







Hochschule für Technik und  
Wirtschaft Dresden  
University of Applied Sciences

## Diplomarbeit

### Entwicklung einer monokularen Personenerkennung für einen Notbremsassistenten

Zur Erlangung des akademischen Grades Diplomingenieur (FH)

---

vorgelegt von:	Elias Stampa
Matrikel-Nummer:	46519
Ort:	HTW Dresden
Fakultät:	Maschinenbau
Studiengang:	Fahrzeugtechnik
Betreuer:	Prof. Dr. rer. nat. Toralf Trautmann Dipl.-Ing. (FH) Franziskus Mendt
Abgabedatum:	18.11.2024

---



---

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>III</b>
<b>Verzeichnis der Formelzeichen und Symbole</b>	<b>VII</b>
<b>Abbildungsverzeichnis</b>	<b>VIII</b>
<b>Tabellenverzeichnis</b>	<b>X</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Zielsetzung . . . . .	2
1.2 Aufbau der Arbeit . . . . .	2
<b>2 Theoretische Grundlagen</b>	<b>3</b>
2.1 Kamera . . . . .	3
2.1.1 Einsatzgebiete von Kameras im Fahrzeug . . . . .	3
2.1.2 Aufbau und Funktionsweise der Kamera . . . . .	4
2.2 LiDAR . . . . .	9
2.3 Künstliche Neuronale Netze . . . . .	10
2.4 Ethernet . . . . .	12
2.5 Controller Area Network . . . . .	12
2.6 Fahrerassistenzsysteme . . . . .	13
2.7 Notbremsassistent . . . . .	16
<b>3 Vorbetrachtungen und Konzept</b>	<b>23</b>
3.1 Konzeptentwicklung . . . . .	23
3.2 Hardware . . . . .	24
3.2.1 Auswahl der Kamera-Hardware . . . . .	24
3.2.2 LiDAR-Sensor . . . . .	27
3.3 MATLAB . . . . .	27
3.4 Algorithmen zur Personenerkennung in MATLAB . . . . .	29
3.4.1 Aggregated Channel Features (ACF)-Detektor . . . . .	29
3.4.2 You Only Look Once (YOLO) . . . . .	33
3.5 Vergleich der Objekterkennungsmethoden . . . . .	34
3.6 Vergleichsversuche der Notbrems-Algorithmen . . . . .	37
3.6.1 Testumgebung und Versuchsablauf der Realfahrversuche . . . . .	38
3.6.2 Testszenario . . . . .	39

<b>4</b>	<b>Umsetzung</b>	<b>41</b>
4.1	Integration der Hardware . . . . .	41
4.1.1	Integration der Gigabit-Ethernet-Kamera . . . . .	41
4.1.2	Integration des LiDAR-Sensors . . . . .	43
4.2	Integration der Objekterkennungen . . . . .	44
4.3	Positionsbestimmung der detektierten Objekte im Raum . . . . .	47
4.3.1	Herausforderungen der monokularen Tiefenbestimmung . . . . .	47
4.3.2	Projektionsgeometrischer Ansatz . . . . .	48
4.3.3	Kamerakalibrierung . . . . .	51
4.3.4	Berechnung der Detektionspositionen in Fahrzeugkoordinaten . . . . .	53
4.3.5	Versuch zur Validierung der Genauigkeit der Entfernungsberechnung . . . . .	53
4.4	Entscheidungslogik für Warnung und Bremsung . . . . .	54
4.5	Tracking . . . . .	59
4.6	Entscheidungslogik für Warnung und Bremsung . . . . .	61
4.7	Integration in die Fahrzeugumgebung . . . . .	63
<b>5</b>	<b>Validierung des Systems</b>	<b>67</b>
5.1	Vergleich der Objektdetektoren . . . . .	67
5.2	Realfahrversuch . . . . .	71
5.2.1	Prüfszenario . . . . .	71
5.2.2	Testdurchführung . . . . .	72
5.3	Auswertung der LiDAR-Punktwolken . . . . .	73
5.3.1	Clustering-Verfahren zur Personendetektion . . . . .	75
5.3.2	Bildbasierte Punktwolkenauswertung . . . . .	75
5.4	Auswertung der Fahrversuche . . . . .	78
5.5	Vergleich zur LiDAR-Personenerkennung . . . . .	82
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>87</b>
	<b>Literatur- und Quellenverzeichnis</b>	<b>89</b>
	<b>Eidesstattliche Erklärung</b>	<b>95</b>
	<b>Anhang</b>	<b>95</b>

---

## Abkürzungsverzeichnis

<b>ACC</b>	Adaptive Cruise Control
<b>ACK</b>	Acknowledge
<b>ACF</b>	Aggregated Channel Features
<b>ADAS</b>	Advanced Driver Assistance Systems
<b>AEB</b>	Automatic Emergency Braking
<b>AEBS</b>	Advanced Emergency Braking System
<b>AP</b>	Average Precision
<b>APS</b>	Active Pixel Sensor
<b>ASIL</b>	Automotive Safety Integrity Levels
<b>BEV</b>	Birds Eye View
<b>Caltech</b>	California Institute of Technology
<b>CAN</b>	Controller Area Network
<b>CCD</b>	Charge Coupled Device
<b>CMOS</b>	Complementary Metal Oxide Semiconductor
<b>COCO</b>	Common Objects in Context
<b>CNN</b>	Convolutional Neural Network
<b>CPU</b>	Central Processing Unit
<b>CRC</b>	Cyclic Redundancy Check
<b>ECU</b>	Electronic Control Unit
<b>EOF</b>	End of Frame
<b>Euro NCAP</b>	European New Car Assessment Programme
<b>FAS</b>	Fahrerassistenzsysteme
<b>FNN</b>	Feedforward Neuronal Network
<b>FPS</b>	Frames per Second
<b>fps</b>	frames per second
<b>FOV</b>	Field of View
<b>FP</b>	False Positive
<b>GPU</b>	Graphics Processing Unit
<b>GigE</b>	Gigabit-Ethernet
<b>IEEE</b>	Institute of Electrical and Electronics Engineers
<b>ID</b>	Identifier

<b>INRIA</b>	Institut national de recherche en informatique et en automatique
<b>IoU</b>	Intersection over Union
<b>IP</b>	Internet Protocol
<b>KNN</b>	Künstliches Neuronales Netz
<b>KI</b>	Künstliche Intelligenz
<b>KITTI</b>	Karlsruhe Institute of Technology and Toyota Technological Institute
<b>LAN</b>	Local Area Network
<b>LDW</b>	Lane Departure Warning
<b>LiDAR</b>	Light Detection and Ranging
<b>LKA</b>	Lane Keeping Assist
<b>ML</b>	Machine Learning
<b>mAP</b>	mean Average Precision
<b>PDC</b>	Park Distance Control
<b>PoE</b>	Power over Ethernet
<b>PPS</b>	Passive Pixel Sensor
<b>Radar</b>	Radio detection and ranging
<b>R-CNN</b>	Region-Based Convolutional Neural Network
<b>RGB</b>	Rot-Grün-Blau
<b>RNN</b>	Recurrent Neural Network
<b>ROI</b>	Region of Interest
<b>ROS</b>	Robot Operating System
<b>RPN</b>	Region Proposal Network
<b>SOF</b>	Start of Frame
<b>SVM</b>	Support Vector Machine
<b>TCP</b>	Transmission Control Protocol
<b>ToF</b>	Time of Flight
<b>TN</b>	True Negative
<b>TP</b>	True Positive
<b>TTC</b>	Time to Collision
<b>UV</b>	Ultra Violett
<b>USB</b>	Universal Serial Bus
<b>V2V</b>	Vehicle-to-Vehicle



<b>V2X</b>	Vehicle-to-Everything
<b>VRU</b>	Vulnerable Road Users
<b>YOLO</b>	You Only Look Once



## Verzeichnis der Formelzeichen und Symbole

Symbol	Einheit	Beschreibung
$c$	-	Lenkübersetzung
$c_0$	m/s	Lichtgeschwindigkeit (300.000 m/s)
$d$	m	Abstand eines Objekts
$f$	mm	Brennweite eines Kameraobjektivs
$h$	m	Höhe der Kamera über der Bodenebene
$k$	-	Blendenzahl
$L$	mm,cm	Achsabstand
$P_{\text{horiz}}$	Pixel/Grad	horizontale Winkelauflösung einer Kamera
$s$	m	Strecke
$s_{\text{brems}}$	m	Distanz zu einem Objekt im Moment des Brems-TTC-Schwellwerts
$R$	m	Kurvenradius
$t$	s	Zeit
$TTC$	s	Zeit bis zur Kollision (Time to Collision)
$u, v$	Pixel	Bildkoordinaten
$u_c, v_c$	Pixel	Bildkoordinaten des Bildhauptpunktes
$v$	km/h, m/s	Geschwindigkeit
$v_r$	m/s	Relative Geschwindigkeit zwischen Fahrzeug und Hindernis
$X_C, Y_C, Z_C$	-	Kamerakoordinaten (optisches Zentrum)
$X_W, Y_W, Z_W$	-	Raumkoordinaten (Bodenebene)
$\alpha$	°	Winkel zwischen optischer Achse und Projektionsstrahl
$\beta$	°	Projektionswinkel zur Bodenebene eines Bildpunktes
$\theta$	°	Bogenmaß zur Bestimmung der Fahrschlauchlänge
$\varphi$	°	Neigungswinkel der optischen Achse

---

## Abbildungsverzeichnis

Abbildung 2.1	Innovative Möglichkeit zur Erfassung der Fahreraktivitäten [3]	3
Abbildung 2.2	Lochkameramodell der projektiven Abbildung [4]	5
Abbildung 2.3	Schematischer Aufbau eines Farbbildsensors [5]	6
Abbildung 2.4	Schematische Darstellung der Linsenarten [8]	7
Abbildung 2.5	Schematische Darstellung der Strahlengänge sphärischer Linsen [9]	8
Abbildung 2.6	Bildwinkel in Abhängigkeit der Brennweite [11]	8
Abbildung 2.7	Schematische Darstellung der Verzeichnung [12]	9
Abbildung 2.8	Schematischer Aufbau eines künstlichen neuronalen Netzes [13]	11
Abbildung 2.9	Schematischer Aufbau einer CAN-Botschaft [14]	12
Abbildung 3.1	Konzeptionelle Darstellung des Informationsflusses und der Verarbeitungsprozesse des umzusetzenden Algorithmus	24
Abbildung 3.2	Sichtbereich des Ouster OS1-64 [29]	28
Abbildung 3.3	Workflow eines ACF-Detektors [33]	30
Abbildung 3.4	Funktionsweise der Sliding Window-Methode [34]	30
Abbildung 3.5	Gradientenhistogramme verschiedener Skalen [35]	32
Abbildung 3.6	Funktionsweise der Objektdetektion mithilfe von Ankerboxen [34]	33
Abbildung 3.7	Berechnung der Intersection over Union [36]	35
Abbildung 3.8	Prüffeld am K-Gebäude der HTW Dresden [37]	38
Abbildung 4.1	Darstellung der physischen Verbindung mehrerer Komponenten über ein Power over Ethernet (PoE)-Ethernet-Switch	41
Abbildung 4.2	Beschreibung von Bildpunkten in einem Graustufen- und einem RGB-Farbbild	42
Abbildung 4.3	Darstellung einer Light Detection and Ranging (LiDAR)-Punktwolke in MATLAB	44
Abbildung 4.4	Darstellung der Detektionen mit Bounding Box-Koordinaten und Klassenbezeichnung	46
Abbildung 4.5	Schematische Darstellung monokularer Tiefenkriterien	48
Abbildung 4.6	Bestimmung der Distanz $d$ über den Neigungswinkel der optischen Achse bei gegebener Fahrbahnorientierung $\varphi$ und Höhe der Kamera $h$	50
Abbildung 4.7	Bestimmung der Distanz $d$ zu einer Bildposition $v$ bei gegebener Fahrbahnorientierung $\varphi$ und Höhe der Kamera $h$	50
Abbildung 4.8	Aufnahme des Kalibrierungsmusters mit detektierten Eckpunkten	52
Abbildung 4.9	Korrigierte Abbildung unter Berücksichtigung der Kameraparameter	52

Abbildung 4.10	Kollage der ermittelten Positionen des Dummys in 5 Meter Schritten . . . . .	54
Abbildung 4.11	Darstellung der Fahrschläuche im Kamerabild für einen Geschwindigkeitswert von 20 km/h . . . . .	56
Abbildung 4.12	Darstellung der Fahrschläuche im Kamerabild für einen Geschwindigkeitswert von 20 km/h bei vollem Lenkeinschlag . . . . .	58
Abbildung 4.13	Extraktion der Keypoints innerhalb der Begrenzungsboxen . . . . .	60
Abbildung 4.14	Kamerabefestigung an der Windschutzscheibe mithilfe einer Saugnapfhalterung . . . . .	63
Abbildung 4.15	Befestigung des Ouster LiDAR-Sensors auf dem Dachträger des Versuchsfahrzeugs . . . . .	64
Abbildung 4.16	Realisierung der Verzögerung über einen CAN-gesteuerten Feststellbrems-Aktuator . . . . .	65
Abbildung 5.1	Grafische Darstellung der gemittelten Werte der errechneten Bewertungsmetriken . . . . .	69
Abbildung 5.2	Erreichter F1-Score in Zusammenhang zur benötigten Detektionszeit . . . . .	70
Abbildung 5.3	Darstellung der ungefilterten Punktwolke einer Messfahrt . . . . .	73
Abbildung 5.4	Darstellung der gefilterten Punktwolke . . . . .	74
Abbildung 5.5	Darstellung der erkannten Person mittels Clustering-Verfahren . . . . .	75
Abbildung 5.6	Erzeugtes Birds Eye View (BEV)-Bild der gefilterten Punktwolke . . . . .	76
Abbildung 5.7	Bereinigtes BEV-Bild nach morphologischer Schließung der verbleibenden Pixelfläche . . . . .	77
Abbildung 5.8	Entfernungsbestimmung im BEV-Bild . . . . .	77
Abbildung 5.9	Gemittelte Distanzen zwischen Fahrzeug und Prüf-Dummy im Auslösemoment des Bremsbefehls . . . . .	79
Abbildung 5.10	Verarbeitete Bilder pro Sekunde während der Messfahrten . . . . .	79
Abbildung 5.11	Regression der Ermittelten Entfernungen über der Verarbeitungsgeschwindigkeit . . . . .	80
Abbildung 5.12	Regression der gemittelten Entfernungen über der Verarbeitungsgeschwindigkeit . . . . .	80
Abbildung 5.13	Verbleibende Time to Collision (TTC)-Werte im Moment des Bremsbefehls . . . . .	81
Abbildung 5.14	Verfahren zur Ermittlung der abbildenden Punkte eines Objektes in LiDAR- und Bilddaten . . . . .	84
Abbildung 5.15	Detektion in LiDAR-Punktwolken mithilfe von complex-yolov4 . . . . .	84

---

## Tabellenverzeichnis

Tabelle 3.1	Entschlüsselung der Bezeichnung der Basler acA1920-40gc . . . . .	26
Tabelle 4.1	Nummerierung der Detektormodelle . . . . .	45
Tabelle 4.2	Erreichte Detektionsgeschwindigkeiten der Detektoren im Vergleich	46
Tabelle 4.3	Vergleich der tatsächlichen und gemessenen Entfernungen mittels Kamera . . . . .	54
Tabelle 4.4	FPS-Werte und Detektionsraten bei Detektion in jedem Bild und bei jeder fünften Bildaufnahme . . . . .	61
Tabelle 5.1	Erreichte Leistungswerte der Detektoren bei jeweiligem Schwellwert von 0,5 für IoU und Aussagewahrscheinlichkeit . . . . .	68
Tabelle 5.2	Erreichte Leistungswerte der Detektoren bei jeweiligem Schwellwert von 0,7 für IoU und Aussagewahrscheinlichkeit . . . . .	70
Tabelle 5.3	gemittelte Messwerte der Versuchsfahrten . . . . .	78
Tabelle 5.4	TTC-Warn-Werte für aller Messfahrten . . . . .	82
Tabelle 5.5	Vergleich der darstellenden Punkte eines Objektes in LiDAR- und Bilddaten . . . . .	83
Tabelle A.1	Messwerte der Realfahrversuche . . . . .	100

---

# 1. Einleitung

Die Sicherheit im Straßenverkehr stellt weltweit eine große Herausforderung dar. Trotz moderner Infrastruktur und zahlreicher gesetzlicher Regelungen sterben laut der Weltgesundheitsorganisation (WHO) jährlich über 1,3 Millionen Menschen bei Verkehrsunfällen, darunter ein erheblicher Anteil ungeschützter Verkehrsteilnehmer wie Fußgänger und Radfahrer, die sogenannten Vulnerable Road Users (VRU). Vor diesem Hintergrund wurde die internationale Initiative Vision Zero ins Leben gerufen, die das Ziel verfolgt, die Zahl der Verkehrstoten und Schwerverletzten signifikant zu reduzieren. Die Vision Zero basiert auf der Überzeugung, dass jeder Verkehrstote vermeidbar ist und setzt auf eine Kombination aus smarter Infrastruktur, Regulierungen und technologischen Innovationen um der Idealvorstellung eines Straßenverkehrssystems ohne Getötete und Schwerverletzte näherzukommen.

In der heutigen Zeit übernehmen Fahrerassistenzsysteme zunehmend Fahraufgaben und entlasten damit die fahrzeugführende Person erheblich. Dies führt nicht nur zu einem erhöhten Fahrkomfort, sondern spielt auch eine entscheidende Rolle bei der Reduzierung von Unfallrisiken. Assistenzsysteme schützen dabei nicht nur die Fahrzeuginsassen, sondern auch andere Verkehrsteilnehmer, insbesondere die ungeschützten VRU. Durch das Übertragen immer größerer Verantwortung auf das Fahrzeug ist es unerlässlich, dass diese Systeme möglichst zuverlässig funktionieren, ohne dabei überempfindlich zu reagieren.

Innovative Fahrerassistenzsysteme, insbesondere Notbremsassistenten, sind ein Schlüsselfaktor, um die Ziele der Vision Zero zu erreichen. Diese Systeme kombinieren modernste Sensortechnologien, künstliche Intelligenz und fortschrittliche Steueralgorithmen, um gefährliche Situationen frühzeitig zu erkennen und entsprechende Maßnahmen einzuleiten. Notbremsassistenten haben das Potenzial, Unfälle mit VRU zu vermeiden oder deren Schwere signifikant zu reduzieren, weshalb diese Systeme seit Juli 2022 für neue Fahrzeugtypen und seit Juli 2024 für jedes neuzugelassene Fahrzeug verpflichtend sind. Dennoch gilt es die Forschung und Entwicklung in diesem Bereich weiter voranzutreiben, um die Zahl der tödlichen Verkehrsunfälle stetig zu senken [1][2].

### **1.1. Zielsetzung**

In der vorliegenden wissenschaftlichen Arbeit zum Thema "Umsetzung einer monokularen Personenerkennung für ein Notbremssystem" wird ein umfassendes Konzept zur Erkennung von Personen im Fahrzeugumfeld mittels einer einzigen Kamera entwickelt und umgesetzt. Ziel ist die Realisierung eines funktionalen Prototyps für Forschung und Lehre, der potenzielle Kollisionssituationen frühzeitig identifiziert und daraufhin einen Notbremsbefehl auslöst, um die Sicherheit von Fahrzeuginsassen und anderen Verkehrsteilnehmern zu erhöhen. Durch die Entwicklung eines offenen und konfigurierbaren Notbremssystems auf Basis einer monokularen Kamera wird eine universelle Integration in verschiedene Fahrzeugumgebungen ermöglicht. Zudem bietet das System eine Plattform für den einfachen Vergleich verschiedener Erkennungsalgorithmen in Versuchen mit einem Versuchsfahrzeug. Damit schafft die Arbeit mehr Transparenz und Nachvollziehbarkeit der Algorithmen eines Notbremssystems und eröffnet neue Möglichkeiten für den Einsatz in der Lehre sowie für weitere Forschungsprojekte.

### **1.2. Aufbau der Arbeit**

Im Anschluss an die Einleitung folgt ein Kapitel zur Erläuterung wichtiger theoretischer Sachverhalte, welche dem Verständnis späterer Umsetzungen und Herleitungen dienen soll. Darauf folgt im dritten Kapitel eine Darstellung des Konzeptes, sowie die Vorstellung der Methodik zur Umsetzung und dem Erreichen der gesetzten Ziele. Dabei wird vor allem auf die eingesetzte Sensorik und die Softwarebasierten Werkzeuge eingegangen, sowie auf die Methoden zur abschließenden Validierung des umgesetzten Systems. Im darauffolgenden vierten Kapitel wird die Vorgehensweise zur Umsetzung der Konzepte unter Verwendung der zuvor beschriebenen Methoden aufgezeigt. Das Fünfte Kapitel widmet sich der Durchführung von Versuchen und Validierungsmethoden der umgesetzten Systeme, sowie der Auswertung sämtlicher ermittelter Messwerte, um eine Aussage über die Funktionalität der umgesetzten Algorithmen treffen zu können und einen Vergleich zu ermöglichen. Abschließend soll in Kapitel sechs eine Zusammenfassung des erreichten Fortschrittes und ein Ausblick auf zukünftige Verbesserungsmöglichkeiten des Systems erfolgen.



---

## 2. Theoretische Grundlagen

In diesem Kapitel werden zunächst die theoretischen und technischen Hintergründe erörtert, welche für spätere Umsetzungen und Erklärungen von zentraler Bedeutung sind. Die folgenden Abschnitte bieten eine systematische Einführung in die wesentlichen Theorien und Modelle, welche den grundlegenden Rahmen für die Analyse, Entwicklung und Interpretation der Ergebnisse dieser Arbeit bilden.

### 2.1. Kamera

Die Kamera hat in den letzten Jahrzehnten eine beeindruckende Entwicklung durchlaufen. Von den ersten Analogen Modellen hat sie sich zu einem allgegenwärtigen digitalen Begleiter entwickelt und zwar nicht nur in Form von reinen Fotoapparaten oder Videokameras. In jedem Smartphone und nahezu jedem Laptop ist heutzutage eine Kamera verbaut und somit ist der Aufwand Bild- oder Videodaten aufzunehmen heutzutage ein sehr viel geringerer, als noch vor einigen Jahren. Diese Technologisierung sorgte nicht nur für eine fortlaufende Verfügbarkeit günstiger Bildsensoren, sondern sorgte auch dafür, dass die Entwicklung der digitalen Kamera immer weiter vorangetrieben wurde und uns somit heute kleine, aber dennoch sehr leistungsfähige und hochauflösende Kameras für jeglichen Einsatzzweck zur Verfügung stehen.

#### 2.1.1. Einsatzgebiete von Kameras im Fahrzeug

Durch die zunehmende Automatisierung moderner Fahrzeuge steigt die Anzahl der eingebauten Sensoren, um ein detailliertes Umweltbild zu erzeugen. Besonders Kameras sind bedeutend, da sie dem menschlichen Auge ähneln und im gleichen Lichtwellenspektrum arbeiten. Ursprünglich erweiterten Kameras das Blickfeld des Fahrers durch Echtzeit-Videodarstellung, wie etwa Rückfahrkameras zum Rangieren und Einparken.

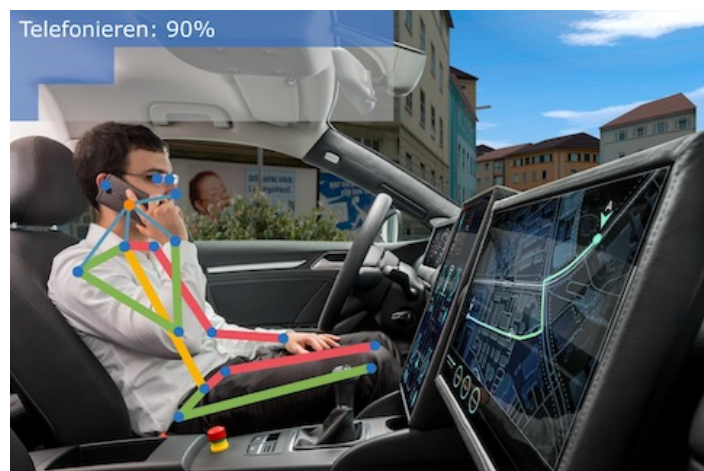


Abbildung 2.1.: Innovative Möglichkeit zur Erfassung der Fahreraktivitäten [3]

Elektronische Außenspiegel ersetzen konventionelle Spiegel durch kleinere, aerodynamische Kameras, was den Luftwiderstand und die Reichweite elektrischer Fahrzeuge verbessert. Mit der Weiterentwicklung extrahieren komplexe Systeme aus Videosignalen Umweltinformationen für innovative Fahrerassistenzsysteme, wie Spurhalteassistenten, Verkehrszeichenerkennung und Parkassistenten. Im Innenraum überwachen Kameras den Fahrerzustand, unterstützen Gestensteuerungen und ermöglichen personalisierte Einstellungen durch Fahreridentifikation. Insgesamt spielen Kameras eine zentrale Rolle in der Fahrzeugentwicklung und bieten in Kombination mit fortschrittlicher Software großes Potenzial für zukünftige Entwicklungen.

### 2.1.2. Aufbau und Funktionsweise der Kamera

Moderne Digitalkameras sind, unabhängig von ihrem Einsatzzweck, in ihrem grundlegenden Aufbau sehr identisch. Ein allgemeiner Einblick in den Aufbau, sowie die Funktionsweise einer Kamera soll hier kurz erläutert werden.

#### Lochkameramodell

Das Lochkameramodell ist eines der grundlegendsten theoretischen Modelle in der Photographie und der Bildverarbeitung, mit welchem sich grundlegende Sachverhalte vereinfacht erklären lassen, um ein besseres Verständnis späterer Zusammenhänge herzustellen. Das Lochkameramodell stellt dabei ein idealisiertes Modell zur Beschreibung der Bildentstehung und der perspektivischen Abbildung einer Kamera dar. Zunächst ist das Verhalten von Licht zu verstehen, um weitere Zusammenhänge der geometrischen Optik nachzuvollziehen. Licht ist eine Form der elektromagnetischen Strahlung, welches im allgemeinen Sprachgebrauch jedoch nur einen gewissen Bereich des Spektrums der elektromagnetischen Wellen beschreibt. Dazu gehört der Bereich, der für das menschliche Auge sichtbar ist und umfasst Wellenlängen von etwa 400 bis 800 *nm*. In der Physik werden auch Wellenbereiche kürzerer Wellen, wie Ultra Violett (UV) und Röntgenstrahlung und längerer Wellenlängen, wie Infrarot- und Terrahertz-Strahlung dazu gezählt. Im Folgenden werden jedoch nur die sichtbaren Anteile der elektromagnetischen Strahlung von Relevanz sein. Die wichtigste Eigenschaft von Licht für die geometrische Optik ist hierbei, dass sich Licht in geradlinigen Strahlen ausbreitet. Beim Lochkameramodell wird die Blendenöffnung, durch die das Licht fällt, als so klein angenommen, dass in der Ebene der Abbildung ein scharfes Bild entsteht. Die Geometrie der Projektion wird dabei mithilfe des dreidimensionalen Kamerakoordinatensystems beschrieben, dessen Ursprung in der Blendenöffnung platziert wird und als optisches Zentrum bezeichnet wird. Die X- und Y-Achse dieses Koordinatensystems liegen parallel zur Bildebene, während die z-Achse senkrecht zur Bildebene steht und als optische Achse bezeichnet wird.

Im Lochkameramodell wird jeder Punkt in der Objektwelt durch einen Lichtstrahl repräsentiert, der durch die Blendenöffnung auf einen entsprechenden Punkt in der Bildebene

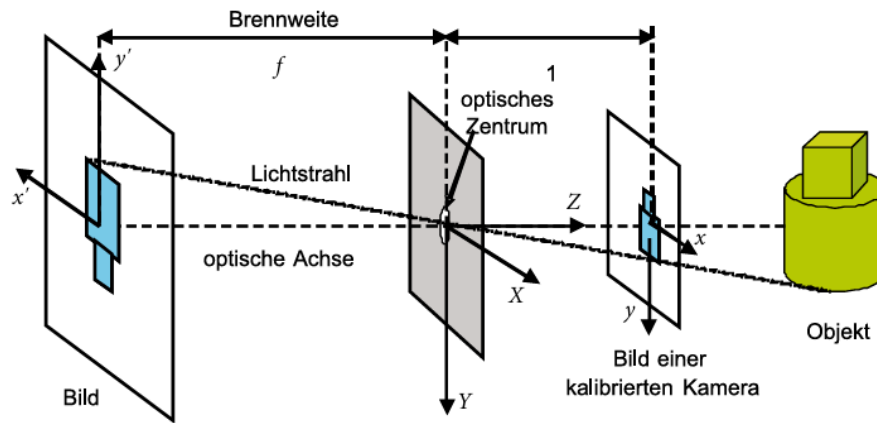


Abbildung 2.2.: Lochkameramodell der projektiven Abbildung [4]

projiziert wird. Diese Projektion erfolgt gemäß der Prinzipien der perspektivischen Abbildung, bei der parallele Strahlen in der Realität auf einen Fluchtpunkt in der Bildebene treffen. Mathematisch lässt sich dies durch die Abbildungsgleichungen des Modells beschreiben, die die Transformation von 3D-Koordinaten in 2D-Bildkoordinaten ermöglichen. Ein wesentliches Merkmal des Lochkameramodells ist die perspektivische Verzerrung, die es ermöglicht, Tiefeninformationen in das zweidimensionale Bild zu integrieren. Dadurch entsteht ein realistischer Eindruck von Raum und Tiefe, was für die Bildverarbeitung komplexerer Systeme von großer Bedeutung ist. Trotz seiner Einfachheit bietet das Lochkameramodell eine wertvolle Grundlage für das Verständnis komplexerer Kameramodelle und Bildverarbeitungsprozesse. In der Praxis werden reale Kameras durch zusätzliche Faktoren wie Linsenverzerrungen, Blendenöffnung und Sensorrauschen beeinflusst. Dennoch ermöglicht das Lochkameramodell eine erste Annäherung an die Bildentstehung und dient als Ausgangspunkt für weiterführende geometrische Zusammenhänge der Projektion der dreidimensionalen Welt auf eine zweidimensionale Abbildung einer Kamera.

## Bildsensor

Der Bildsensor stellt das Herzstück einer modernen Digitalkamera dar, welcher in Verbindung mit einem Speichermedium den fotografischen Film bzw. die Fotoplatte analoger Kameras ersetzt. Der Complementary Metal Oxide Semiconductor (CMOS)-Sensor ist hierbei der derzeit am weitesten verbreitetste Bildsensor und hat mit der Zeit den bis dato viel verbauten Charge Coupled Device (CCD)-Sensor, welcher dem CMOS-Sensor lange Zeit in Hinblick auf das Rauschverhalten überlegen waren, abgelöst. Das schlechtere Rauschverhalten konnte durch innovative Entwicklungen kompensiert werden und heute überwiegen die Vorteile des CMOS-Sensors, speziell des sogenannten Active Pixel Sensor (APS). Neben dem APS, wird auch der Passive Pixel Sensor (PPS) in der CMOS-Bauweise gefertigt. Der größte Vorteil des APS ist sein Aufbau, welcher dem eines Computer-Mikroprozessors sehr nahe kommt und damit sehr günstig in Massenproduktion gefertigt werden kann und zusätzlich deutlich energieeffizienter ist, als vergleichbare CCD-Sensoren. APS-Sensoren sind

hierbei jedoch besonders gefragt, da diese im Gegensatz zum PPS- und CCD-Sensor sogenannte aktive Pixel haben. Die aktiven Pixel bieten den Vorteil, dass das aufgenommene Bildsignal direkt im Bildsensor in digitaler Form vorliegt. Somit können zusätzliche Funktionen, wie eine automatisierte Rauschreduzierung und andere Bildverarbeitungsprozesse direkt auf dem Sensor realisiert werden.

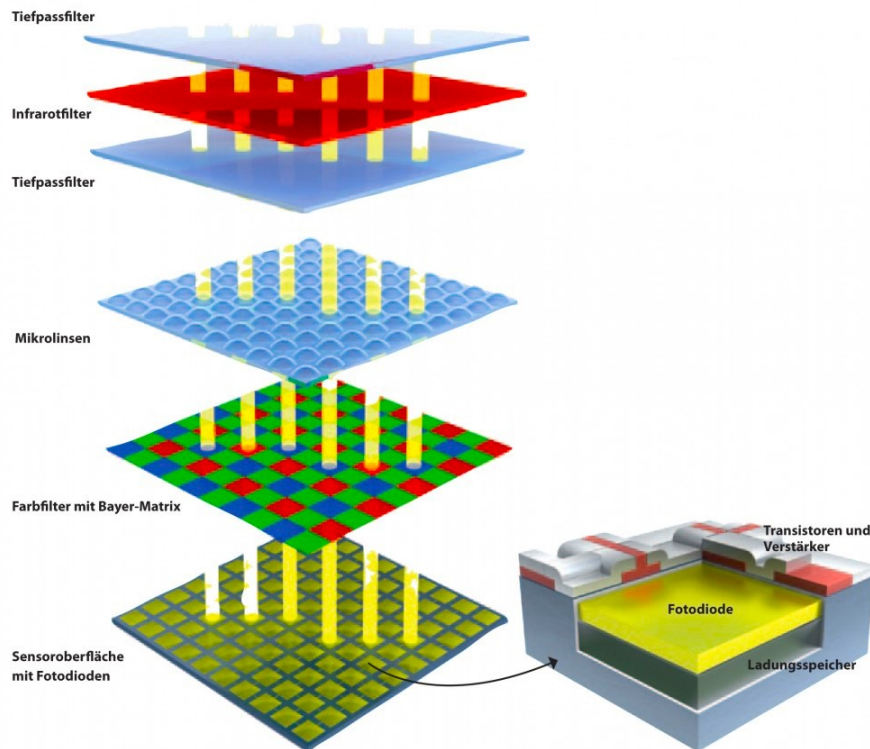


Abbildung 2.3.: Schematischer Aufbau eines Farbbildsensors [5]

Der Sensor selbst ist aus mehreren Schichten aufgebaut. Die grundlegendste und wichtigste Schicht ist hierbei die Schicht, auf die das Licht zuletzt fällt. Diese ist die eigentliche Sensoroberfläche, welche aus mehreren Millionen Fotodioden besteht, wobei jede Fotodiode einem Pixel entspricht. Diese Fotodioden sind in der Lage die Photonen des Lichtes, gemäß des photoelektrischen Effekts, in Elektronen umzuwandeln. Wenn Licht auf das Halbleitermaterial des Sensors trifft, wird die Energie der Photonen von Elektronen im Material absorbiert, was energetisch bereits genügt, um die Elektronen aus ihren Bindungen zu lösen und somit als freie Elektronen im Material zu bewegen. Jedes Pixel im Sensor sammelt die Elektronen über einen vorgegebenen Zeitraum, der sogenannten Belichtungszeit oder Verschlusszeit, wobei die Menge an freien Elektronen der Intensität des Lichtes entspricht, welches auf das spezifische Pixel trifft. Diese Elektronen werden dann in ein elektrisches Signal umgewandelt, das der Helligkeit des Pixels entspricht. Mit diesem Signal lässt sich jedoch nur ein Bild darstellen, welches der Lichtintensität entspricht, also ein reines Graustufenbild. Um auch Farbintensitäten aufnehmen zu können werden daher verschiedene Farbfilter auf die einzelnen Pixel aufgebracht, um ausschließlich das Licht einer bestimmten Wellenlänge, folglich einer bestimmten Farbe auf die Fotodiode passieren zu lassen. Die verschiedenen

Farben werden hierbei in entsprechenden Matrizen angeordnet. Der am häufigsten verwendete Farbfilter moderner Bildsensoren ist die Bayer-Matrix, welche in Abbildung 2.3 als Farbfilter dargestellt ist. Die Bayer-Matrix besteht aus den drei Grundfarben Rot, Grün und Blau, häufig als RGB zusammengefasst, wobei der grüne Anteil 50% ausmacht und Rot und Blau jeweils zu 25% vertreten sind. Dies hat den Grund, dass das menschliche Auge besonders empfindlich für grüne Lichtwellen ist, da etwa 60% der vom menschlichen Auge wahrnehmbaren Lichtwellen im grünen Lichtspektrum liegen. Damit ist der Mensch in der Lage Unterschiede in den grünen Wellenlängen feiner zu unterscheiden, wodurch die Wahrnehmung von Bilddetails und Bildschärfe genauer wird [6] [7].

### Objektiv / Optik

Nicht weniger relevant ist auch die Optik, bzw. das Objektiv einer Kamera. Hierbei handelt es sich um ein sammelndes optisches System, welches maßgeblich die Bilddarstellung auf dem Bildsensor steuert. Das Objektiv befindet sich im Strahlengang zwischen dem Objekt und der Abbildung, welche bei modernen Kameras auf dem Bildsensor realisiert wird. Die einfachste Form eines Objektivs besteht aus einer einzigen Sammellinse, moderne Objektivsysteme bestehen jedoch häufig aus mehreren Linsen und zusätzlich können Spiegel oder auch Beugungsgitter in den Strahlengang eingebracht werden. Die Komponenten sind in der Regel in einem Tubus untergebracht, der innen schwarz gehalten ist, um Streulicht zu vermeiden. Hauptsächlich werden die Eigenschaften eines Kameraobjektives jedoch durch die verschiedenen Linsen beeinflusst. Dabei sind die beiden Radien, die Dicke der Linse, sowie der Brechungsindex und die Dispersion des Materials entscheidend. Sphärische Linsen, also Linsen mit Kugeloberflächen, können in Sammellinsen (Konvexlinsen) und Streulinsen (Konkavlinsen) unterschieden werden, wobei eine Sammellinse im Zentrum dicker ist, als am Rand, während dies bei Streulinsen genau andersherum ist.

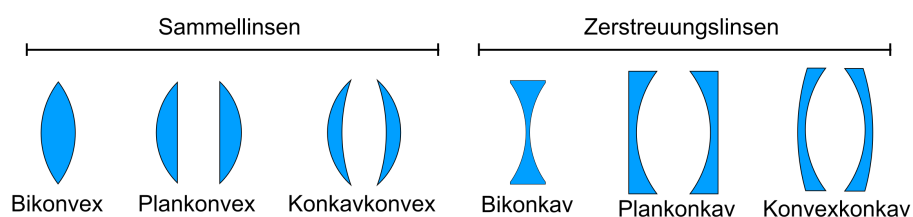


Abbildung 2.4.: Schematische Darstellung der Linsenarten [8]

Die verschiedenen Linsenformen haben einen maßgeblichen Einfluss auf die Veränderung des Strahlengangs, des Lichtes, welches durch die Linse fällt. Diese Unterschiede sind in Abbildung 2.5 schematisch dargestellt.

