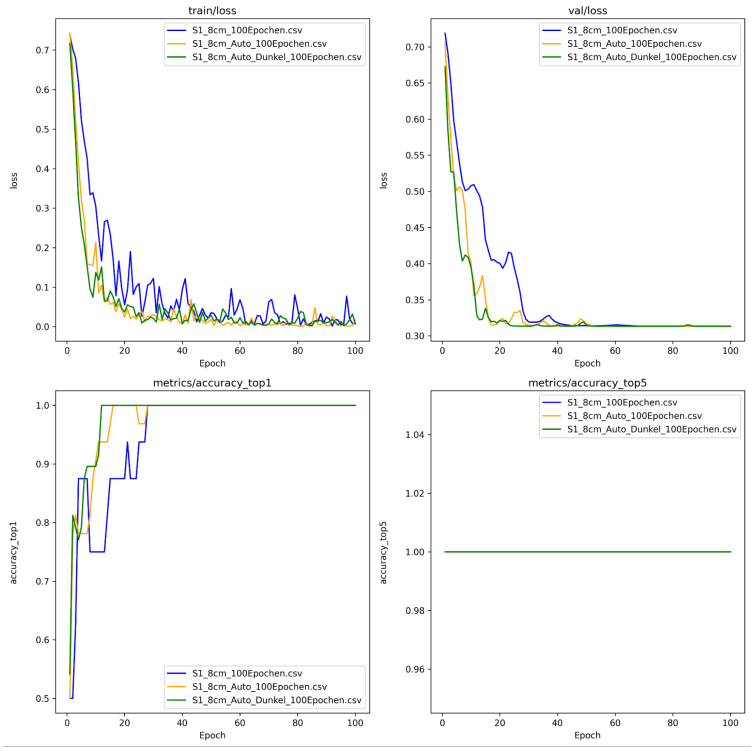


Abbildung 4.1: Auswertung Sektion eins, in zwei Teilabschnitte mit jeweils acht Zentimeter Länge geteilt

Die Evaluationsdaten der einzelnen Modelle in Abb. 4.1 zeigen, dass bereits nach wenigen Trainingsdurchläufen ein Übertraining einsetzt, was auf ein „Auswendiglernen“ der Trainingsdaten hinweist. In diesem spezifischen Fall ist ein solches Verhalten jedoch bis zu einem gewissen Grad erwünscht. Deutlich wird auch, dass die Erweiterung des Trainingsdatensatzes um ein Bild pro Zentimeter zu einer signifikanten Leistungssteigerung führt. Die Hinzunahme eines zweiten Bildes pro Zentimeter verbessert die Ergebnisse weiter, jedoch fällt der Zugewinn im Vergleich zum ersten Schritt deutlich geringer aus. Aus diesem Grund wurde im weiteren Verlauf auf eine Erweiterung über drei Bilder pro Zentimeter und Sektion hinaus verzichtet. Zwar könnte dies die Robustheit des Modells potenziell weiter erhöhen, im Rahmen dieser Arbeit ist jedoch keine signifikante Verbesserung mehr zu erwarten.

Da das Modell bereits bei wenigen Trainingsdurchläufen eine Genauigkeit von 100% erreicht, wurde Sektion eins anschließend in kleinere Teilabschnitte unterteilt. Basierend auf den bisherigen Erkenntnissen erfolgte das weitere Training nicht mehr mit getrennten

Datensätzen, sondern mit dem vollständigen Datensatz, bestehend aus den Kategorien Hell, Auto und Dunkel.

Testfall	**Genauigkeit**
S1_4cm_20Epochen_Autos_Dunkel	56.25%
S1_4cm_50Epochen_Autos_Dunkel	93.75%
S1_4cm_100Epochen_Autos_Dunkel	93.75%
S1_2cm_20Epochen_Autos_Dunkel	43.75%
S1_2cm_50Epochen_Autos_Dunkel	93.75%
S1_2cm_100Epochen_Autos_Dunkel	93.75%

Tabelle 4.2: Genauigkeit pro Testfall unter Dunkelheitsbedingungen

Tab. 4.2 verdeutlicht, dass die Genauigkeit bei 20 Epochen im Vergleich zum vorherigen Testfall zunächst deutlich abfällt. Bereits nach 50 Epochen zeigt das Modell jedoch eine starke Erholung und liefert in nur einem Testfall eine falsche Vorhersage.