Durch gezielte Anordnungen der verschiedenen Linsenformen lässt sich somit das Verhalten des Lichtes innerhalb des Objektivs auf verschiedene Weisen steuern und damit können unterschiedliche Brennweiten realisiert werden. Die Brennweite  $f$  ist das Hauptmerkmal eines Objektivs und beschreibt die Entfernung zwischen der Mitte der Linse und dem Punkt, in dem sich parallele Strahlen nach dem Passieren einer Linse schneiden, beziehungsweise

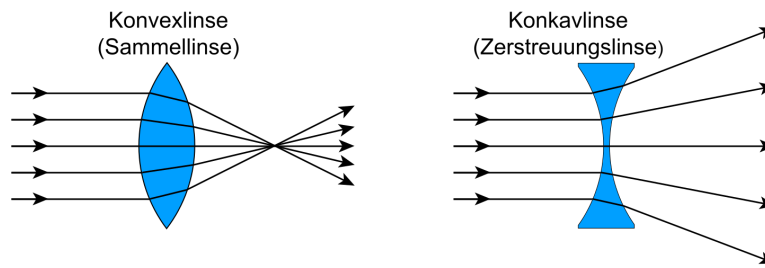


Abbildung 2.5.: Schematische Darstellung der Strahlengänge sphärischer Linsen [9]

bündeln, dem sogenannten Brennpunkt. Über die Brennweite lässt sich damit der ungefähre Abbildungsmaßstab eines Objekts für einen gegebenen Objektstand bestimmen. Zu beachten ist dabei, dass sich mit steigender Brennweite der Bildwinkel verkleinert, also das Sichtfeld, Field of View (FOV) genannt, der Kamera verkleinert wird. Dieser Sachverhalt ist in Abbildung 2.6 schematisch dargestellt. Das menschliche Auge kann im unbewegten Zustand in etwa einen Bereich erfassen, der einem Bildwinkel von rund 53 Grad entspricht, womit sich für den gleichen Bildwinkel bei einem Vollformat-Bildsensor eine Brennweite von etwa 43 mm ergibt [10]. Diese Brennweiten um etwa 50 mm werden somit als Normalbrennweiten bezeichnet, wobei hier auch die Größe des Bildsensors eine entscheidende Rolle spielt, da bei kleineren Sensoren immer nur ein Ausschnitt des technisch möglichen dargestellt wird und die Normalbrennweite für kleinere Sensoren damit auch kürzer ausfällt. Dieses Verhältnis kann über den Crop-Faktor der spezifischen Sensorgröße ermittelt werden. Brennweiten, die unterhalb der Normalbrennweiten liegen bezeichnet man somit als weitwinkelig, da der Bildwinkel erhöht wird und Objektive mit höheren Brennweiten, als die Normalbrennweite, werden als Teleobjektive bezeichnet, da zwar der Bildwinkel verkleinert wird, jedoch weit entfernte Objekte stark vergrößert dargestellt werden können.

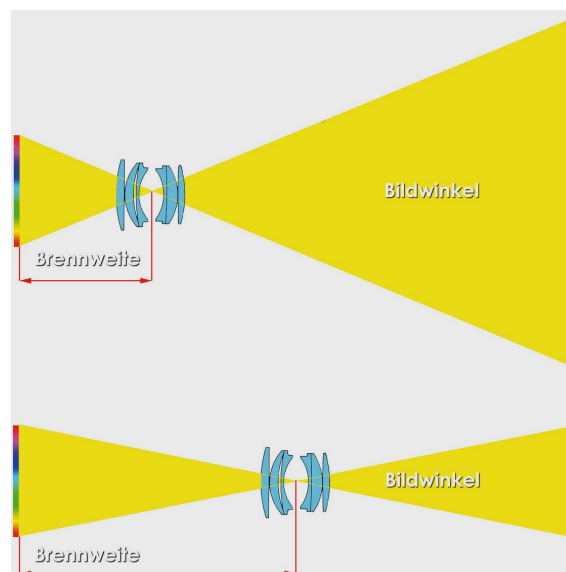


Abbildung 2.6.: Bildwinkel in Abhängigkeit der Brennweite [11]

Während Objektive häufig recht große Brennweitenbereiche abdecken können, werden im Automobilbereich häufig Optiken mit Festbrennweite verbaut, da eine Kameramodul im

Fahrzeug meist nur einen Anwendungsfall abdeckt und somit die Anforderungen an das System zuvor festgelegt sind.

Durch die Unvollkommenheit von Linsen entstehen Abbildungsfehler, auch Abberationen genannt. Diese lassen sich in dem meisten Fällen durch die Verwendung mehrerer Linsen stark verringern oder gar gänzlich kompensieren.

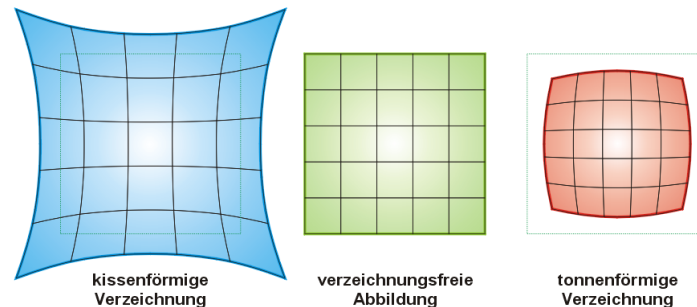


Abbildung 2.7.: Schematische Darstellung der Verzeichnung [12]

Dennoch können Verzerrungen, die sogenannte Verzeichnung oder Distortion, auftreten die besonders in Anwendungen, bei denen geometrische Informationen aus der Umwelt erfasst werden sollen, beachtet werden müssen. Bei der Verzeichnung werden gerade Linien der echten Welt, gekrümmt in der Projektion der Kamera dargestellt. Die am häufigsten auftretenden Distortionen sind dabei die kissenförmige und tonnenförmige Verzeichnung, dargestellt in Abbildung 2.7.

## 2.2. LiDAR

LiDAR bezeichnet ein optisches Messverfahren, welches im automotiven Bereich meist auf der Time of Flight (ToF)-Messung basiert. Es werden elektromagnetische Impulse in Form von Infrarotem, Ultraviolettem oder sichtbarem Licht ausgesendet und von vorhandenen Objekten reflektiert, bevor die reflektierten Strahlen von einem photosensitiven Sensor aufgenommen werden. Dabei wird die Zeit zwischen Aussenden des Lichtimpulses und dem Empfangen dessen Reflexion gemessen, wobei die Zeitdifferenz proportional zur Entfernung des reflektierenden Objektes ist und kann nach Formel 2.1 berechnet werden.

$$d = \frac{c_0 \cdot t}{2} \quad (2.1)$$

$d$  = Abstand in m

$c_0$  = Lichtgeschwindigkeit (300.000 m/s)

$t$  = Zeit in s

Die gemessene Zeit entspricht dabei jedoch der doppelten Strecke, da das Licht die Strecke zweimal zurücklegt, bis es wieder von dem Sensor empfangen wird und muss daher halbiert werden, um auf die einfache Strecke zu schließen. LiDAR-Sensoren können in ihrem

Aufbau stark variieren. Während es LiDAR-Sensoren gibt, die über eine Lichtquelle, welche lediglich einen Lichtstrahl aussendet und eine photosensitive Diode verfügen, gibt es ebenso hochkomplexe Systeme, welche im 360-Grad-Winkel in einer Vielzahl von Ebenen Lichtstrahlen aussenden und diese mit hochsensiblen Optiksystemen empfangen können. Solche mehrschichtigen Systeme geben ihre Daten als Punktwolken aus, in der sämtliche Reflexionen aus der Umwelt in dreidimensionalen Koordinaten hinterlegt sind und somit ein räumliches Abbild der Umgebung erzeugen können. Die Reichweiten moderner Sensoren betragen dabei bis zu mehrere hunderte Meter, während die Messgenauigkeit nur im Millimeterbereich liegt. Während LiDAR-Systeme im Gegensatz zur Kamera eine unabhängige Tiefenwahrnehmung bieten und nicht von den Lichtverhältnissen der Umgebung abhängig sind, besteht allerdings eine starke Abhängigkeit des Systems von Witterungsbedingungen. Bei Nebel oder starkem Niederschlag können die Lichtstrahlen stark von Tropfen abgelenkt werden und somit die Leistung des Sensors negativ beeinflussen.

### 2.3. Künstliche Neuronale Netze

Ein Künstliches Neuronales Netz (KNN), auch Neuronales Netz genannt, ist ein mathematisches Computermodell, welches stark vom Aufbau und der Funktionsweise des menschlichen Gehirns inspiriert ist und das Ziel verfolgt, komplexe Problemstellungen zu lösen, welche nicht, wie bei Computerprogrammen üblich, mit vorgegebenen Regeln beschrieben werden können. Künstliche Neuronale Netze bestehen in der Regel aus sehr vielen Neuronen, welche ähnlich wie die Neuronen im menschlichen Gehirn, miteinander verbunden sind. Die Neuronen eines KNN sind in mehreren Schichten angeordnet, wobei die Anzahl der Schichten und der darin enthaltenen Neuronen, je nach Komplexität des Anwendungsfalls variieren kann. Das einzelne künstliche Neuron kann hierbei als sehr einfache Recheneinheit betrachtet werden und hat die Funktion ein Eingangssignal in bestimmter Weise, entsprechend einer mathematischen Funktion, Aktivierungsfunktion genannt, zu modifizieren und dieses veränderte Signal anschließend entsprechend der Vernetzung an das in der nächsten Schicht liegende Neuron weiterzuleiten. Jede Verbindung zwischen den einzelnen Neuronen hat hierbei eine Gewichtung, welche beschreibt, wie stark ein Neuron das nächste beeinflusst. Die Schichten sind hintereinander angeordnet, wobei ein KNN üblicherweise aus einer Eingabeschicht (Inputlayer), einer oder mehreren verborgenen Schichten (Hidden Layer) und aus einer Ausgabeschicht (Outputlayer) besteht.

Die Neuronen einer Schicht können mit jedem Neuron, der jeweils nächsten Schicht verbunden sein (Fully Connected Layer), aber es gibt auch Strukturen, wo nicht alle Neuronen der aufeinanderfolgenden Schichten miteinander vernetzt sind (Sparse Connection). Neuronen innerhalb einer Schicht sind in der Regel nicht miteinander verbunden. Je nach Komplexität des Anwendungsfalls eines Neuronalen Netzes, steigt auch die Anzahl der Schichten und Verzweigungen des Netzes. Neuronale Netze bilden heutzutage die Grundlage vieler Anwendungen aus dem Bereich der künstlichen Intelligenz (KI) und kommen in sehr vielfältigen Bereichen zum Tragen. Da verschiedene Anwendungsbereiche auch verschiedene Anforde-



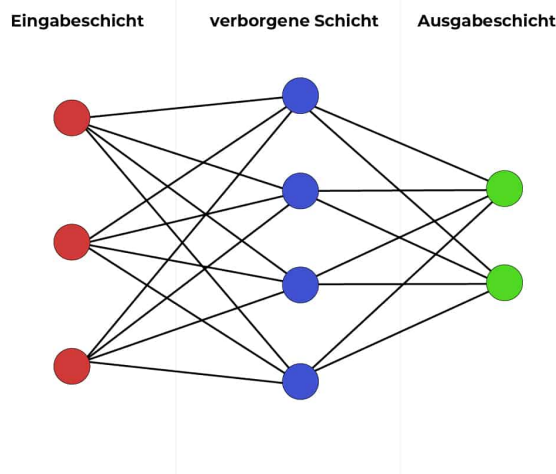


Abbildung 2.8.: Schematischer Aufbau eines KNN [13]

rungen an die KNNs stellen und diese daher auch in ihrer Architektur stark variieren können, soll hier ein kurzer Überblick über die verbreitetsten Netzarten gegeben werden:

- **Feedforward Neuronal Network (FNN)**

Dies ist die einfachste Form eines KNNs. Die Informationen fließen bei einem FNN nur in eine Richtung, es existieren keine Rückkopplungen zu vorherigen Schichten und keine Schleifen. Diese Art von Netzen ist durch seine sehr einfache Struktur für grundlegende Aufgaben, wie Mustererkennung, Regressions- und Klassifikationsprobleme geeignet, jedoch sind derartige Netze meist sehr ineffizient bei der Verarbeitung komplexerer Daten, wie Bildern oder Sprache.

- **Recurrent Neural Network (RNN)**

RNNs sind darauf ausgelegt sequenzielle Daten zu verarbeiten. Im Gegensatz zu FNNs verfügen diese Netze über Rückkopplungen zwischen den Schichten, was ihnen ermöglicht, Informationen über vorherige Zeitpunkte zu behalten und kurzzeitig zwischenspeichern. Durch diese Art der Gedächtnisfunktion sind RNNs besonders gut geeignet für Anwendungen im Bereich der Spracherkennung, des maschinellen Übersetzens oder der Zeitreihenanalyse.

- **Convolutional Neural Network (CNN)**

CNNs sind besonders geeignet für die Verarbeitung von Bild- und Videodaten. Sie nutzen sogenannte Faltungsschichten, welche wie Filter funktionieren und relevante Merkmale, wie Kanten oder Muster, aus den Daten extrahieren, wodurch die Menge der Daten stark reduziert wird, wobei die wichtigsten Informationen erhalten bleiben. Typischerweise bestehen derartige KNNs aus mehreren Faltungs- und Pooling-Schichten, welche die Daten schrittweise verarbeiten und immer weiter vereinfachen, bevor sie am Ende in vollständig verbundenen Schichten eine Entscheidung treffen, beispielsweise, um vorherzusagen, welches Objekt in einem Bild abgebildet ist.

Damit ein KNN präzise Aussagen treffen kann, muss nicht nur die Architektur des Netzes dem Anwendungsfall angepasst werden, auch muss das Netz zuvor sorgfältig mit möglichst

vielen und sehr variablen Daten trainiert werden, um die Robustheit bei Veränderungen der Randbedingungen zu steigern. Im Falle der Objekterkennung wird das Training eines KNNs mithilfe von gelabelten Bildern durchgeführt, in welchen die Positionen der zu erkennenden Objekte, mithilfe von Begrenzungsrahmen vorgegeben werden. Das Labeln der Daten kann manuell oder auch in automatisierten Prozessen stattfinden.

## 2.4. Ethernet

Ethernet ist eine grundlegende Netzwerktechnologie, die ursprünglich dazu gedacht war, Daten in kabelgebundenen Netzwerken, den sogenannten Local Area Networks (LANs), zu übertragen und spezifiziert dabei sowohl die Netzwerkprotokolle, als auch die eingesetzte Hardware. Es existiert eine Vielzahl an Standards für Ethernet, jedoch hat sich der IEEE 802.3 Standard als wichtigste Ethernet-Spezifikation durchgesetzt. Dabei werden die Daten mithilfe des Transmission Control Protocol (TCP) unterteilt und in Pakete (Packets) verpackt und versendet. Die Pakete enthalten neben der eigentlichen Information (Payload) immer einen Header, welcher Informationen über das verwendete Netzwerkprotokoll enthält, sowie die IP-Adressen des Empfängers und des Versenders und Informationen, welche entscheidend sind, um die Gesamtdatei aus den Datenpaketen zu rekonstruieren. Datenpakete, die über mehrere Netzwerke hinweg übertragen werden sollen, werden auf der Data Link Layer zusätzlich in sogenannte Frames (Rahmen) eingebettet, welche weiterhin MAC-Adressen

Zusätzlich zur Datenübertragung bietet der Ethernet-Standard die Möglichkeit, Geräte direkt über das achtadrige Ethernet-Kabel mit Strom zu versorgen. Das nennt sich PoE und kann in der aktuellsten Generation bis zu 100W liefern.

## 2.5. Controller Area Network

Das Controller Area Network (CAN) ist ein zuverlässiges Kommunikationsprotokoll, das für die effiziente Vernetzung von Steuergeräten (ECUs) in Fahrzeugen und industriellen Systemen entwickelt wurde.

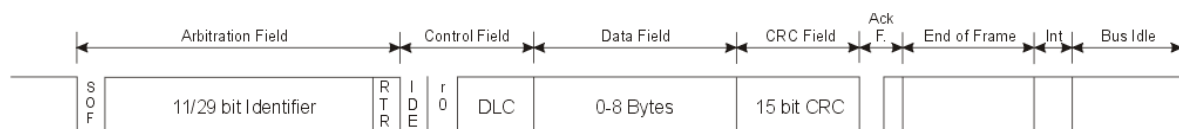


Abbildung 2.9.: Schematischer Aufbau einer CAN-Botschaft [14]

Eine CAN-Nachricht beginnt mit einem Start of Frame (SOF) enthält einen Identifier (ID), der die Priorität der Nachricht bestimmt. Niedrigere IDs haben höhere Priorität und werden bevorzugt behandelt, was eine effektive und zeitkritische Kommunikation ermöglicht.

Das Data Field überträgt die eigentlichen Nutzdaten und wird durch einen Cyclic Redundancy Check (CRC) auf Fehler überprüft. Der sogenannte Acknowledge (ACK) bestätigt den Empfang der Nachricht, während der End of Frame (EOF) das Ende markiert.

Durch seine prioritätsbasierte Kommunikation gewährleistet CAN, dass kritische Nachrichten schnell und zuverlässig übermittelt werden, wodurch es besonders für sicherheits- und zeitkritische Anwendungen wie Fahrzeugsteuerungen und industrielle Prozesse geeignet ist.

### 2.6. Fahrerassistenzsysteme

Fahrerassistenzsysteme (FAS) sind technologische Lösungen, die den Fahrer bei der Fahrzeugführung unterstützen und die Sicherheit im Straßenverkehr erhöhen sollen. Sie können sowohl in die Längs- als auch in die Querverführung des Fahrzeugs eingreifen und bieten Funktionen, die von einfachen Warnungen bis hin zu teilautomatisierten Fahrfunktionen reichen. Zu den bekanntesten FAS gehören Systeme wie der Notbremsassistent (AEBS/AEB), der Spurhalteassistent (LDW/LKA), der Abstandsregeltempomat (ACC) oder auch die Einparkhilfe (PDC).

Das Hauptziel von Fahrerassistenzsystemen ist es, die fahrzeugführende Person in kritischen Situationen zu entlasten und im besten Fall Unfälle zu verhindern. Indem sie kontinuierlich, mithilfe verschiedener Sensorik, Informationen aus der Umgebung des Fahrzeugs erfassen, können FAS frühzeitig auf potenzielle Gefahrensituationen aufmerksam machen und somit die Reaktionszeit des Fahrers erheblich verkürzen. Studien haben gezeigt, dass moderne Assistenzsysteme das Unfallrisiko erheblich reduzieren können, insbesondere bei Auffahrunfällen und gefährlichen Spurwechseln.

Zusätzlich tragen viele der Systeme zu einer deutlichen Erhöhung des Fahrkomforts bei. Assistenzsysteme wie der Abstandsregeltempomat Adaptive Cruise Control (ACC) oder die Stauassistent nehmen dem Fahrer anstrengende und vor allem monotone Aufgaben ab, was zu einer geringeren physischen und mentalen Belastung führt. Langfristig ebnen diese Technologien den Weg für das hochautomatisierte oder sogar autonome Fahren, indem sie immer komplexere Verkehrssituationen selbstständig bewältigen und immer mehr unterstützende Systeme für spezifische Fahraufgaben dazukommen.

Fahrerassistenzsysteme sind die Grundlage für automatisierte und autonome Fahrfunktionen, die nach dem **SAE-Standard J3016** in fünf Automatisierungsstufen unterteilt werden. Diese Klassifizierung basiert auf dem Grad der Eingriffe, die das System eigenständig vornimmt, sowie:

- **Stufe 0** – Keine Automatisierung: Der Fahrer hat die volle Kontrolle über das Fahrzeug, während FAS nur warnend oder unterstützend wirken, wie etwa ein Spurwechselassistent.
- **Stufe 1 – Fahrerunterstützung:** Hier übernimmt das System entweder die Längs- oder Querverführung des Fahrzeugs, beispielsweise durch das ACC oder einen Spurhalteassistenten, wie die Lane Departure Warning (LDW), die Lenkimpulse gibt. Der Fahrer muss jedoch stets bereit sein, die Kontrolle zu übernehmen.

- **Stufe 2 – Teilautomatisierung des Fahrens:** Systeme dieser Stufe können sowohl die Längs- als auch die Querverführung übernehmen. Der Fahrer muss jedoch aufmerksam bleiben und das System überwachen, beispielsweise bei einem Stauassistenten, der autonom beschleunigt und abbremst.
  - **Stufe 3 – Bedingte Fahrautomatisierung:** Das Fahrzeug kann in bestimmten Situationen, wie etwa im Stau oder auf der Autobahn, für einen gewissen Zeitraum selbstständig fahren. Der Fahrer muss jedoch bereit sein, auf Anforderung wieder einzugreifen.
  - **Stufe 4 – Hochautomatisiertes Fahren:** In fest definierten Einsatzbereichen (z.B. auf Autobahnen oder in Parkhäusern) kann das Fahrzeug alle Fahraufgaben selbstständig übernehmen, ohne dass der Fahrer eingreifen muss. Eine manuelle Übernahme ist nur noch in Ausnahmefällen erforderlich.
  - **Stufe 5 – Vollautomatisiertes Fahren:** In dieser Stufe ist kein menschlicher Eingriff mehr erforderlich, das Fahrzeug kann in allen Verkehrssituationen autonom fahren.
- [15]

Die derzeit auf dem Markt befindlichen Fahrerassistenzsysteme bewegen sich größtenteils zwischen den Automatisierungsstufen 1 und 3, wobei Mercedes mit dem Drive Pilot 95, welcher der Automatisierungsstufe 3 entspricht und bis zu einer Geschwindigkeit von 95 km/h zugelassen wurde, das derzeit fortschrittlichste, ins Deutschland in einem Serien-PKW zugelassene System bietet. Vollautomatisierte Fahrzeuge, welche auf den Stufen 4 oder 5 operieren, sind aktuell noch Gegenstand von Forschung und Erprobung.

Umfelderfassende Fahrerassistenzsysteme funktionieren, indem sie Daten, die sie von verschiedensten Sensoren im Fahrzeug sammeln, verarbeiten und analysieren. Zu den meistverbauteiten Sensortypen zählen Kameras, Radar- und Lidarsensoren, sowie Ultraschallsensoren. Diese Sensoren liefern sämtliche Informationen über die Fahrzeugumgebung, wie beispielsweise die Entfernung zu anderen Verkehrsteilnehmern und Hindernissen, Informationen über die Umgebungsbedingungen, wie Dunkelheit und Witterung oder auch die Erkennung von Fahrspuren und entgegenkommenden Fahrzeugen bei Nacht. Aber auch Fahrzeuginterne Informationen, wie beispielsweise die Raddrehzahlen und Beschleunigungswerte werden kontinuierlich gemessen und verarbeitet um permanent einen stabilen Fahrzustand zu gewährleisten.

Je nach Art und Funktionsweise des Assistenzsystems, werden diese Daten von einem zentralen Steuergerät (ECU) verarbeitet, das gegebenenfalls entsprechende Eingriffe einleitet oder dem Fahrer Warnsignale übermitteln kann. Ein Notbremsassistent beispielsweise kann im Ernstfall eine vollständig autonome Gefahrenbremsung einleiten, um eine Kollision zu verhindern oder zumindest die Kollisionsenergie zu mindern. Der Spurhalteassistent nutzt Kameras, um Fahrbahnmarkierungen zu erkennen und kann bei Abweichungen von der Spurmitte korrigierende Lenkmanöver vorschlagen (LDW) oder diese automatisch ausführen

(Lane Keeping Assist (LKA)). Ein Totwinkelassistent überprüft schwer einsehbare Bereiche mit Ultraschallsensoren und warnt den Fahrer mittels einer Warnleuchte im Außenspiegel, falls sich dort ein Verkehrsteilnehmer befindet. All diese Systeme zielen vor allem auf die Sicherheit im Straßenverkehr ab, während Systeme wie beispielsweise der adaptive Abstandstempomat (ACC) unter der Vorgabe einer gewünschten maximalen Geschwindigkeit in einem gewissen Rahmen selbstständig die gesamte Längsführung des Fahrzeugs übernehmen. Dadurch werden sowohl der Komfort auf langen Strecken erhöht als auch die Konzentration des Fahrers geschont, was wiederum das Sicherheitspotential erhöht.

Manche Fahrerassistenzsysteme sind in modernen Fahrzeugen mittlerweile gesetzlich vorgeschrieben, insbesondere im Rahmen der EU-Verkehrssicherheitsvorschriften. Eine zentrale Rolle spielt dabei die 'General Safety Regulation', welche für alle neu zugelassenen Fahrzeuge seit Juli 2024 gilt. Diese Verordnung verlangt, dass Fahrzeuge mit einer Reihe von Fahrerassistenzsystemen ausgestattet sein müssen, darunter:

- **Notbremsassistent (Advanced Emergency Braking System, AEBS):** Dieses System muss in der Lage sein, automatisch zu bremsen, wenn eine Kollision erkannt wird. Es soll helfen, Auffahrunfälle zu verhindern oder deren Schwere zu verringern. Der AEBS ist sowohl für Pkw als auch für Lkw und Busse vorgeschrieben.
- **Aktiver Spurhalteassistent (Lane Keeping Assist, LKA):** Fahrzeuge müssen über ein System verfügen, das den Fahrer warnt, wenn das Fahrzeug unbeabsichtigt die Spur verlässt. Zudem kann der Assistent bei Bedarf leichte Lenkeingriffe vornehmen, um das Fahrzeug in der Spur zu halten.
- **Müdigkeitserkennung (Driver Drowsiness and Attention Warning):** Dieses System überwacht das Verhalten des Fahrers und warnt, wenn Anzeichen von Müdigkeit oder Ablenkung erkannt werden.
- **Rückfahrassistent (Rear-view Camera or Detection System):** Alle Fahrzeuge müssen mit einem System ausgestattet sein, das Hindernisse beim Rückwärtsfahren erkennt und den Fahrer warnt oder Eingriffe vornimmt, um Zusammenstöße zu vermeiden.

Zusätzlich sieht die Verordnung vor, dass Fahrzeuge zukünftig auch mit einem Intelligenten Geschwindigkeitsassistenten (ISA) ausgestattet sein müssen, der den Fahrer auf Geschwindigkeitsbegrenzungen hinweist und unterstützt, diese einzuhalten. Diese Systeme sollen nicht nur den Fahrer warnen, sondern können, je nach Ausprägungsstufe, auch aktiv die Geschwindigkeit anpassen.

Trotz der vielen Vorteile, welche derartige Assistenzsysteme mit sich bringen, entstehen auch viele Herausforderungen mit der Anwendung derartigen Technologisierungen. Eine der größten ist die zuverlässige Erkennung von Objekten und Gefahren, vorallem bei schlechter Sicht oder unübersichtlichen Verkehrssituationen. Dadurch ist eine kontinuierliche Verbesserung der Sensorik, sowie der angewandten Algorithmen notwendig, um diese Systeme weiterhin

zu verbessern und die Zuverlässigkeit, mit Hinblick auf das große Ziel des autonomen Fahrens, stetig zu erhöhen. Moderne Systeme machen sich verstärkt der Kombination mehrerer Sensoren (Sensorfusion) zu Nutze, um eine robustere und genauere Umfelderkennung zu ermöglichen. Beispielsweise werden Kameras häufig in Kombination mit Sensoren zur Distanzmessung verwendet, da eine einzelne Kamera nicht in der Lage ist, valide Tiefeninformationen zu liefern. Zusätzlich wird kontinuierlich an der Verbesserung der Mensch-Maschine-Schnittstelle gearbeitet, um sicherzustellen, dass der Fahrer in jeder Situation ausreichend informiert und eingebunden bleibt. Eine weitere große Herausforderung ist die Akzeptanz der Technologien seitens der Nutzer. Viele Menschen begegnen derartigen automatisierten Fahrfunktionen häufig mit Skepsis und Zurückhaltung. Deshalb ist es besonders wichtig, diese Systeme vertrauenswürdig und benutzerfreundlich zu gestalten.

Durch die starken Fortschritte im Bereich der Künstlichen Intelligenz (KI) und des maschinellen Lernens und Deep Learning der letzten Jahre konnte auch die Erkennungsfähigkeit der FAS erheblich verbessert werden. Zusätzlich können durch den Einsatz solcher Systeme im Straßenverkehr, immer mehr Daten gesammelt werden, welche zusätzlich die Entwicklung und Anpassung der Algorithmen vorantreiben. Ein weiterer vielversprechender Ansatz besteht in der Entwicklung von innovativen Kommunikationstechnologien zwischen den Fahrzeugen untereinander (Vehicle-to-Vehicle (V2V)), sowie zusätzlich mit Infrastrukturkomponenten und Daten aus dem Internet (Vehicle-to-Everything (V2X)). Diese Technologien könnten nicht nur die Sicherheit im Straßenverkehr erhöhen, sondern zusätzlich den Verkehrsfluss verbessern.

### **2.7. Notbremsassistent**

Der Notbremsassistent (Advanced Emergency Braking System (AEBS) oder Automatic Emergency Braking (AEB)) ist ein sehr bedeutsames Fahrerassistenzsystem geworden, das darauf abzielt, Kollisionen mit anderen Verkehrsteilnehmern und Objekten zu verhindern oder die Schwere des Zusammenstoßes zu reduzieren und somit die Sicherheit im Straßenverkehr, sowohl für Fahrzeuginsassen, als auch für ungeschützte Verkehrsteilnehmer, signifikant zu erhöhen. Wenn das System eine drohende Kollision erkennt, wird zunächst der Fahrer durch akustische, visuelle oder haptische Signale gewarnt. Wird auf diese Warnungen nicht entsprechend reagiert oder ist die Zeit bis zur Kollision zu gering, leitet das System automatisch eine Bremsung ein, um das Fahrzeug rechtzeitig zum Stillstand zu bringen oder die Aufprallgeschwindigkeit möglichst stark zu verringern. Während zu Beginn noch zwischen einem 'Notbremsassistenten', welcher lediglich warnt und das Bremssystem vorkonditioniert, um schneller den vollen Bremsdruck zu erreichen und einem 'autonomen Notbremssystem' unterschieden wurde, ist heute nur noch das vollautonome System von Relevanz.

Untersuchungen zeigen, dass ein Großteil schwerer Verkehrsunfälle auf menschliches Versagen zurückzuführen ist, insbesondere in Situationen, in denen eine schnelle und präzise Reaktion erforderlich ist. Hier stößt der Mensch aufgrund physiologischer und psychologi-

scher Einschränkungen, wie verzögerter Reaktionszeit oder Panik, häufig an seine Grenzen. Der Notbremsassistent kann in solchen kritischen Momenten als zuverlässiger Helfer agieren, da er unabhängig von äußeren Einflüssen oder emotionalem Zustand stets objektiv und sofort reagiert.

Darüber hinaus haben statistische Auswertungen bereits gezeigt, dass Fahrzeuge mit AEBS-Systemen signifikant weniger Auffahrunfälle und Kollisionen mit Fußgängern oder Radfahrern verursachen. Diese Systeme tragen somit nicht nur zur Verringerung der Unfallzahlen bei, sondern reduzieren auch die Unfallfolgen, wie schwere Verletzungen oder Todesfälle [16] [17]. Vor diesem Hintergrund ist der Einsatz von Notbremsassistenten nicht nur eine technologische Verbesserung, sondern eine essentielle Maßnahme zur Erhöhung der allgemeinen Verkehrssicherheit.

### **Gesetzliche Vorschriften**

Aufgrund des nachweisbaren Nutzens und der potentiellen Rettung von Menschenleben sind Notbrems-Assistenzsysteme seit dem 06. Juli 2022 für alle neuzugelassenen Fahrzeugtypen in Europa, im Rahmen der Typgenehmigung, Pflicht und seit dem 07. Juli 2024 flächendeckend für alle neuzugelassenen Fahrzeuge verpflichtend.

Geregelt werden diese Vorgaben durch verschiedene europäische und internationale Verordnungen und Regelungen, die die technischen Anforderungen und Pflichten zur Ausstattung von Fahrzeugen mit Notbremsassistentensystemen festlegen.

Eine der zentralen europäischen Vorschriften ist die bereits erwähnte **EU-Verordnung 2019/2144**, auch bekannt als die "General Safety Regulation". Diese Verordnung schreibt vor, dass alle neuen Fahrzeugtypen ab dem 6. Juli 2022 und alle neu zugelassenen Fahrzeuge ab dem 7. Juli 2024 mit fortschrittlichen Sicherheitssystemen ausgestattet sein müssen, zu denen auch Notbremsassistentensysteme gehören. Ziel dieser Verordnung ist es, die Verkehrssicherheit zu erhöhen und die Zahl der Verkehrsunfälle mit Todesfolge oder schweren Verletzungen signifikant zu reduzieren.

Die **Verordnung (EG) Nr. 661/2009** des Europäischen Parlaments und des Rates vom 13. Juli 2009 legt allgemeine Sicherheitsanforderungen für Kraftfahrzeuge fest. Sie bildet die Grundlage für die Typgenehmigung von Fahrzeugen hinsichtlich ihrer Sicherheitsausstattung und beinhaltet unter anderem Bestimmungen zu Notbremsassistentensystemen. Diese Verordnung zielt darauf ab, technische Hindernisse im Fahrzeugbau zu beseitigen und ein hohes Maß an Fahrzeugsicherheit zu gewährleisten.

Auf internationaler Ebene ist die **UN-Regelung Nr. 152** von Bedeutung, die einheitliche Vorschriften für die Genehmigung von Fahrzeugen in Bezug auf fortschrittliche Notbremsassistentensysteme festlegt. Sie definiert die technischen Anforderungen, Leistungsstandards und Testverfahren für AEBS, um sicherzustellen, dass diese Systeme weltweit einheitlichen Sicherheitsstandards entsprechen.

Des Weiteren ist die **ECE-Regelung R131** relevant, die spezifische Anforderungen an Notbremsassistentensysteme für schwere Nutzfahrzeuge stellt. Diese Regelung legt fest, unter

welchen Bedingungen das System aktiv werden muss, welche Objekte es erkennen soll und wie die Reaktionszeiten und Bremsleistungen gestaltet sein müssen. Ziel ist es, die Sicherheit im Schwerlastverkehr zu erhöhen und Unfälle durch Auffahrkollisionen zu verhindern.

Die **ISO 26262** ergänzt diese Vorschriften, indem sie die funktionale Sicherheit von elektrischen und elektronischen Systemen in Fahrzeugen sicherstellt. Für AEBS-Systeme gewährleistet die ISO 26262, dass diese zuverlässig und fehlerfrei arbeiten, indem sie den gesamten Sicherheitslebenszyklus von der Entwicklung bis zur Produktion abdeckt. Durch die Festlegung von Automotive Safety Integrity Levels (ASIL) werden angemessene Sicherheitsmaßnahmen basierend auf Risikobewertungen implementiert, wodurch die Systeme hohen Sicherheitsstandards entsprechen.

Diese Vorschriften beschreiben dabei die Mindestanforderungen, die ein Notbremssystem erfüllen muss, um auf dem europäischen Markt genehmigt zu werden.

### **Bewertung und Tests durch Euro NCAP**

Neben den gesetzlichen Vorschriften spielt das European New Car Assessment Programme (Euro NCAP) eine entscheidende Rolle bei der Bewertung und Sicherstellung der Wirksamkeit von Fahrerassistenzsystemen. Euro NCAP führt umfangreiche Tests durch, um die Leistung und Zuverlässigkeit von Notbremsassistenzsystemen sowie anderen Sicherheitstechnologien unter realen Fahrbedingungen zu prüfen.

Durch standardisierte Crashtests und Prüfscenarien, sowie spezifische Bewertungsprotokolle bewertet Euro NCAP, wie effektiv diese Systeme bei der Vermeidung von Unfällen und der Reduzierung von Aufprallkräften sind. Bei der Prüfung von AEBS-Systemen werden unterschiedliche Tests berücksichtigt, die sowohl die Interaktion mit anderen Fahrzeugen als auch mit verschiedenen VRUs wie Fußgängern, Radfahrern und Motorradfahrern umfassen. Dafür werden verschiedene Szenarien mit statischen und bewegten Prüfzielen, sogenannten Dummies, nachgestellt [18] [19]. Die Testergebnisse werden in Form von Sternbewertungen veröffentlicht, die Verbrauchern als Orientierungshilfe bei der Auswahl sicherer Fahrzeuge dienen [20]. Darüber hinaus motivieren die Bewertungen Hersteller, kontinuierlich in die Weiterentwicklung und Verbesserung ihrer Sicherheitssysteme zu investieren, um höhere Bewertungen zu erzielen und sich im Wettbewerb zu differenzieren. Dennoch zeigen unabhängige Tests, dass viele Notbremssysteme, speziell auf die Anforderungen der European New Car Assessment Programme (Euro NCAP)-Tests zugeschnitten sind und bei geringfügigen Abweichungen deutliche Defizite aufweisen [21].

Die Zusammenarbeit zwischen gesetzlichen Vorgaben und den unabhängigen Bewertungen von Euro NCAP trägt maßgeblich zur Steigerung der Verkehrssicherheit bei. Während die gesetzlichen Vorschriften die Mindeststandards festlegen, gehen die Euro NCAP Tests darüber hinaus, indem sie die tatsächliche Leistungsfähigkeit der Systeme unter verschiedenen Szenarien bewerten und somit eine zusätzliche Sicherheitsebene für die Verbraucher bieten.



### Funktionsweise

Ein Notbremssystem umfasst eine Vielzahl an Prozessschritten, die permanent ablaufen, um dauerhaft die Umgebung des Fahrzeuges zu beobachten und die Situationen einzuschätzen und gegebenenfalls einzugreifen. Dazu sind folgende Teilprozesse notwendig:

#### 1. Umgebungserfassung:

- **Sensorik:** Das System nutzt Sensoren wie Radar, Lidar, Ultraschall und Kameras oder auch eine Kombination dieser (Sensorfusion), um die Umgebung vor dem Fahrzeug kontinuierlich zu überwachen.
- **Datenaufnahme:** Diese Sensoren sammeln Informationen über Objekte, deren Abstand, Geschwindigkeit und Bewegungsrichtung.

#### 2. Objekterkennung und -klassifizierung:

- **Datenverarbeitung:** Die gesammelten Rohdaten werden verarbeitet, um Objekte zu identifizieren.
- **Klassifizierung:** Identifizierte Objekte werden als Fahrzeuge, Fußgänger, Radfahrer oder Hindernisse klassifiziert.

#### 3. Gefahrenanalyse:

- **Relativgeschwindigkeit:** Berechnung der relativen Geschwindigkeit zwischen dem eigenen Fahrzeug und dem erkannten Objekt.
- **Kollisionswahrscheinlichkeit:** Einschätzung, ob bei gleichbleibender Fahrtrichtung und Geschwindigkeit eine Kollision droht.
- **Zeit bis zur Kollision (TTC):** Berechnung der verbleibenden Zeit bis zum möglichen Aufprall.

#### 4. Fahrerwarnung:

- **Warnstufen:** Je nach Dringlichkeit können mehrere Warnstufen durchlaufen werden.
  - **Visuelle Warnung:** Anzeige im Kombiinstrument oder Head-up-Display.
  - **Akustische Warnung:** Warnsignale oder Sprachhinweise.
  - **Haptische Warnung:** Vibration im Lenkrad oder kurzer Bremsruck.
- **Ziel:** Den Fahrer zur eigenständigen Reaktion animieren.

#### 5. Vorbereitung des Bremssystems:

- **Vorkonditionierung:** Erhöhung des Bremsdrucks, um eine schnellere Reaktion zu ermöglichen.
- **Empfindlichkeit:** Anpassung der Bremsempfindlichkeit, um bei Betätigung sofort maximale Bremswirkung zu erzielen.

## 6. Automatischer Bremsengriff:

- **Teilbremsung:** Falls der Fahrer nicht reagiert, wird zunächst eine automatische Teilbremsung eingeleitet, um die Geschwindigkeit zu reduzieren.
- **Vollbremsung:** Bei weiterhin drohender Kollision erfolgt eine automatische Vollbremsung.
- **Bremskraftsteuerung:** Das System passt die Bremskraft dynamisch an die Situation an.

Somit ist eine präzise Umfelderkennung essenziell, damit das System eine zuverlässige Aussage darüber treffen kann, wie kritisch eine Situation ist. Eine Kombination verschiedener Sensortechnologien bietet somit die höchste Zuverlässigkeit. In der Praxis werden hierbei häufig Kamerasensoren, welche besonders gut für die Klassifikation von Objekten geeignet ist, mit entfernungsmessenden Verfahren, wie Radar kombiniert, um präzise Informationen über die Objektklasse, sowie den Abstand des Objektes zu generieren. Die Gefahrenanalyse und die Entscheidung über die Notwendigkeit eines Bremsengriffs des Systems wird dann hochdynamisch anhand von Entfernungs- und Geschwindigkeitsrelationen getroffen.

Dieses Verhältnis wird mithilfe der sogenannten TTC beschrieben und wird zur Ermittlung der Kollisionsgefahr genutzt. Die TTC beschreibt die Zeit, die bis zur Kollision verbleibt, wenn sich beide Kollisionsgegner mit unveränderter Bewegungsrichtung und -geschwindigkeit fortbewegen und berechnet sich nach Formel 2.2.

$$TTC = \frac{d}{v_r} \quad (2.2)$$

$d$  : Relativer Abstand zwischen Fahrzeug und Hindernis (in Metern)

$v_r$  : Relative Geschwindigkeit zwischen Fahrzeug und Hindernis (in m/s)

Die TTC stellt somit ein direktes Verhältnis des Abstands zweier Kollisionsgegner im Verhältnis der relativen Geschwindigkeit zueinander dar.