Modell	**Testbild**	**Tatsache**	**Vorhersage**
S1_4cm_50pochen_Autos_Dunkel	S1_12	9b12	13b16
S1_4cm_100pochen_Autos_Dunkel	S1_12	9b12	13b16
S1_2cm_50pochen_Autos_Dunkel	S1_12	9b12	13b14
S1_2cm_100pochen_Autos_Dunkel	S1_12	9b12	13b14

Tabelle 4.3: Vergleich der Vorhersagen für verschiedene Modelle unter Dunkelheitsbedingungen

Dem Vergleich der verschiedenen Modelle in Tab. 4.3 ist zu entnehmen, dass die einzige falsche Vorhersage immer das gleiche Bild betrifft. Dies könnte ein Indiz für eine Ungenauigkeit bei der Aufzeichnung des Testfotos sein.

4.2 Vergleich aller Sektionen

Testfall	Genauigkeit
S1_2cm_100Epochen	93.75%
S2_2cm_100Epochen	93.75%
S3_2cm_100Epochen	66.67%
S4_2cm_100Epochen	81.25%
S5_2cm_100Epochen	83.33%

Tabelle 4.4: Die Vorhersagegenauigkeit

Der Vergleich aller Sektionen (siehe Tab. 4.4) mit Teilabschnitten von jeweils zwei Zentimetern Länge zeigt, dass die Genauigkeit in geradlinigen Streckenabschnitten höher ausfällt. Auffällig ist der deutliche Unterschied in der Modellgenauigkeit zwischen Sektion zwei und Sektion fünf, obwohl beide Sektionen vergleichbare Abschnitte abbilden und sich in ihren Straßenumgebungsmerkmalen nur geringfügig unterscheiden.

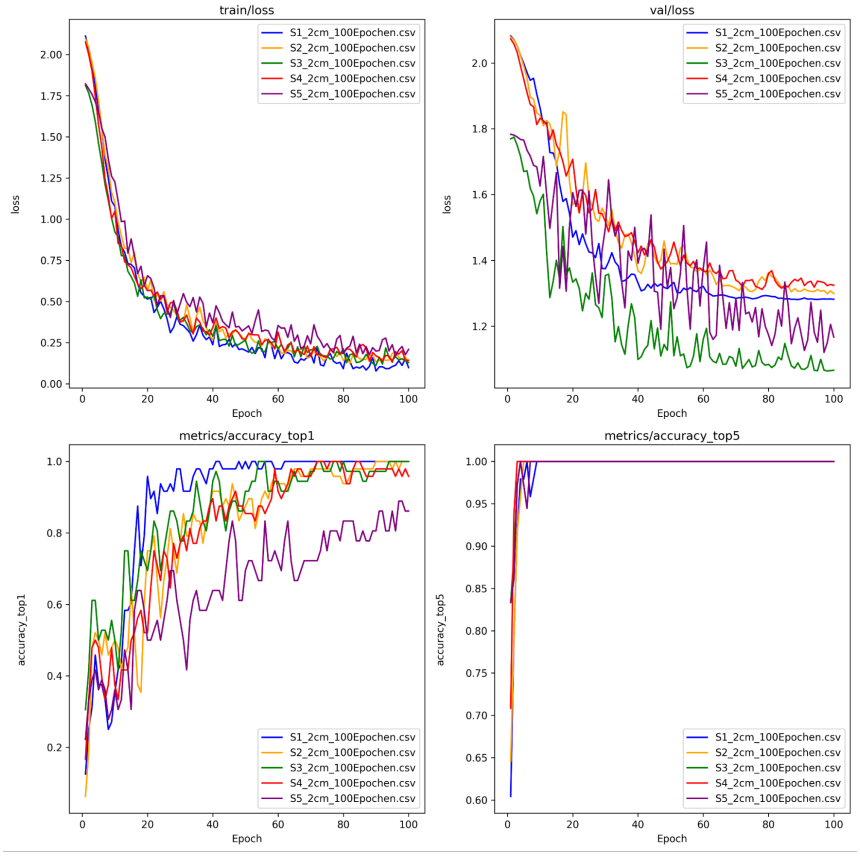


Abbildung 4.2: Vorhersagegenauigkeit über alle Sektionen hinweg

In Abb. 4.2 ist zu erkennen, dass insbesondere die Top-1-Genauigkeit des Modells aus Sektion fünf deutlich abfällt. Selbst nach 100 Trainingsepochen erreicht dieses Modell keine höhere Treffergenauigkeit als 85%, während alle anderen Modelle Werte nahe 100% erzielen. Auch die Entwicklung des Validierungsverlusts bestätigt diese Auffälligkeit: Das Modell aus Sektion fünf zeigt über den gesamten Trainingsverlauf hinweg starke Schwankungen. Ähnliche Schwankungen treten auch bei Sektion drei auf, allerdings ist der Validierungsverlust hier insgesamt deutlich geringer als in allen anderen Sektionen.

Die Gegenüberstellung von Trainings- und Validierungsverlust liefert zudem Hinweise auf Overfitting. Während sich der Trainingsverlust bei etwa 0,25 stabilisiert, liegt der Validierungsverlust der Modelle im Bereich zwischen 1,1 und 1,4.

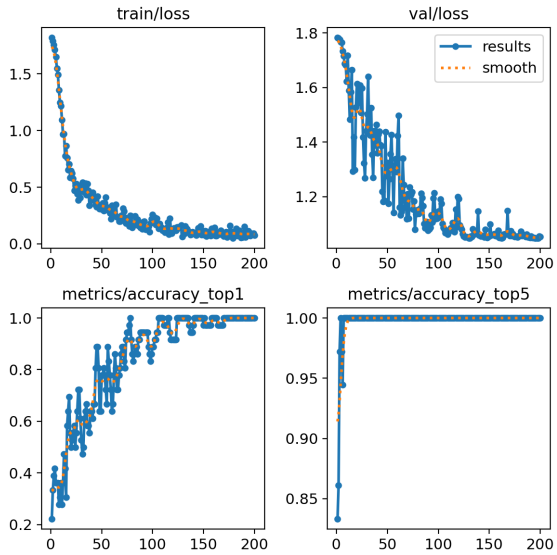


Abbildung 4.3: Auswertung Sektion 5 200 Epochen

In der Auswertung des Modells von Sektion fünf über eine Dauer von 200 Trainingsepisoden (siehe Abb. 4.3) wird deutlich, dass eine Top-1-Genauigkeit von 100% erst knapp nach 100 Episoden erreicht wird. Vergleicht man nun auch die Genauigkeit der Vorhersagen des Modells mit 100 Trainingsepisoden mit dem, welches 200 zu verzeichnen hat, so ist mit einer Genauigkeit von rund 83% kein Unterschied zu erkennen.

Bildname	**Tatsache**	**Vorhersage**
S1_12	S1_CM_11_12	S1_CM_13_14
S2_3	S2_CM_3_4	S2_CM_1_2
S3_5	S3_CM_5_6	S3_CM_3_4
S3_7	S3_CM_7_8	S3_CM_5_6
S4_11	S4_CM_11_12	S4_CM_9_10
S4_15	S4_CM_15_16	S4_CM_13_14
S4_3	S4_CM_3_4	S4_CM_1_2
S4_9	S4_CM_9_10	S4_CM_7_8
S5_11	S5_CM_11_12	S5_CM_9_10
S5_3	S5_CM_3_4	S5_CM_1_2
S5_5	S5_CM_5_6	S5_CM_3_4

Tabelle 4.5: Übersicht der falschen Vorhersagen, sektionsübergreifend

Die Zusammenfassung aller Testfälle mit fehlerhaften Vorhersagen (vgl. Tab. 4.5) zeigt, dass die vorhergesagten Teilabschnitte jeweils direkt an den tatsächlichen Abschnitt angrenzen. Besonders auffällig ist, dass die falschen Zuordnungen stets am Übergang zwischen zwei benachbarten Teilabschnitten auftreten. Ein exemplarischer Fall ist das Bild `S1_12`, das fälschlicherweise der Klasse `S1_CM_13_14` zugeordnet wurde. Daraus lässt sich ableiten, dass die Positionsgenauigkeit des Modells im Bereich von bis zu 2 cm liegt.

4.3 Sektionen-übergreifendes Modelltraining

Im weiteren Verlauf der Arbeit wurden die Datensätze aller Sektionen zusammengeführt und erneut eine Vielzahl von Testfällen durchgeführt. Ziel war es unter anderem zu prüfen, ob das Modell durch Fahrzeuge, die während des Trainings in einer bestimmten Sektion vorhanden waren und in Testbildern anderer Sektionen auftreten, in seiner Vorhersagegenauigkeit beeinträchtigt wird.