Für diese TTC werden Schwellwerte festgelegt, anhand derer das System entscheidet, ob eine Warnung, eine leichte oder sogar eine Vollbremsung ausgelöst wird [22] [23] [1].

Dabei gibt es drei charakteristische Bereiche der TTC:

- ca.  $1,0\text{s} < TTC < \text{ca. } 1,6\text{s}$ : Ausweichmanöver möglich, aber gefährlich
- ca.  $0,6\text{s} < TTC < \text{ca. } 1,0\text{s}$ : Ausweichen physikalisch möglich, vom Fahrer praktisch nicht mehr geleistet
- $TTC < \text{ca. } 0,6\text{s}$ : Ausweichen physikalisch nicht mehr möglich

Somit kann sich zur Festlegung von Schwellwerten zur automatisierten Bremsung an diesen Werten orientiert werden, da bei TTC-Werten kleiner  $1,0\text{s}$  nicht mehr zwangsläufig davon ausgegangen werden kann, dass der Fahrer eine Kollision eigenständig vermeidet [4][1].

Eine besondere Schwierigkeit stellen ungeschützte Verkehrsteilnehmer, sogenannte VRU, dar, da diese zum einen deutlich schneller schwere Verletzungen von einer Kollision davontragen, als auch deutlich agiler sind als motorisierte Verkehrsteilnehmer. Fußgänger können beispielsweise sehr plötzlich ihre Bewegungsrichtung ändern und unerwartet auf die Fahrbahn treten. Daher ist eine hohe Abtastrate der Sensorik, sowie eine schnelle Verarbeitung der Daten notwendig, um in Echtzeit reagieren zu können. Echtzeit beschreibt in der Datenverarbeitung die Fähigkeit eines Systems, Eingaben (z.B. Sensordaten) innerhalb einer streng definierten und konstant kurzen Zeitspanne zu verarbeiten und entsprechende Ausgaben zu generieren. Dabei unterscheidet man zwischen harter und weicher Echtzeitanforderung. Während ein System, welches der harten Echtzeitanforderung entspricht, garantiert wird, dass eine festgesetzte Reaktionszeit niemals überschritten wird, wird bei der weichen Echtzeit das Erreichen des festgesetzten Wertes nur statistisch garantiert. Das bedeutet, dass beispielsweise der Mittelwert der Verarbeitungszeiten dem festgesetzten Wert entspricht und somit Schwankungen zulässig sind, solange das Ziel erreicht wird. Da ein Notbremssystem ein sicherheitskritisches System darstellt, wo die Verarbeitungszeit eine entscheidende Rolle bei der Vermeidung einer Kollision spielt, unterliegen diese Systeme der harten Echtzeitanforderung.

### **Monokulare Notbremssysteme**

Während der Trend bei sämtlichen Fahrerassistenzsystemen, die auf eine Umfelderkennung angewiesen sind, stark in die Richtung der Sensorfusion geht, wobei möglichst viele Sensortechnologien kombiniert werden, um ein noch detaillierteres und zuverlässigeres Abbild der Umwelt zu erschaffen, gibt es dennoch Hersteller, die bei ihren Notbremssystemen auf reine Kamerasensorik setzen. Zu nennen ist hierbei Tesla, die die Philosophie verfolgen sämtliche automatisierte Fahrfunktionen lediglich auf einem Netzwerk von Kameras umzusetzen. Tesla verbaut dabei jedoch alleine schon drei Frontkameras, welche für unterschiedliche Reichweiten ausgelegt sind. Weiterhin bietet der Automobilhersteller Subaru mit 'Eye-Sight' ein Notbremssystem, sowie weitere Assistenzsysteme, welche lediglich die Daten einer Stereo-Kamera nutzen. Kia bietet mit dem 'Forward Collision-Avoidance Assist (FCA) (Front camera only)' ein Notbremssystem an, welches basierend auf einer Monokamera operiert, dieses funktioniert allerdings ausdrücklich nicht für Fußgänger und Fahrradfahrer. Das Unternehmen mobileye gilt seit langem als führend im Bereich der bildbasierten Fahrerassistenzsysteme und setzt dabei auf innovative Computer-Vision Technologien und modernste KI-Ansätze.

Das Unternehmen liefert dabei, mit ihrer EyeQ-Serie, für viele Automobilhersteller und Zulieferer System-on-Chip-Lösungen für Frontkameras. Zusätzlich hat das System mit dem MobilEye 630 eine neuartige und innovative Nachrüstlösung entwickelt, welche über eine monokulare Kamera eine Vielzahl an Fahrerassistenzsystemen in so gut wie jeder Fahrzeugklasse und -baujahr ermöglicht. Das System liefert dabei auch eine Möglichkeit der Fußgängererkennung, allerdings warnt dieses System lediglich visuell und akustisch, ein

Bremseingriff erfolgt nicht. Obwohl das Unternehmen die fortschrittlichsten Lösungen für monokulare Smartkameras im Advanced Driver Assistance Systems (ADAS)-Segment bietet, werden diese zumindest im Bereich der Notbremssysteme immernoch mit Radar- oder LiDAR-Sensorik oder einer zweiten Kamera für Stereo-Vision kombiniert. Monokulare Systeme, welche ohne weitere Sensorik den heutigen Anforderungen eines AEBS entspricht, sind somit keine gängige Lösung, welche in der Praxis Anwendung findet.

---

## 3. Vorbetrachtungen und Konzept

In diesem Kapitel sollen zunächst erste Überlegungen zur Umsetzung und zu den bestehenden Möglichkeiten erläutert werden.

Konkret sollen die folgenden Ziele erreicht werden:

### 3.1. Konzeptentwicklung

Folgend sollen die notwendigen Teilschritte und Methoden zum Erreichen der festgesetzten Ziele dieser Arbeit aufgezeigt werden.

Um eine Personenerkennung für ein Notbremssystem über eine monokulare Kamera zu realisieren ist zunächst die Auswahl geeigneter Sensorik, in Form einer Kamera mit entsprechender Kameraoptik notwendig, um eine präzise Umfelderkennung zu gewährleisten. Diese Daten müssen anschließend in verschiedenen Prozessen weiterverarbeitet werden, um aus den Rohdaten wichtige Informationen zu erhalten, weshalb ein echtzeitfähiger Algorithmus erstellt werden muss, welcher all diese Prozesse steuert und miteinander kombiniert. Dazu ist eine zuverlässige Objekterkennung und -klassifikation notwendig, um Personen innerhalb des Bildes zu erkennen und zu lokalisieren. Anschließend müssen Positionsinformationen über die Personen gewonnen werden. Da eine Kamera keine entfernungsmessende Sensorik, wie beispielsweise ToF-Sensorik, wie Radar oder LiDAR, darstellt, muss eine Möglichkeit geschaffen werden anhand der zur Verfügung stehenden Sensordaten der Kamera Tiefeninformationen zu generieren. Um eine Entscheidungsfindung des Systems zur Warnung oder für einen Bremsenriff zu realisieren, muss zusätzlich ein Bezug zur relativen Geschwindigkeit zwischen Zielobjekt und Fahrzeug geschaffen werden, weshalb Informationen zur Fahrzeuggeschwindigkeit in den Algorithmus einfließen müssen. Anschließend ist eine Logik zu implementieren, welche anhand der vorhandenen Informationen der Abstands- und Geschwindigkeitsreaktionen, eine Entscheidung darüber trifft, inwiefern das System in Form einer Warnung oder eines Bremsenriffes, zur Vermeidung einer Kollision reagiert. Dazu müssen die genannten Prozesse in einer sinnvollen Reihenfolge nacheinander ablaufen.

Der Aufbau des Algorithmus sollte demnach grob entsprechend der in Darstellung 3.1 abgebildeten Logik erfolgen, in der Zunächst die Bilddaten von der Kamera abgerufen werden, bevor die weiterverarbeitenden Prozesse, wie Detektion und Lokalisierung der Personen, stattfinden und zusätzlich die Fahrzeuggeschwindigkeit mit in den Algorithmus einfließen muss, bevor eine Entscheidung getroffen werden kann und entsprechend, wenn notwendig, ein Befehl für die Warnung oder die Ansteuerung der Bremse ausgegeben wird. Dieser Prozess läuft dabei, wie zu erkennen, in einer Schleife ab, die wiederum damit beginnt, dass neue, aktualisierte Bilddaten in den Algorithmus mit einfließen und diese ausgewertet werden. Da der Algorithmus möglichst konfigurierbar gestaltet werden soll und die Aufgabenstellung einen Vergleich verschiedener Algorithmen verlangt, sollen einzelne Bausteine des Algorithmus, allem voran die Objektdetektion, austauschbar sein.

Um den Algorithmus zu validieren und die verschiedenen Konfigurationen miteinander zu

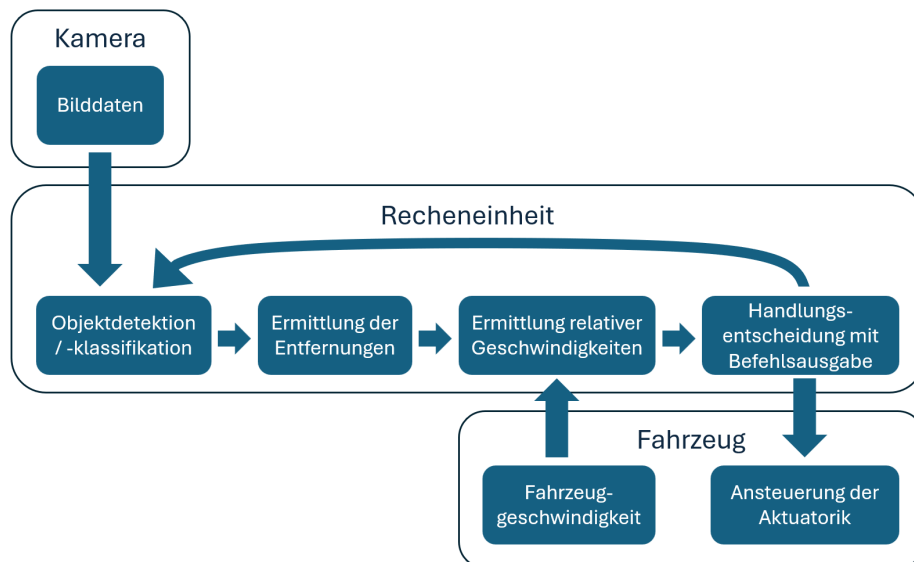


Abbildung 3.1.: Konzeptionelle Darstellung des Informationsflusses und der Verarbeitungsprozesse des umzusetzenden Algorithmus

vergleichen, sollen letztendlich Fahrversuche durchgeführt werden, um das System im Kontext nachgestellter Szenarien, innerhalb der Fahrzeugumgebung zu testen. Vergleichbare Größen sind hierbei die Schnelligkeit des Systems, die zeitgerechte Auslösung anhand zuvor definierter Bedingungen, sowie das Überprüfen des Systems auf unerwünschte Fehlauflösungen.

Aufgrund der aktuellen Relevanz wird abschließend ein Vergleich der Implementierbarkeit von kameragestützter gegenüber LiDAR-basierter Personenerkennung durchgeführt. Außerdem wird die Eignung beider Sensortechnologien für die Anwendung in einem Notbremsystem zur Personenerkennung analysiert.

## 3.2. Hardware

### 3.2.1. Auswahl der Kamera-Hardware

Die Kamera für die Umsetzung der Personenerkennung sollte eine möglichst weitreichende Parametrierbarkeit und Konfigurierbarkeit bieten, um möglichst viel Einfluss auf die aufgenommenen Bilddaten zu erhalten und diese den Anforderungen dieser Umsetzung bestmöglich anzupassen. Das erspart rechenintensive Nachbearbeitungsprozesse und bietet deutlich mehr Nachvollziehbarkeit, sowie Optimierungsmöglichkeiten in der weiteren Umsetzung. Sogenannte Industriekameras zeichnen sich durch eben dieses hohe Maß an Parametrierbarkeit aus und bieten dabei eine umfassende Auswahl verschiedener Schnittstelle, Auflösungen und Bildwiederholraten, sowie eine hohe Kompatibilität mit verschiedensten Programmiersprachen. Da es für diese Umsetzung keine spezifischen Limitierungen durch den Bauraum gibt, macht sich diese Kamerakategorie somit besonders bedeutsam für diesen Anwendungsfall. Da die meisten Objektdetektoren auf Farbbildern trainiert sind und somit auch viele wichtige Bildmerkmale aus den Farbkanälen entnehmen, soll die Kamera ebenfalls Farbinformatio-

nen liefern können, um das Potential der Detektoren möglichst gut auszuschöpfen. Dabei ist jedoch zu beachten, dass Farbbilder bei gleicher Auflösung ein Vielfaches an Datenmenge umfassen, was mehr Rechenressourcen in Anspruch nimmt. Daher ist es wichtig die Kameraauflösung so zu wählen, dass sie so niedrig wie möglich ist, jedoch immer noch hoch genug, um präzise Erkennungen zu ermöglichen. Bei der Auslegung einer Front View-Kamera spielt vor allem das Sichtfeld eine große Rolle, welches von der Brennweite des Objektivs abhängt. Besonders das horizontale Sichtfeld sollte möglichst groß gewählt werden, um auch Personen, welche nah vor dem Fahrzeug auf die Fahrbahn geraten, rechtzeitig erfassen zu können. Hierbei sind Werte von mindestens 40 Grad erforderlich, in der Praxis sind hier Werte zwischen 50 und 110 Grad verbreitet [4] [24] [25]. Es ist wichtig, das Verhältnis zwischen Sensorauflösung und dem FOV, also dem Sichtfeld, zu berücksichtigen, da bei gleichbleibender Auflösung eine Zunahme des Sichtfelds zu einer geringeren Detailauflösung von Objekten in einer bestimmten Entfernung führt. Daher sollte die Auflösung in Pixel pro Grad, die sogenannte Winkelauflösung, berücksichtigt werden, um eine angemessene Detailgenauigkeit sicherzustellen. Für eine Kennzeichenerkennung sind hier beispielsweise Werte größer als 15 Pixel pro Grad erforderlich [4]. Da Personen zwar größer sind, aber aus einer deutlich höheren Entfernung sicher erkannt werden müssen, sollte dieser Wert daher als Mindestanforderung betrachtet werden.

Ein Weiteres Kriterium bei der Auswahl der Kamera stellt die Bildwiederholrate, gemessen in Herz (Hz), bzw. frames per second (fps) dar. Dabei ist eine möglichst hohe zeitliche Auflösung anzustreben, um möglichst schnell auf kritische Situationen reagieren zu können. Wie bei der räumlichen Auflösung gilt jedoch auch hier, dass mit ansteigender Auflösung die Datenmenge und somit die Anforderungen an die weiterverarbeitende Hardware, sowie die Algorithmen steigt. Eine zeitliche Auflösung von etwa 30 bis 40 fps sollte für eine solche Anwendung angestrebt werden [24] [4]. Die Kameraschnittstelle, welche die Kamera mit der Recheneinheit verbindet ist ebenfalls ein Kriterium, welches berücksichtigt werden sollte. Die Schnittstelle muss sowohl eine ausreichende Übertragungsbandbreite bieten, als auch eine möglichst universelle Anbindung an die, in der Lehre verwendeten, Hardware ermöglichen. Die Datenübertragung über Ethernet bietet einige Vorteile gegenüber anderen Übertragungstechnologien, wie beispielsweise USB, FireWire oder CameraLink. Zu nennen sind hier die hohen Bandbreiten von bis zu 100 Gbit/s im Kamerabereich mit dem Gigabit-Ethernet (GigE)100-Standard, die langen Übertragungsentfernungen von bis zu hundert Metern ohne Signalverstärker, sowie die Nutzung vorhandener Ethernet-Infrastruktur in vielen industriellen Umgebungen, da dies ein sehr verbreiteter Standard ist, sowie die Integration mehrerer Kameras in ein Netzwerk [26]. Im Falle der Umsetzung einer Notbremsfunktion in einem experimentellen Rahmen sprechen vor allem die Zuverlässigkeit und Robustheit der Ethernet-Datenübertragung, was ideal für die Verwendung in kritischen Anwendungen ist, als auch das standardisierte Internet-Protokoll, sowie die Verfügbarkeit an geeigneten Kameras und der kompatiblen Hardware im MechLab der HTW Dresden. Ein weiterer Vorteil ist, dass die Kameras PoE unterstützen und damit sowohl die Datenübertragung als auch

die Stromversorgung über ein Kabel realisiert werden können.

Die Auswahl fiel hierbei auf eine bereits im MechLab vorhandene GigE-Vision Industriekamera des Herstellers Basler, welche den zuvor definierten Anforderungen entspricht. Diese Kamera trägt die Modellbezeichnung Basler ace acA 1920-40gc, welche sich folgendermaßen entschlüsselt:

Tabelle 3.1.: Entschlüsselung der Bezeichnung der Basler acA1920-40gc

Bezeichnung	Bedeutung
<b>acA</b>	ac = steht für die Produktreihe ace A = Area-Scan-Camera (Flächenkamera mit Global Shutter-Technologie)
<b>1920</b>	Horizontale Standardauflösung der Kamera in Pixeln 1920 Pixel.
<b>40</b>	Maximale Bildrate der Kamera in den Standardeinstellungen.
<b>gc</b>	g = GigE-Schnittstelle c = Color (Farbkamera)

Speziell gehört diese Kamera der Produktfamilie ace U von Basler an, welche sich durch eine überarbeitete, moderne Sensorpalette auszeichnet. Die Kamera hat einen Sony IMX249 CMOS-Sensor verbaut, welcher mit einer maximalen Auflösung von 1936 × 1216 Pixeln bis zu 42 Bilder pro Sekunde liefert. Somit ergibt sich eine maximale Gesamtauflösung von rund 2,3 Megapixeln, welche mit bis zu einem Gigabit pro Sekunde übertragen werden können. Der Sensor arbeitet nach dem Global Shutter-Prinzip, wobei alle Pixel der Sensorfläche gleichzeitig belichtet werden, wodurch sich die Kamera besonders für schnell bewegte Szenen sehr gut eignet. Der Kamerasensor hat eine Größe von 11,33 mm × 7,13 mm. Anhand der Sensorgröße lässt sich nun die erforderliche Brennweite bestimmen, um sowohl die Anforderungen des horizontalen Sichtfeldes, als auch der räumlichen Auflösung zu erfüllen. Das horizontale Blickfeld  $FOV_{horizontal}$  lässt sich nach folgender Formel berechnen:

$$FOV_{horizontal} = 2 \times \arctan\left(\frac{w_{sensor}}{2 \times f}\right) \quad (3.1)$$

Dabei ist:

- $w_{sensor}$  - Breite des Bildsensors
- $f$  - Brennweite in mm

Stellt man die Formel nun nach der gesuchten Brennweite  $f$  um und setzt für  $FOV_{horizontal}$  den gewünschten mindestwert von 50°, sowie die gebene Sensorbreite  $w_{sensor}$  von 11,33 mm ein, so erhält man:

$$f = \frac{11.33 \text{ mm}}{2 \times \tan\left(\frac{50^\circ}{2}\right)} \approx \frac{11.33}{0.9326} \approx 12.15 \text{ mm} \quad (3.2)$$

Um ein horizontales Sichtfeld von mindestens 50° zu erhalten ist bei gegebener Sensorgröße somit eine Brennweite von höchstens 12 mm zu wählen. Dabei ist zu beachten, dass mit steigender Brennweite das Sichtfeld abnimmt und geringere Brennweiten zu einem größeren Blickwinkel der Kamera führen. Als Kameraoptiken kamen daher zwei vorhandene Objektive



in Frage, welche diesem Wert recht nahe kommen. Eines mit 16 mm Brennweite, wofür sich nach Formel 3.1 ein horizontales Sichtfeld von etwa  $37,1^\circ$  ergibt und ein deutlich weitwinkeliges mit 8 mm Brennweite, welches ein Sichtfeld von etwa  $70,4^\circ$  ermöglicht [27] [28].

Da nun die horizontalen Blickwinkel beider Objektive bekannt sind, lassen sich unter Beachtung der horizontalen Auflösung des Sensors von 1920 Pixeln, die Winkelauflösungen beider Brennweiten berechnen ganz einfach durch Division berechnen:

#### **Horizontale Pixel pro Grad für $f = 8$ mm:**

$$P_{\text{horiz}}(f = 8 \text{ mm}) = \frac{1920}{70.4} \approx 27.27 \text{ Pixel/Grad} \quad (3.3)$$

#### **Horizontale Pixel pro Grad für $f = 16$ mm:**

$$P_{\text{horiz}}(f = 16 \text{ mm}) = \frac{1920}{37.1} \approx 51.75 \text{ Pixel/Grad} \quad (3.4)$$

Beide Objektive gewährleisten somit in Kombination mit der gewählten Kamera eine horizontale Winkelauflösung, welche hoch genug sein sollte, um auch in größerer Entfernung eine Person so hoch aufzulösen, dass eine Detektion umsetzbar ist. Da beide Objektive geeignet sind, sollen auch beide Objektive in der Umsetzung berücksichtigt werden und ein einfacher Wechsel der Brennweiten im Algorithmus ermöglicht werden.

#### **3.2.2. LiDAR-Sensor**

Zur Aufnahme der Punktwolken der Versuchsfahrten soll ein LiDAR-Sensor des Herstellers Ouster zum Einsatz kommen. Dieser Sensor trägt die Bezeichnung Ouster OS1-64, wobei 'OS1' für die Produktlinie 'Ouster Sensor 1' steht und '64' für die Anzahl an vertikalen Scanebenen bzw. Kanälen, die der Sensor erfassen kann. Der Sensor ist ein rotierender LiDAR-Sensor, welcher seine Umgebung im  $360^\circ$  Winkel erfassen kann, wobei das vertikale FOV bis zu  $45^\circ$  abdeckt.

Der Sensor hat eine Reichweite von etwa 110 Metern und erfasst seine Umgebung mit Drehraten von 10 oder 20 Hz, wobei Auflösungen von 512, 1024 oder 2048 Pulsen pro Umdrehung pro Kanal möglich sind. Somit können bis zu 1.310.720 Messpunkte pro Sekunde erfasst werden. Mit dieser Auflösung und Reichweite ist der Sensor der Klasse der Mid-Range-High-Resolution-Sensoren zuzuordnen.

### **3.3. MATLAB**

Da ein offenes und konfigurierbares System geschaffen werden soll, welches in der Lehre und für eigene Forschungsprojekte im MechLab an der HTW Dresden gedacht ist, war es von hoher Relevanz, eine Softwareumgebung zu nutzen, welche im MechLab und vor allem für die Studierenden frei zugänglich ist. Die Wahl fiel hierbei auf die Plattform MATLAB

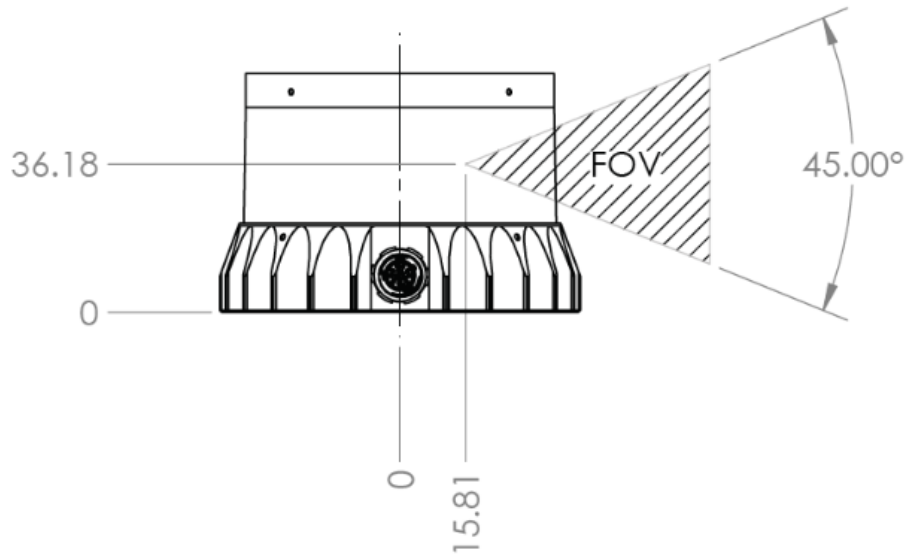


Abbildung 3.2.: Sichtbereich des Ouster OS1-64 [29]

(Matrix Laboratory) der Firma MathWorks. Für Studierende der HTW Dresden besteht der kostenfreie Zugang zu dieser Software durch eine Campus-Lizenz, da diese bereits fester Bestandteil in der Lehre des Studienganges Fahrzeugtechnik ist. MATLAB ist ebenfalls in der freien Wirtschaft eine etablierte Umgebung, welche zur Entwicklung und Simulation von neuen Systemen genutzt wird. In MATLAB stehen viele themenspezifische vorgefertigte Funktionspakete in Form von Toolboxen zur Verfügung und für viele relevante Themengebiete stehen seitens MathWorks Lernkurse und Beispielumsetzungen zur Verfügung, welche die Nachvollziehbarkeit in der Lehre zusätzlich stark fördert. MATLAB wurde ursprünglich entwickelt, um mit Matrizen und Arrays zu arbeiten, was es besonders geeignet für lineare Algebra und numerische Berechnungen macht. Zusätzlich bietet MATLAB leistungsstarke Tools zur grafischen Darstellung der Ergebnisse, sowohl in 2D, als auch in 3D. Die Programmiersprache in MATLAB gilt als benutzerfreundlich und verhältnismäßig leicht zu erlernen. Der Code lässt sich in Funktionen und Skripte verpacken, welche in einen Workflow eingebunden werden können. Zusätzlich ist Matlab über vorgefertigte Compiler mit sämtlichen gängigen Programmiersprachen, sowie dem Robot Operating System (ROS) kompatibel und kann auf verschiedenster Hardware, wie Microcontrollern und verschiedenen Embedded Systems angewandt werden. Obwohl es geeigneter Softwareumgebungen für die Anwendung in sicherheitskritischen Systemen mit harter Echtzeitanforderung gibt, da MATLAB zum einen eine Programmiersprache ist, welche während der Ausführung zunächst interpretiert werden muss und somit für Latenzen sorgt und einen deutlich höheren Ressourcenverbrauch hat, als kompakte, systemnahe Sprachen, eignet sich MATLAB besonders durch die zuvor genannten Gründe vor allem für das Prototyping von Algorithmen in der Lehre, welche auch in späteren Entwicklungsstadien noch in echtzeitfähige Sprachen überführt werden können [30]. Für die Umsetzung dieser Arbeit wurde MATLAB in der Version R2024a verwendet. Eine ausführliche Überprüfung der Funktionalität der Algorithmen in der kürzlich erschie-

nen Version R2024b ist noch nicht erfolgt. In Matlab stehen verschiedene vorgefertigte Modelle zur Objekterkennung, speziell zur Personenerkennung bereit, welche verschiedene Herangehensweisen der Objektdetektion verfolgen. Besonders der `peopledetectorACF` und der `yolov4ObjectDetector` sollen aufgrund ihrer Verschiedenheit im Fokus dieser Umsetzung stehen. Diese Modelle sind vortrainiert und können somit ohne weiteren Aufwand in die Umsetzung implementiert werden [31], [32].

Die verschiedenen Modelle, sowie deren Funktionsweise sollen zum allgemeinen Verständnis zunächst etwas genauer beleuchtet werden.

## 3.4. Algorithmen zur Personenerkennung in MATLAB

Zur Detektion von Personen aus Bilddaten stehen in MATLAB verschiedene vorgefertigte Ansätze zur Verfügung, welche über die *Computer Vision Toolbox* bereits implementiert sind und zum Teil speziell für die Personenerkennung vortrainiert sind, als Multiobjektdetektoren für eine Vielzahl an Objektklassen trainiert sind und demzufolge noch für den spezifischen Anwendungsfall angepasst werden müssen oder lediglich ein Modell liefern, welches auf die eigenen Bedürfnisse trainiert werden muss, bevor man diese anwenden kann. Die Detektoren, welche für diese Umsetzung am sinnvollsten erachtet wurden, sollen daher in Bezug auf ihre Funktionsweisen und Besonderheiten etwas näher beleuchtet werden.

### 3.4.1. ACF-Detektor

Der Aggregated Channel Features (ACF)-Detektor ist eine Objekterkennungsmethode, welche spezifische Objekte innerhalb eines Bildes anhand von sogenannten Kanalmerkmalen detektiert und die Position des Objektes im Bild vorhersagen kann. Hierzu werden verschiedene Informationskanäle des Bildes, wie beispielsweise der Luminanzkanal, welcher Informationen über die Helligkeit in Form von Graustufen liefert, die verschiedenen Farbkkanäle, sowie daraus entstehende Gradientengrößen und Gradientenausrichtungen und deren Histogramme, aus welchen auf Objektkanten innerhalb des Bildes geschlossen werden kann, untersucht und daraus sogenannte Merkmale interpretiert. Bei den aggregierten Kanalmerkmalen handelt es sich um eine Weiterentwicklung der integrierten Kanalmerkmale oder auch Integralbilder. Die Merkmale beschreiben hierbei Informationen, welche bestimmte Eigenschaften, Strukturen oder Muster des Bildes hervorheben. Diese Merkmale werden für kleinere Pixelbereiche des Bildes zusammengefasst und als aggregierte Merkmalsvektoren zwischengespeichert, wodurch eine recht hohe Einsparung an Rechenleistung gewonnen wird (Abb. 3.3).

Ein Fenster vordefinierter Größe, welche an das zu detektierende Objekt angepasst ist, bei Personen könnte dieses beispielsweise  $64 \times 128$  Pixel umfassen, wird dann systematisch in kleinen Abständen über das Bild geschoben und die zuvor zwischengespeicherten Merkmalsvektoren derjenigen Pixelbereiche, die in diesem Fenster liegen, auf das Vorhandensein eines Objektes untersucht. Diese Methode wird auch als „Sliding Window“ (engl.

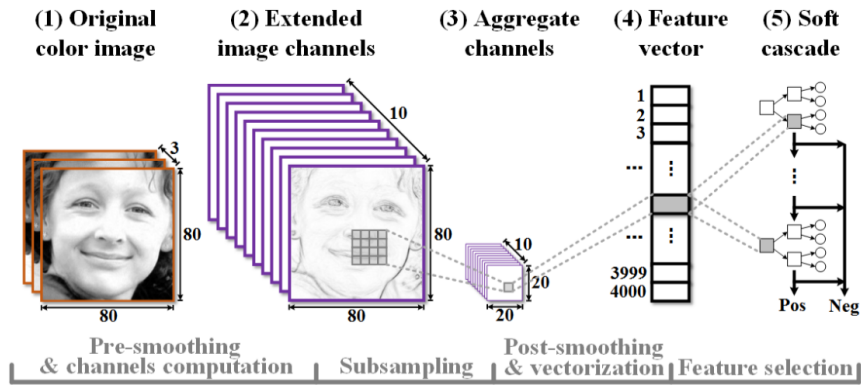


Abbildung 3.3.: Workflow eines ACF-Detektors [33]

Schiebefenster oder Gleitfenster) bezeichnet. Die Entscheidung, ob sich in diesem Bildbereich ein zu detektierendes Objekt befindet, trifft ein auf den spezifische Anwendungsfall zugeschnittener Klassifikator, welcher auf Basis von Maschinellern trainiert wurde, wie beispielsweise eine Support Vector Machine (Support Vector Machine (SVM)) oder ein Ada-Boost-Klassifikator, welche alle zuvor generierten Kanalmerkmale mit bekannten Merkmalsanhäufungen aus dem Training vergleicht und eine Aussage darüber trifft, mit welcher Wahrscheinlichkeit sich in diesem Bildbereich ein relevantes Objekt befindet. Ein Schwellwert hilft hierbei Falschdetektionen zu vermeiden.

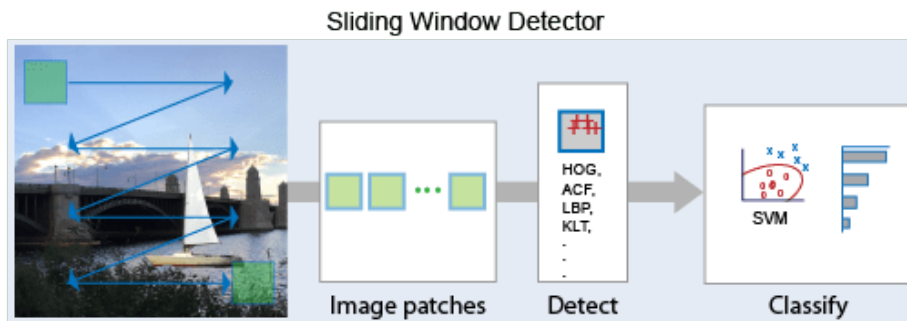


Abbildung 3.4.: Funktionsweise der Sliding Window-Methode [34]

Da das Schiebefenster oftmals mit recht kleinen Schrittweiten über das Bild bewegt wird, besteht die Möglichkeit, dass sich die untersuchten Bildbereiche überlappen und verschiedene Ausschnitte desselben Objekts mehrfach vom Schiebefenster erfasst werden. Dies führt potenziell zu mehreren Detektionsaussagen ein und desselben Objekts, da die anteiligen Merkmalsvektoren somit für das Vorkommen eines relevanten Objekts im Fenster sprechen. Um diese redundanten Erkennungen zu verhindern, insbesondere bei einer starken Anhäufung von hohen Wahrscheinlichkeitswerten, wird eine Konsolidierung vorgenommen. Dabei wird ausschließlich diejenige Position des Fensters als tatsächliches Objekt anerkannt, die den höchsten Wahrscheinlichkeitswert aufweist. Diese Vorgehensweise minimiert die Wahrscheinlichkeit von Mehrfachdetektionen und verbessert somit die Präzision der Objekterkennung, macht es dem Detektor jedoch sehr schwer mehrere nah beieinander liegende

Objekte zu erkennen. Durch die Sliding Window-Methode kann außerdem eine Aussage über die genaue Position der detektierten Objekte im Bild getroffen werden. Da dieses Schiebefenster jedoch eine vordefinierte Größe aufweist, werden meist nur Objekte, welche annähernd die Größe des Fensters haben erkannt. Objekte, die weit vorn erscheinen, könnten sehr viel mehr Bildfläche einnehmen, als das Schiebefenster erfassen kann, während Objekte, welche sich sehr weit im Hintergrund befinden, nur sehr klein in dem Bild erscheinen. Beides kann bewirken, dass die Merkmale in diesem Bildbereich dadurch statistisch nicht den Merkmalen entsprechen, welche dem Klassifikator für das spezifische Objekt aus dem Training bekannt sind und damit keine korrekte Detektion der Objekte erfolgt. Um diese Problematik zu umgehen ist es in der Praxis üblich die zu verarbeitenden Eingabebilder in verschiedene Größen zu skalieren. Zwar könnte man auch das Schiebefenster in verschiedene Größen skalieren, was als „Multi-Scale Sliding Window“ (engl. Mehrskaliges Gleitfenster) bekannt ist, jedoch ist diese Methode meist deutlich rechenintensiver, da für jede Position und Skalierung des Fensters die Merkmale neu berechnet werden müssten. Um Objekte verschiedener Größen in Bildern zu detektieren, werden daher die Bilder in verschiedene Skalen transformiert und sogenannte Bildpyramiden erstellt. Aus diesen Bildpyramiden werden dann jeweils die Merkmale für jedes skalierte Bild aus den verschiedenen Kanälen extrahiert und in sogenannten „Feature Pyramids“, also Merkmalspyramiden gesammelt. Diese Pyramiden bestehen aus mehreren Ebenen, wobei jede Ebene die Merkmale eines bestimmten Bildmaßstabs enthält. Bei dieser Methode werden für jede Bildgröße die Merkmale nur ein einziges Mal berechnet und diese lediglich mithilfe der Gleitfenster-Methode für jeden Schritt mit den Trainingswerten „verglichen“. Dadurch lässt sich, ähnlich wie bei der bereits beschriebenen Mehrfachdetektion durch geringe Schrittweiten, auch die mehrfache Merkmalsberechnung der gleichen Stelle vermeiden, wodurch bessere Laufzeiten der Objekterkennung realisiert werden können. Da die Neuberechnung eines Bildes in verschiedene Auflösungen, sowie die Merkmalsauswertung der verschiedenen Skalierungen dennoch ein sehr rechenintensiver Prozess ist und somit auch kostbare Laufzeit in Anspruch nimmt, gab es dennoch Bedarf diesen Arbeitsschritt noch weiter zu beschleunigen. In Ihrem weitverbreiteten und vielzitierten Paper „Fast Feature Pyramids for Object Detection“ [35], von 2014, stellen die Wissenschaftler Piotr Dollar et al. sowohl einen Ansatz zur effizienteren Berechnung der Merkmalspyramiden vor, der auch heute noch in praktischen Anwendung Einsatz findet, sowie weitere Methoden, die zur Beschleunigung von Objekterkennungsalgorithmen beitragen. Die Erkenntnisse, die aus diesem Forschungsprojekt hervorgehen, haben die Entwicklung im Bereich der echtzeitfähigen Objekterkennung zu seinerzeit drastisch vorangetrieben. Die Autoren erläutern zunächst, wie sogenannte „übereinstimmende“ oder „reichhaltigere“ Bilddarstellungen, welche mehrfache Auflösungen, sowie mehrere Orientierungen eines Bildes enthalten, wie beispielsweise die Bildpyramiden, die Genauigkeit und Robustheit der Objektdetektoren drastisch erhöht. Diese Erhöhung der Robustheit und besonders der Genauigkeit der Detektoren ist jedoch zuvor immer zulasten des Rechenaufwandes und somit auch der Schnelligkeit dieser Detektoren gefallen. Die Schreiber erklären

weiterhin, dass die Verbesserungen der Genauigkeit nicht zwangsläufig mit Abstrichen in der Geschwindigkeit einhergehen müssen, da es zum Erstellen einer feinabgetasteten Merkmalspyramide nicht zwingend nötig ist, für jede Ebene der Pyramide die Merkmale einzeln aus den Bildern zu extrahieren. Aus Versuchen und Analysen der Forscher geht hervor, dass Bilder sogenannte fraktale Statistiken besitzen, die genutzt werden können, um bestimmte Strukturen und Bildmuster über verschiedene Skalen hinweg zuverlässig vorherzusagen. Es stellte sich heraus, dass der Informationsgehalt eines hochgerechneten Bildes, also eines Bildes dessen Auflösung künstlich erhöht wird, konstant bleibt, während der Informationsgehalt, vor allem bei hochfrequenten und feinstrukturierten Inhalten, mit fortschreitender Reduktion der Auflösung sinkt. Dieser Informationsverlust ist jedoch konsistent und folgt dabei einem Potenzgesetz, unter dessen Berücksichtigung sich die Merkmalsstatistiken auch für geringe Auflösungen extrapolieren lassen.