Abbildung 4.4: Darstellung der in den Testfällen verwendeten Fahrzeuge

So wurde beispielsweise der DHL-Lastwagen (vgl. Abb. 4.4), der im Training der ersten Sektion verwendet wurde, gezielt in Bildern der zweiten Sektion platziert. Umgekehrt erscheinen der weiße Transporter und ein kleines schwarzes Auto aus Sektion zwei (vgl. Abb. 4.4) auf Testbildern der ersten Sektion. Der daraus entstandene zweite Testdatensatz umfasst insgesamt 28 Bilder, die in verschiedenen Teilabschnitten aufgenommen wurden und unterschiedliche Distanzen zu den Fahrzeugen abbilden.

Für weiterführende Tests wurden zudem alle ursprünglich für die jeweiligen Sektionen vorgesehenen Testbilder zusammengeführt. Das Modell wurde anschließend auf Basis dieses kombinierten Testdatensatzes hinsichtlich seiner Gesamtgenauigkeit evaluiert.

Testfall	**Genauigkeit**
AlleSektionen	84.72%
AlleSektionen_andereAutos	57.14%

Tabelle 4.6: Evaluierung der Modellgenauigkeit bei sektionsübergreifendem Trainingsdatensatz

Ähnlich wie bei den Vorhersagegenauigkeiten der einzelnen Sektionen erreicht auch das Modell, das auf Bildern aller Sektionen trainiert wurde, eine Gesamtgenauigkeit von etwa 85% (vgl. Tab.4.6). Der Durchschnitt der Genauigkeiten der einzelnen Sektionen liegt bei 83,75% und unterscheidet sich damit nur geringfügig. Deutlich schlechter fällt hingegen die Modellleistung bei Testbildern aus, die Fahrzeuge enthalten, die zuvor nur in anderen Sektionen verwendet wurden.

4.4 Evaluierung der Lokalisierungsgenauigkeit mit YOLOv8

Die Analyse der Vorhersagewahrscheinlichkeiten für Testbild `S1_12` (vgl. Abb. 4.5a) zeigt eine Verteilung von 52 % für Klasse 13b14 und 48 % für Klasse 11b12. Das Modell war damit bei der Klassifizierung nicht eindeutig.

(a) S1_12 (b) S1_0_14 (c) S1_1_14

Abbildung 4.5: Vergleich Testbild zu Trainingsbildern Sektion eins

Der Vergleich des Testbildes mit zwei der drei zugehörigen Trainingsbilder (vgl. Abb. 4.5b und Abb. 4.5c) deutet auf eine mögliche Fehlerquelle hin. Während anhand der grauen Fläche vor der Kirche eine größere Distanz erkennbar ist, wurde das Foto offenbar aus einem leicht abweichenden Winkel aufgenommen. Auffällig ist, dass das Kirchendach auf allen Bildern nahezu an derselben Stelle abgeschnitten wird, was eine präzise Klassifizierung erschwert.

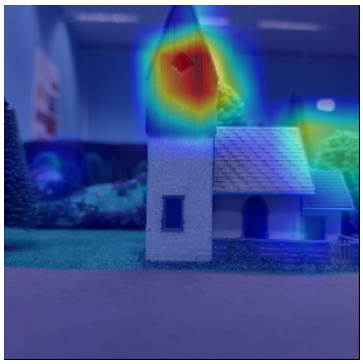


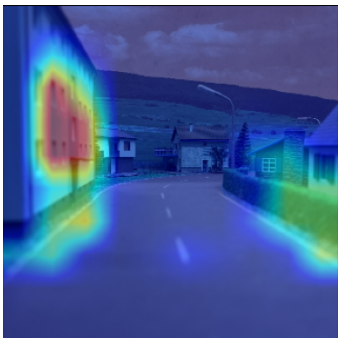
Abbildung 4.6: Eigen-CAM S1_12

Zieht man nun die Eigen-CAM hinzu, lässt sich die Vermutung bestätigen. In Abb. 4.6 wird deutlich, dass das Modell hauptsächlich die obere Hälfte des Kirchturms nutzt, um das Bild zu klassifizieren.