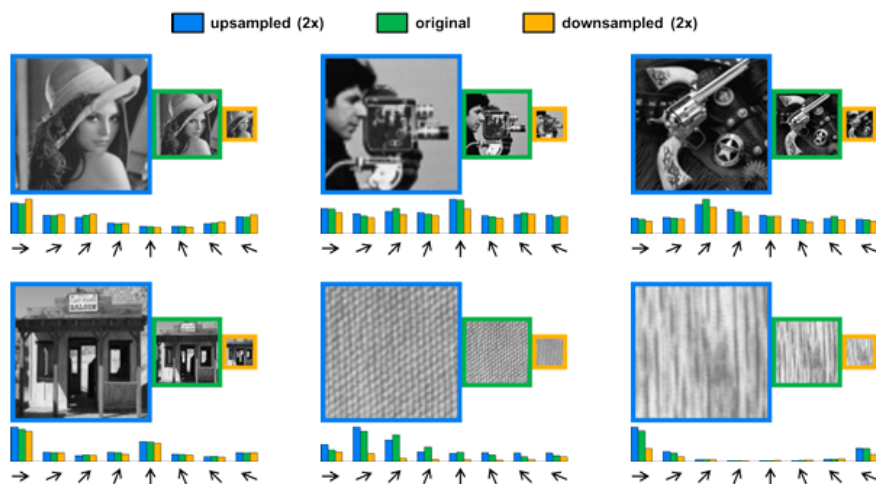


Abbildung 3.5.: Gradientenhistogramme verschiedener Skalen [35]

Abbildung 3.5 stellt die Histogramme der verschiedenen Gradientenausrichtungen der Bilder in den verschiedenen Skalen dar. Zu erkennen ist hier die abweichende Merkmalsverteilung der runtergerechneten Bilder in den letzten beiden Bildern, welche eine atypische Verteilung der Gradientenhistogramme aufweisen. Durch die Anwendung dieses Ansatzes konnten sie die Anzahl der notwendigen Berechnungen für die Merkmalspyramiden erheblich reduzieren, was zu einer signifikanten Beschleunigung der Objektdetektion führte. Dies wird erreicht, indem die Merkmale nur für eine begrenzte Anzahl von Skalen explizit berechnet werden und die fehlenden Skalen durch eine schnelle und genaue Approximation ergänzt werden. Diese Methode ermöglicht es, die Rechenzeit zu minimieren, ohne die Genauigkeit der Detektion wesentlich zu beeinträchtigen. Der ACF-Detektor profitiert somit von diesen Optimierungen und stellt eine effektive Lösung für Echtzeit-Objekterkennungsaufgaben dar, bei denen sowohl Genauigkeit als auch Geschwindigkeit von entscheidender Bedeutung sind. Obwohl neuere Methoden auf tiefen neuronalen Netzen basieren, bleibt der ACF-Detektor aufgrund seiner Einfachheit und Effizienz in bestimmten Anwendungsbereichen relevant, insbesondere dort, wo begrenzte Rechenressourcen zur Verfügung stehen [35].

### 3.4.2. YOLO

Im Gegensatz zum zuvor beschriebenen ACF-Detektor, der handgestaltete Merkmale und klassische maschinelle Lernverfahren nutzt, basieren modernere Objekterkennungsmethoden, wie auch die Methode des You Only Look Once (YOLO)-Detektors, auf tiefen neuronalen Netzen. Der grundlegende Aufbau, sowie die Funktionsweise neuronaler Netze wurde bereits im Abschnitt 2.3 der theoretischen Grundlagen erläutert.

Die ursprünglichen regionbasierten CNN-Detektoren wie Region-Based Convolutional Neural Network (R-CNN) und dessen Weiterentwicklungen Fast R-CNN und Faster R-CNN, erstellen zunächst Regionenvorschläge über ein Region Proposal Network (RPN) und in einem zweiten Arbeitsschritt folgt die Klassifikation der Objekte in diesen Regionen, wodurch diese Detektoren als Two-Stage-Detektoren bezeichnet werden. YOLO führt die Objekterkennung in Form der Lokalisierung und Klassifizierung, in einem einzigen Schritt durch, was die Verarbeitungsgeschwindigkeit erheblich erhöht. Somit zählt YOLO zu den One-Stage-Detektoren

YOLO betrachtet das gesamte Bild auf einmal und teilt es in ein Gitter aus gleich großen Zellen auf. Jede Zelle des Gitters ist dafür verantwortlich, Vorhersagen darüber zu treffen, ob sich ein Objekt in ihr befindet, welche Klasse es hat und wo es sich genau befindet. Dabei verwendet YOLO sogenannte Ankerboxen (*Anchor Boxes*), die als vordefinierte Formen dienen, um die Bounding Boxes besser an verschiedene Objektgrößen und -formen anzupassen. Diese Ankerboxen ermöglichen es dem Modell, die Position und die Größe der Objekte präziser vorherzusagen, indem sie verschiedene Standardgrößen und -verhältnisse bereitstellen, die häufig in den Trainingsdaten vorkommen.

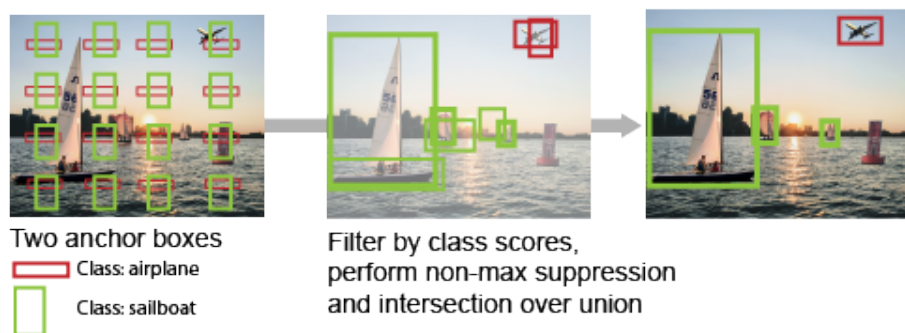


Abbildung 3.6.: Funktionsweise der Objektdetektion mithilfe von Ankerboxen [34]

Für diese Umsetzung soll speziell YOLOv4 angewendet werden, was für Version 4, also die vierte Generation des Detektors steht. Ein zentraler Bestandteil von YOLOv4 ist der Backbone des Netzwerks. Der Backbone kann als das Grundgerüst des neuronalen Netzes verstanden werden, das dafür zuständig ist, aus dem Eingabebild wichtige Merkmale und Muster zu extrahieren. In YOLOv4 wird hierfür standardmäßig CSPDarknet53 verwendet, ein effizientes CNN, das die Verarbeitungsgeschwindigkeit erhöht und gleichzeitig die Genauigkeit der Merkmalsextraktion verbessert. Der Backbone verarbeitet das Bild durch mehrere Schichten von Convolutional Layers, um abstrakte Merkmale zu erkennen, die für

die Objekterkennung notwendig sind. CSP steht bei der Netzbezeichnung für Cross Stage Partial, was eine Optimierung des Merkmalsflusses im CNN beschreibt, wobei ein Teil der extrahierten Merkmale an Netzschichten vorbeigeleitet werden und erst in späteren Schichten mit in das Netz einfließen, wodurch die Effizienz deutlich gesteigert werden kann. Diese Neuerung wurde mit dieser vierten Version von YOLO eingeführt.

Zusätzlich nutzt YOLOv4 eine Sammlung von speziellen Methoden und Techniken, die als 'Bag of Specials' bezeichnet werden. Diese beinhalten verschiedene Optimierungen und Tricks, die die Genauigkeit und Effizienz des Modells weiter verbessern. Wichtige Bestandteile des Bag of Specials sind:

- **Spatial Pyramid Pooling (SPP):** Diese Technik ermöglicht es dem Netzwerk, Merkmale aus unterschiedlichen Auflösungen zu kombinieren. Dadurch kann das Modell Objekte in verschiedenen Größen besser erkennen, da es Informationen aus verschiedenen Skalierungsstufen des Bildes berücksichtigt.
- **Path Aggregation Network (PANet):** Dieses Netzwerk verbessert den Informationsfluss zwischen den verschiedenen Ebenen des neuronalen Netzes. Es hilft dabei, feine Details aus den frühen Schichten mit abstrakteren Informationen aus späteren Schichten zu verbinden, was besonders bei der Erkennung kleiner Objekte von Vorteil ist.
- **Mosaic Data Augmentation:** Diese Methode kombiniert vier Trainingsbilder zu einem einzigen, wodurch das Modell robuster gegenüber unterschiedlichen Objektgrößen und -positionen wird.
- **Self-Adversarial Training (SAT):** Eine Technik, bei der das Modell sich selbst herausfordert, um die Erkennungsfähigkeiten zu verbessern und die Robustheit gegenüber schwierigen Szenarien zu erhöhen.

In MATLAB steht dieser YOLOv4 als potenter, vortrainierter Multiklassen-Detektor zur Anwendung bereit. Insgesamt stellt YOLOv4 noch heute einen bedeutenden Fortschritt in der Objekterkennung dar, indem es durch innovative Techniken die Leistung von One-Stage-Detektoren weiter verbessert. Die in YOLOv4 eingeführten Verbesserungen trugen dazu bei, die Lücke zwischen Geschwindigkeit und Genauigkeit weiter zu schließen und bieten eine robuste Grundlage für anspruchsvolle Echtzeitanwendungen, weshalb dieser Detektor noch heute sowohl in der Forschung, als auch in der Praxis breite Anwendung findet.

### 3.5. Vergleich der Objekterkennungsmethoden

Um die Unterschiede der zuvor beschriebenen Objekterkennungsalgorithmen durch den Vergleich rationaler Werte zu veranschaulichen und somit die geeignetste Methode für den hier beschriebenen spezifischen Anwendungsfall zu identifizieren, müssen zunächst aussagekräftige Vergleichswerte geschaffen werden. Um derartige Evaluationsmetriken zu erzeugen ist es



in der Praxis üblich, den Objektdetektor auf einem Bilddatensatz anzuwenden, für welchen auch die sogenannten Grundwahrheiten, (engl. Ground-Truth) vorliegen. Die Grundwahrheiten enthalten dabei zu jedem Objekt eines Bildes Positions- und Klasseninformationen. Diese Informationen liegen zumeist als Koordinaten der jeweiligen, zuvor festgelegten, Bounding-boxen vor, welche in einem manuellen oder bei größeren Datensätzen auch automatisierten Verfahren, dem sogenannten Labeln, den entsprechenden Objekten zugeordnet werden. Durch den Abgleich der Vorhersagen des Objektdetektors mit den gelabelten Ground-Truth-Daten lässt sich ableiten, ob der Detektor das Objekt korrekt erkannt hat und wie präzise er es im Bild lokalisiert. Zusätzlich ist es von Bedeutung zu analysieren, wie viele der vorhergesagten Boundingboxen einem tatsächlichen Objekt zugeordnet werden können und welche ohne tatsächliches Objekt an zufälligen Positionen im Bild liegen. Um diese Aspekte numerisch auszudrücken, wird zunächst die **Intersection over Union (IoU)** für jede Boundingbox berechnet. Diese Kennzahl beschreibt die Übereinstimmung einer vorhergesagten Boundingbox mit der Ground-Truth-Boundingbox. Genauer gesagt ist die IoU das Verhältnis zwischen der Überlappungsfläche (Area of Overlap) der beiden Boundingboxen und der Gesamtfläche, die durch beide Boxen zusammen umschlossen wird (Area of Union).


$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


Abbildung 3.7.: Berechnung der Intersection over Union [36]

Anhand eines selbst gewählten Schwellwerts für die IoU lässt sich nun eine genaue Aussage darüber treffen, ob eine Vorhersage des Detektors korrekt ist oder nicht. Wenn die IoU einer Vorhersage-Boundingbox mit einer Ground-Truth-Boundingbox also größer oder gleich dem Schwellwert ist, so bezeichnet man diese korrekte Vorhersage als **True Positive (TP)**. Desweiteren kann eine Aussage **False Positive (FP)** sein, was bedeutet, dass der Detektor ein Objekt erkannt hat, wo keines ist, also eine Fehldetektion. Wenn der Detektor ein tatsächlich vorhandenes Objekt nicht erkannt hat bzw. die Positionen aller Vorhersagen des Objektes so stark abweichen, dass keine Vorhersage den IoU-Schwellwert einer Grundwahrheit übersteigt, so nennt man das **FN! (FN!)**, also eine Nichterkennung. Zwar gibt es bei typischen binären Klassifikationsproblemen auch das Kriterium True Negative (TN), jedoch machte es in der Objekterkennung bei den meisten Anwendungen keinen Sinn, zu bestimmen, wo der Detektor korrekterweise kein Objekt vorhergesagt hat, da es dafür meist unendlich viele mögliche Bereiche im Bild gibt. Daher beschränkt man sich bei

der Bewertung von Objekterkennungsmethoden eher auf die relevanten Metriken TP, FP und **FN!**.

#### Evaluationsmetriken für Objekterkennungsverfahren

Viele der gängigen Bewertungsmetriken für Objektdetektoren basieren auf den zuvor eingeführten Kriterien True Positive, False Positive und False Negative. Zwei besonders häufig verwendete Kennzahlen sind die Precision (Präzision) und der Recall (Trefferquote), die sich wie folgt berechnen lassen:

- Die **Präzision** beschreibt den Anteil der korrekt erkannten Objekte an allen vom Detektor vorhergesagten Objekten. Sie gibt also an, wie viele der vorhergesagten Objekte tatsächlich existieren. Die Formel für die Präzision lautet:

$$\text{Präzision} = \frac{TP}{TP + FP} \quad (3.5)$$

- Die **Trefferquote** hingegen gibt an, wie viele der tatsächlich vorhandenen Objekte vom Detektor erkannt wurden. Sie drückt aus, wie vollständig die Erkennung eines Detektors ist und berechnet sich folgendermaßen:

$$\text{Trefferquote} = \frac{TP}{TP + FN} \quad (3.6)$$

Da Präzision und Trefferquote oft im Widerspruch zueinander stehen – eine hohe Präzision bedeutet oft eine geringere Trefferquote und umgekehrt – wird häufig der sogenannte **F1-Score** verwendet, um einen Kompromiss zwischen beiden Kennzahlen darzustellen. Der F1-Score ist das harmonische Mittel von Präzision und Trefferquote und lässt sich mit der folgenden Formel berechnen:

$$F1 = 2 \times \frac{\text{Präzision} \times \text{Trefferquote}}{\text{Präzision} + \text{Trefferquote}} \quad (3.7)$$

Ein weiterer wichtiger Bewertungsmaßstab ist die **mean Average Precision (mAP)**, also die durchschnittliche Präzision, die über verschiedene IoU-Schwellen hinweg die Leistung eines Detektors bewertet. Die mAP stellt hierbei das arithmetische Mittel der durchschnittlichen Präzisionen über alle Objektklassen hinweg dar, während die Average Precision (AP) die mittlere Präzision einer Objektklasse beschreibt. Diese Metrik liefert ein umfassenderes Bild der Modellleistung, indem sie die Präzision für verschiedene Vorhersage-Schwellenwerte berechnet und mittelt. Da in dem hier beschriebenen Anwendungsfall lediglich die eine Objektklasse 'Person' von Relevanz ist, sind die Bezeichnung mAP und AP in den folgenden Betrachtungen und Auswertungen gleichzusetzen.

Zusätzlich zur mAP wird oft auch die **Verarbeitungszeit** eines Detektors in Betracht gezogen, um zu beurteilen, wie schnell und effizient ein Algorithmus in Echtzeitanwendungen

funktioniert. Besonders in sicherheitskritischen Anwendungen, wie dem Notbremsassistenten, spielt neben der Erkennungsgenauigkeit auch die Geschwindigkeit eine wesentliche Rolle. Da die Bildverarbeitung meist mit großen Datenmengen verbunden ist und die Prozesse oft sehr komplex sind, können die Laufzeiten der Objekterkennungsmethoden, je nach genutzter Hardware, sehr unterschiedlich ausfallen. Es hat sich gezeigt, dass Objekterkennungsalgorithmen, welche die Verarbeitung auf einer Graphics Processing Unit (GPU) ausführen, große Vorteile gegenüber solchen, welche ihre Prozesse lediglich auf der Central Processing Unit (CPU) ausführen, aufweisen. Das ist damit zu erklären, dass die GPU, welche speziell für grafische Berechnungen konzipiert ist, viel mehr Berechnungsschritte parallelisiert ausführen kann. Grund dafür ist, dass die GPU tausende kleine Rechenkerne besitzt, wohingegen eine CPU im Regelfall wenige, dafür deutlich leistungsstärkere Rechenkerne besitzt. Zusätzlich ist die Speicherbandbreite einer GPU meist deutlich höher, als die des Arbeitsspeichers einer CPU und es gibt viele innovative Technologien, wie Tensorkerne, CUDA-Kerne und Software basierte Beschleunigungen, welche die Vorteile der Architektur einer GPU noch weiter fördern, um derartige Prozesse noch effektiver und schneller ausführen zu können. Die Betrachtung dieser Metriken soll zum einen einen Zusammenhang zwischen verschiedenen Eigenschaften eines Detektors, wie beispielsweise der Rechenzeit und der Erkennungsgenauigkeit und seiner Eignung für den Einsatz in einem AEBS herstellen. Zum anderen ermöglichen die Metriken, insbesondere die Präzision, eine Einschätzung der Falschauslösungen der Detektoren im Einsatzgebiet eines Notbremsystems, da dies anhand von Realfahrversuchen nicht zuverlässig ermittelt werden kann.

#### **3.6. Vergleichsversuche der Notbrems-Algorithmen**

Um die verschiedenen Detektionsmethoden, sowie die implementierte Entscheidungslogik zur Auslösung eines Notbremsbefehls in praxisnahen Szenarien innerhalb des Kontextes eines Notbrems-Assistenzsystems abschließend zu überprüfen, sind Realfahrversuche, trotz der Möglichkeit moderner und innovativer Simulationsumgebungen, unerlässlich. Üblicherweise werden daher genormte Prüfzenarien inszeniert, in welchen das Fahrzeug mit vordefinierten Geschwindigkeiten auf eine potentielle Kollisionsstelle zufährt, in welcher ein weiches Prüfziel, also ein Fahrzeug- oder Personen-Dummy, je nach Testfall statisch oder dynamisch einen Positionskonflikt mit der Fahrspur des Fahrzeuges erzeugt. Das Prüfzenario wird dabei als bestanden bewertet, wenn der Dummy entweder nicht erfasst wurde oder die Kollisionsgeschwindigkeit des Fahrzeuges einen bestimmten Wert unterschreitet. Dabei kann das System selbstständig den Zeitpunkt zur Auslösung des Notbremssignals, sowie die Stärke der Verzögerung wählen. Da das Bremsmanöver, für sich, keinen Teil dieser Umsetzung darstellt und es in dieser Ausarbeitung vielmehr um die Qualität der Erkennung einer potentiellen Gefahrensituation, sowie der damit verbundene Zeitpunkt der Ausgabe des Bremssignals geht, soll daher die Bewertung dieses Aspektes eine tragende Rolle zur Validierung der Systeme spielen.

#### 3.6.1. Testumgebung und Versuchsablauf der Realfahrversuche

Zur Umsetzung der Realfahrversuche steht an der HTW Dresden ein Prüffeld zur Verfügung, welches sich für derartige Fahrversuche sehr gut eignet. Zum einen, da das Prüffeld einen nahezu ebenen, gut asphaltierten Untergrund, als auch genügend Platz bietet, um sämtliche relevante Prüfscenarien, die den gesetzlich vorgeschriebenen sehr nahe kommen, durchzuführen. Diese Testumgebung stellt zudem eine übersichtliche Szene dar, durch welche unerwünschte Fehldetektionen durch Passanten oder störende Objekte weitestgehend vermieden werden können. Das Prüffeld umfasst eine große freie Fläche, die bis auf einen großzügig umrandeten Baum, keine störenden Hindernisse aufweist. Mit maximalen Ausmaßen von etwa 70 x 55 m ergibt sich eine nutzbare Diagonale von etwa 75 m was genügend Platz bietet, um aussagekräftige Ergebnisse zu generieren.



Abbildung 3.8.: Prüffeld am K-Gebäude der HTW Dresden [37]

Auf der Fläche existieren zudem Fahrbahnmarkierungen, die unter anderem Kreuzungsszenarien darstellen, was möglicherweise genutzt werden kann, um die Verschiedenen Szenarien reproduzierbar zu gestalten und sich somit an festen, permanent vorhandenen Markierungen zu orientieren. Die erlaubte Geschwindigkeit auf dem Prüffeld ist auf 30 km/h begrenzt, womit auch die maximale Prüfgeschwindigkeit der Versuche auf diese Geschwindigkeit limitiert ist.

Um die Qualität der Algorithmen in ihrem Gesamtumfang vergleichen zu können, soll besonders der Auslösezeitpunkt des Notbremsignals, in Bezug zum vordefinierten TTC-Schwellwert im Fokus der Versuche stehen. Diese Latenz soll eine Aussage über die Qualität der Algorithmen zulassen. Diese Latenzzeit zwischen TTC-Schwellwert und Auslösezeitpunkt spiegelt dabei zum einen die Schnelligkeit der Algorithmen wieder und zeigt zum

anderen, ob die Erkennungsgenauigkeit hoch genug ist. Um den Auslösezeitpunkt zu bestimmen soll eine Betrachtung der Zusammenhänge zur Berechnung der TTC dienen.

$$\text{TTC} = \frac{d}{v_r} \quad (3.8)$$

Die TTC ergibt sich nach Formel 3.8 aus dem Abstand  $d$  und der relativen Geschwindigkeit  $v_r$  zweier Verkehrsteilnehmer.

Somit lässt sich bei Kenntnis über den Abstand  $d$  beider Verkehrsteilnehmer, sowie einer bekannten relativen Geschwindigkeit  $v_r$  zum Auslösezeitpunkt der TTC-Wert berechnen. Dazu soll eine Möglichkeit implementiert werden, eine LiDAR-Punktwolke zu diesem Zeitpunkt aufzunehmen und daraus die Entfernung zwischen Testfahrzeug und Prüfziel zu ermitteln.

#### 3.6.2. Testszenario

Während die Prüfverfahren gemäß den gesetzlichen Vorgaben der UN-Regelung Nr. 152, sowie die standardisierten Tests des Euro NCAP zur Überprüfung der Notbremssysteme ausschließlich Testfälle mit bewegten Prüfzielen umfassen, wird in dieser Arbeit ein vereinfachtes Szenario mit einem statischen Prüfobjekt verwendet. Diese Vorgehensweise dient der grundlegenden Validierung der Systeme unter kontrollierten Bedingungen.

Der Einsatz eines statischen Prüfobjekts bietet mehrere Vorteile:

- **Reduzierte Komplexität:** Durch die Eliminierung der Bewegungsvariablen können einzelne Systemkomponenten isoliert und deren Funktionalität detaillierter untersucht werden. Das erleichtert das Identifizieren spezifischer Schwachstellen des Systems.
- **Kosten- und Zeitersparnis:** Statische Tests benötigen weniger Ressourcen und können schneller durchgeführt werden, was insbesondere in frühen Entwicklungsphasen eines solchen Systems von großer Bedeutung ist.
- **Verbesserte Fehlererkennung:** Die Erkennung von Abweichungen in den Messwerten wird deutlich verbessert, wodurch aussagekräftigere Erkenntnisse und Vergleiche zwischen den einzelnen Algorithmen möglich ist.

Obwohl Tests mit dynamischen Prüfzielen eine realistischere Abbildung der realen Einsatzbedingungen widerspiegeln, soll dieses Testverfahren entsprechend der Aufgabenstellung und der Zielsetzung dieser Arbeit keine Eignung des Systems für den Einsatz in der Praxis darstellen, sondern lediglich die generelle Funktionalität des Systems aufzeigen, sowie einen Vergleich der verschiedenen umgesetzten Algorithmen untereinander zulassen.

Das Testszenario soll dabei so gestaltet werden, dass über alle Messfahrten hinweg eine möglichst konstante Geschwindigkeit angestrebt wird, um die Vergleichbarkeit zwischen den Messungen möglichst hoch zu gestalten.

Um eine möglichst lange Prügerade zu erhalten, wird daher die Diagonale des Prüffeldes genutzt, um möglichst lange mit konstanter Geschwindigkeit auf das Prüfziel zuzufahren.

Für jeden Algorithmus sollen drei Messungen durchgeführt werden, um einen aussagekräftigen Mittelwert zu bilden.

---

## 4. Umsetzung

In diesem Kapitel wird die praktische Umsetzung der zuvor erläuterten Konzepte vorgestellt. Dabei soll auf die Implementierung der benötigten Hardware, sowie der Detektionsmodelle, in MATLAB, sowie die Umsetzung einer sinnvollen Logik zur Entscheidungsfindung und der abschließenden Integration des Gesamtsystems in die Fahrzeugumgebung, eingegangen werden.

### 4.1. Integration der Hardware

#### 4.1.1. Integration der Gigabit-Ethernet-Kamera

Die für diese Umsetzung ausgewählte GigE-Vision Kamera der Marke Basler, wird wie bereits beschrieben, über Ethernet mit dem Host-PC verbunden. Da die Kamera durch PoE ihren Strom direkt über das Ethernetkabel bezieht und gängige Rechner in der Regel nicht in der Lage sind eine Stromversorgung über Ethernet bereitzustellen ist ein PoE-fähiger Ethernet-Switch notwendig, um die Kamera in Betrieb zu nehmen. Dieser Switch bietet sowohl die Möglichkeit, die Stromversorgung der PoE-Geräte zu gewährleisten, als auch mehrere Ethernet-Geräte mit dem Host-PC zu verbinden.

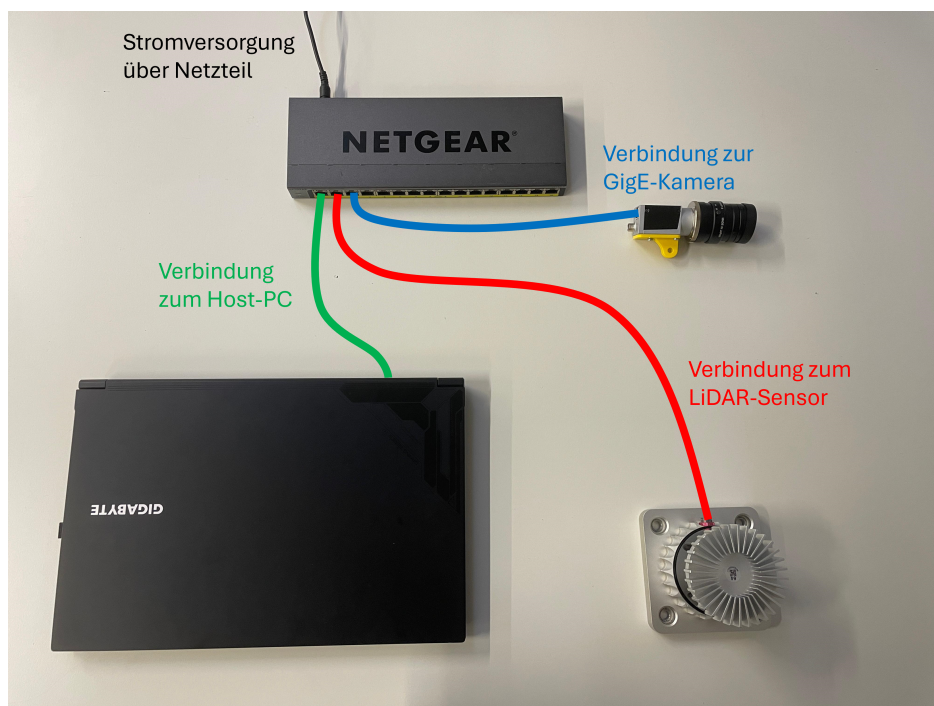


Abbildung 4.1.: Darstellung der physischen Verbindung mehrerer Komponenten über ein PoE-Ethernet-Switch

Um neben der physischen Verbindung auch eine Datenverbindung herzustellen, ist es wichtig die IP-Adressen, sowie die Netzmasken des Host-PC und der Kamera aneinander anzupassen. Zusätzlich ist zu beachten, dass der Netzwerkadapter des Host-PCs so konfiguriert ist, dass er mit den hohen Datenmengen und entsprechend großen Datenpaketen der Kamera

umgehen kann. Für erste Konfigurationen der Kamera, wie beispielsweise der Festlegung der IP-Adresse des Gerätes oder bei relevanten Fehlersuchen kann es hilfreich sein die herstellereigene Software pylon Viewer zu nutzen, welche eine übersichtliche und vereinfachte Verwaltung und Konfiguration der Basler-Produkte über eine grafische Benutzeroberfläche bietet. Der pylon Viewer ist Bestandteil des Camera Software Suits von Basler, welcher zusätzlich sämtliche Treiber für die Basler-Kameras enthält, welche ebenfalls benötigt werden. Um auch über MATLAB auf die Kamera zugreifen zu können, ist es notwendig die entsprechenden Toolboxen und Support-Packages aus dem MATLAB Add-On Explorer zu installieren. Allen voran sind hier die *Computer Vision Toolbox* und die *Image Acquisition Toolbox*, sowie die *Image Processing Toolbox* zu erwähnen, welche einen großen Umfang an Funktionen zur Aufnahme und Verarbeitung von Bilddaten liefern. Um nun einen direkten Zugriff auf die Basler-Kamera in MATLAB zu erhalten ist das '*Image Acquisition Toolbox Support Package for GigE Vision Hardware*', sowie der '*Basler camera driver*' notwendig. Innerhalb von MATLAB wird die Kamera als *gigecam-object* erstellt und unter einer beliebigen Variable abgelegt. Mithilfe des Aufrufs dieser Variable lässt sich das Kamera-Objekt konfigurieren und durch den '*Basler camera driver*' erhält man Zugriff auf den gesamten Funktionsumfang der Kamera mit den equivalenten Anpassungsmöglichkeiten, wie im pylon Viewer. Da sich die benötigten Rechenressourcen zum Empfangen und Verarbeiten des Datenstroms der Kamera in MATLAB gegenüber der Basler-Software stark erhöht und es dadurch zu Abbrüchen der Datenübertragung kommen kann, ist es notwendig die maximale Paketgröße (Packetsize) und die Paketverzögerung (Packetdelay) entsprechend anzupassen. Das kann sich allerdings auf die maximal erreichbare Übertragungsbandbreite auswirken und eine hohe Paketverzögerung erzeugt einen gewissen Versatz zwischen Aufnahmezeitpunkt der Bilddaten und dem Zeitpunkt des Eintreffens der Bilddaten im Host-PC, was bei echtzeitnahen Anwendungen unerwünscht ist und daher vermieden werden sollte. Da diese Umsetzung eine weitere Verarbeitung der Bilddaten vorsieht, wird der Algorithmus in MATLAB so gestaltet, dass nach vorheriger Konfiguration und Initialisierung sämtlicher Hardware und der nötigen Variablen, eine Hauptschleife ausgeführt wird, in welcher als erster Hauptschritt ein Bild von der Kamera abgerufen wird, bevor alle weiteren Verarbeitungsprozesse in der Hauptschleife ausgeführt werden (Abb. 3.1). Die Bildaufnahme erfolgt über die Methode '*snapshot*' unter der Angabe des entsprechenden Kamera-Objektes.

I: 62	I: 62	I: 61
I: 61	I: 61	I: 61

R: 40 G: 61 B: 124	R: 40 G: 61 B: 124	R: 39 G: 60 B: 123
R: 39 G: 60 B: 123	R: 39 G: 60 B: 123	R: 39 G: 60 B: 123

Abbildung 4.2.: Beschreibung von Bildpunkten in einem Graustufen- und einem RGB-Farbbild



Die Bilddaten werden unter einer zuvor festgelegten Variable als zweidimensionale Matrix im Falle eines Graustufenbildes und als dreidimensionale Matrix im Falle eines RGB-Farbbildes abgelegt, wobei die dritte Dimension die drei Farbkanäle Rot, Grün und Blau darstellt. Die Zeilen- und Spaltenanzahl der Matrix entspricht dabei den Pixelzeilen und -spalten des aufgenommenen Bildes. Die einzelnen Pixelwerte sind dabei standardmäßig als `uint8`, also als vorzeichenlose Ganzzahlen im 8-Bit-Wertebereich, welcher von 0 bis 255 reicht, beschrieben.

In Abbildung 4.2 sind die Dimensionsunterschiede sehr gut zu erkennen. Während im Graustufenbild (links) jedes Pixel nur durch einen Intensitätswert beschrieben wird, sind im Farbbild (rechts) drei Werte notwendig, um einen Bildpunkt zu definieren, wodurch folglich die dreifache Menge an Daten anfällt.

Um die Kamera dem Anwendungsfall entsprechend zu konfigurieren wurden lediglich kleine Anpassungen von den Standardeinstellungen vorgenommen. Beispielsweise wurde das Pixelformat der Kamera auf das Profil 'RGB8' gestellt, in welchem Farbinformationen übermittelt werden, diese jedoch auf 8 Bit statt 12 Bit aufgelöst werden. Das reduziert den Datenstrom und erhöht die Verarbeitungsgeschwindigkeit. Zusätzlich wurde die Verstärkung der Kamera auf automatisch gestellt, um eine automatische nachträgliche Regelung der Helligkeit zu gewährleisten, um auch bei wechselnden Beleuchtungssituationen eine annähernd gleiche Bildhelligkeit zu gewährleisten. Zusätzlich wurde die automatische Anpassung der Belichtungszeit des Sensors ausgeschaltet und diese auf einen festen Wert gesetzt, da im automatischen Modus starke Defizite in der Bildfrequenz der Kamera in MATLAB festgestellt wurden.

### 4.1.2. Integration des LiDAR-Sensors

Um den Ouster LiDAR-Sensor mit in das System einzubinden, kann dieser, wie in Abbildung 4.1 dargestellt, über Ethernet mit dem Switch verbunden werden, wodurch sowohl die Daten des Laserscanners, als auch die der Kamera gleichzeitig vom Host-PC empfangen werden können. Wie auch bei der GigE-Kamera ist es wichtig zu beachten, dass sich alle drei Geräte des Netzwerks im selben IP-Raum befinden und für die identische Netzmaske konfiguriert sind. Der Zugriff auf den LiDAR-Sensor über Matlab erfolgt über die Erstellung eines `ousterlidar`-Objektes unter Angabe einer Konfigurationsdatei, welche sämtliche Einstellungs-Parameter des Sensors enthält. Ein Abruf der Daten des Sensors ist in MATLAB mit dem Befehl `read` unter Angabe des `ousterlidar`-Objektes möglich.

Die Daten des Sensors werden in MATLAB als `pointCloud`-Objekt formatiert. Diese Klasse enthält eine  $N \times 3$  Matrix mit den x-,y- und z-Koordinaten der Punkte, die Gesamtanzahl aller Punkte, sowie die Minimal- und Maximalwerte jeder Achsenrichtung und eine  $N \times 1$  Matrix, welche die Reflexionsintensitäten aller Punkte enthält. Da der Ouster-Sensor auf 64 Ebenen scannt und pro Ebene bis zu 1024 Pulse in einer Umdrehung sendet und empfängt, sind die Punktwolken dieses Sensors als  $64 \times 1024 \times 3$  Matrix formatiert. Die dreidimensionale Darstellung erfolgt in MATLAB über den Befehl `pcshow`. Eine solche Darstellung ist in

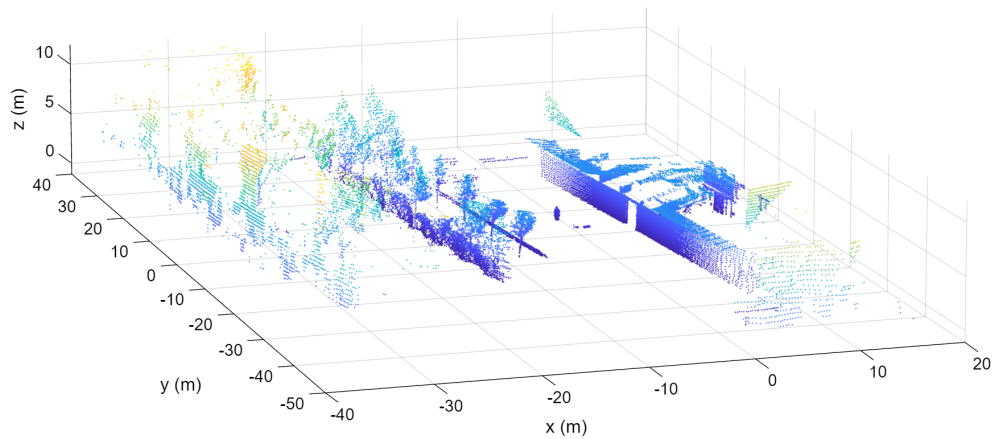


Abbildung 4.3.: Darstellung einer LiDAR-Punktwolke in MATLAB

Abbildung 4.3 zu sehen, wobei die Farben die Reflexionsintensität darstellen.

### 4.2. Integration der Objekterkennungen

Um die Bilddaten der Kamera direkt im Anschluss zu verarbeiten und in Echtzeit Informationen über vorhandene Objekte und die Position dieser im Bild zu erhalten, muss die Detektion ebenfalls in der zuvor implementierten Hauptschleife ausgeführt werden. Dazu muss zunächst der entsprechende Objektdetektor initialisiert werden. Wie bereits in den Vorbetrachtungen beschrieben, sollen die vortrainierten Modelle des YOLOv4-Detektors und des ACF-Detektors zum Einsatz kommen. Während die YOLOv4 Implementierung auf einem Datensatz trainiert wurde, welcher mehrere Objektklassen enthält und somit einen Multiklassen-Detektor darstellt, ist der ACF-Detektor aufgrund seiner Architektur nur in der Lage, Aussagen über eine Objektklasse zu treffen. In MATLAB ist mit dem 'peopleDetectorACF' ein für die Objektklasse 'person' vortrainierter ACF-Detektor verfügbar. Für diesen Detektor kann man zwischen den Modellen 'inria-100x41' und 'caltech-50x21' wählen. Diese unterscheiden sich zum einen in der Größe des Fensters, welches in der Sliding-Window-Methode über das Bild geschoben wird und zum anderen in dem Datensatz, auf welchem diese auf basis von maschinellem Lernen trainiert wurden. Das Modell 'inria-100x41' besitzt demnach eine Fenstergröße von 100x41 Pixeln und wurde auf dem INRIA-Personen-Datensatz trainiert, wohingegen 'caltech-50x21' eine Fenstergröße von 50x21 Pixeln nutzt und auf dem California Institute of Technology (Caltech)-Pedestrian-Datensatz angeleitet wurde. Eine klare Vorhersage, welches der Modelle präziser oder schneller arbeitet, ist hierbei nicht zu treffen, da dies stark vom Anwendungsfall und der genutzten Hardware abhängig ist. Höhere Fenstergrößen können hier bei leistungsstarker Hardware einen Vorteil in der Schnelligkeit erreichen, während die kleine Fenstergröße auf leistungsschwachen Recheneinheiten besser abschneidet. Der Caltech-Pedestrian-Datensatz ist jedoch weitaus umfangreicher und bietet größere Variationen in den Posen der abgebildeten Personen und den dargestellten Szenarien, als der Institut national de recherche en informatique et en automatique (INRIA)-

Personen-Datensatz. Im Falle von YOLO kann hier zwischen 'csp-darknet53-coco' als präzisere, aber rechenintensivere und 'tiny-yolov4-coco', als weniger komplexe Variante, gewählt werden. Diese unterscheiden sich im Backbone, also der Architektur des künstlichen neuronalen Netzes, welches für die Merkmalsextraktion und -auswertung genutzt wird. Beide Modelle wurden auf dem Common Objects in Context (COCO)-Datensatz trainiert, welcher 80 verschiedene Objektklassen umfasst.

Der Algorithmus dieser Umsetzung soll möglichst konfigurierbar und modular gestaltet werden, weshalb eine schnelle und simple Möglichkeit zum Wechsel zwischen den verschiedenen Detektormodellen integriert wurde. Die Auswahl des Detektors erfolgt dabei zu Beginn des Skriptes über die Variable '*personDetector*'. Die Detektoren sind dabei beginnend bei 1 durchnummeriert. Eine Übersicht der Detektormodelle und deren Nummerierung ist folgend zusammengetragen.