(a) S2_0_1

(b) S2_0_2

(c) S2_0_3



(d) Eigen-CAM S2_3

(e) S2_3

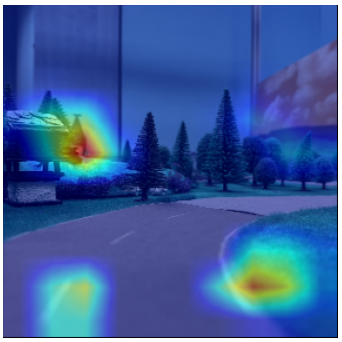
Abbildung 4.7: Vergleich der Traingsbilder mit Testbild inkl. Eigen-CAM in Sektion zwei

Auch bei Testbild `S2_3` (vgl. Abb. 4.7e) trifft das Modell eine falsche Vorhersage. Im Gegensatz zum Fall bei Bild `S1_12` ist die Wahrscheinlichkeit für Klasse `S1_CM_1_-2` hier bei 100 %. Durch eine Analyse mit Hilfe der Eigen-CAM (vgl. Abb. 4.7d) wird deutlich, dass die Hauswand als hauptsächliche Quelle der Entscheidungsfindung genutzt wird. Betrachtet man die Trainingsbilder für Teilabschnitt eins, zwei und drei (vgl. Abb. 4.7a, Abb. 4.7b und Abb. 4.7c), wird deutlich, dass die Unschärfe der Hauswand und der Winkel sich sehr stark dem des Testbildes `S2_3` ähneln. Das Trainingsbild des dritten Teilabschnitts ist in diesem Bereich scharf und weist zudem einen leicht veränderten Blickwinkel auf. Es ist zu schlussfolgern, dass auch hier eine Ungenauigkeit bei der Aufzeichnung des Fotos die ausschlaggebende Ursache für die falsche Vorhersage darstellt.

(a) S3_0_5

(b) S3_0_6

(c) S3_0_7



(d) Eigen-CAM S3_7

(e) S3_7

Abbildung 4.8: Vergleich der Traingsbilder mit Testbild inkl. Eigen-CAM in Sektion drei

Bei einer Analyse der fehlerhaften Vorhersagen in Sektion drei und der dazugehörigen Bilder lässt sich keine offensichtliche Fehlerquelle erkennen. So haben die markierten Bereiche der Eigen-CAM (vgl. Abb. 4.8d) auch keine auffällig großen Unterschiede. In diesem Fall könnte die geringe Anzahl an Trainings- und Validierungsdaten und der kurvige Bereich mit wenigen markanten Objekten für eine eindeutige Identifikation ein Grund für diese falsche Vorhersage sein.

(a) Eigen-CAM S4_3 (b) Eigen-CAM S4_11 (c) Eigen-CAM S4_15

Abbildung 4.9: Vergleich Eigen-CAM Sektion vier

Interessant ist auch zu sehen, dass das Modell in der gleichen Sektion verschiedene Regionen nutzt, um eine Klassifizierung vorzunehmen (vgl. Abb.4.9). Anhand der Regionen ist aber auch zu erkennen, dass eine nicht konsistente Aufzeichnung der Fotos eine korrekte Klassifizierung erschwert. Dies ist voraussichtlich mit einem größeren Datensatz mit mehr Varianz bezüglich des Blickwinkels ein deutlich geringeres Problem. Im Falle der vierten Sektion ist bei einem Vergleich zwischen Trainings- und Testbildern kein augenscheinlicher Grund für die falsche Klassifizierung herausgestochen. Bedeutend war ausschließlich, dass der Blickwinkel stark schwankte und somit die Merkmale schwerer zu erlernen waren. Dies könnte zum einen an der Kurve liegen, zum anderen musste bei der Aufzeichnung einiger Fotos die Laterne auf der rechten Straßenseite entfernt werden. Aufgrund dessen musste das iPhone kurzzeitig weggelegt werden.

(a) Eigen-CAM S5_3 (b) S5_0_3 (c) S5_1_3

Abbildung 4.10: Vergleich Eigen-CAM-Testbild mit zwei ähnlichen Fotos

In Sektion fünf ist die relevante Region ein Teil des Mittelstreifens. Da alle Bilder in diesem Abschnitt sich ähneln, kann dies ein Grund für die falsche Vorhersage sein. Betrachtet man die Trainingsbilder genauer, so ist die Distanz zum ersten Streifen auch leicht verändert. Dies liegt erneut an dem Winkel der Kamerapositionierung. Eine korrekte Klassifizierung mit wenigen Trainingsbildern ist somit erschwert.

Abschließend lässt sich festhalten, dass die geringe Anzahl an Trainingsbildern und die damit verbundene geringe Varianz der unterschiedlichen Winkel der entscheidene Faktor für eine falsche Klassifizierung ist. Es ist dennoch erstaunlich, wie genau das Modell bereits mit drei Bildern pro CM die Position bestimmen kann.