Tabelle 4.1.: Nummerierung der Detektormodelle

Detektor	Modell	Nummer
yolov4ObjectDetector	'csp-darknet53-coco'	1
yolov4ObjectDetector	'tiny-yolov4-coco'	2
peopleDetectorACF	'caltech-50x21'	3
peopleDetectorACF	'inria-100x41'	4

Über die Variable '*personDetector*' wird zwischen den jeweiligen cases gewitched. Bei Bedarf können hier weitere Detektoren hinzugefügt werden. Die Initialisierung des Detektors ist nur einmal erforderlich und findet daher außerhalb der Hauptschleife statt. Die eigentliche Detektion findet mithilfe der '*detect*'-Methode des Detektors innerhalb der Schleife statt. Dazu muss der Funktion der Detektor und die zuletzt erfassten Bilddaten übergeben werden. Als Ausgabe erhält man die Begrenzungsrahmen, meist als '*bboxes*' bezeichnet, die '*scores*', also Vertrauenswerte und je nach Detektormodell die '*labels*', also die Klassenbezeichnungen der Detektionen. Während YOLO ein Multiklassen-Detektor ist und somit die Klassenbezeichnungen als Labels ausgibt, erkennt der hier verwendete ACF-Detektor nur Personen als Objekte und gibt daher keine Labels aus. Die Begrenzungsboxen sind in MATLAB dabei als N-by-4 Matrix strukturiert, wobei N der Anzahl der Detektionen eines Bildes entspricht. Die Koordinaten der Boxen werden dabei als [x, y, width, height] formatiert. Dabei ist:

- x** - x-Koordinate der oberen linken Ecke der Box
- y** - y-Koordinate der oberen linken Ecke der Box
- width** - Breite der Begrenzungsbox
- height** - Höhe der Begrenzungsbox

In Abbildung 4.4 sind die Detektionen des YOLO-Detektors dargestellt. In den Bounding Box-Beschriftungen sind sowohl die erkannten Objektklassen, als auch die Koordinaten dargestellt. Die x- und y-Koordinaten entsprechen dabei den rot markierten Pixelkoordinaten.

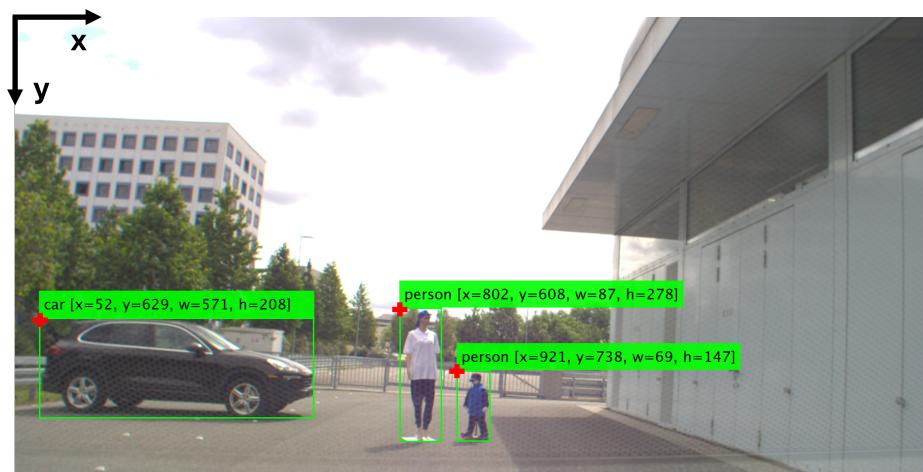


Abbildung 4.4.: Darstellung der Detektionen mit Bounding Box-Koordinaten und Klassenbezeichnung

Die Breite in Pixeln ist mit einem  $w$  bezeichnet und die Höhe mit einem  $h$ . Die Ausgabe der Klassenbezeichnungen der Detektionen als Labels sind bei einem Multiklassen-Detektor, wie YOLO notwendig, um die Detektionen, die der Objektklasse 'person' angehören zu identifizieren. Nur diese sollen in dieser Umsetzung zum Auslösen des Notbremsignals führen, weshalb alle Detektionen anderer Objektklassen ignoriert und nur die relevanten herausgefiltert werden müssen. Da die Detektoren große Unterschiede in den Rechenanforderungen aufweisen, variieren auch die resultierenden Laufzeiten sehr stark. Die Laufzeit eines Schleifendurchlaufes lässt sich dabei in MATLAB ganz einfach über die Befehle *tic* und *toc* ermitteln. Da jeder Schleifendurchlauf der Aufnahme und Verarbeitung eines Bildes entspricht, kann der Kehrwert der ermittelten Zeiten in fps angegeben werden.

Tabelle 4.2.: Erreichte Detektionsgeschwindigkeiten der Detektoren im Vergleich

Detektor	Modell	fps
yolov4ObjectDetector	'csp-darknet53-coco'	7,49
yolov4ObjectDetector	'tiny-yolov4-coco'	9,18
peopleDetectorACF	'caltech-50x21'	2,02
peopleDetectorACF	'inria-100x41'	4,5
R-CNN	'alexNet'	0,67
Faster R-CNN	'resNet18'	3,45

Die erreichten fps-Wert der verwendeten Detektoren sind in der Tabelle 4.2 zusammengetragen. Die Werte wurden bei der Verarbeitung der Biddaten der GigE-Kamera ermittelt. Zum Vergleich sind die ermittelten Werte selbsttrainierter R-CNN-Detektoren dargestellt. Es ist hierbei ein klarer Unterschied zwischen den YOLO- und den ACF-Detektoren zu erkennen. Die YOLO-Detektoren erreichen hierbei teilweise mehr als die doppelte Geschwindigkeit aller weiteren Detektoren und selbst zwischen den beiden YOLO-Modellen ist ein Unterschied zu erkennen.

### 4.3. Positionsbestimmung der detektierten Objekte im Raum

Die Tiefeninformationen sind für ein Notbrems-Assistenzsystem essenziell, da dieses nur auslösen darf, wenn die Geschwindigkeit des Fahrzeuges sowie der Abstand des potenziellen Kollisionsgegners in einem Verhältnis stehen, das eine Kollision ohne Eingriff des Systems unvermeidbar macht. Dieses Verhältnis wird mithilfe der TTC, gemäß Formel 2.2 beschrieben. Daher ist es notwendig, Informationen über die Entfernung des Objektes, in diesem Fall der Person, zu generieren.

#### 4.3.1. Herausforderungen der monokularen Tiefenbestimmung

Eine einzelne Kamera kann ohne zusätzliche Maßnahmen keine verlässlichen Tiefeninformationen liefern, da bei der Bilderzeugung die Tiefendimension der realen Umgebung verloren geht. Dadurch ist es schwierig, aus der zweidimensionalen Abbildung Rückschlüsse auf die dritte Dimension zu ziehen. In der Praxis wird häufig eine Kombination aus Kamera und ToF-Sensorik, wie Radio detection and ranging (Radar) oder LiDAR, verwendet, um über Sensorfusion die Objektdetektionen mit direkten Entfernungsangaben zu kombinieren. Alternativ kommen Stereo-Kamerasysteme zum Einsatz, die durch die Disparität zweier Perspektiven auf die Räumlichkeit schließen lassen.

Für diesen Anwendungsfall musste jedoch eine andere Lösung gefunden werden, um mithilfe einer einzelnen Kamera und im besten Fall aus nur einer Bildaufnahme auf die Entfernungen der Detektionen zu schließen. Zwar existieren mittlerweile aufwändige KI-gestützte Methoden, um Tiefeninformationen nachträglich durch Schätzungen zu regenerieren, diese sind jedoch in der Regel sehr rechenintensiv und durch die Nutzung neuronaler Netze kaum bis nicht nachvollziehbar und bleiben lediglich Schätzungen. Weiterhin gibt es die Möglichkeit, die Pixelbewegung aufeinanderfolgender Bilder zu analysieren und daraus Entfernungen abzuschätzen. Dieses Verfahren ist als optischer Fluss (*optical flow*) bekannt. Der große Nachteil hierbei ist, dass die Entfernungsberechnung abhängig von der relativen Bewegung der Szene zur Kamera ist und die Berechnung auf mehrere Bilder angewiesen ist, um überhaupt Daten liefern zu können. Zudem kann diese Methode sehr rechenintensiv sein, wenn viel Bewegung im Bild stattfindet und eine hohe Bildauflösung erforderlich ist. Durch die Berechnung auf Basis aufeinanderfolgender Bilder entsteht eine Art Trägheit gegenüber Verfahren, die Entfernungen auf Grundlage eines Einzelbildes bestimmen können, was die Reaktionsfähigkeit des Systems mindert, wenn sich ein Objekt schnell in den Sichtbereich des Sensors bewegt.

Schaut man sich die grundlegendsten monokularen Tiefenkriterien an, welche vor allem im Bereich der Wahrnehmungspsychologie zum Tragen kommen, so sind besonders zwei dieser Kriterien für diese Umsetzung relevant.

Abbildung 4.5 zeigt die drei grundlegendsten Kriterien der monokularen Raumwahrnehmung:

- **Relative Größe:** Abbildung 4.5a) stellt die relative Größe zweier tatsächlich gleich-

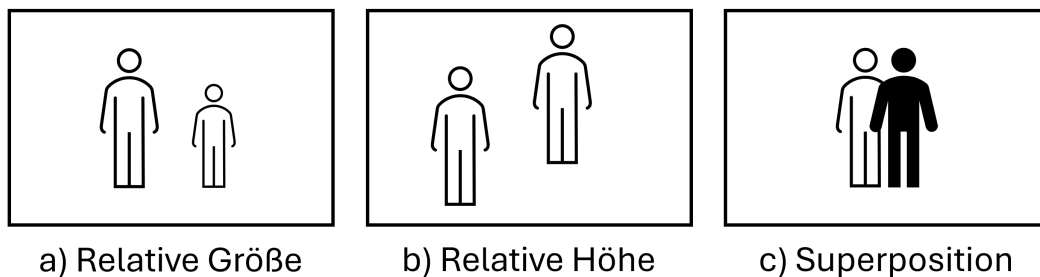


Abbildung 4.5.: Schematische Darstellung monokularer Tiefenkriterien

großer Objekte im Raum dar. Die relative Größe besagt, dass Objekte, die weiter vom Projektionszentrum entfernt sind, in der Abbildung kleiner dargestellt werden als nahe Objekte. Dabei besteht ein inverser Zusammenhang zwischen Abbildungsgröße und Entfernung des Objektes. Um genaue Entfernungswerte abzuleiten, ist allerdings vorausgesetzt, dass die tatsächliche Größe des Objektes bekannt ist. Dies ist in dieser Umsetzung nicht der Fall, und die Annahme eines pauschalen Wertes für die Körpergröße ist nicht möglich, da die Körpergrößen von Menschen stark variieren. Dies würde zu sehr hohen Abweichungen in der Entfernungsabschätzung führen. Besonders kleine Personen, wie Kinder, wären gefährdet, da sie durch ihre Körpergröße stets als weiter entfernt wahrgenommen würden als größere Personen in derselben Entfernung.

- **Superposition (Überdeckung):** Die in Abbildung 4.5c) dargestellte Superposition oder auch Überdeckung genannt ist ebenfalls ein bedeutendes Tiefenkriterium. Sie lässt sich allerdings für eine genaue Entfernungsabstufung nicht nutzen, da aus diesem Kriterium lediglich hervorgeht, ob ein Objekt näher oder ferner am Betrachter positioniert ist als ein anderes, und eine Überdeckung nur selten im realen Straßenverkehr gegeben ist.
- **Relative Höhe:** Ein weiteres wichtiges Tiefenkriterium in der monokularen Optik ist die relative Höhe bzw. die projizierte Höhe eines Objektes im Bild, dargestellt in Abbildung 4.5b). Diese besagt, dass Objekte, die weiter entfernt sind, unter der Voraussetzung, dass das Projektionszentrum oberhalb der Bodenebene liegt, weiter oben im Bild dargestellt werden als Objekte, die sich näher an der Kamera befinden.

#### 4.3.2. Projektionsgeometrischer Ansatz

Zunächst sollen, zum besseren Verständnis der nachfolgenden Herleitungen, die verwendeten Koordinatensysteme definiert werden.

- **Raumkoordinatensystem ( $X_W, Y_W, Z_W$ )**
  - Ursprung befindet sich auf der Bodenebene
  - $X_W$ : Parallel zur Bodenebene
  - $Y_W$ : Senkrecht zur Bodenebene

- $Z_W$ : Parallel zur Bodenebene
- **Bildkoordinatensystem  $(u, v)$** 
  - Ursprung befindet sich in der oberen linken Bildecke
  - $u$ : Horizontale Pixelkoordinate von links nach rechts
  - $v$ : Vertikale Pixelkoordinate von oben nach unten
- **Kamerakoordinatensystem  $(X_C, Y_C, Z_C)$** 
  - Ursprung befindet sich im optischen Zentrum der Kamera im Abstand  $f$  zur Bildebene
  - $X_C$ : Parallel zur  $u$ -Achse des Bildkoordinatensystems
  - $Y_C$ : Parallel zur  $v$ -Achse des Bildkoordinatensystems
  - $Z_C$ : Senkrecht zur Bildebene, in Blickrichtung der Kamera (optische Achse)

Um die folgenden Herleitungen übersichtlicher zu gestalten, sollen die Bildkoordinaten, wie es im Bereich der Bildverarbeitung gebräuchlich ist mit  $u$  und  $v$  bezeichnet werden. Dabei entspricht  $u$  der sonst gebräuchlichen x-Achse und  $v$  dementsprechend der y-Achse des Bildkoordinatensystems.

Um einen mathematischen Zusammenhang zwischen Abbildungshöhe und Entfernung eines Objektes herzustellen, dient die Betrachtung des in Abschnitt 2.1.2 beschriebenen Lochkameramodells. Daraus geht hervor, dass jede Kamera ein optisches Zentrum besitzt, welches den Ursprung des Kamerakoordinatensystems bildet, dessen Z-Achse, die sogenannte optische Achse, orthogonal zur Bildebene steht und diese im sogenannten Bildhauptpunkt durchstößt. Bei einer Rotation um die X-Achse des Kamerakoordinatensystems im mathematisch negativen Drehsinn, neigt sich demzufolge die optische Achse um den sogenannten Elevationswinkel  $\varphi$  in Richtung der Bodenebene und durchstößt diese in einem Punkt  $P$ , welcher in den Raumkoordinaten  $(X_W, Y_W, Z_W)$  beschrieben werden kann. Fällt man nun das Lot vom optischen Zentrum zum Boden, so bildet sich, unter der Annahme einer idealisierten ebenen Bodenebene, entsprechend Abbildung 4.6, ein rechtwinkliges Dreieck.

Ist dabei die Y-Position des optischen Zentrums in Bezug zur Bodenebene, welche der lotrechten Höhe  $h$  entspricht, sowie der Neigungswinkel der optischen Achse  $\varphi$  bekannt, so lässt sich mit einfachen trigonometrischen Zusammenhängen die horizontale Distanz  $d$  als Ankathete des rechtwinkligen Dreiecks wie folgt berechnen:

$$d = \frac{h}{\tan(\varphi)} \quad (4.1)$$

Da die optische Achse die Bildebene senkrecht im sogenannten Bildhauptpunkt  $c$ , mit den Bildkoordinaten  $u_c, v_c$ , durchstößt kann somit ein Zusammenhang zwischen der vertikalen Koordinate  $v_c$  und dem horizontalen Abstand  $d$  des Punktes  $P$  auf der Bodenebene hergestellt werden.

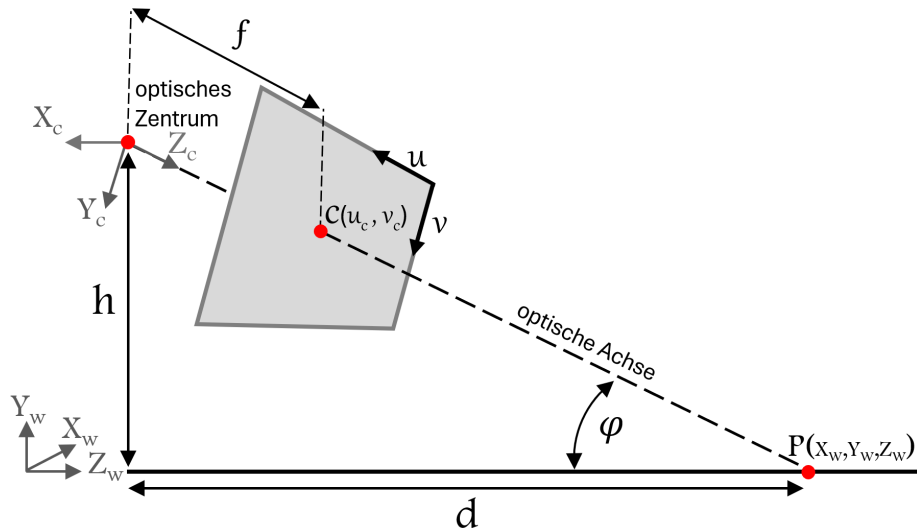


Abbildung 4.6.: Bestimmung der Distanz  $d$  über den Neigungswinkel der optischen Achse bei gegebener Fahrbahnorientierung  $\varphi$  und Höhe der Kamera  $h$

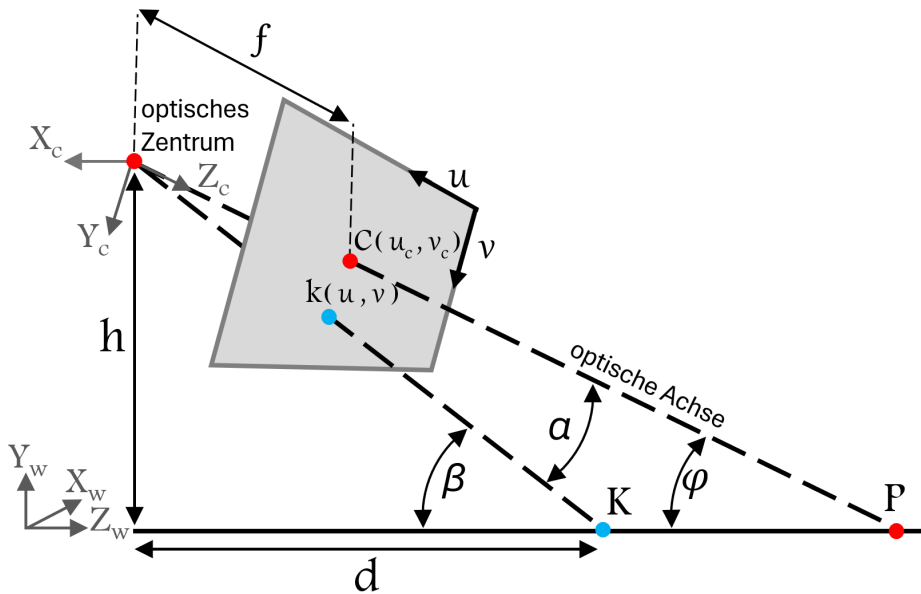


Abbildung 4.7.: Bestimmung der Distanz  $d$  zu einer Bildposition  $v$  bei gegebener Fahrbahnorientierung  $\varphi$  und Höhe der Kamera  $h$

Unter der Voraussetzung, dass der Abstand zwischen der Bildebene und dem optischen Zentrum, die sogenannte Brennweite  $f$  bekannt ist, kann somit gemäß Abbildung 4.7 für jeden weiteren Punkt  $K$  der echten Welt, der im Raumkoordinatensystem  $(X_w, Y_w, Z_w)$  definiert ist, über dessen vertikale Koordinate  $v$  in der Abbildung, der Projektionswinkel  $\alpha$  relativ zur optischen Achse wie folgt bestimmt werden:

$$\alpha = \arctan\left(\frac{v - c_v}{f}\right) \quad (4.2)$$

Somit setzt sich der Projektionswinkel dieser Bildkoordinate  $v$  relativ zur Bodenebene, bezeichnet als  $\beta$ , aus den bereits bekannten Winkeln  $\varphi$  und  $\alpha$ , wie folgt zusammen:



$$\beta = \varphi + \alpha \quad (4.3)$$

Für diesen ermittelten Winkel  $\beta$  kann weiterhin mithilfe der Tangensfunktion ein Rückschluss auf die horizontale Distanz  $d$  des Punktes  $K$  gezogen werden.

$$\tan(\beta) = \frac{h}{d} \quad \Rightarrow \quad d = \frac{h}{\tan(\beta)} \quad (4.4)$$

$$d = \frac{h}{\tan\left(\varphi + \arctan\left(\frac{v-c_v}{f}\right)\right)} \quad (4.5)$$

Somit gilt zusammengefasst folgender Zusammenhang:

- **Niedrigere Punkte im Bild (größere  $v$ -Werte):** Diese Punkte liegen weiter unten im Bild und repräsentieren Objekte, die näher an der Kamera sind.
- **Höhere Punkte im Bild (kleinere  $v$ -Werte):** Diese Punkte liegen weiter oben im Bild und repräsentieren Objekte, die weiter entfernt sind.

Analog zu diesem Vorgehen kann ebenfalls die X-Koordinate des Punktes  $K$  im Raum über die horizontale Bildkoordinate  $u$  ermittelt werden. Zu beachten ist hierbei, dass sich für die Richtungen  $u$  und  $v$  unterschiedliche Brennweiten  $f_u$  und  $f_v$  ergeben können, da die abbildenden Bildpixel nicht zwangsläufig quadratisch sein müssen. Diese für die Berechnung notwendigen Brennweiten, sowie die Koordinaten des Bildhauptpunktes können mithilfe einer Kamerakalibrierung bestimmt werden.

### 4.3.3. Kamerakalibrierung

Durch die Verwendung von Linsensystemen in Kameras wird der Strahlengang des Lichtes manipuliert, um ein scharfes und qualitatives Bild zu erzeugen. Das führt dazu, dass die Projektion einer Kamera von dem vereinfachten und idealisierten Lochkammermodell abweicht und nicht durch dieses beschrieben werden kann. Der Strahlengang wird weitaus komplexer und somit auch die Beziehungen zwischen Punkten, welche im Objektraum liegen und deren projizierten Bildpunkten. Durch die Verwendung verschiedener Linsen wird die Brennweite verändert, es können Abbildungsfehler, wie die in Abschnitt 2.1.2 beschriebenen Verzeichnungen, auftreten und die Lage des Bildhauptpunktes kann vom Mittelpunkt des Bildsensors abweichen. Um Kenntnis über diese Parameter zu erhalten und somit dennoch einen möglichst genauen Rückschluss zwischen Bildpunkten und Raumkoordinaten zu ermöglichen, ist es daher üblich eine Kamerakalibrierung durchzuführen. Dazu können Objekte bekannter Größe genutzt werden, um die Relation zwischen Bildkoordinaten und Raumkoordinaten herzustellen. Sehr verbreitet ist die Nutzung eines Schachbrettmusters mit definierter Kachelgröße. Dieses wird aus verschiedenen Winkeln und möglichst so, dass das Muster in jedem Bildbereich mal vorkommt, fotografiert.

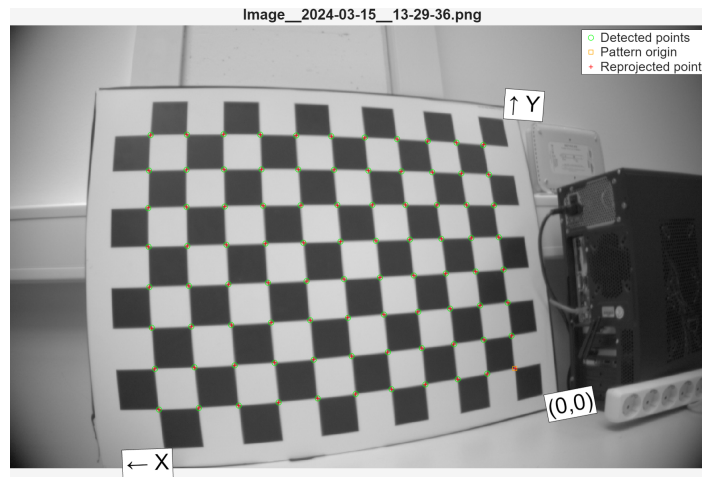


Abbildung 4.8.: Aufnahme des Kalibrierungsmusters mit detektierten Eckpunkten

In MATLAB kann mithilfe der *Camera Calibrator*-App eine solche Kamerakalibrierung anhand eines vordefinierten Kalibrierungsmusters recht simpel durchgeführt werden. Die App detektiert dazu selbstständig die Eckpunkte der Kacheln, wie in Abbildung 4.8 dargestellt. Mithilfe dieser vordefinierten Abstände kann nun zunächst auf die äußere Orientierung der Kamera geschlossen werden, also die Translationen und Rotationen des Kamerakoordinatensystems gegenüber der Kalibriertafel. Diese Werte werden als Extrinsische Kameraparameter bezeichnet. Mit diesen Parametern kann dann ebenfalls auf die radiale und tangentielle Verzerrung der Abbildung, die Koordinaten des Bildhauptpunktes und die Brennweiten in vertikaler und horizontaler Richtung geschlossen werden. Diese Werte nennt man innere Orientierung oder auch intrinsische Kameraparameter.

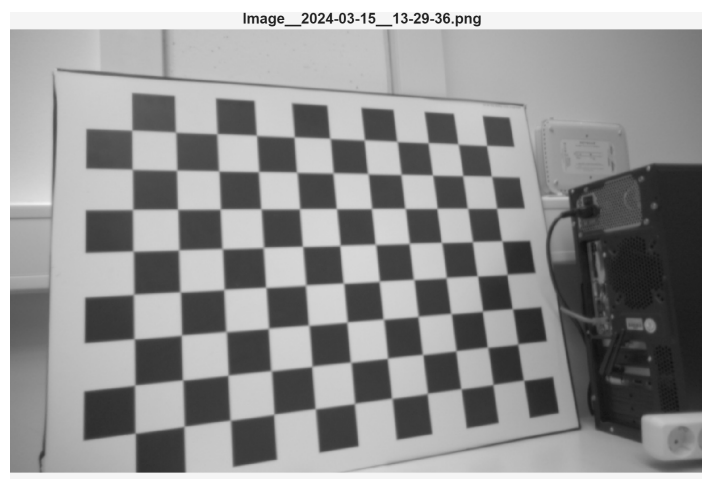


Abbildung 4.9.: Korrigierte Abbildung unter Berücksichtigung der Kameraparameter

Abbildung 4.9 zeigt dieselbe Bildaufnahme, wie in Abbildung 4.8, jedoch geometrisch korrigiert anhand der ermittelten intrinsischen Kameraparameter. Während die durch das Objektiv verursachte tonnenförmige Verzerrung in der unkorrigierten Abbildung gut an den Kanten der Kalibriertafel zu erkennen ist, konnte diese durch die Korrektur scheinbar gänzlich kompensiert werden. Somit ist nun auch ein zuverlässiger Rückschluss auf die Positionen der Detektionen und Raumkoordinaten, über die gesamte Bildfläche hinweg gewährleistet

#### 4.3.4. Berechnung der Detektionspositionen in Fahrzeugkoordinaten

Mithilfe der aus der Kamerakalibrierung gewonnen Parameter lässt sich in MATLAB ein *monoCamera*-Objekt erstellen, welches die Höhe der Kamera über dem Boden und die intrinsischen Kameraparameter berücksichtigt, sowie weitere extrinsische Parameter der Kamera. Mit diesem Sensorobjekt lassen sich mithilfe der Funktionen *'imageToVehicle'* und *'vehicleToImage'* Bildkoordinaten, unter Anwendung der zuvor erläuterten Zusammenhänge, direkt in Fahrzeugkoordinaten umrechnen und umgekehrt.

Um die Positionen der detektierten Objekte zu bestimmen muss folglich ein Punkt des Objektes gefunden werden, welcher möglichst nah an der Bodenebene liegt, da die Berechnung der Entfernung nur dann möglichst genau ist. Da Fußgänger in der Regel mindestens in einem Punkt Kontakt zum Boden haben, bzw. selbst im Laufschrift nur geringfügig von diesem entfernt sind, kann somit die Entfernung anhand des tiefsten Punktes des Begrenzungsrahmens der Detektion bestimmt werden. Wichtige Voraussetzung hierbei ist folglich eine präzise Detektion der gesamten Person, speziell der Füße als Kontaktpunkt der Person zur Bodenebene. Zur Bestimmung dieses Punktes wird in MATLAB für jede Begrenzungsbox die Koordinate des untersten horizontalen Mittelpunktes berechnet und dieser dann mithilfe der *'imageToVehicle'*-Funktion in Fahrzeugkoordinaten umgerechnet. Somit lässt sich für jedes erkannte Objekt die Position relativ zum Fahrzeug bestimmen und erhält die x- und y-Koordinate des Objektes in Fahrzeugkoordinaten [38].

#### 4.3.5. Versuch zur Validierung der Genauigkeit der Entfernungsberechnung

Die Genauigkeit dieser Methode zur Entfernungsermittlung soll anhand eines kleinen Versuchsaufbaus überprüft und validiert werden. Dazu wurde die Kamera auf einer Höhe von 1,5 Metern und einem Neigungswinkel von 6 Grad an einer Halterung angebracht und das Sensor-Objekt in MATLAB dementsprechend konfiguriert. Verwendet wurde hierfür das Objektiv mit 16 mm Brennweite. Um die Entfernungswerte zu überprüfen wurde mit einem Maßband der Abstand zur Kamera in 5-Meter-Schritten bestimmt und mit Markierungen am Boden festgehalten. Zusätzlich wurde an jeder Markierung ein gelber Ball zur besseren Visualisierung platziert.

Im Anschluss wurde ein Dummy entsprechend an jeder Position einmalig platziert und die Entfernung über das umgesetzte Verfahren ermittelt. Die Labels der Detektionen enthalten dabei die ermittelten Positionen in Fahrzeugkoordinaten, wobei die x-Achse nach vorn und die positive y-Achse nach links zeigt. Dementsprechend stellt die x-Koordinate den horizontalen Abstand zwischen dem optischen Zentrum der Kamera und dem Dummy dar. In Abbildung 4.10 sind beispielhaft die Messungen der ungeraden Werte abgebildet. Die ermittelten Werte über den gesamten Messbereich von 5 bis 35 Metern in 5 Meter Schritten sind in Tabelle 4.3 zusammengetragen.

Während der ermittelte Wert bei 30 Metern mit 6,8 % am stärksten abweicht, beträgt die Abweichung bei 35 Metern lediglich 1,34 %, wodurch eher ein proportionales Verhalten

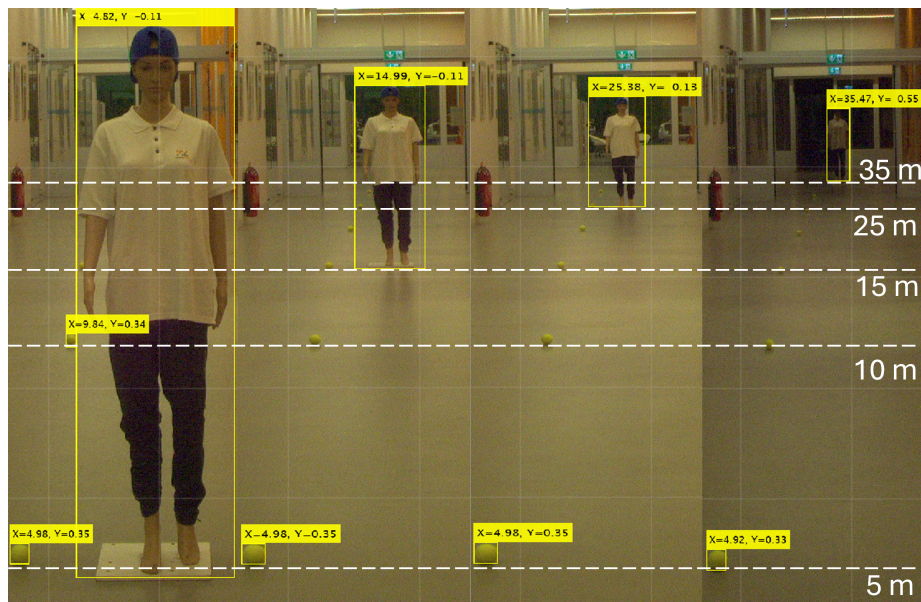


Abbildung 4.10.: Kollage der ermittelten Positionen des Dummys in 5 Meter Schritten

Tabelle 4.3.: Vergleich der tatsächlichen und gemessenen Entfernungen mittels Kamera

Tats. Entfernung (m)	Gem. Entfernung (m)	Fehler (m)	Fehler (%)
5,00	4,82	-0,18	-3,60 %
10,00	10,00	0,00	0,00 %
15,00	14,99	-0,01	-0,07 %
20,00	19,97	-0,03	-0,15 %
25,00	25,38	+0,38	+1,52 %
30,00	32,04	+2,04	+6,80 %
35,00	35,47	+0,47	+1,34 %

der Abweichung bei in Bezug zur Entfernung ausgegangen werden kann und der ermittelte Wert lediglich einen Ausreißer darstellt. Der durchschnittliche absolute Fehler über alle Werte hinweg liegt dabei bei rund 1,93 % was einen akzeptablen Wert darstellt. Dennoch zeigte sich, dass die Genauigkeit stark vom Detekormodell abhängt. Die in Tabelle 4.3 dargestellten Werte wurde mit dem 'csp-darknet53-coco'-Modell des YOLO-Detektors, also dem genauesten Detektor ermittelt. Da die anderen Modelle, besonders in den größeren Entfernungen, Probleme hatte die Füße des Dummys konstant zu erkennen, wiesen diese somit höhere Abweichungen dar.

#### 4.4. Entscheidungslogik für Warnung und Bremsung

Da nun die Positionen der detektierten Personen, sowohl in Bild-, als auch in Fahrzeugkoordinaten bekannt sind, sowie die Bereiche festgelegt wurden, welche zu einer kritischen Situation führen können, wenn sich eine Person darin befindet, kann nun eine Logik implementiert werden, die die Entscheidung darüber trifft, ob ein Eingriff des Systems notwendig ist. Dazu soll eine zweistufige Reaktionsstrategie implementiert werden, die in Abhängig-

keit vom Gefährdungsgrad der Situation aktiviert wird. In der ersten Stufe soll dazu eine Warnung in Form eines akustischen Signals, sowie eines kurzen Bremsrucks als haptischer Hinweis, erfolgen. Reagiert der Fahrer auf die Warnungen nicht ausreichend, sodass es dazu kommt, dass die Bedingungen der zweiten Stufe erfüllt werden, so wird eine Bremsung bis zum Stillstand des Fahrzeuges veranlasst. Die Unterscheidung zwischen den beiden Stufen erfolgt hierbei über die beiden zuvor erzeugten Fahrschläuche für den Warn- und den Notbremsbereich.

### **Adaptiver Fahrschlauch zur Entscheidungsfindung**

Um die örtliche Abgrenzung zwischen kritischen Bereichen und solchen, die als unkritische Positionen der Detektionen angesehen werden können, zu realisieren, ist es erforderlich, den kritischen Fahrbereich des Fahrzeuges als Fahrschlauch zu beschreiben. Es kann somit überprüft werden, ob sich eine gefährdete Person im Fahrbereich des Fahrzeuges befindet und somit ein Eingriff des Notbremssystems notwendig ist. Dazu ist zunächst die Berechnung der Koordinaten des Fahrschlauchs notwendig, sowie eine weiterführende Entscheidungslogik, die anhand des Fahrschlauches und den ermittelten Positionen der detektierten Objekte entscheidet, ob eine Warnung oder sogar ein Notbremseingriff notwendig ist.

### **Berechnung der Fahrschlauchkoordinaten**

Damit der Fahrschlauch in jedem Fahrzustand des Fahrzeuges den kritischen Bereich beschreibt, muss dieser dynamisch an die Längsführung und die Querführung des Fahrzeuges angepasst werden. Sowohl der Kurvenradius, als auch die Geschwindigkeit, welche ein Fahrzeug fährt, hat Einfluss auf die Position dieses für Fußgänger kritischen Bereichs relativ zum Fahrzeug. Daher muss die Berechnung der Koordinaten an diese Größen permanent angepasst werden. Zunächst soll daher die Abgrenzung dieses Bereichs in Fahrzeuglängsrichtung erfolgen.

### **Längenbegrenzung der Fahrschläuche**

Die Länge des Fahrschlauches beschreibt den längsdynamischen Aspekt des Notbremssystems. Somit stellt diese die Hauptkomponente bei der Bewertung von potentiellen Gefahren dar. Die Entscheidung, ob eine Bremsung erforderlich ist, sowie der Zeitpunkt der Warnung wird dabei in der Praxis anhand von festgelegten TTC-Schwellwerten getroffen. Diese berechnet sich nach Formel 3.8 und stellt somit einen zeitlichen Bezug zum Kollisionsmoment dar. Die TTC ist dabei abhängig vom räumlichen Abstand und der relativen Geschwindigkeit beider Kollisionsgegner.

$$d = \text{TTC} \cdot v_r$$

Stellt man die Formel nach dem Abstand um, so lassen sich für feste Schwellwerte der TTC geschwindigkeitsabhängige Entfernungswerte ermitteln. Anhand dieser lässt sich so-

mit ein räumlicher Bezug zur Bewertung der Kritikalität einer Situation herstellen. Somit kann dieser ermittelte Entfernungswert als räumlicher Schwellwert für das umgesetzte Notbremsssystem genutzt werden, um eine Entscheidung zu treffen. Da die Warnung, wenn möglich, deutlich vor der Bremsung erfolgen muss, sind somit zwei Schwellwerte für die TTC erforderlich

Diese liegen in der Praxis bei bis zu 1,6 Sekunden für den ersten Bremsingriff des Systems und bei etwa 2,5 Sekunden für die Warnung des Fahrers[4]. Aufgrund der verhältnismäßig hohen Laufzeiten der Detektoren sollen diese Werte zunächst mit einem Korrekturwert beaufschlagt werden. Die endgültigen Werte gilt es dann mithilfe von Messungen der Latenz des Systems und anhand Versuchsfahrten zu ermitteln.

Um nun den Fahrschlauch dynamisch an die Fahrzeuggeschwindigkeit zu koppeln, muss die Länge des Fahrschlauches entsprechend der Geschwindigkeit angepasst werden. Das bedeutet der Fahrschlauch muss weiter vorausschauen, je schneller das Fahrzeug fährt. Die Geschwindigkeitswerte werden dabei direkt vom CAN-Bus des Fahrzeuges abgegriffen, da diese Werte ohnehin vorliegen und eine gewisse Genauigkeit liefern, welche für diese Umsetzung als ausreichend betrachtet wird. Eine detaillierte Beschreibung zur Erfassung und Auswertung der CAN-Signale folgt im Abschnitt 4.7 Integration in die Fahrzeugumgebung.

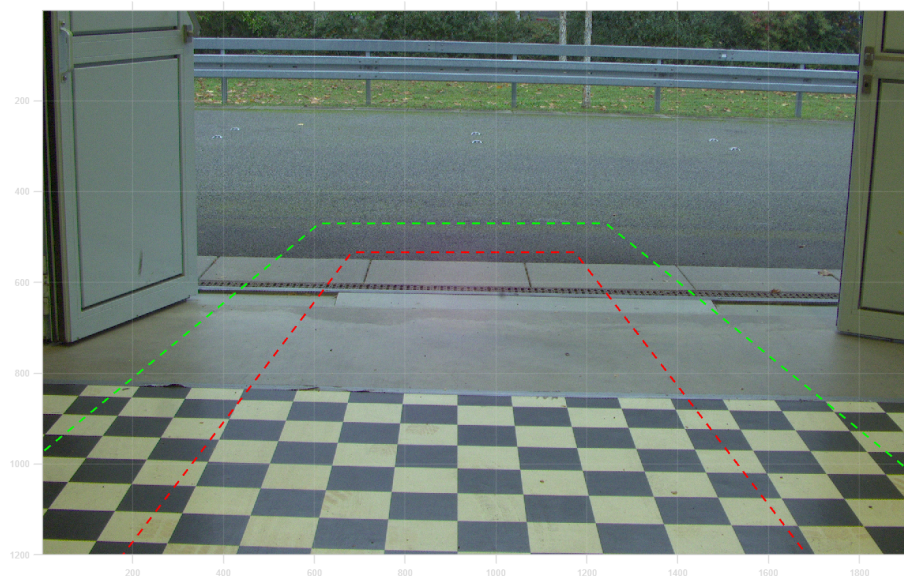


Abbildung 4.11.: [Darstellung der Fahrschläuche im Kamerabild für einen Geschwindigkeitswert von 20 km/h

### Seitliche Begrenzung der Fahrschläuche

Um den Fahrschlauch auch in Bezug der Querverführung des Fahrzeuges dynamisch der Fahrtrichtung in Kurvenfahrt anzupassen, ist der Lenkwinkel des Fahrzeuges relevant, um daraus den Kurvenradius zu berechnen. Dazu wurden in dem Fall die Lenkwinkelwerte des Lenkrades, nach dem selben Prinzip, wie zuvor die Geschwindigkeitswerte, vom CAN-Bus des Fahrzeuges abgegriffen und für jeden Schleifendurchlauf ein Mittelwert gebildet. Um aus

den Lenkradwinkelwerten ( $lw$ ) nun auf den mittleren Lenkwinkel der Vorderräder zu schließen, muss die Lenkübersetzung berücksichtigt werden. Bei einem BMW i3 ab Modelljahr 2019 liegt dieses Übersetzungsverhältnis  $c$  bei 14:1 [39]. Daher ergibt sich der Lenkwinkel  $\theta$  der Vorderräder in Bogenmaß zu:

$$\theta = \frac{lw}{c} \times \frac{\pi}{180} \quad (\text{Bogenmaß}) \quad (4.6)$$

Unter Berücksichtigung des Achsabstandes  $L$  lässt sich der Kurvenradius  $R$  der Fahrzeugmitte in  $m$  bestimmen:

$$R = \frac{L}{\tan(\theta)} \quad (4.7)$$

Der Achsabstand eines BMW i3 beträgt 2570 mm = 257 cm, bei einem Passat der Baureihe B8 beträgt dieser 2791 mm = 279,1 cm [39].

Damit nun auch die Länge des gekrümmten Fahrschlauches auf dem Kurvenradius  $R$  abgetragen werden kann, wird der Bogenwinkel  $\alpha$  auf Grundlage der zuvor ermittelten Distanz aus der TTC berechnet:

$$\alpha = \frac{s_{\text{brems}}}{R} \quad (4.8)$$

Basierend auf dem Kurvenradius und dem Bogenwinkel werden die x- und y-Koordinaten der Mittellinie der Fahrspur berechnet:

$$x_{\text{Center}} = R \cdot \sin(\theta_{\text{Points}}) \quad (4.9)$$

$$y_{\text{Center}} = R \cdot (1 - \cos(\theta_{\text{Points}})) \quad (4.10)$$

Hierbei ist  $\theta_{\text{Points}}$  ein Vektor mit gleichmäßig verteilten Winkeln von 0 bis  $\alpha$ , welcher in MATLAB über die Funktion *linspace* für jeden Schleifendurchlauf neu berechnet wird. Die Anzahl der Punkte kann hierbei frei gewählt werden. Um die linke und rechte Begrenzungslinie der Fahrspur darzustellen, müssen nun noch Offset-Koordinaten entsprechend der Fahrspurbreite (*laneWidth*) in  $m$  berechnet werden.

$$\text{offsetX} = \left( \frac{\text{laneWidth}}{2} \right) \cdot \cos(\theta_{\text{Points}}) \quad (4.11)$$

$$\text{offsetY} = \left( \frac{\text{laneWidth}}{2} \right) \cdot \sin(\theta_{\text{Points}}) \quad (4.12)$$

Damit lassen sich entsprechend die Koordinaten der linken und rechten Begrenzung der Fahrspur berechnen.

$$x_{\text{Left}} = x_{\text{Center}} + \text{offsetY} \quad (4.13)$$

$$y_{\text{Left}} = y_{\text{Center}} - \text{offsetX} \quad (4.14)$$

$$x_{\text{Right}} = x_{\text{Center}} - \text{offsetY} \quad (4.15)$$

$$y_{\text{Right}} = y_{\text{Center}} + \text{offsetX} \quad (4.16)$$

Die Begrenzungspunkte werden zu einem geschlossenen Polygon kombiniert, welches die Fahrspur in Fahrzeugkoordinaten darstellt. Anhand dieses Polygons kann anhand eines räumlichen Abgleichs der Positionen der Detektionen mit den Polygonkoordinaten, bestimmt werden, ob eine kritische Situation vorliegt, welche das Eingreifen des Systems erforderlich macht.

Mit der Funktion *vehicleToImage* lassen sich die Polygonkoordinaten in MATLAB, unter Berücksichtigung des zuvor beschriebenen sensor-objects, in Bildkoordinaten übertragen, wodurch der Fahrschlauch auch im Ausgabebild angezeigt werden kann.

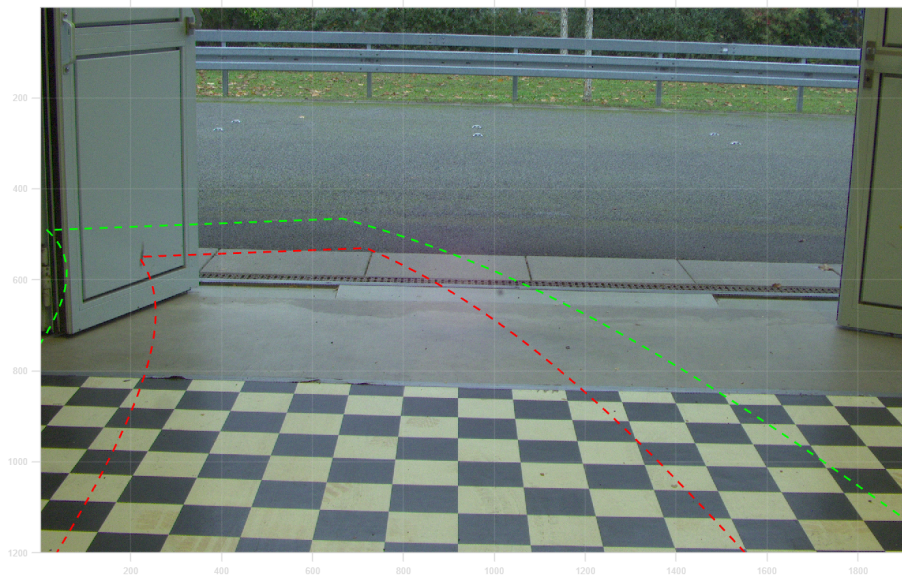


Abbildung 4.12.: Darstellung der Fahrschläuche im Kamerabild für einen Geschwindigkeitswert von 20 km/h bei vollem Lenkeinschlag

Da die gesetzlichen Rahmenbedingungen vorschreiben, dass der Fahrer zunächst durch das System gewarnt werden soll, bevor die eigentliche Notbremsung ausgelöst wird, soll diese Anforderung auch in dieser Umsetzung Bestand haben. Dazu wurde analog zum vorherigen Fahrschlauch ein zweiter, etwas breiterer und längerer, Fahrschlauch gebildet, welcher den Warnbereich darstellt. Dieser Bereich stellt eine Zone um die tatsächliche Fahrspur des Fahrzeuges dar, welche nicht zwangsläufig eine Kollision mit einem darin befindlichen Objekt nach sich zieht, jedoch kann es in diesem Bereich dennoch zu einer Kollision kommen, wenn entweder das Objekt seine Position ändert oder das Fahrzeug seinen Kurs ändert, bzw mit unveränderter Geschwindigkeit auf das Objekt zufährt. Um diesen breiteren und längeren Fahrschlauch des Warnbereiches zu erzeugen, wurden lediglich eine größere Spurweite



*laneWidthWarn*, welche sich aus der ursprünglichen Spurweite *laneWidth* und einem aufaddierten Wert in Metern ergibt. Außerdem wurde die Länge des Warnbereichs gegenüber dem ursprünglichen Fahrschlauch erhöht, indem der errechnete Bremsweg *brakingDist* mit dem Produkt aus Fahrzeuggeschwindigkeit *speed* und einer zuvor gewählten Zeitspanne *tReact* beaufschlagt wird. Dadurch wird gewährleistet, dass der Fahrer über den gesamten Geschwindigkeitsbereich hinweg die gleiche Zeitspanne zum selbstständigen Eingreifen hat, bevor die automatische Notbremsung eingeleitet wird. Das funktioniert lediglich nur, wenn das Objekt früh genug detektiert wird, also beispielsweise bei frontalem Zufahren auf das Objekt. Bewegt sich das Objekt schnell von der Seite in den Fahrschlauch, so ist auch der Warnbereich hinfällig.

Durch diese Berechnungen des Fahrspur- und Warnbereichspolygon ist das System nun in der Lage, lokal zwischen, der Fahrsituation entsprechend kritischen, potenziell kritischen und unkritischen Bereichen zu unterscheiden, was die Grundlage für die Entscheidungsfindung bildet, ob das System hart eingreifen muss, lediglich auf eine Situation aufmerksam machen soll oder kein Handlungsbedarf für das System besteht.

### 4.5. Tracking

Bei Systemen, die verschiedene Sensortechnologien kombinieren, erfolgt die Validierung einer Detektion durch den Abgleich der Objektpräsenz in den Daten aller Sensoren. Solch ein Abgleich ist bei monokularen Systemen hingegen nicht möglich. Dies ist aber relevant, um ein Auslösen des Systems durch Fehldetektionen zu vermeiden. Da Fehldetektionen meist nur kurzzeitig, innerhalb einer Bildaufnahme, auftreten und im nächsten Bild nicht mehr vorhanden sind, kann die Validierung einer Detektion somit über das Verfolgen der Objekte über mehrer Bilder hinweg realisiert werden. Wenn eine Detektion somit nur in einem Schleifendurchlauf vorhanden ist, wird diese daher als Fehldetektion identifiziert, womit diese nicht zum Auslösen des Systems führt. Zwar mindert dies die Reaktionsfähigkeit des Systems, dies muss jedoch zur korrekten Funktionalität des Gesamtsystems in Kauf genommen werden. Um die detektierten Personen nun also über mehrer Bilder hinweg immer wieder zu identifizieren und somit einen Zusammenhang zwischen detektionen aufeinanderfolgender Bilder zu erzeugen, muss ein Tracking-Verfahren implementiert werden.

Bei der hier eingesetzten Tracking-Methode *MultiObjectTrackerKLT* handelt es sich um ein vorgefertigtes Modell, welches ein Point-Tracking-Verfahren mit den Vorteilen des Tracking-by-Detection vereint. Diese Implementierung wurde ursprünglich zum Tracking von Gesichtern genutzt und ist über den MATLAB File-Exchange verfügbar [40]. Das Point-Tracking ist im Gegensatz zu Verfahren, die lediglich die Positionen der Bounding-Box-Koordinaten verfolgen, recht rechenaufwändig. Bei dieser Implementierung wird das Point-Tracking allerdings nur innerhalb der Begrenzungsboxen der Detektionen angewandt. In Kombination mit der Detektion bietet dieses Verfahren daher eine recht hohe Genauigkeit, bei deutlich reduzierter Rechenanforderung zum herkömmlichen Point-Tracking. Diese Implementierung basiert auf dem *vision.PointTracker*, welcher bereits in MATLAB verfügbar ist. Dieser ist

jedoch ein reiner Point-Tracker, welcher über das gesamte Bild hinweg Punkte verfolgt. Dabei kann dieser nicht zwischen verschiedenen Objekten unterscheiden. Das Multi-Objekt-Tracking wird durch das Verfolgen der Punkte innerhalb der Begrenzungsboxen des Detektors ermöglicht [41].



Abbildung 4.13.: Extraktion der Keypoints innerhalb der Begrenzungsboxen

In Abbildung 4.13 ist gut zu erkennen, wie der Tracker innerhalb der Begrenzungsboxen markante Key-Points extrahiert, welche in den darauffolgenden Bildern reidentifiziert und verfolgt werden. Diese Key-Points liegen meist an herausstechenden Pixelgradienten, wie beispielsweise der Kontur eines schwarzen Monitors vor hellem Hintergrund. An der Person im Hintergrund ist zu sehen, dass trotz der Bewegung durch das Bild, die Punkte immer wieder zugeordnet werden können. Da somit eine Beziehung zwischen den Detektionen aufeinanderfolgender Bilder hergestellt werden kann, ist es nun möglich den Detektionen eine ID zuzuordnen. Damit ergeben sich weitere Möglichkeiten, wie beispielsweise die Unterscheidung zwischen einer kurzzeitigen Fehldetektion oder einer länger vorhandenen, validen Detektion. Weiterhin ermöglicht das Tracking, durch die Verfolgung der Objekte, die Ermittlung der Positionsänderung eines Objektes zwischen den Bildern. Somit ist unter Berücksichtigung des zeitlichen Abstandes der Bildaufnahmen, eine Geschwindigkeitsberechnung der detektierten Objekte realisierbar.

Da diese Implementierung des Trackers große Probleme aufwies, wenn bereits getrackte Detektionen das Bild verlassen oder Objekte später in das Bild treten, war es notwendig das Initialisierungsverhalten des Trackers zu überwachen und zu steuern. Dafür wurde eine neue Methode integriert, welche sicherstellt, dass nur dann das Tracking durchgeführt wird, wenn dieser und die Detektionen in der Tracker-Logik korrekt initialisiert sind. Damit wird sichergestellt, dass diesbezügliche Fehler, die zur Unterbrechung des Algorithmus führen, vermieden werden. Zusätzlich stellte sich heraus, dass das Tracking bei sehr großen Positionsänderungen eines Objektes zwischen zwei Bildern eine große Herausforderung darstellt. Dies tritt auf, wenn sich ein Objekt sehr schnell durch das Bild bewegt oder die Verarbeitungsfrequenz der Bild sehr niedrig ist. Besonders bei dem Caltech-Modell des ACF-Trackers

fürte dies häufig dazu, dass der Tracker die Objekte häufig verlor, was sich in der Vergabe verschiedener IDs eines Objektes äußerte. Um das Tracking robuster zu gestalten, wurde daher eine Logik in dem Algorithmus implementiert, welche mithilfe eines Zählers die Bilder zählt und nur bei jedem n-ten Bild eine Detektion durchführt, während die Objekte in den dazwischenliegenden Bildern weiterhin getracked werden. Dadurch konnte die Zahl der verarbeiteten Bilder pro Sekunde deutlich erhöht werden, wodurch die Positionsänderungen zwischen den Bildern bei gleicher Bewegung der Objekte geringer ausfällt. Dadurch konnte ein deutlich besseres Tracking-Ergebnis erzielt werden. Hierbei ist jedoch zu beachten, dass in Abhängigkeit des Wertes n, welcher festlegt bei jedem wievielten Schleifendurchlauf die Detektion stattfindet, die Detektionsrate verringert wird und das System somit verzögert auf neue Objekte im Bild reagieren kann. Eine gute Reaktionsfähigkeit ist jedoch besonders bei einem Notbremssystem von hoher Relevanz, weshalb der Wert n für den Counter möglichst gering zu wählen ist.

Tabelle 4.4.: FPS-Werte und Detektionsraten bei Detektion in jedem Bild und bei jeder fünften Bildaufnahme

Modell	FPS			Detektionsrate (D/s)		
	n=1	n=5	Steigerung (%)	n=1	n=5	Differenz (D/s)
Csp	4,82	8,38	≈ 73,8 %	4,82	1,68	-3,14
Tiny	5,86	10,56	≈ 80,3 %	5,86	2,11	-3,75
Inria	4,41	9,38	≈ 113,3 %	4,41	1,88	-2,53
Caltech	2,15	6,65	≈ 209,3 %	2,15	1,33	-0,82
R-CNN	3,55	8,81	≈ 148,2 %	3,55	1,76	-1,79

## 4.6. Entscheidungslogik für Warnung und Bremsung

Da nun die Positionen der detektierten Personen, sowohl in Bild-, als auch in Fahrzeugkoordinaten bekannt sind, sowie die Bereiche festgelegt wurden, welche zu einer kritischen Situation führen können, wenn sich eine Person darin befindet, kann nun eine Logik implementiert werden, die die Entscheidung darüber trifft, ob ein Eingriff des Systems notwendig ist. Dazu soll eine zweistufige Reaktionsstrategie implementiert werden, die in Abhängigkeit vom Gefährdungsgrad der Situation aktiviert wird. In der ersten Stufe soll dazu eine Warnung in Form eines akustischen Signals, sowie eines kurzen Bremsrucks als haptischer Hinweis, erfolgen. Reagiert der Fahrer auf die Warnungen nicht ausreichend, sodass es dazu kommt, dass die Bedingungen der zweiten Stufe erfüllt werden, so wird eine Bremsung bis zum Stillstand des Fahrzeuges veranlasst. Die Unterscheidung zwischen den beiden Stufen erfolgt hierbei über die beiden zuvor erzeugten Fahrschläuche für den Warn- und den Notbremsbereich.

### 1. Einstufung der Gefahrensituation:

- Für jede detektierte Person wird überprüft, ob sie sich im definierten Warn- oder Notbremsbereich befindet. Das wird in MATLAB anhand der Funktion *inpolygon* umgesetzt. Damit wird ermittelt, ob sich eine der ermittelten Koordinaten in einem der Polygone befindet, welches die beiden Bereiche darstellen.
- Befindet sich eine Person im Warnbereich, wird ein Zähler (*isInLaneWarnCounter*) für diese Person erhöht. Sobald der Zähler einen definierten Schwellenwert erreicht (z. B. nach mindestens zwei Schleifendurchläufen), wird die Person endgültig als im Warnbereich befindlich (*isInLaneWarnFinal*) klassifiziert. Die Verfolgung der Detektionen innerhalb des Polygons wird anhand der IDs des Trackers ermöglicht und soll das Auslösen durch Fehldetektionen vermeiden.
- Falls sich eine Person zusätzlich im Notbremsbereich befindet und zuvor bereits im Warnbereich klassifiziert wurde, wird ein weiterer Zähler (*isInLaneCounter*) erhöht. Personen, deren Zähler einen weiteren Schwellenwert erreicht, werden schließlich als im kritischen Notbremsbereich (*isInLaneFinal*) betrachtet.

### 2. Warnung in der ersten Stufe:

- Sobald mindestens eine Person als im Warnbereich befindlich erkannt wird (*isInLaneWarnFinal*), wird eine Warnung ausgelöst, die ein akustisches Signal und einen kurzen Bremsruck umfasst. Dies dient als Hinweis für den Fahrer, die Situation zu entschärfen.

### 3. Automatische Notbremsung in der zweiten Stufe:

- Falls der Fahrer nicht ausreichend reagiert und eine Person als im Notbremsbereich erkannt wird, erfolgt eine automatische Bremsung bis zum Stillstand. Diese wird durch das Senden eines spezifischen CAN-Signals an die Bremsaktuatorik ausgelöst.
- Diese Reaktion dient als letzter Schritt, um eine Kollision zu vermeiden, falls der Fahrer nicht rechtzeitig auf die Warnung reagiert.

### 4. Kontinuierliche Aktualisierung:

- Die Erkennung und Klassifizierung erfolgt in Echtzeit über mehrere Frames hinweg. Dies stellt sicher, dass nur Personen berücksichtigt werden, die tatsächlich eine Gefährdung darstellen und sich konsistent im Warn- oder Notbremsbereich befinden.
- Die Detektionsdaten, einschließlich der Position und Geschwindigkeit der Personen, werden fortlaufend aktualisiert und in der Tracking-Struktur gespeichert.

### 5. Situationsabhängige Aktivierung:

- Beide Stufen sind an die aktuelle Fahrzeuggeschwindigkeit gekoppelt. Liegt die Geschwindigkeit bei null, wird keine Warnung oder Bremsung ausgelöst.

Diese Logik berücksichtigt dynamische Faktoren wie die Fahrzeuggeschwindigkeit, Lenkwinkel und die konsistente Anwesenheit von Personen in definierten Gefahrenbereichen. Durch die zweistufige Reaktionsstrategie wird die Sicherheit maximiert und gleichzeitig unnötige Eingriffe minimiert. Diese Entscheidungslogik

### 4.7. Integration in die Fahrzeugumgebung

Damit das System in der Praxis zur Anwendung kommen kann, muss eine Schnittstelle zwischen Fahrzeug und Algorithmus geschaffen werden, sowie die Hardware in das Versuchsfahrzeug eingebunden werden. Damit die Kamera zuverlässig gute Bilddaten liefern kann, muss diese an geeigneter Stelle positioniert werden. Um die Entfernungserfassung möglichst präzise zu halten ist daher eine möglichst hohe Einbauposition wünschenswert. Üblicherweise werden Frontkameras in Fahrzeugen auf Höhe des Rückspiegels hinter der Windschutzscheibe verbaut. Der Vorteil hier ist, dass die Kamera vor harten Umwelteinflüssen, wie Schmutz, Feuchtigkeit und Vereisung geschützt ist. Da die Windschutzscheibe frei von Vereisung und starken Verschmutzungen zu halten ist und dieser Bereich vom Scheibenwischer überwacht wird, ist hier eine freie Sicht der Kamera, ohne weiteren Aufwand, am ehesten zu gewährleisten. Da sich diese Einbauposition sehr gut eignet, wurde auch in diesem Fall die Kameraposition so gewählt, dass sie möglichst zentral und möglichst weit oben hinter der Windschutzscheibe verbaut wird. Dazu wurde die Kamera, wie in Abbildung 4.15 zu sehen, mithilfe einer Saugnapfhalterung neben dem Rückspiegelgehäuse an der Windschutzscheibe befestigt. Die Neigung der Kamera lässt sich durch die quaderförmige Gehäuseform sehr gut messen. Hierzu kann der Gyroskopsensor eines modernen Smartphones genutzt werden, um den Neigungswinkel mit zureichender Genauigkeit zu bestimmen.



Abbildung 4.14.: Kamerabefestigung an der Windschutzscheibe mithilfe einer Saugnapfhalterung

Um die beste Kameraposition zu erreichen, musste diese aufgrund des Neigungswinkels der Windschutzscheibe, über Kopf hängend verbaut werden. Daher müssen die Achsen der Kamera gespiegelt werden, um die richtige Orientierung der Kamerabilder zu erhalten. Das erfolgt in MATLAB, indem die Eigenschaften *ReverseX* und *ReverseY* des GigE-Kameraobjektes auf "True" gesetzt werden. Die Verkabelung der Kamera erfolgt wie gehabt über den Ethernet-Switch an den Host-PC. Die Stromversorgung des Host-PC kann entweder über das Bordnetz des Fahrzeuges oder über einen mobilen Akku erfolgen. Der verwendete Ouster LiDAR-Sensor zum Aufnehmen der Punktwolken im Prüfmodus wurde für eine möglichst freie Sicht oberhalb des Prüffahrzeuges mit einer entsprechenden Halterung für den Dachträger befestigt.

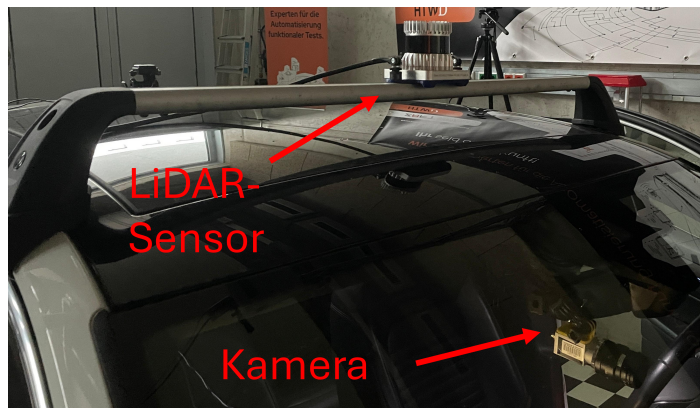


Abbildung 4.15.: Befestigung des Ouster LiDAR-Sensors auf dem Dachträger des Versuchsfahrzeugs

Da das Prüffahrzeug eine Straßenzulassung besitzt und somit Manipulationen des Betriebsbremssystems unzulässig sind, war ein direktes Ausführen des Notbremsbefehls über das Betriebsbremssystem des Fahrzeuges nicht möglich. Um dennoch eine Verzögerung des Fahrzeuges umzusetzen, wurde ein bereits zur Verfügung stehender Aktuator eingesetzt, welcher die elektrische Feststellbremse des Fahrzeuges betätigt, indem er mechanisch den entsprechenden Taster in der Mittelkonsole über einen Servomotor nach oben zieht. Angesteuert wird dieser Aktuator über CAN-Signale.

Um die Geschwindigkeits- und Lenkwinkelwerte eines Fahrzeuges zu erfassen, ist eine Schnittstelle zum CAN-Bus des Fahrzeuges erforderlich, um die von den Steuergeräten gesendeten Signale abzugreifen. Hierfür wird das VN1630A CAN-Interface von Vector Informatik eingesetzt. Dieses Interface ermöglicht das direkte Mitlesen der Signale vom CAN-Kabelstrang und kann über Universal Serial Bus (USB) mit dem Host-PC verbunden werden. Die *Vehicle Network Toolbox* stellt dabei eine Vielzahl vorgefertigter Funktionen zur Verfügung, die die Verarbeitung und Analyse der empfangenen CAN-Daten in MATLAB erleichtern. Um die CAN-Signale zu entschlüsseln und interpretieren zu können, liegen für die Versuchsfahrzeuge fahrzeugspezifische CAN-Datenbanken vor. Unter Angabe des Interfaces und des entsprechenden Kanals, über welchen die Daten am Interface empfangen werden, kann in MATLAB ein CAN-Channel erstellt werden. Dieser Kanal ist bezüglich der Baudrate des Netzwerkes und der Datenbank zu konfigurieren, um auswertbare Werte zu erhalten. Um nur die relevan-



**Abbildung 4.16.:** Realisierung der Verzögerung über einen CAN-gesteuerten Feststellbrems-Aktuator

ten CAN-Messages auszuwerten, kann mithilfe der CAN-Datenbank nach den spezifischen IDs der Lenkradwinkel- und Geschwindigkeitswerte gefiltert werden. Da die Sendefrequenz der Signale deutlich höher ist als die Verarbeitungsfrequenz des Algorithmus, werden die entsprechenden Werte über einen Schleifendurchlauf hinweg als Vektor zwischengespeichert und anschließend der Mittelwert gebildet, um zuverlässigere und weniger nervöse werte zu erhalten. Innerhalb jedes Schleifendurchlaufes wird anhand der gemittelten Werte der Geschwindigkeit und des Lenkwinkels, gemäß der Berechnungen aus Abschnitt 4.4, die Länge und die Krümmung beider Fahrschläuche ermittelt





---

## 5. Validierung des Systems

Um die verschiedenen Algorithmen miteinander vergleichen zu können, sollen die Objekterkennungsmethoden sowohl bezüglich ihrer Leistungsfähigkeit mit den bereits zuvor beschriebenen gängigen Vergleichsmetriken Präzision, Trefferquote und dem F1-Score validiert werden, als auch ihre Eignung für den Einsatz in einem Notbrems-Assistenzsystem in praktischen Fahrversuchen geprüft und verglichen werden.

### 5.1. Vergleich der Objektdetektoren

Zunächst sollen die verwendeten Detektoren auf Basis eines eigens erstellten und gelabelten Datensatzes angewandt werden, um mithilfe der im Abschnitt 3.5 beschriebenen Metriken einen Vergleich zwischen den Detektoren herstellen zu können. Damit soll zunächst ein erster Überblick über die Leistungsfähigkeit, sowie die Laufzeiten der ausgewählten Detektoren und deren verschiedenen Modellen geschaffen werden. Der Datensatz umfasst 143 Bilder, welche einen Umfang an Bilddaten beinhaltet, der neben gut belichteten Aufnahmen auch bewusst unter- und überbelichtete Bilder, sowie stark verdeckte Personen und Dummies abbildet, um besonders aussagekräftige Werte in Bezug auf die Robustheit der Modelle zu erhalten. Da besonders im Automotiven Anwendungsgebiet mit stark wechselnden Umgebungsbedingungen zu rechnen ist, sollte daher dieser Aspekt in gewisser Form mit in die Untersuchung einfließen. Die Bilder entstammen zum Teil eigenen Aufnahmen und zum Teil dem Karlsruhe Institute of Technology and Toyota Technological Institute (KITTI)-Personendatensatz [42]. In MATLAB stehen bereits vorgefertigte Funktionen zur Verfügung, welche das Berechnen der Bewertungsmetriken erleichtern. Diese stellen beispielsweise die Funktionen *evaluateDetectionPrecision* und *evaluateDetectionMissrate* dar. Diese Funktionen benötigen die Aussagen der Detektoren, sowie die Ground Truth-Daten als Eingabe und berechnen automatisiert die relevanten Metriken, wie Precision, Recall, FPPI und Missrate. Zusätzlich kann im Anschluss entsprechend der F1-Score berechnet werden und die benötigten Laufzeiten mithilfe der *tic* und *toc* -Befehle ermittelt werden. Desweiteren lassen sich Schwellwerte für den Prediction-Score, also die Aussagewahrscheinlichkeit, sowie für die Intersection over Union festlegen. Allerdings hat sich gezeigt, dass bereits die voreingestellten Standard-Schwellwerte von jeweils 0,5 Ergebnisse liefern, welche eine allgemeine Aussage über die Qualität der Detektionsvorhersagen der einzelnen Modelle repräsentieren. Für die Generierung der Vergleichswerte wurde daher ein MATLAB-Skript geschrieben, in welchem nacheinander zunächst beide Modelle des YOLO-Detektors und anschließend beide Modelle des ACF-Detektors auf dem Datensatz angewandt werden und die Metriken berechnet werden. Dieser Prozess wird dreimal wiederholt, um im Anschluss einen Mittelwert der drei Durchläufe für jeden Detektor zu errechnen. Somit sollen Abweichungen durch Ausreißer besser kompensiert werden. Die Ergebnisse aller Durchläufe, sowie die errechneten Mittelwerte werden als Textdatei gespeichert und können dem beigelegten Datenträger entnommen werden. Eine Übersicht der Ergebnisse mit den Standard-Schwellwerten ist in

Tabelle 5.1 dargestellt, wobei der jeweils beste erreichte Wert in grün und der schlechteste in rot dargestellt ist.

Tabelle 5.1.: Erreichte Leistungswerte der Detektoren bei jeweiligem Schwellwert von 0,5 für IoU und Aussagewahrscheinlichkeit

Detektor	YOLOv4	YOLOv4	ACF	ACF
Modell	tiny-yolov4	csp-darknet53	caltech-50x21	inria-100x41
Präzision	0,3733	0,7299	0,0260	0,5435
F1-Score	0,3104	0,5178	0,0331	0,4476
Fehlerrate	0,6612	0,3770	0,9706	0,4928
Laufzeit (s)	6,15	15,88	35,55	16,78

Aus den in Tabelle 5.1 dargestellten Ergebnissen lassen sich bereits Tendenzen ableiten, wie die Modelle im Vergleich zueinander abschneiden. Während YOLOv4 mit dem DarkNet-Backbone in sämtlichen Leistungsmetriken Bestwerte erzielt und somit das höchste Maß an Genauigkeit bei den Aussagen bietet, hat der YOLO-Detektor mit dem schmalen Modell 'tiny-yolov4-coco' die kürzeste Laufzeit von 6,15 Sekunden, womit es mindestens doppelt so schnell ist, wie alle anderen Modelle. Die auf dem YOLOv4-basierenden Detektoren haben hierbei einen klaren Vorteil, da diese auf der GPU des Test-Setups ausgeführt werden können, während die ACF-Detektoren lediglich die CPU des Rechners nutzen. Desweiteren sticht ins Auge, dass der ACF-Detektor auf dem Caltech-Datensatz trainiert, in sämtlichen Kategorien mit großem Abstand am schlechtesten Abschneidet, wobei dieser nicht nur sehr schlechte Erkennungswerte liefert, sondern auch die mit Abstand längste Rechenzeit benötigt.

Aus der grafischen Darstellung der erreichten Werte der Bewertungsmetriken in Abbildung 5.1 sind die Unterschiede zwischen den verschiedenen Modellen weitaus besser ersichtlich. Hier zeigt sich deutlich die Überlegenheit des YOLOv4-Detektors mit dem leistungsfähigeren DarkNet-backbone gegenüber den anderen Detektoren. Zu beachten ist, dass bei der Präzision und dem F1-Score ein hoher Wert für ein gutes Ergebnis spricht, während bei der Fehlerrate möglichst niedriger Wert anzustreben ist. Besonders aussagekräftig für die allgemeine Leistungsfähigkeit ist der F1-Score. Dieser spiegelt sowohl wieder, wie viele der tatsächlich vorhandenen Objekte erkannt wurden, aber auch wie viele der Aussagen des Detektors einem tatsächlich vorhandenen Objekt entsprechen. Ein Detektor, welcher zwar alle Objekte erkennt, aber zu viele Vorhersagen ausgibt und somit zu Falschauslösungen des Systems führen könnte, erreicht hier ebenso ein schlechtes Ergebnis, wie ein Detektor, welcher kaum vorhandene Objekte detektiert.

Die erreichten F1-Scores sind für diese Umsetzung besonders interessant im Zusammenhang mit der benötigten Detektionszeit, wie es in Abbildung 5.2 dargestellt ist. Hierbei stellen Werte, die möglichst weit in der oberen linken Ecke liegen die besten Ergebnisse dar. Diese Ecke repräsentiert die höchsten Genauigkeitswerte bei kürzester Rechenzeit. Auch hier stellen beide Modelle des YOLOv4-Detektors die besten Leistungen dar. Das Modell

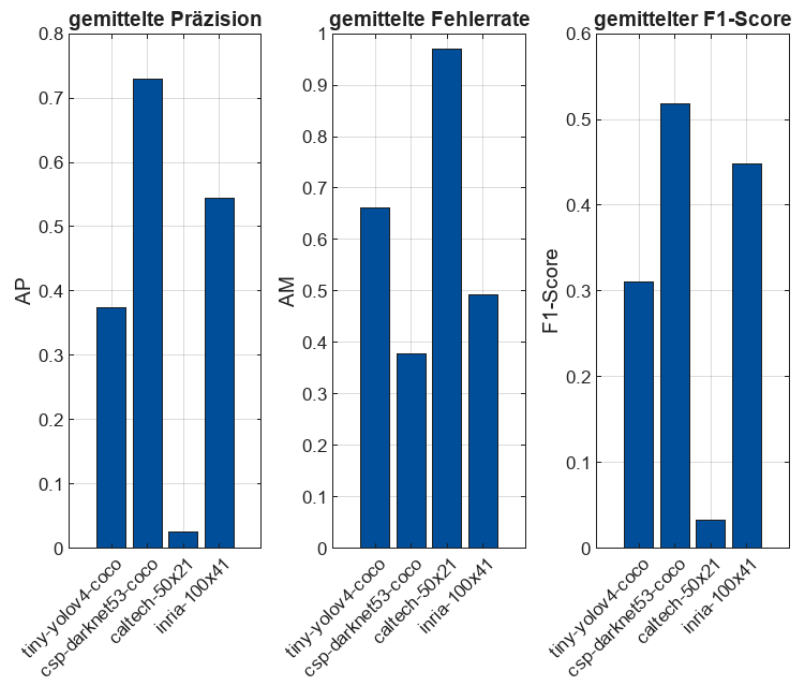


Abbildung 5.1.: Grafische Darstellung der gemittelten Werte der errechneten Bewertungsmetriken

'tiny-yolov4-coco' ist hierbei durch seine unübertroffene Schnelligkeit besonders interessant, während das Modell 'csp-darknet53-coco' die zweitschnellste Detektion bei den höchsten Genauigkeitswerten bietet.

Diese Metriken wurden ebenfalls für verschiedene Schwellwerte des Wahrscheinlichkeit-Scores und der IoU ermittelt. Dabei stellte sich heraus, dass bereits eine Anhebung des Schwellwerts für den Vertrauens-Score auf 0,55 bei gleichbleibendem IoU-Schwellwert erheblichen Einfluss auf die Ergebnisse hat. Während alle anderen Modelle annähernd gleiche Werte lieferten, versagte der ACF-Detektor 'inria-100x41' vollständig und lieferte keine Detektionen, die den Kriterien entsprachen. Daraus lässt sich schließen, dass dieser Detektor zwar verhältnismäßig präzise Aussagen treffen kann, jedoch nur mit einem geringen Vertrauens-Score und der Detektor sich also selbst bei sehr genauen Erkennungen von Objekten nicht wirklich sicher ist, ob die Aussagen korrekt sind. Dies kann in sicherheitskritischen Systemen ein Ausschlusskriterium eines solchen Detektors darstellen, ist jedoch bei weitem besser, als ein System, welches sehr ungenaue Aussagen trifft, sich dabei aber sicher ist, richtig zu liegen, so wie es beim Caltech-Modell der Fall ist. Dieser konnte bei einem Wahrscheinlichkeit-Schwellwert von 0,6 und gleichbleibendem IoU-Schwellwert keine entsprechenden Erkennungen mehr vorweisen. Da es bei dieser Umsetzung besonders auf eine präzise Lokalisierung der Objekte ankommt, da darüber auch die Entfernungsbestimmung und die Aussage darüber, ob eine Person sich in der Fahrspur des Fahrzeuges befindet, realisiert wird, soll hier ein besonderes Augenmerk auf hohe IoU-Werte gelegt werden.

Diese sind der Tabelle 5.2 zu entnehmen.

Erhöht man die Schwellwerte für den Wahrscheinlichkeit-Score der Aussagen und die IoU

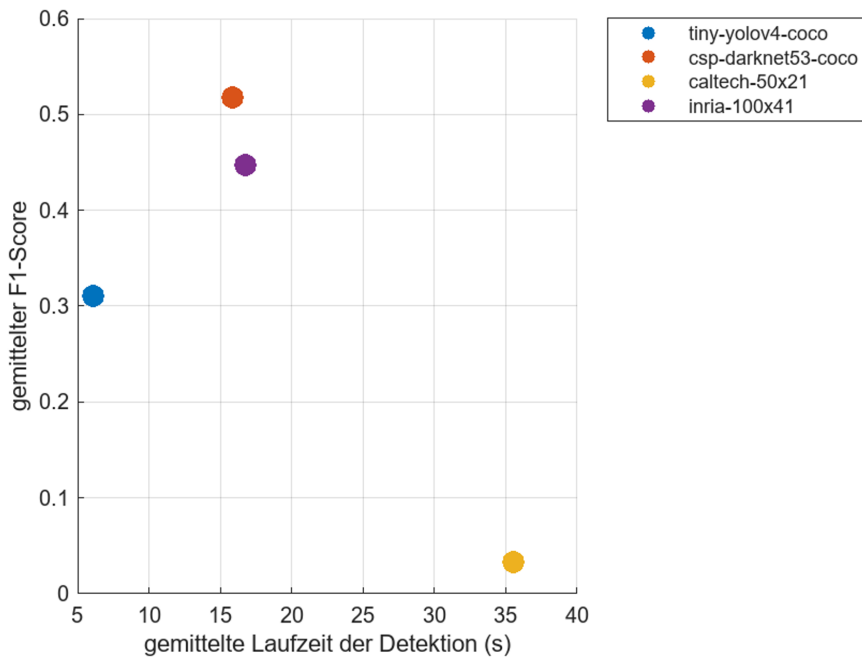


Abbildung 5.2.: Erreichter F1-Score in Zusammenhang zur benötigten Detektionszeit

Tabelle 5.2.: Erreichte Leistungswerte der Detektoren bei jeweiligem Schwellwert von 0,7 für IoU und Aussagewahrscheinlichkeit

Detektor	YOLOv4	YOLOv4	ACF	ACF
Modell	tiny-yolov4	csp-darknet53	caltech-50x21	inria-100x41
Präzision	0,2165	0,5608	0,0000	0,0000
F1-Score	0,2172	0,4482	0,0000	0,0000
Fehlerrate	0,8261	0,5910	0,0000	0,0000
Laufzeit (s)	5,70	14,36	36,47	17,41

jeweils auf 0,7, so ist den ermittelten Werten aus Tabelle 5.2 zu entnehmen, dass beide Modelle des ACF-Detektors keine Aussagen treffen konnten, die beide dieser Schwellwerte übersteigen. Das ist gerade bei dem Modell, welches auf dem INRIA-Datensatz basiert bemerkenswert, da dieses bei den festgelegten Schwellwerten von 0,5 sogar besser abschnitt, als der YOLO in der schmalen Variante mit dem Modell 'tiny-yolov4-coco'. Weiterhin erreicht YOLOv4 mit dem leistungsstärkeren Netz die besten Ergebnisse in der Genauigkeit. Die Unterschiede in Bezug auf die Laufzeiten bleiben verständlicherweise unverändert.