5 Fazit

Die vorliegende Arbeit befasste sich mit der Untersuchung der Möglichkeiten zur kamerabasierten Lokalisierung in einer Miniaturumgebung unter Verwendung von YOLOv8-Klassifikationsmodellen. Ziel war es, die Position eines Systems mit einer Genauigkeit von wenigen Zentimetern zu bestimmen, wobei Straßenumgebungsmerkmale als Grundlage für die Lokalisierung dienten. Zu diesem Zweck wurden verschiedene Experimente durchgeführt, um die Präzision und Robustheit des Modells unter unterschiedlichen Bedingungen zu bewerten.

Hierfür wurden über 1000 Bilder im Umfeld des Mikrowunderlands der HAW mit einem iPhone 15 Pro aufgenommen und anschließend mithilfe von Python-Skripten annotiert. Ein strukturiertes Schema wurde entwickelt, um die Sektionen und Datensätze voneinander zu unterscheiden. Diese klare Struktur bildete die Basis für ein systematisches Training und die anschließende Auswertung.

Das YOLOv8-Modell wurde auf einer Vielzahl von Datensätzen trainiert, um die Vorhersagegenauigkeit in unterschiedlichen Szenarien zu evaluieren. Der Erfolg des Modells wurde anhand der Genauigkeit auf Testbildern gemessen und die Statistiken der Trainingsläufe analysiert. Dabei zeigte sich, dass mit steigender Anzahl von Trainingsbildern und Epochen die Modellleistung zunahm, jedoch auch erste Anzeichen für leichtes Übertraining erkennbar wurden.

Die ersten Tests in Sektion eins zeigten, dass bereits mit wenigen Trainingsbildern und einer geringen Anzahl an Epochen gute Vorhersagen erzielt werden konnten. Es wurde jedoch deutlich, dass mindestens zwei Bilder pro Teilabschnitt erforderlich sind, um verlässliche Ergebnisse zu gewährleisten. Bei Sektionen in Kurvenbereichen sank die Vorhersagegenauigkeit leicht ab. Eine mögliche Ursache hierfür könnte in der Qualität der Trainingsdaten liegen, da in diesen Sektionen größere Variationen innerhalb der Bildmerkmale festgestellt wurden. Insbesondere die Schwankungen des Kamerawinkels bei

der Bildaufnahme in Kurven scheinen die Modellleistung zu beeinflussen, während auf geraden Strecken konsistentere Ergebnisse erzielt wurden.

Insgesamt konnte eine Lokalisierungsgenauigkeit von etwa zwei Zentimetern erreicht werden. Die Analyse der Fehlklassifikationen zeigte, dass diese meist an den Übergängen zwischen benachbarten Teilabschnitten auftraten. Das deutet darauf hin, dass das Modell in der Lage ist, die ungefähre Position des Fahrzeugs zuverlässig einzugrenzen, jedoch bei minimalen Abweichungen in Perspektive oder Beleuchtung zu Unsicherheiten neigt. Besonders die Ergebnisse bei Testbildern mit neuen Fahrzeugen oder leicht veränderten Kamerawinkeln offenbaren die begrenzte Generalisierungsfähigkeit des Modells. Hier traten teils deutliche Genauigkeitseinbußen auf, was zeigt, dass das Modell stark auf eine konstante Bildaufnahmesituation angewiesen ist.

Die Ergebnisse verdeutlichen, dass die Qualität und Konsistenz der Trainingsdaten einen entscheidenden Einfluss auf die Modellleistung haben. Insbesondere varianzarme Daten mit gleichbleibender Kameraposition führen zu stabileren Vorhersagen. Für eine höhere Robustheit des Modells wäre eine gezielte Erweiterung des Datensatzes um perspektivisch und beleuchtungsmäßig variierende Aufnahmen sinnvoll.

Abschließend lässt sich festhalten, dass das entwickelte System unter kontrollierten Bedingungen eine zuverlässige Lokalisierung im Zentimeterbereich ermöglicht. Für den praktischen Einsatz in komplexeren oder dynamischeren Szenarien sind jedoch Erweiterungen und Verbesserungen erforderlich, insbesondere im Hinblick auf Generalisierungsfähigkeit und Robustheit gegenüber Störungen. Die Arbeit legt damit eine fundierte Basis für weitere Forschung und Entwicklung im Bereich bildbasierter Lokalisierung auf Miniaturmaßstab.

5.1 Ausblick

Diese Arbeit hat gezeigt, dass die kamerabasierte Lokalisierung mit einem Klassifikationsmodell eine valide Option ist. Das Training benötigt wenig Aufwand und war mit einer geringen Anzahl von Trainingsbildern bereits erfolgreich. In der Zukunft könnte man den Ansatz auf andere Klassifikationsnetze wie beispielsweise ResNet erweitern und diese vergleichen.