Aus den hier gewonnenen Erkenntnissen lässt sich ein erster Eindruck gewinnen, wie leistungsstark und schnell die Detektoren im Vergleich zueinander sind. Diese Werte basieren allerdings auf einem Datensatz, welcher zu großen Teilen sehr schwer detektierbare Objekte enthält, da viele Bilder überdurchschnittlich unter- oder überbelichtet sind und manche Zielobjekte so stark verdeckt sind, dass nur kleine Teile von ihnen wahrgenommen werden können. Diese Bilder spiegeln somit Situationen wieder, welche in herkömmlichen Anwendungen im Fahrzeug verhältnismäßig selten vorkommen. Dennoch können derartige Situationen auftreten, beispielsweise beim Verlassen oder Einfahren in einen Tunnel oder im Fall der

Verdeckung, wenn eine Person zügig hinter einem parkenden Fahrzeug hervortritt. Besonders in solchen Fällen müssen diese Systeme allerdings auch zuverlässig funktionieren, da diese Situationen genauso herausfordernd für einen menschlichen Fahrer sind und es hier schnell zum Versagen des Fahrers kommen kann. Genau in solchen Fällen unterscheidet sich eine gute Detektion von einem weniger robusten Modell. Aus den hier ausgearbeiteten Vergleichswerten lässt sich allerdings die Überlegenheit moderner, KI-gestützter Detektionsmethoden, welche auf künstlichen neuronalen Netzen basieren, gegenüber herkömmlichen Machine Learning (ML)-basierter Bildverarbeitungsmethoden zur Detektion, klar erkennen. Dennoch lässt sich mit diesen Ergebnissen noch keine endgültige Aussage treffen, welches dieser Systeme sich am besten für das Anwendungsgebiet eines Notbrems-Assistenzsystems eignet. Es kann sein, dass 'tiny-yolov4-coco' durch seine Schnelligkeit am besten im AEBS performt, es kann aber auch sein, dass beispielsweise 'csp-darknet53-coco' Personen durch seine hohe Genauigkeit und Robustheit bereits aus viel größerer Entfernung detektiert und somit einen Vorteil bei hohen Geschwindigkeiten bietet. Um eine Umfassende Aussage darüber treffen zu können, sind Realfahrversuche im Gesamtkontext des Notbremsalgorithmus unabdingbar.

### 5.2. Realfahrversuch

Anhand eines Realfahrversuches soll der umgesetzte Algorithmus auf seine Funktionsfähigkeit geprüft werden. Damit soll herausgefunden werden, wie gut das Gesamtsystem im praktischen Einsatz unter realitätsnahen Bedingungen performt.

#### 5.2.1. Prüfszenario

Als Prüfszenario soll ein grundlegendes Standard-Szenario, welches die Mindestanforderungen an ein solches System stellt, nachgebildet werden. Dazu wird mit einer vordefinierten Geschwindigkeit, möglichst lange und konstant auf ein statisches Prüfziel, einen Personendummy, welcher die Statur einer durchschnittlichen erwachsenen Frau darstellt, zugefahren, bis das System reagiert. Da das tatsächliche Verzögern des Fahrzeuges keinen relevanten Bestandteil des Eigenanteils dieser Umsetzung darstellt, sondern lediglich die vorhergehenden Prozesse, bis zum Auslösen des Bremsbefehls, sollen auch nur diese Teil der Untersuchungen sein. Die Realisierung des Bremsmanövers über die zuvor genannte Möglichkeit des Aktuators an der elektrischen Feststellbremse ist weiterhin kaum nachvollziehbar in Hinsicht der Konsistenz der erreichten Bremsverzögerungswerte, sowie möglicher Latenzen und Schwellzeiten der Bremskraft und könnte somit die Prüfergebnisse stark beeinflussen. Daher wird das Szenario so gestaltet, dass möglichst lange auf den Dummy zugefahren wird und so spät wie möglich ein Ausweichmanöver stattfindet, um erst nach Auslösen des Systems die Geschwindigkeit zu verringern. Um ein rechtzeitiges Ansprechen des Systems auf das Prüfziel zu gewährleisten, galt es entsprechende TTC-Schwellwerte für den Bremsbefehl und die Warnung des Fahrers festzulegen, welche denen in der Praxis verwendeten Werten

möglichst nahe kommen und dennoch eine sichere Versuchsdurchführung ohne Kollisionen und Schäden zu ermöglichen. Eine durch ein AEBS ausgelöste Bremsung darf laut den gesetzlichen Vorgaben der ECE-Regelung 131 nicht zu früh auslösen. Speziell wird hier ein TTC-Wert von weniger als 3,0 Sekunden festgelegt, ab wann das System eingreifen darf [43]. Daher sollte auch hier die Bremsung nicht vorher ausgelöst werden. Um einen konkreten Wert festzulegen, wurde daher die Berechnung des Anhalteweges für die festgelegte Prüfgeschwindigkeit nach der allgemeingültigen Faustformel zur Orientierung genutzt. Der Anhalteweg beträgt bei einer Geschwindigkeit von 30 km/h demnach 12 Meter. Da sich die Position der Kamera nicht an der Front des Fahrzeuges befindet, muss der horizontale Abstand zwischen Fahrzeugbeginn und Kameraposition von 2,10 m zusätzlich berücksichtigt werden. Um zusätzliche Sicherheitsreserven zu gewährleisten, wurde der Wert auf 15m aufgerundet. Demnach ergibt sich nach Formel 2.2 ein TTC-Schwellwert von 1,8 Sekunden für das Auslösen der Bremsung. Für die Zeitdifferenz zwischen Bremsbefehl und Warnung wurde für die Versuchsfahrten ein unrealistisch hoher Wert gewählt, um die Leistungsfähigkeit des Systems auf lange Distanzen zu testen. Dazu wurde eine Zeitdifferenz von 4 Sekunden festgelegt, womit sich der TTC-Schwellwert für die Warnung zu 5,8 Sekunden ergibt. Das ergibt bei 30 km/h eine Entfernung von etwa 48,33 Metern, was nahezu die gesamte Prüfgerade umfasst. Durch diese Erkenntnisse der Funktionalität des Systems auf größeren Entfernungen, lässt sich somit auch auf die Eignung des System bei höheren Geschwindigkeiten schließen, die im Rahmen dieser Prüfumgebung nicht getestet werden können.

### 5.2.2. Testdurchführung

Um die Testergebnisse möglichst vergleichbar zu halten und möglichst geringe Abweichungen durch Veränderungen der Umgebungsbedingungen, wie beispielsweise der Lichtsituation oder verschiedenen Witterungen zu gewährleisten, ist wichtig die Fahrversuche zeitlich möglichst nah beieinander durchzuführen. Da der umgesetzte Algorithmus jedoch auch zum Teil sehr rechenaufwendige Modelle zur Detektion vorsieht, wurde dennoch darauf geachtet, zwischen den Messungen kurze, konstante Abkühlzeiten der Hardware zu gewährleisten, damit ein Modell, welches an der Leistungsgrenze des Systems arbeitet keinen thermischen Einfluss auf die nächste Messfahrt hat und das nächste Modell möglicherweise schneller in den Bereich der thermalen Leistungsbegrenzung kommt, durch welche die benötigte Laufzeit des Algorithmus negativ beeinflusst werden könnte. Daher wurden zunächst einige Läufe durchgeführt, um auch für die ersten Prüffahrten die selben Bedingungen zu schaffen, wie bei den letzten Messfahrten. Zusätzlich wurde die Reihenfolge der angewandten Modelle so gewählt, dass eine Messreihe jeweils alle Modelle einmal testet, bevor die nächste Messreihe aufgenommen wird, um die Umgebungsbedingungen innerhalb einer Messreihe möglichst konstant zu halten. Dadurch soll ein möglichst fairer Vergleich gewährleistet werden. Um mögliche Ausreißer und Abweichungen identifizieren zu können, sollen für jedes Testszenario pro Detektormodell drei Messfahrten stattfinden. Da der Auslösezeitpunkt des Systems

stark Geschwindigkeitsabhängig ist, war es wichtig eine möglichst identische und konstante Geschwindigkeit innerhalb der verschiedenen Messfahrten zu garantieren. Dafür wurde in dem Testfahrzeug der Speed-Limiter, also eine Geschwindigkeitsbegrenzung, aktiviert. Diese ist bei dem Prüffahrzeug erst ab einer Geschwindigkeit von 30 km/h einsetzbar. Daher wurde die Geschwindigkeiten für die Testfahrten auf diese Geschwindigkeit festgelegt. Da das Prüffeld nur eine begrenzte Größe bietet, wurde als Prüfgerade die Diagonale des Areal genutzt, um ein möglichst langes und konstantes Zufahren auf das Prüfziel zu ermöglichen. Um bereits auf der Diagonale möglichst schnell die Prüfungsgeschwindigkeit von 30 km/h zu erreichen, war es notwendig, zuvor eine kleine Schleife zu fahren, um bereits beim Einbiegen auf die Diagonale eine gewisse Startgeschwindigkeit zu erreichen. Anschließend galt es möglichst gerade auf den Dummy zuzufahren, um die Ausrichtung des Fahrschlauches permanent und möglichst früh auf den Dummy zu lenken, um ein Auslösen des Systems zu ermöglichen, sobald der Dummy in den kritischen Bereich gerät.

### 5.3. Auswertung der LiDAR-Punktwolken

Um Aussagen über die Leistungsfähigkeit der Modelle im Kontext eines Notbrems-Assistenzsystems treffen zu können, müssen zunächst die aufgezeichneten LiDAR-Punktwolken ausgewertet werden, um Informationen über die Entfernung des Zielobjektes im Auslösemoment des Systems zu erhalten. Dazu wurde ein MATLAB-Skript geschrieben, welches dem beigelegten Datenträger entnommen werden kann.

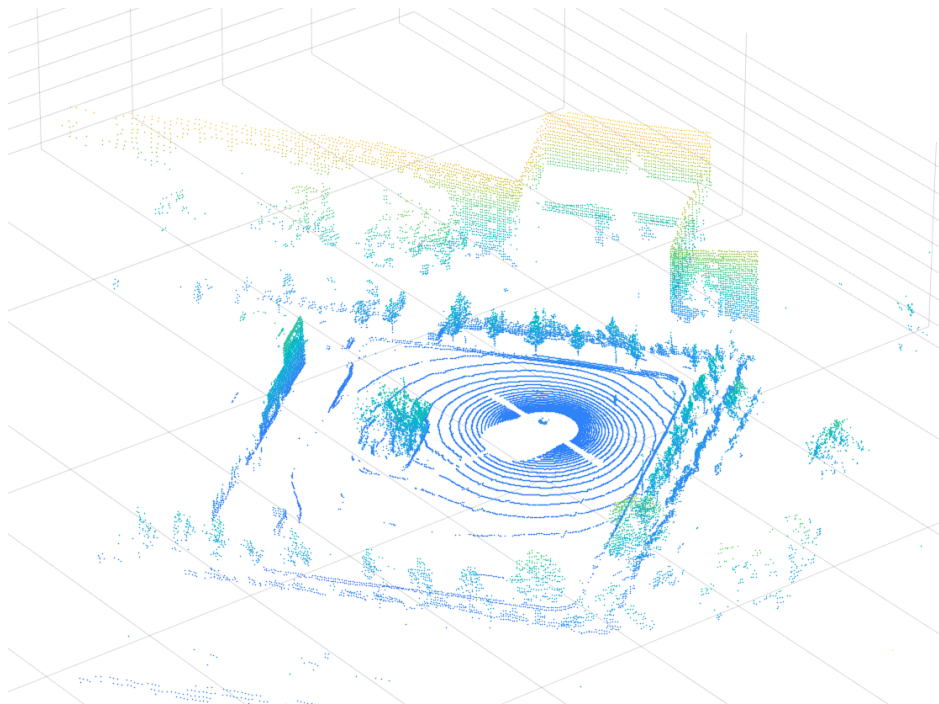


Abbildung 5.3.: Darstellung der ungefilterten Punktwolke einer Messfahrt

Wie in Abbildung 5.3 dargestellt, enthält die unbearbeitete und ungefilterte Punktwolke eine Vielzahl an Punkten von Bereichen, die für die weitere Auswertung nicht von Relevanz

sind und das Extrahieren des Zielobjekts erschweren. Daher wird in diesem Skript zunächst eine Region of Interest (ROI), also ein relevanter Bereich festgelegt, der jeweils 4 m zu jeder Seite des Fahrzeuges, also in Y-Richtung und 30 m in X-Richtung, also vor das Fahrzeug ragt. In Z-Richtung reicht dieser Bereich bis 1,1 m unterhalb und 1 m oberhalb des Sensors, welcher auf einer Höhe von etwa 1,4 m auf dem Dach des Versuchsfahrzeuges angebracht ist. Durch die Begrenzung in negativer Z-Richtung wird sichergestellt, dass keine Punkte, die vom Boden reflektiert werden, zu Störungen in weiteren Verarbeitungsprozessen führen. Somit fließen nur die Punkte in die weitere Auswertung mit ein, die dem Bereich entsprechen, wo der Dummy zu erwarten ist.

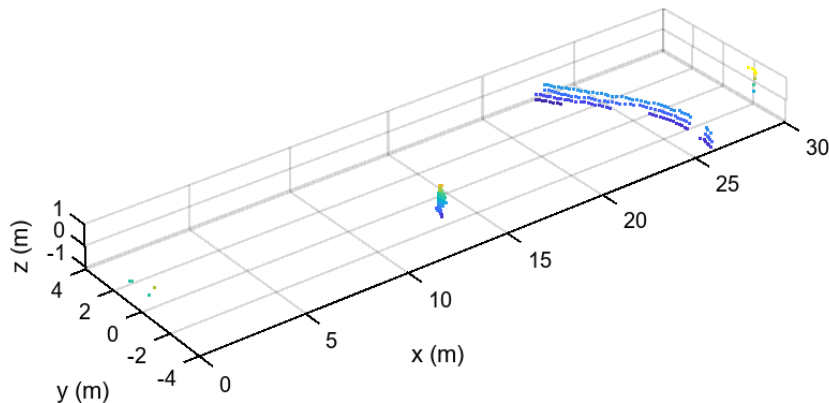


Abbildung 5.4.: Darstellung der gefilterten Punktwolke

In Abbildung 5.4 ist zu erkennen, dass nach dem Entfernen der Punkte, die nicht der ROI entsprechen, bereits der Dummy als signifikantes Objekt in den verbleibenden Punkten zu erkennen ist. Aus dieser gefilterten Punktwolke lassen sich die Punkte, welche der Reflexion des Dummies entsprechen noch besser extrahieren. Dazu wurden zwei unterschiedliche Methoden angewandt, die unabhängig voneinander die Punkte, die zum Dummy gehören herausfiltern und die Entfernung zwischen Sensorkoordinaten-Ursprung und dem Dummy ermitteln. Somit erhält man zwei unabhängig voneinander ermittelte Entfernungswerte, welche miteinander verglichen werden können, um bei zureichender Ähnlichkeit beider Werte einen Mittelwert zu berechnen. Dies dient zum einen dazu die Genauigkeit zu erhöhen und vor allem um bei einer automatisierten Auswertung vieler Punktwolken leichter Fehldetektionen durch abweichende Entfernungswerte zu identifizieren.



### 5.3.1. Clustering-Verfahren zur Personendetektion

Zum einen wird ein klassisches Clustering-Verfahren eingesetzt, das Punktansammlungen basierend auf spezifischer Höhe und Breite sowie einer festgelegten Punktzahl und -dichte herausfiltert, die den Zielobjekten entsprechen. In Abbildung 5.4 ist gut zu erkennen, dass neben der Punktanhäufung, welche dem Dummy entspricht, nur die Schutzplanke hinter dem Dummy als weitere Anhäufung von Punkten in dem Bereich liegt, sowie einige vereinzelte Punkte. Die vereinzelten Punkte lassen sich hierbei mithilfe eines Schwellwerts für die Mindestanzahl an Punkten des gewünschten Clusters herausfiltern. Um nun noch die Punkte der Schutzplanke zu eliminieren, können Schwellwerte für Höhe und Breite des gewünschten Clusters genutzt werden, um durch die unterschiedlichen geometrischen Ausdehnungen beider Punktanhäufungen zwischen den Objekten zu differenzieren. Der Dummy ist sowohl deutlich höher als die Begrenzung, als auch weitaus schmaler.

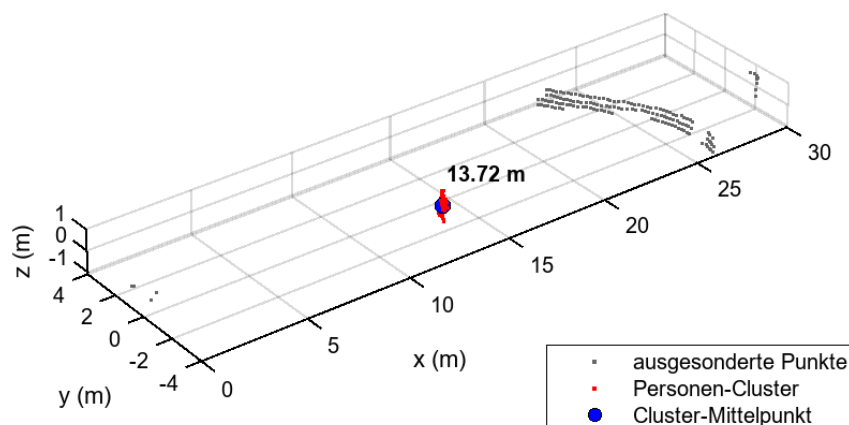


Abbildung 5.5.: Darstellung der erkannten Person mittels Clustering-Verfahren

Im Anschluss kann für die Punktanhäufung, welche den Kriterien entspricht, der Mittelpunkt, bzw. der Schwerpunkt aller zugehörigen Punkte, berechnet werden, um einen Punkt zu erhalten der zur Entfernungsbestimmung dient. In Abbildung 5.5 kann man sehen, dass lediglich die Punkte, welche der Anhäufung des Dummys entsprechend rot markiert sind und alle weiteren Punkte grau dargestellt sind.

### 5.3.2. Bildbasierte Punktwolkenauswertung

Als weiteres Verfahren zur Erkennung des Dummys aus der Punktwolke und der anschließenden Entfernungsermittlung wird eine Methode, welche auf verschiedenen Bildverarbeitungsmethoden beruht, eingesetzt. Dazu wird aus der gefilterten Punktwolke zunächst ein

sogenanntes BEV-Image, also zweidimensionales Bild, welches die Punkte in einer Vogelperspektive darstellt, erzeugt. Dadurch, dass das Bild quadratisch ist, die ROI jedoch deutlich größere Ausdehnung in x-Richtung, als in y-Richtung aufweist, ist die Darstellung verzerrt, was in der Entfernungsermittlung berücksichtigt werden muss. Dieses BEV-Bild ist am Beispiel der Punktwolke der ersten Messfahrt in Abbildung 5.6 zu sehen. Das weitere Vorgehen ist dem des Clustering-Verfahrens sehr ähnlich, basiert jedoch auf einer zweidimensionalen Abbildung.

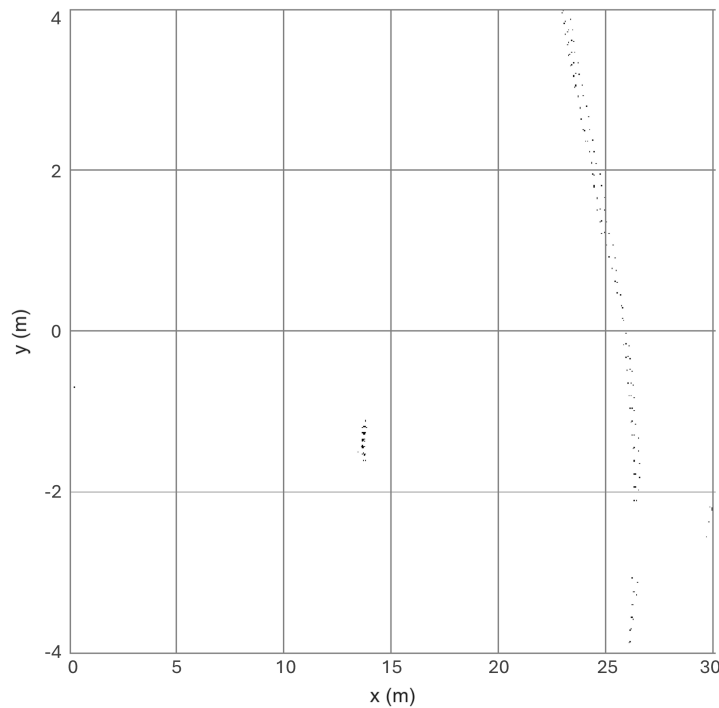
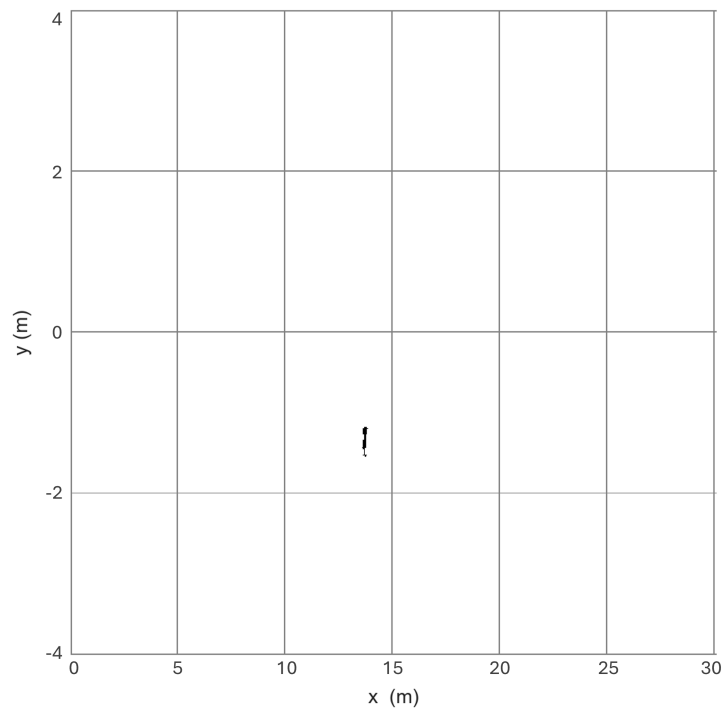
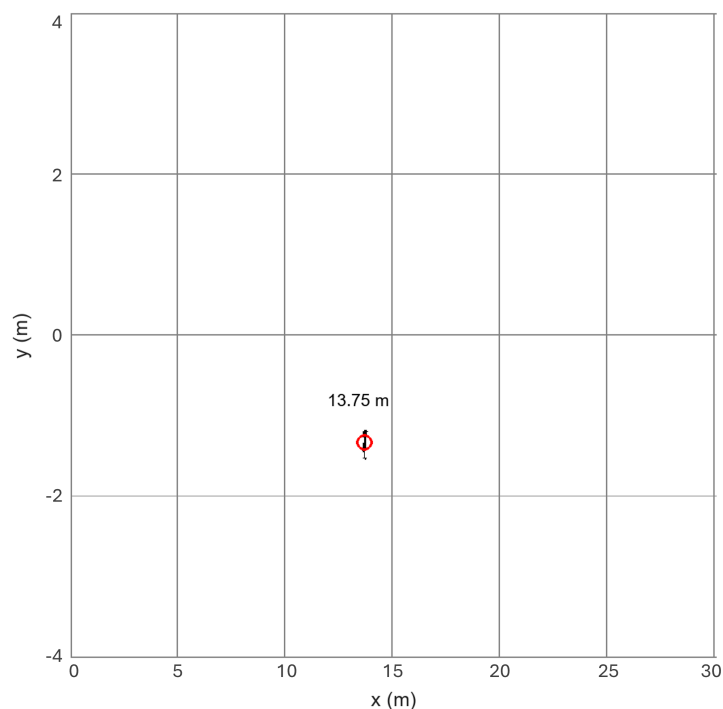


Abbildung 5.6.: Erzeugtes BEV-Bild der gefilterten Punktwolke

Die Abbildung der Reflexionen ist zunächst noch sehr zerklüftet und enthält viele Punkte, welche nicht zum Zielobjekt gehören. Mithilfe eines Schwellwertes für die Mindestanzahl zusammenhängender Pixel, können die Bildpunkte, welche die Reflexionen der Schutzplanke darstellen herausgefiltert werden, da diese deutlich weniger zusammenhängend positioniert sind, als die des Dummys. Anschließend können mithilfe einer morphologischen Operation die Lücken zwischen den Pixeln geschlossen werden, um eine zusammenhängende Pixelfläche des zu erkennenden Objektes zu erhalten. Es entsteht somit eine Abbildung aus der Vogelperspektive, welche lediglich die Reflexionen des Dummys als zusammenhängende Pixelfläche in einer sonst von Störpunkten befreiten Umgebung, wie in Abb. 5.7, darstellt. Für diese Pixelfläche lässt sich nun ebenfalls ein Mittelpunkt errechnen, dessen Position zunächst in Bildkoordinaten vorliegt. Durch die vorherige Vorgabe der ROI-Maße in Raumkoordinaten und der bekannten Anzahl an Pixelreihen und Pixelspalten des BEV-Bildes lässt sich nun ein Zusammenhang zwischen Bildkoordinaten und Raumkoordinaten des LiDAR-Sensors herstellen und somit der Abstand des Dummys vom Sensor ermitteln. Diese Methode liefert für die Punktwolke der ersten Messfahrt einen Entfernungswert des Dummys von 13.75 m zum Sensor. Damit weicht dieser Wert gerade einmal 3 cm von dem



**Abbildung 5.7.:** Bereinigtes BEV-Bild nach morphologischer Schließung der verbleibenden Pixelfläche



**Abbildung 5.8.:** Entfernungsbestimmung im BEV-Bild

Entfernungswert von  $13,72 \text{ m}$  ab, welcher über die Clustering-Methode ermittelt wurde. Dies kann für diesen Anwendungsfall als zureichend genau betrachtet werden. Mithilfe dieser Methode lässt sich also ein auf unabhängige Weise berechneter zweiter Wert für die Entfernung zwischen LiDAR-Sensor und Zielobjekt berechnen, womit sich auch direkt im Prüfgeschehen eine automatisierte Auswertung der Messwerte realisieren lässt, wobei eine

falsche Detektion eines der beiden Verfahren schneller erkannt werden kann.

#### 5.4. Auswertung der Fahrversuche

Nach der Auswertung der Punktwolken sind nun die Entfernungswerte zwischen Dummy und Sensor-Mittelpunkt zum Zeitpunkt des Auslösens des Notbremssignals bekannt. Anhand dieser lassen sich die verschiedenen Algorithmen nun in Relation zueinander setzen. Zusätzlich können die aufgezeichneten Kennwerte, wie die erreichten Schleifendurchläufe pro Sekunde, die Fahrzeuggeschwindigkeiten und die Zeitdifferenzen zwischen Warnung und Bremsbefehl dazu dienen, die Werte besser nachzuvollziehen und die Vergleichbarkeit der Testfahrten zu validieren, sowie weitere Erkenntnisse über die Leistungsfähigkeit der Modelle zu erhalten. Dazu soll zunächst eine Übersicht der gemittelten Werte über die Versuchsfahrten jedes Detektors einen Überblick über die Ergebnisse ermöglichen. Sämtliche Einzelwerte der Messfahrten sind dem Anhang A zu entnehmen.

Tabelle 5.3.: gemittelte Messwerte der Versuchsfahrten

<b>Detektor Modell</b>	<b>YOLOv4</b>	<b>YOLOv4</b>	<b>ACF</b>	<b>ACF</b>
	<b>tiny-yolov4</b>	<b>csp-darknet53</b>	<b>inria</b>	<b>caltech</b>
Entfernung (m)	13,48	13,37	13,08	10,94
FPS	10,73	9,41	9,89	6,59
Zeitdifferenz (s)	3,4741	3,8961	3,4863	2,204
Geschwindigkeit (km/h)	31,4	31,37	31,57	31,27

In Tabelle 5.3 sind die gemittelten Werte der jeweiligen drei Messfahrten jedes Modells dargestellt. Die gemittelten Geschwindigkeitswerte weisen dabei nur sehr geringe Unterschiede auf und auch in den Einzelmesswerten gab es keine größeren Abweichungen, weshalb die Geschwindigkeiten der verschiedenen Messfahrten als konstant angenommen werden können.

Die gemittelten Geschwindigkeitswerte zeigen zunächst, dass alle Werte sehr nah beieinander liegen und auch innerhalb der Messreihen gab es aufgrund des aktivierten Geschwindigkeitsbegrenzers lediglich vernachlässigbar kleine Differenzen. Die Werte liegen dabei aber konsistent oberhalb von 30 km/h, was durch ein Überschwingen der Geschwindigkeitsregelung des Limiters verursacht wird, da aufgrund der geringen Distanz auf dem Prüffeld ein schnelles Beschleunigen auf die Prüfgeschwindigkeit notwendig ist. Dennoch ist dieses Verhalten sehr konsistent und ermöglicht somit eine Vergleichbarkeit der Messreihen.

Betrachtet man nun die ermittelten Entfernungen, welche einen Bezug zum Zeitpunkt des Bremsbefehls darstellen, so kann man erkennen, dass alle Detektoren recht nah beieinander liegen, lediglich das Caltech-Modell des ACF-Detektors erreichte geringere Werte, was für einen späteren Auslösezeitpunkt spricht. Begutachtet man die Fehlerbalken des Diagramms, so sind auch hier nur geringe Abweichungen zwischen den Messungen zu erkennen. Tiny-YOLO und Caltech liefern hierbei die konstantesten Ergebnisse, während YOLO, mit DarkNet53 als rechenaufwändigstes Modell, die höchsten Abweichungen aufweist. Das liegt

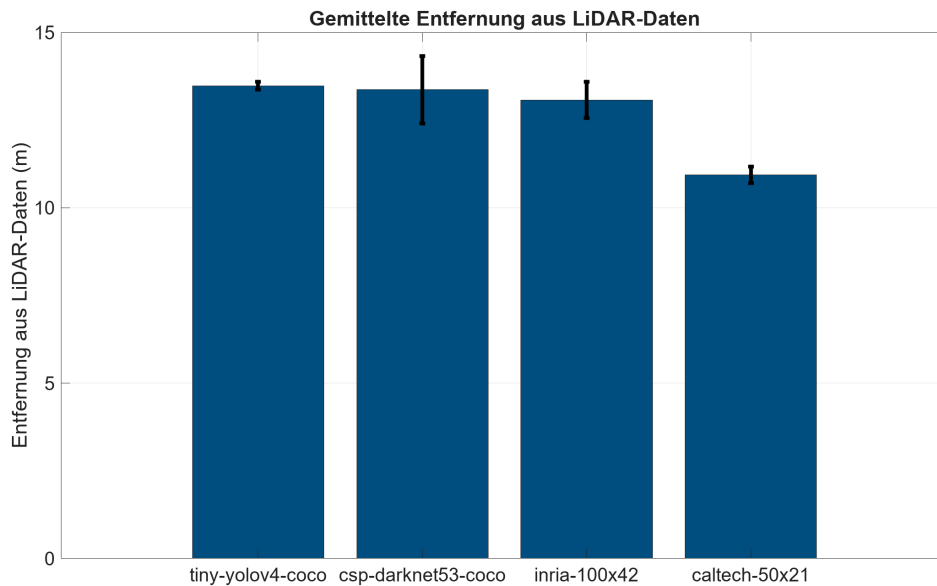


Abbildung 5.9.: Gemittelte Distanzen zwischen Fahrzeug und Prüf-Dummy im Auslösemoment des Bremsbefehls

vermutlich darin begründet, dass dieses Modell die Verarbeitungs-Hardware am meisten auslastet und die Performance des Systems unter der Verwendung dieses Modells stark von der Dauer abhängt, wie lange der Algorithmus auf dem System ausgeführt wird. Mit steigender Zeit sinkt hierbei die Anzahl der verarbeiteten Bilder pro Sekunde, da die verwendete Hardware die Leistung immer weiter begrenzt.

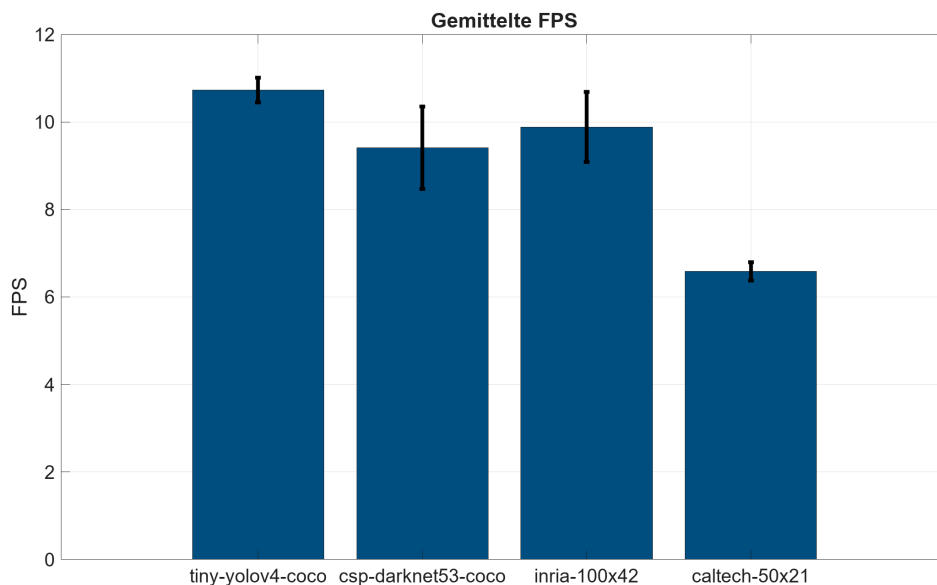


Abbildung 5.10.: Verarbeitete Bilder pro Sekunde während der Messfahrten

Schaut man auf die in Abbildung 5.10 dargestellten erreichten durchschnittlichen Frames per Second (FPS) der Versuchsfahrten, so spiegeln diese einen Zusammenhang zu den ermittelten Laufzeiten wieder, welche im Abschnitt 5.1 bei der Anwendung auf einem Datensatz gemessen wurden, wobei zu beachten ist, dass die FPS den Kehrwert der dort ermittelten Laufzeiten darstellt. Die Unterschiede sind hierbei jedoch weitaus geringer, da die Detektion im Gesamtkontext nur noch einen gewissen Teil der Laufzeiten ausmacht. Zu erkennen

ist, dass die anhand der Fehlerbalken dargestellten Abweichungen bei den Rechenzeiten innerhalb der Versuchsfahrten eine Proportionalität zu denen in Abbildung 5.9 aufweisen. Dadurch lässt sich ein Zusammenhang zwischen der Verarbeitungsgeschwindigkeit und der dem Auslösezeitpunkt des Systems vermuten.

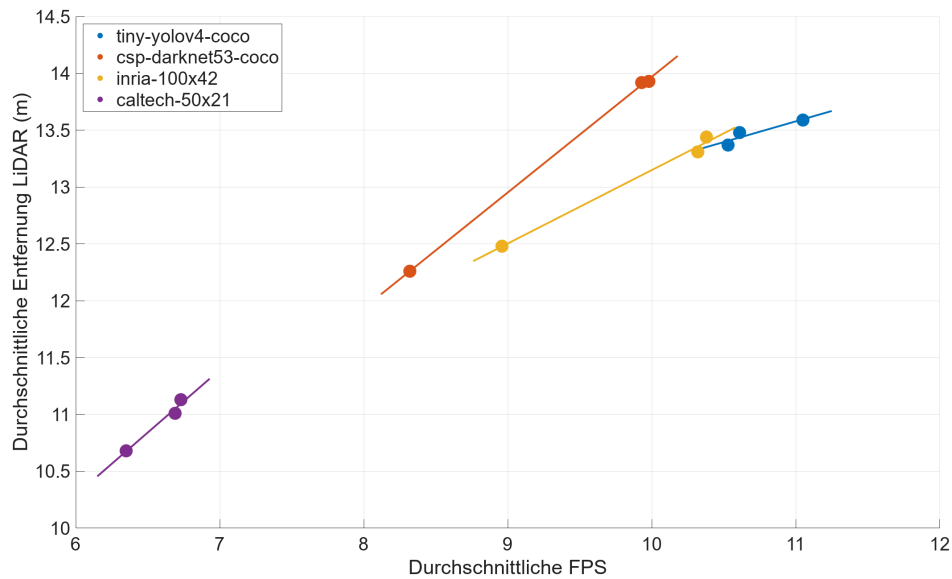


Abbildung 5.11.: Regression der Ermittelten Entfernungen der Einzelmesswerte über der Verarbeitungsgeschwindigkeit

In Abbildung 5.11 sind die gemessenen Entfernungen über der Verarbeitungsgeschwindigkeit in FPS dargestellt. Anhand der Regressionsgeraden lässt sich dabei ein annähernd linearer Zusammenhang zwischen dem Auslösezeitpunkt und den durchschnittlichen FPS innerhalb der Messungen eines Detektormodells, aber auch über alle Detektormodelle hinweg, erkennen. Diese Relation ist noch deutlicher anhand einer Darstellung der gemittelten Entfernungswerte zu erkennen.

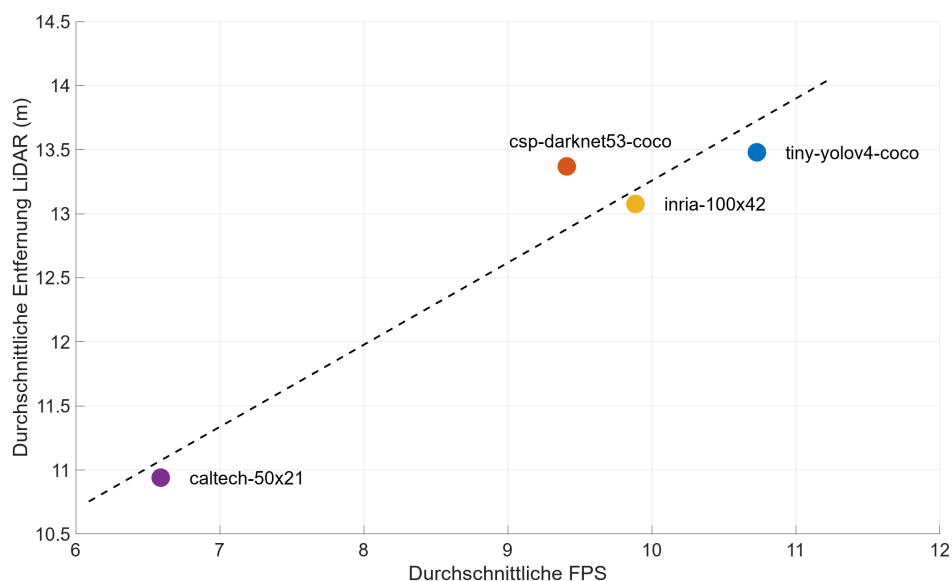


Abbildung 5.12.: Regression der gemittelten Entfernungen über der Verarbeitungsgeschwindigkeit

Die Darstellung in Abbildung 5.12 zeigt sehr deutlich, dass ein annähernd lineares Verhalten zwischen der Anzahl der verarbeiteten Bilder und den Entfernungswerten besteht. Lediglich das DarkNet-Modell des YOLO-Detektors weicht davon ab und liegt weit oberhalb der Regressionsgerade. Dieses Modell erreicht somit weitaus bessere Ergebnisse für die verhältnismäßig geringe FPS-Zahl, was mit großer Wahrscheinlichkeit auf die besonders hohe Erkennungsgenauigkeit des Detektors zurückzuführen ist.

Um die Ergebnisse in einen zeitlichen Bezug zu bringen, lassen sich aus den Entfernungswerten, unter Berücksichtigung der Geschwindigkeitswerte, die TTC-Werte für den Moment der Auslösung des Bremsbefehls berechnen. Diese sind in Abbildung 5.13 dargestellt.