Es wurde auch deutlich, dass die größte Fehlerquelle die Aufzeichnung der Fotos war und bereits kleine Abweichungen des Neigungswinkels zu einer falschen Vorhersage führen

können. Aus diesem Grund ist eine Erweiterung des Datensatzes wichtig, um die Robustheit und Vorhersagegenauigkeit des Modells weiter zu fördern. Hierbei kann auch eine Möglichkeit geschaffen werden, bei der Aufzeichnung der Fotos genauer zu arbeiten. Da die Anlage über eine Overhead-Kamera verfügt, könnte man das Auto tracken und Fotos über die Onboard-Kamera aufzeichnen und gleichzeitig mit der Position versehen.

Eine weitere Herausforderung könnte die Integration des Modells für Echtzeitverarbeitung sein. Auch hier bestehen verschiedene Möglichkeiten. Es kann untersucht werden, ob eine Verarbeitung und Auswertung direkt auf dem Auto oder auf einem externen Rechner möglich ist. Daraufhin könnte eine Kommunikation zwischen mehreren Autos realisiert werden, um anschließend den Verkehr zu steuern.

Die Ergebnisse dieser Arbeit stellen somit eine solide Grundlage für zukünftige Entwicklungen. Mit gezielten Erweiterungen des Datensatzes, präziseren Aufzeichnungsmethoden und der Integration in Echtzeit-Umgebungen können die Ansätze weiter optimiert und neue Anwendungsmöglichkeiten erschlossen werden.

Literaturverzeichnis

[1] Agrawal, Motilal ; Konolige, Kurt ; Blas, Morten: CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching, 10 2008, S. 102–115. – ISBN 978-3-540-88692-1

[2] Arandjelović, Relja ; Gronat, Petr ; Torii, Akihiko ; Pajdla, Tomas ; Sivic, Josef: *NetVLAD: CNN architecture for weakly supervised place recognition.* 2016. – URL https://arxiv.org/abs/1511.07247

[3] Blaer, P. ; Allen, P.: Topological mobile robot localization using fast vision techniques. In: *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)* Bd. 1, 2002, S. 1031–1036 vol.1

[4] Bresson, Guillaume ; Alsayed, Zayed ; Yu, Li ; Glaser, Sébastien: Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving. In: *IEEE Transactions on Intelligent Vehicles* 2 (2017), Nr. 3, S. 194–220

[5] Chen, Zetao ; Lam, Obadiah ; Jacobson, Adam ; Milford, Michael: *Convolutional Neural Network-based Place Recognition.* 2014. – URL https://arxiv.org/abs/1411.1509

[6] Choudhary, Siddharth ; Narayanan, P. J.: Visibility Probability Structure from SfM Datasets and Applications. In: Fitzgibbon, Andrew (Hrsg.) ; Lazebnik, Svetlana (Hrsg.) ; Perona, Pietro (Hrsg.) ; Sato, Yoichi (Hrsg.) ; Schmid, Cordelia (Hrsg.): *Computer Vision – ECCV 2012.* Berlin, Heidelberg : Springer Berlin Heidelberg, 2012, S. 130–143. – ISBN 978-3-642-33715-4

[7] Eade, E. ; Drummond, T.: Scalable Monocular SLAM. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)* Bd. 1, 2006, S. 469–476

[8] Irschara, Arnold ; Zach, Christopher ; Frahm, Jan-Michael ; Bischof, Horst: From structure-from-motion point clouds to fast location recognition. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, S. 2599–2606

[9] Krizhevsky, Alex ; Sutskever, Ilya ; Hinton, Geoffrey: ImageNet Classification with Deep Convolutional Neural Networks. In: *Neural Information Processing Systems* 25 (2012), 01

[10] Kröse, B.J.A ; Vlassis, N ; Bunschoten, R ; Motomura, Y: A probabilistic model for appearance-based robot localization. In: *Image and Vision Computing* 19 (2001), Nr. 6, S. 381–391. – URL https://www.sciencedirect.com/science/article/pii/S026288560000086X. – ISSN 0262-8856

[11] Maeda, S. ; Kuno, Y. ; Shirai, Y.: Active navigation vision based on eigenspace analysis. In: *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97* Bd. 2, 1997, S. 1018–1023 vol.2

[12] Muhammad, Mohammed B. ; Yeasin, Mohammed: Eigen-CAM: Class Activation Map using Principal Components. In: *2020 International Joint Conference on Neural Networks (IJCNN)*, IEEE, Juli 2020, S. 1–7. – URL http://dx.doi.org/10.1109/IJCNN48605.2020.9206626

[13] Oliva, Aude ; Torralba, Antonio: Torralba, A.: Building the GIST of a scene: The role of global image features in recognition. Progress in Brain Research 155, 23-36. In: *Progress in brain research* 155 (2006), 02, S. 23–36. ISBN 9780444519276

[14] RangeKing: *Model structure of YOLOv8 detection models(P5) - yolov8n/s/m/l/x:*. https://private-user-images.githubusercontent.com/27466624/239739723-57391d0f-1848-4388-9f30-88c2fb79233f.jpg?jwt=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiJnaXRodWIuY29tIiwiYXVkIjoicmF3LmdpdGh1YnVzZXJjb250ZW50LmNvbSIsImtleSI6ImtleTUiLCJleHAiOjE3NDU5MTczODksIm5iZiI6MTc0NTkxNzA4OSwicGF0aCI6Ii8yNzQ2NjYyNC8yMzk3Mzk3MjMtNTczOTFkMGYtMTg0OC00Mzg4LTlmMzAtODhjMmZiNzkyMzNmLmpwZz9YLUFtei1BbGdvcml0aG09QVdTNC1ITUFDLVNIQTI1NiZYLUFtei1DcmVkZW50aWFsPUFLSUFWQ09EWUxTQTUzUFFLNFpBJTJGMjAyNTA0MjklMkZ1cy1lYXN0LTElMkZzMyUyRmF3czRfcmVxdWVzdCZYLUFtei1EYXRlPTIwMjUwNDI5VDA4NTgwOVomWC1BbXotRXhwaXJlcz0zMDAmWC1BbXotU2lnbmF0dXJlPThkZTQzZWViODAwMjFkMWM0NGM3NDI1NDZiODNlMjUwM

DFkZWY0NTViYjZmYzg1ZjE5Yzg0NzIzNTRlOTYwN2MmWC1BbXotU2lnbmVkSGVhZGVycz1ob3N0In0.lBYK0VjS7dk7C5EmnH2NAbyMExOddeTA_LzYUTP4GJs. 2023. – Abgerufen am 29.04.2024

[15] Sattler, Torsten ; Leibe, Bastian ; Kobbelt, Leif: Efficient Effective Prioritized Matching for Large-Scale Image-Based Localization. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (2017), Nr. 9, S. 1744–1756

[16] Simonyan, Karen ; Zisserman, Andrew: *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. – URL https://arxiv.org/abs/1409.1556

[17] Sivic ; Zisserman: Video Google: a text retrieval approach to object matching in videos. In: *Proceedings Ninth IEEE International Conference on Computer Vision*, 2003, S. 1470–1477 vol.2

[18] Sünderhauf, Niko ; Dayoub, Feras ; Shirazi, Sareh ; Upcroft, Ben ; Milford, Michael: *On the Performance of ConvNet Features for Place Recognition*. 2015. – URL https://arxiv.org/abs/1501.04158

[19] Sünderhauf, Niko ; Protzel, Peter: BRIEF-Gist - closing the loop by simple means. In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, S. 1234–1241

[20] Thoma, Janine ; Paudel, Danda P. ; Chhatkuli, Ajad ; Probst, Thomas ; Gool, Luc V.: Mapping, Localization and Path Planning for Image-Based Navigation Using Visual Features and Map. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, S. 7375–7383

[21] Zhang, Xiwu ; Wang, Lei ; Zhao, Yan ; Su, Yan: Graph-Based Place Recognition in Image Sequences with CNN Features. In: *J. Intell. Robotics Syst.* 95 (2019), August, Nr. 2, S. 389–403. – URL https://doi.org/10.1007/s10846-018-0917-2. – ISSN 0921-0296

A Anhang

A.1 Verwendete Hilfsmittel

In der Tabelle A.1 sind die im Rahmen der Bearbeitung des Themas der Bachelorarbeit verwendeten Werkzeuge und Hilfsmittel aufgelistet.

Tabelle A.1: Verwendete Hilfsmittel und Werkzeuge

Tool	Verwendung
LaTeX	Textsatz- und Layout-Werkzeug verwendet zur Erstellung dieses Dokuments

Erklärung zur selbständigen Bearbeitung

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort	Datum	Unterschrift im Original