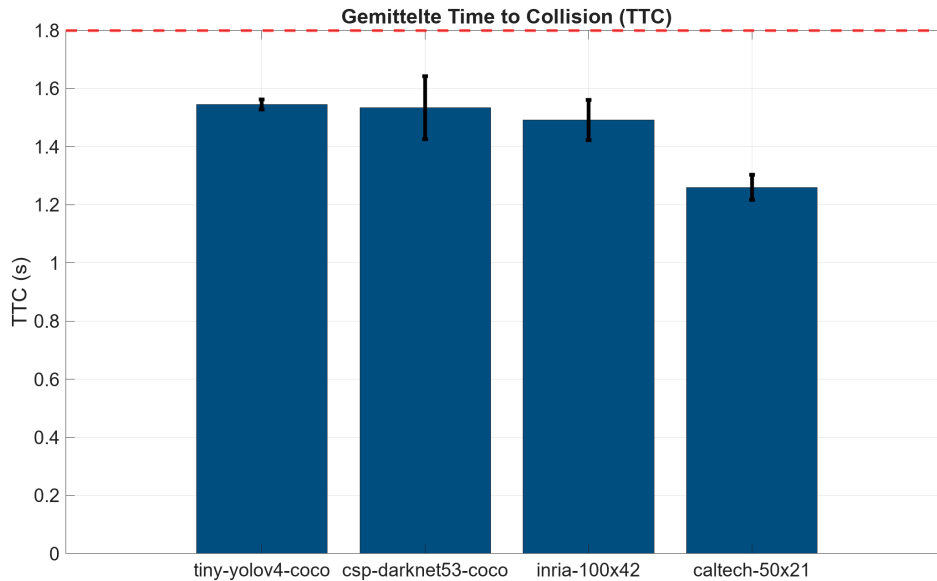


Abbildung 5.13.: Verbleibende TTC-Werte im Moment des Bremsbefehls

Die rot gestrichelte Linie symbolisiert dabei den Wert des Schwellwerts der TTC, bei welchem das System frühestens auslösen soll. Zu sehen ist, dass alle Algorithmen dabei eine Verzögerung von über 0,2 Sekunden aufweisen. Dies lässt sich durch mehrere Zusammenhänge erklären. Zum einen entsteht grundlegend durch die verhältnismäßig geringe Verarbeitungsfrequenz bereits eine recht niedrige Abtastrate der Umgebung, wodurch derartige Latenzen entstehen. Zum anderen sieht die Entscheidungslogik des Algorithmus vor, dass die Detektion sich mindestens 2 Schleifendurchläufe im kritischen Bereich befinden muss, um Falschauslösungen des Systems zu vermeiden. Dadurch ist ebenfalls eine gewisse Latenz unabdingbar. Desweiteren besteht aufgrund der Verarbeitungszeiten innerhalb der Schleife eine weitere Latenz zwischen Bildaufnahme und der letztendlichen Aufzeichnung der LiDAR-Daten. Diese Abweichungen verhalten sich jedoch größtenteils konsistent, weshalb die Anpassung des TTC-Schwellwertes um einen Korrekturwert, welcher der ermittelten Latenz entspricht, Abhilfe verschaffen kann. Dennoch sind die Größenordnungen der Latenzen in einem Bereich angesiedelt, welcher besonders in Anbetracht der erreichten FPS-Zahlen, vertretbar ist. Dennoch kann diese Latenzzeit weiter gesenkt werden, indem schnellere Verarbeitungszeiten erreicht werden.

Die Betrachtung der errechneten TTC-Werte im Moment der Warnung soll nun eine Aus-

sage über das Potential der Detektoren über größeren Erkennungsdistanzen ermöglichen. Dazu wurden die Zeitdifferenzen zwischen dem Moment des Bremsbefehls und der initialen Warnung gemessen. Addiert man nun diese Zeitdifferenz auf den TTC-Wert der Bremsung, so erhält man den TTC-Wert im Moment der Warnung, mit welchem sich unter Einbezug der Geschwindigkeit eine Aussage über die Entfernung zum Dummy im Moment der Auslösung des Warnsignals treffen lässt. Diese Werte sind der 5.4 zu entnehmen.

Tabelle 5.4.: TTC-Warn-Werte für aller Messfahrten

Detektormodell	MF 1 (s)	MF 2 (s)	MF 3 (s)	Mittelwert (s)
tiny-yolov4-coco	4,454	5,203	5,402	5,020
csp-darknet53-coco	5,478	5,304	5,263	5,348
inria-100x42	5,390	4,531	5,013	4,978
caltech-50x21	3,414	3,611	3,367	3,464

Die Werte geben einen Aufschluss auf die Reichweite der Modelle. Hierbei zieht sich, dass das Darknet-Modell mit der höchsten Erkennungsgenauigkeit am besten abschneidet. Zusammenfassen lässt sich sagen, dass YOLO-Detektoren, welche auf tiefen neuronalen Netzen basieren einen großen Leistungsvorteil gegenüber den ACF-Detektoren aufzeigen, nicht nur in der Detektionsgenauigkeit, sondern auch in der Verarbeitungsgeschwindigkeit. Desweiteren ließ sich Schlussfolgern, dass sowohl eine hohe Erkennungsgenauigkeit, als auch eine schnelle Verarbeitungszeit notwendig ist, um eine Zuverlässige Personendetektion für einen Notbremsassistenten zu implementieren. Dabei ließ sich ein proportionales Verhältnis zwischen dem Auslösezeitpunkt und der erreichten fps der Detektoren erkennen. Generell zeigt sich die starke Weiterentwicklung dieser Technologien in den letzten Jahren, was sehr gut an dem Leistungsunterschied der beiden Ansätze ersichtlich wird.

## 5.5. Vergleich zur LiDAR-Personenerkennung

Abschließend soll aufgrund der aktuell recht starken Verbreitung von LiDAR-Sensorik zur Umfelderkennung im Automobilbereich ein Vergleich der Umsetzbarkeit einer Personenerkennung aus Bild- und LiDAR-Daten erfolgen. Dabei soll zusätzlich auf die Eignung beider Sensortechnologien im ADAS-Kontext eingegangen werden. Dazu dient zunächst die Betrachtung der unterschiedlichen Beschaffenheit der jeweiligen Daten, die beide Sensoriken liefern. Während die Bilddaten der Kamera eine detaillierte zweidimensionale Projektion der realen Welt in Form eines Rasters aus Bildpunkten darstellen, welche auch Farbinformationen enthalten können, liefert der LiDAR-Sensor eine zumeist weniger detaillierte und informationshaltige, dafür jedoch dreidimensionale Abbildung der Umwelt.

Dadurch ist die grundlegende Objekterkennung aus LiDAR-Punktwolken deutlich einfacher, da durch die Dreidimensionalität der Daten auf triviale Weise, anhand von Gradientenanalyse der Koordinaten, zwischen Objekten im Vordergrund und dem Hintergrund unterschieden werden kann. Während die Unterscheidung zwischen Objekt und Hintergrund aus visuellen



Informationen für den Menschen spielend einfach scheint, ist dieser Prozess für Recheneinheiten, aufgrund der fehlenden dritten Dimension, weitaus komplexer, als bei Punktwolken und mit rechenaufwändigen Algorithmen und statistischen Verfahren verbunden.

Somit reichen oftmals einfache Cluster-Methoden, damit aus Punktwolken nicht nur auf das Vorhandensein eines Objektes geschlossen werden kann, sondern zeitgleich die Position dessen im Raum bestimmt werden kann. Da die Tiefeninformationen der Sensorik für viele Fahrerassistenzsysteme essenziell sind und eine einzelne Kamera diese nicht so präzise und zuverlässig liefern kann, wie ein Laserscanner, bietet diese Technologie, daher viele Vorteile für diesen Einsatzzweck. In Bezug auf die Rechenanforderung kann eine Objekterkennung in LiDAR-Daten somit deutlich einfacher umgesetzt werden, als über Bilddaten, schlichtweg aufgrund der Datenmenge und Beschaffenheit. Begrenzender Faktor ist hierbei die Abtastrate des Sensors von maximal 20 Hz.

Da es speziell bei einer Personenerkennung allerdings viel mehr auf den Informationsgehalt der Abbildungen ankommt, soll hier ein Vergleich der Daten beider Sensoriken, für die Eignung aufgezeigt werden. Daher wurde bei dem Versuchsaufbau aus dem Abschnitt 4.3.5 zur Überprüfung der Genauigkeit der Methode zur Entfernungsermittlung ebenfalls bei jeder geprüften Entfernung eine LiDAR-Punktwolke von dem Dummy aufgenommen. Verwendet wurde dazu der Ouster OS1-64, welcher auch zur Aufnahme der Punktwolken der Versuchsfahrten genutzt wurde. Die Punktwolken wurden auf der maximalen Auflösung des Sensors von 2048 Punkten pro Kanal pro Bild, bei einer Drehrate von 10 Hz zur präziseren Darstellung aufgenommen. Die Punktwolken wurden im Anschluss ausgewertet, indem die Punkte, welche die Reflexionen des Dummys darstellen, herausgefiltert wurden. Dadurch ließ sich auch die Zahl der Punkte bestimmen, welche den Dummy darstellen. Zusätzlich ließen sich in MATLAB, anhand eines zuvor maskierten Bildes, durch Binarisierung, die Bildpunkte bestimmen, welche auf dem Dummy liegen. Abbildung 5.14 zeigt die Methoden zur Ermittlung der Anzahl der abbildenden Punkte beider Verfahren für einen Objektabstand von 5 Metern. Die ermittelten Werte sind in Tabelle 5.5 zusammengetragen.

Tabelle 5.5.: Vergleich der darstellenden Punkte eines Objektes in LiDAR- und Bilddaten

Entfernung	Punkte LiDAR	Bildpunkte
5 m	415	73.402
10 m	111	—
15 m	56	10.972
20 m	29	—
25 m	19	4.198
30 m	11	—
35 m	9	1.383

Aus den zusammengetragenen Werten in Tabelle 5.5 lässt sich der Unterschied der räumlichen Auflösung beider Technologien gut erkennen. Dabei ist zu beachten, dass die Bildaufnahmen zwar mit der Brennweite von 16mm aufgenommen wurde, wodurch entfernte Objekte deutlich höher aufgelöst werden, als mit beispielsweise einer 8mm Optik, dennoch

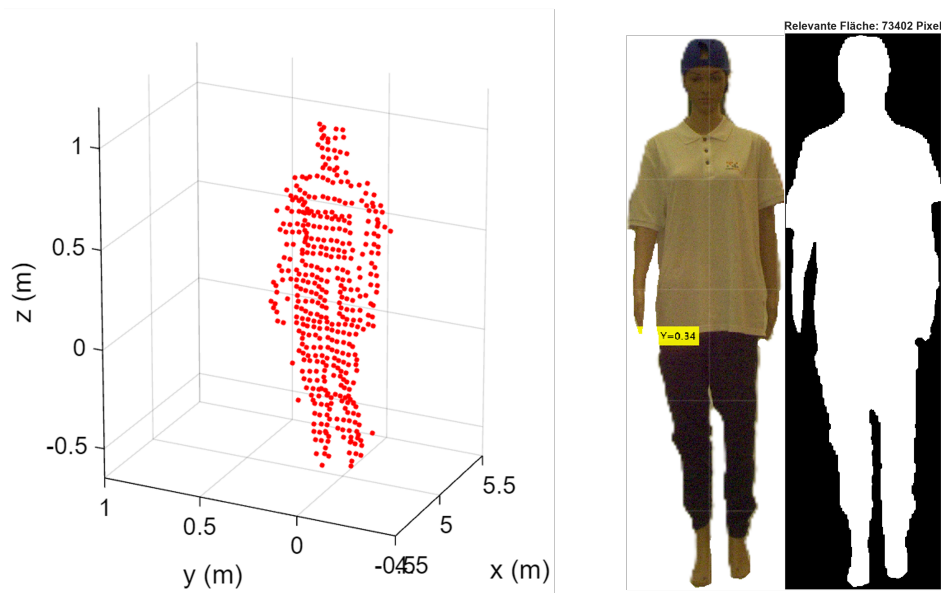


Abbildung 5.14.: Verfahren zur Ermittlung der abbildenden Punkte eines Objektes in LiDAR- und Bilddaten

würde sich die räumliche Auflösung maximal halbieren. Somit wäre der Unterschied dennoch erheblich, in Anbetracht dessen, dass die Kameraauflösung von 1920x1200 Pixeln verhältnismäßig gering ist und technisch weitaus mehr möglich ist, wohingegen die Auflösung des Ouster-Sensors im Vergleich heutzutage eher im hochauflösenden Bereich anzusiedeln ist. Besonders, da in Entfernungen um die 30 Meter, welche im Hinblick auf den Einsatz im ADAS-Bereich viel zu gering ist, gerade einmal etwa 10 Punkte auf dem Dummy liegen. Ein Kinder-Dummy in derselben Entfernung wurde in der Punktwolke mit etwa 3 Punkten abgebildet womit eine zuverlässige Detektion nicht möglich ist.

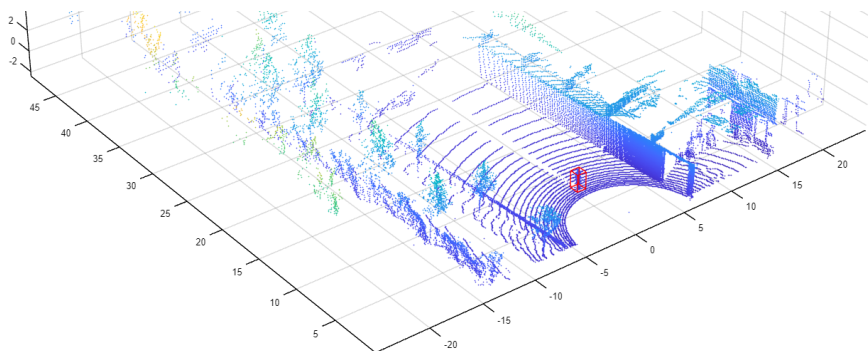


Abbildung 5.15.: Detektion in LiDAR-Punktwolken mithilfe von complex-yolov4

Desweiteren wurde für die Auswertung der Punktwolken eine KI-basierte LiDAR-Objekterkennung in Erwägung gezogen, welcher die Punktwolke in ein BEV-Bild umwandelt und mithilfe eines YOLO-Detektors diese Bilder analysiert und auswertet. Dieser nennt sich '*complex-yolov4*' und funktionierte sehr zuverlässig für die Erkennung von Fahrzeugen und Personen im näheren Umfeld, jedoch für die Auswertung der Punktwolken der Messfahrten hat dieses System versagt, da schlichtweg zu wenig Punkte auf dem Dummy lagen und die Erkennung somit nicht möglich war. Somit kann geschlussfolgert werden, dass LiDAR eine sehr

gute Möglichkeit zur Objekterkennung bietet, jedoch oftmals die Auflösung der derzeit verfügbaren Sensoren nicht ausreicht, um kleinere Objekte in weiterer Entfernung zuverlässig detektieren zu können. Eine Klassifikation der Objekte ist ebenfalls weitaus komplexer, da Objekte häufig nur durch ihre Breite und Höhe unterschieden werden können. Somit kann beispielsweise eine am Straßenrand stehende Mülltonne schnell mit einem Kind verwechselt werden, aufgrund ähnlicher geometrischer Eigenschaften. Hingegen ist die Erkennung über Bilddaten weitaus komplexer, jedoch über die reichhaltigere Datenstruktur, durch Farbinformationen und die in der regel höhere Auflösung ist die Klassifikation nach einer erfolgten Lokalisierung eines Objektes deutlich präziser möglich. Die Festlegung auf eine der Sensortechnologie, vermeidet daher viel Potential, welches die Kombination beider Technologien aufgrund der Vorteile ihrer jeweiligen Datenstrukturen bieten. Gerade im Bereich von Notbremssystemen, wo es auf eine präzise Objekterkennung und Klassifizierung ankommt und dennoch hoch genaue Entfernungswerte notwendig sind, bietet die Kombination dieses beiden Technologien eine wertvolle Symbiose für die Umfelderkennung.



---

## 6. Zusammenfassung und Ausblick

In dieser Arbeit wurde ein umfassendes Konzept zur Entwicklung eines monokularen Personenerkennungssystems für einen Notbremsassistenten vorgestellt und umgesetzt. Ziel war es, ein System zu schaffen, das mithilfe einer einzigen Kamera potenziell gefährliche Verkehrssituationen frühzeitig erkennt und daraufhin einen automatischen Notbremsbefehl auslöst, um die Sicherheit der Fahrzeuginsassen und anderer Verkehrsteilnehmer, insbesondere ungeschützter Verkehrsteilnehmer wie Fußgänger, zu erhöhen.

Zur Umsetzung des Systems wurde sowohl eine geeignete Hardware- als auch eine Softwarestruktur entwickelt. Die Auswahl und Integration der Kameratechnologie, sowie die Implementierung der Objekterkennungsalgorithmen standen hierbei im Fokus. Für die Objekterkennung wurden verschiedene Verfahren untersucht und verglichen, um eine möglichst hohe Erkennungsgenauigkeit bei gleichzeitig niedrigen Verarbeitungszeiten zu erreichen. Hierbei zeigte sich die Überlegenheit moderner KI-basierter Ansätze.

Die Validierung des Systems erfolgte anhand von Testfahrten, bei denen die Funktionsfähigkeit der monokularen Personenerkennung sowie der Entscheidungslogik des Notbremsassistenten unter Verwendung der zuvor getesteten Objekterkennungsmethoden überprüft wurde. Die Ergebnisse zeigen, dass das System in der Lage ist, potenzielle Kollisionssituationen zuverlässig zu erkennen und angemessene Notbremsmanöver einzuleiten. Die Auswertung der Versuchsdaten zeigte darüber hinaus, dass die umgesetzte Lösung trotz der stark verminderten Systemkomplexität eine vielversprechende Grundlage für eine zukünftige Weiterentwicklung bietet.

Weiterhin erfolgte ein Vergleich der Kamera-basierten Personenerkennung mit einem LiDAR-basierten System. Dieser Vergleich verdeutlichte die Stärken und Schwächen der jeweiligen Technologien. Während die monokulare Kamera besonders gut zur Klassifikation von Objekten geeignet ist, zeigte der LiDAR-Sensor Vorteile bei der präzisen Abstandsmessung. Eine zukünftige Erweiterung des umgesetzten Systems mit einer LiDAR-basierten Objekterkennung könnte daher die Funktionalität und Zuverlässigkeit des Systems weiter erhöhen. Im Ausblick lassen sich mehrere Ansätze für weitere Forschungs- und Entwicklungsarbeiten ableiten, die an dieser Stelle aufgezeigt werden sollen:

- Die Untersuchung hat gezeigt, dass die Verarbeitungsfrequenz einen großen Einfluss auf die Leistung des Systems hat. Eine Überführung des Algorithmus in eine schnellere, systemnahe Programmiersprache könnte daher Vorteile bringen.
- Die Implementierung einer weniger anfälligen Entfernungsmessung, die nicht stark von engen Randbedingungen abhängig ist. Dafür könnte der Einsatz einer weiteren Sensortechnologie erforderlich sein.
- Die Implementierung einer zuverlässigen Geschwindigkeitsbestimmung der detektierten Objekte, um den Bremszeitpunkt besser an die relative Geschwindigkeit beider potenzieller Kollisionsgegner anzupassen. Zudem könnte dies ermöglichen, bereits im

Voraus zu bestimmen, ob ein Fußgänger die Fahrbahn betreten wird oder diese möglicherweise bereits verlassen hat, bevor das Fahrzeug die Kollisionsstelle erreicht. Ein verbessertes Tracking-Verfahren könnte hier die Lösung bieten.

Zusammenfassend lässt sich sagen, dass die entwickelte monokulare Personenerkennung für einen Notbremsassistenten eine solide Grundlage für die Integration in Fahrzeugumgebungen darstellt. Das System bietet Potenzial für den Einsatz in Forschung und Lehre sowie für weitergehende Entwicklungsarbeiten. Mit gezielten Optimierungen und Erweiterungen könnte das System einen wesentlichen Beitrag zur Erhöhung der Verkehrssicherheit leisten und damit den Zielen der Vision Zero näherkommen

---

## Literatur- und Quellenverzeichnis

- [1] L. Stark:  
Quantifizieren der Vision Zero. Hrsg. von L. Stark. 1. Aufl. AutoUni - Schriftenreihe. Springer Vieweg Wiesbaden, 2022. ISBN: 978-3-658-40636-3. DOI: <https://doi.org/10.1007/978-3-658-40636-3>.
- [2] o.V.:  
Handelsblatt - WHO-Bericht - Jedes Jahr 1,35 Millionen Verkehrstote weltweit. Website. 2018. URL: <https://www.handelsblatt.com/politik/international/who-bericht-jedes-jahr-1-35-millionen-verkehrstote-weltweit/23731714.html> (besucht am 09.11.2024).
- [3] o.V.:  
RELAI-Projekt zeigt Möglichkeiten des Computer-Sehens. Webseite. URL: <https://www.safetywissen.com/object/A11/A11.arn738750ssvv7hkt3767252x04y8s63827980852/safetywissen> (besucht am 12.11.2024).
- [4] H. Winner und S. Hakuli und G. Wolf:  
Handbuch Fahrerassistenzsysteme. Hrsg. von H. Winner und S.Hakuli und F. Lotz und C.Singer. 3. Aufl. ATZ/MTZ-Fachbuch. Springer Vieweg Wiesbaden, 2015. ISBN: 978-3-658-05733-6. DOI: <https://doi.org/10.1007/978-3-658-05734-3>.
- [5] Andreas Jordan:  
Bildsensoren: So funktionieren sie. Website. 2017. URL: <https://www.fotomagazin.de/test-technik/wissen/bildsensoren-so-funktionieren-sie/> (besucht am 27.06.2024).
- [6] C. Neumeyer:  
Farbsehen – Wie und warum wir die Welt farbig wahrnehmen. Hrsg. von C.Neumeyer. 1. Aufl. Springer Spektrum Berlin, Heidelberg, 2024. ISBN: 978-3-662-69340-7. DOI: <https://doi.org/10.1007/978-3-662-69341-4>.
- [7] Viktor Bulova:  
Licht- und Farbwahrnehmung. Website. 2001. URL: [https://homepages.thm.de/~hg10013/Lehre/MMS/SS01\\_WS0102/Farbmodelle/Kapitel/Kapitel2.html#Marke22](https://homepages.thm.de/~hg10013/Lehre/MMS/SS01_WS0102/Farbmodelle/Kapitel/Kapitel2.html#Marke22) (besucht am 05.09.2024).
- [8] o.V.:  
Weitere Linsenformen. Website. 2018. URL: <https://www.leifiphysik.de/optik/optische-linsen/ausblick/weitere-linsenformen> (besucht am 03.09.2024).
- [9] o.V.:  
Linsenformen. Website. 2018. URL: <https://www.leifiphysik.de/optik/optische-linsen/grundwissen/linsenformen> (besucht am 03.09.2024).

- [10] Lars Theiß:  
Warum ein 50 mm Objektiv? – Die Normalbrennweite. Website. 2021. URL: <https://www.fotomagazin.de/test-technik/wissen/warum-ein-50-mm-objektiv-die-normalbrennweite/> (besucht am 03.09.2024).
- [11] Anselm F. Wunderer:  
Fotografie - klipp und klar. 1. Aufl. mitp Verlag, 2015. ISBN: 9783958450417. URL: [https://www.oreilly.com/library/view/fotografie-klipp/9783958450417/Text/final\\_stage2\\_03.htm](https://www.oreilly.com/library/view/fotografie-klipp/9783958450417/Text/final_stage2_03.htm).
- [12] o.V.:  
Wikipedia - Verzeichnung. Website. URL: <https://upload.wikimedia.org/wikipedia/commons/0/0a/Verzeichnung3.png> (besucht am 18.11.2024).
- [13] Laurenz Wuttke:  
Künstliche Neuronale Netzwerke: Definition, Einführung, Arten und Funktion. Website. 2024. URL: <https://datasolut.com/neuronale-netzwerke-einfuehrung/> (besucht am 02.09.2024).
- [14] o.V.:  
Botschaftsaufbau. Website. URL: <https://www.caneasy.de/caneasyhilfe/botschaftsaufbau.htm> (besucht am 11.09.2024).
- [15] o.V.:  
Automatisierungsstufen. Website. URL: [https://www.kba.de/DE/Themen/Marktueberwachung/Produktpruefungen/AutomatisiertesAutonomesFahren/Automatisierungsstufen/Automatisierungsstufen\\_node.html](https://www.kba.de/DE/Themen/Marktueberwachung/Produktpruefungen/AutomatisiertesAutonomesFahren/Automatisierungsstufen/Automatisierungsstufen_node.html) (besucht am 29.09.2024).
- [16] J. Redlich et al.:  
Assistenzsysteme machen Autofahren sicherer, verbreiten sich aber nur langsam. Webseite. 2022. URL: <https://www.gdv.de/gdv/medien/medieninformationen/assistsenzsysteme-machen-autofahren-sicherer-verbreiten-sich-aber-nur-langsam-73164> (besucht am 12.06.2024).
- [17] T. Hummel et al.:  
Fahrerassistenzsysteme - Ermittlung des Sicherheitspotenzials auf Basis des Schadensgeschehens der Deutschen Versicherer. Forschungsbericht. 2011. URL: <https://www.udv.de/resource/blob/78756/0ffa5e1dc48329809a4aa0d69499e43f/12-fahrerassistenzsysteme-data.pdf> (besucht am 19.06.2024).
- [18] Euro NCAP.:  
Euro NCAP - TEST PROTOCOL – AEB Car-to-Car systems. Prüfbericht. 2023. URL: <https://www.euroncap.com/media/79864/euro-ncap-aeb-c2c-test-protocol-v43.pdf> (besucht am 12.06.2024).



- [19] Euro NCAP.:  
Euro NCAP - TEST PROTOCOL – AEB/LSS VRU systems. Prüfbericht. 2024. URL: <https://www.euroncap.com/media/80156/euro-ncap-aeb-lss-vru-test-protocol-v451.pdf> (besucht am 12.06.2024).
- [20] Euro NCAP.:  
Euro NCAP. Webseite. URL: <https://www.euroncap.com/en/> (besucht am 12.11.2024).
- [21] auto mobil - das VOX-Automagazin:  
Notbremsassistenten im Ländervergleichstest - Welches Auto schneidet am besten ab? | auto mobil. YouTube. 2024. URL: <https://www.youtube.com/watch?v=Inrp8ZbWw6Y> (besucht am 05.10.2024).
- [22] o.V.:  
Time to Collision definition. Website. o.J. URL: <https://www.lawinsider.com/dictionary/time-to-collision> (besucht am 09.06.2024).
- [23] S. Naseralavi et al.:  
„A general formulation for time-to-collision safety indicator“. In: *Proceedings of the ICE - Transport* 166 (Okt. 2013), S. 294–304. DOI: 10.1680/tran.11.00031.
- [24] Robert Bosch GmbH.:  
Multifunktionskamera. Website. 2024. URL: <https://www.bosch-mobility.com/de/loesungen/kamera/multifunktionskamera/> (besucht am 10.11.2024).
- [25] Continental.:  
Multifunktions Mono Kamera MFC527. Website. 2024. URL: <https://www.continental-automotive.com/de/komponenten/kameras/multi-function-mono-camera-mfc527.html> (besucht am 10.11.2024).
- [26] o.V.:  
100GigE-Flächenkameras der Zenith-Serie. Website. 2024. URL: <https://emergentvisiontec.com/de/products/zenith-hz-100gige-cameras-rdma-area-scan/> (besucht am 12.08.2024).
- [27] o.V.:  
Weiterführende Informationen zu Brennweite und Bildfeld. Edmund Website. URL: <https://www.edmundoptics.de/knowledge-center/application-notes/imaging/understanding-focal-length-and-field-of-view/> (besucht am 15.11.2024).
- [28] o.V.:  
Definition - field of view (FOV). TechTarget Webseite. URL: <https://www.techtarget.com/whatis/definition/field-of-view-FOV> (besucht am 15.11.2024).

- [29] o.V.:  
Ouster - OS1 - Mid-Range High-Resolution Imaging Lidar. Datasheet. URL: <https://data.ouster.io/downloads/datasheets/datasheet-revd-v2p0-os1.pdf> (besucht am 05.11.2024).
- [30] o.V.:  
MATLAB Mathematik. Grafiken. Programmierung. Website. URL: <https://de.mathworks.com/products/matlab.html> (besucht am 25.07.2024).
- [31] o.V.:  
MATLAB Dokumentation - yolov4ObjectDetector. Website. URL: <https://www.mathworks.com/help/vision/ref/peopledetectoracf.html#d126e349786> (besucht am 06.05.2024).
- [32] o.V.:  
MATLAB Dokumentation - yolov4ObjectDetector. Website. URL: <https://www.mathworks.com/help/vision/ref/yolov4objectdetector.html> (besucht am 06.05.2024).
- [33] Bin Yang et al.:  
Aggregate channel features for multi-view face detection. Abstract. 2014. URL: <https://doi.org/10.48550/arXiv.1407.4023> (besucht am 11.11.2024).
- [34] o.V.:  
Anchor Boxes for Object Detection. MathWorks Help Center. URL: <https://www.mathworks.com/help/vision/ug/anchor-boxes-for-object-detection.html> (besucht am 11.06.2024).
- [35] Piotr Dollár et al.:  
„Fast Feature Pyramids for Object Detection“. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.8 (2014), S. 1532–1545. DOI: 10.1109/TPAMI.2014.2300479.
- [36] Adrian Rosebrock:  
Intersection over Union (IoU) for object detection. Website. 2016. URL: <https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/> (besucht am 09.09.2024).
- [37] o.V.:  
Google Maps - HTW Dresden K-Gebäude. Google Maps. URL: [https://www.google.de/maps/place/HTW+Dresden+K-Geb%C3%A4ude/@51.0346505,13.7396313,150m/data=!3m1!1e3!4m6!3m5!1s0x4709c5a6ce259cfd:0xcee999449e140e7e!8m2!3d51.03512!4d13.73889!16s%2Fg%2F1115j9t2pq?entry=ttu&g\\_ep=EgoyMDIOMTEExMi4wIKXMDSOASAFQAw%3D%3D](https://www.google.de/maps/place/HTW+Dresden+K-Geb%C3%A4ude/@51.0346505,13.7396313,150m/data=!3m1!1e3!4m6!3m5!1s0x4709c5a6ce259cfd:0xcee999449e140e7e!8m2!3d51.03512!4d13.73889!16s%2Fg%2F1115j9t2pq?entry=ttu&g_ep=EgoyMDIOMTEExMi4wIKXMDSOASAFQAw%3D%3D) (besucht am 05.10.2024).

- [38] o.V.:  
Visual Perception Using Monocular Camera. MathWorks Help Center. URL: <https://www.mathworks.com/help/driving/ug/visual-perception-using-monocular-camera.html> (besucht am 11.06.2024).
- [39] o.V.:  
Technische Daten. BMW i3 (120 Ah). Datenblatt. 2018. URL: <https://www.press.bmwgroup.com/deutschland/article/attachment/T0284828DE/421717&ved=2ahUKEwjograJzKeJAxWNgf0HHVL8FNUQFnoECDYQAQ&usg=A0vVaw27p0fz1TmQnNR0cDmfZC8V> (besucht am 27.06.2024).
- [40] Dima Lisin:  
Detect and Track Multiple Faces. MATLAB File Exchange. 2024. URL: <https://www.mathworks.com/matlabcentral/fileexchange/47105-detect-and-track-multiple-faces> (besucht am 12.11.2024).
- [41] MathWorks Inc.:  
vision.PointTracker System Object. Online-Dokumentation. 2012. URL: <https://www.mathworks.com/help/vision/ref/vision.pointtracker-system-object.html> (besucht am 12.11.2024).
- [42] o.V.:  
KITTI - Object Detection Evaluation 2012. Website. 2012. URL: [https://www.cvlibs.net/datasets/kitti/eval\\_object.php?obj\\_benchmark=2d](https://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=2d) (besucht am 09.11.2024).
- [43] Wirtschaftskommission für Europa der Vereinten Nationen (UN/ECE):  
Regulation No. 131 concerning automated driving systems. Regelung. 2021. URL: [https://www.unece.org/fileadmin/DAM/trans/main/wp29/wp29regs/R131\\_e.pdf](https://www.unece.org/fileadmin/DAM/trans/main/wp29/wp29regs/R131_e.pdf) (besucht am 09.06.2024).



---

## Eidesstattliche Erklärung

Das vorliegende Dokument wurde an der Hochschule für Technik und Wirtschaft Dresden unter der Leitung von Prof. Dr. rer. nat. Toralf Trautmann und Dipl.-Ing. (FH) Franziskus Mendt angefertigt. Hiermit erkläre ich, dass ich die vorliegende Arbeit zum Thema

**„Entwicklung einer monokularen Personenerkennung für einen  
Notbremsassistenten“**

selbstständig und ohne Benutzung anderer Quellen und Hilfsmittel als angegeben angefertigt habe. Insbesondere versichere ich, dass ich alle wörtlichen und sinngemäßen Übernahmen aus anderen Werken als solche kenntlich gemacht habe. Ferner gestatte ich der Hochschule für Technik und Wirtschaft Dresden, die vorliegende Diplomarbeit unter Beachtung insbesondere urheber-, datenschutz-, und wettbewerbsrechtlicher Vorschriften für Lehre und Forschung zu nutzen.

---

Ort, Datum

---

Elias Stampa



---

## Verzeichnis der Anhänge

<b>A</b>	<b>Messwerte</b>	<b>99</b>
<b>B</b>	<b>Programmablaufpläne</b>	<b>101</b>
B.1	Hauptprogramm Notbremssystem . . . . .	101
B.2	Unterprogramm Entscheidungsfindung für Warnung und Bremsbefehl . . .	102
<b>C</b>	<b>Datenträger</b>	<b>103</b>







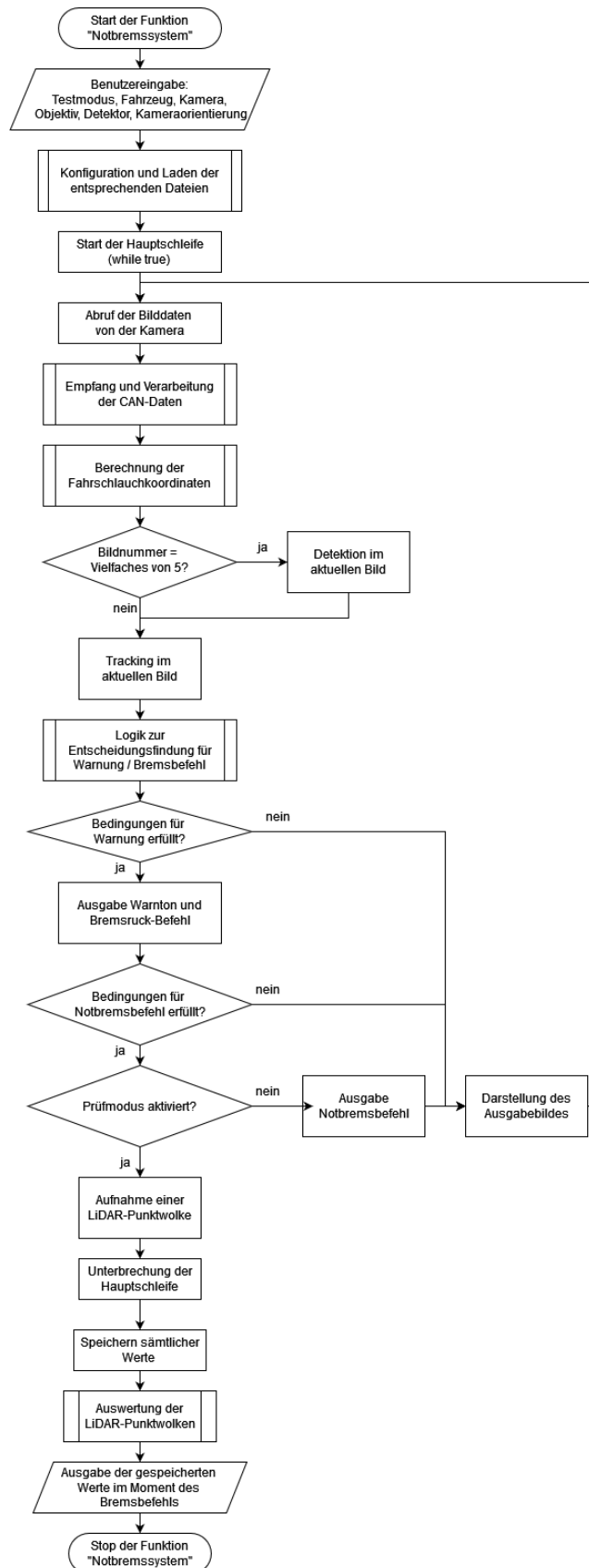
## A. Messwerte

Tabelle A.1.: Messwerte der Realfahrerprobe

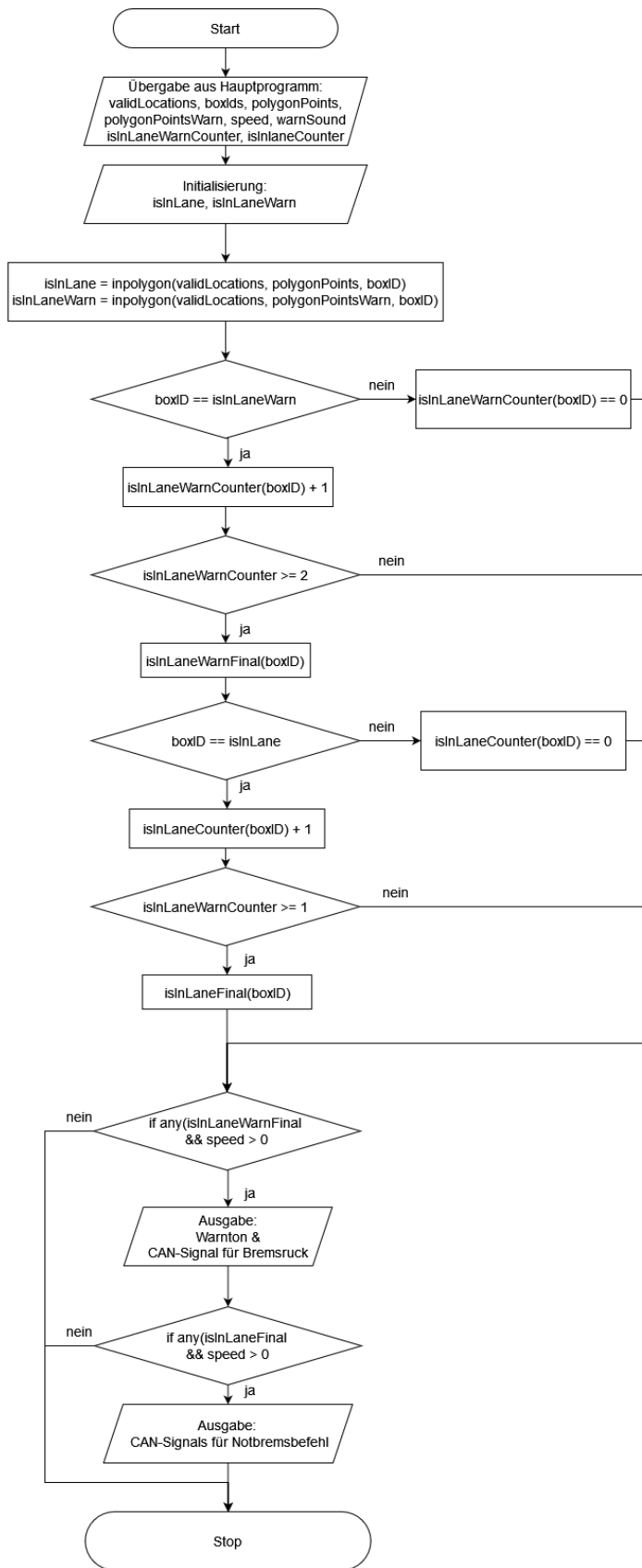
Detektormodell	Messung	Distanz (m)	Geschw. (km/h)	TTC (s)	Zeitdiff. (s)	TTC-Warn (s)	Dist-Warn (m)
tiny-yolov4-coco	MF 1	13,37	31,5	1,528	2,926	4,454	38,973
	MF 2	13,59	31,3	1,5631	3,6399	5,203	45,237
	MF 3	13,48	31,4	1,5455	3,8565	5,402	47,117
	<b>Mittelwert</b>	13,48	31,4	1,5455	3,4741	5,0196	43,776
csp-darknet53-coco	MF 1	13,93	31,7	1,582	3,8956	5,4776	48,233
	MF 2	12,26	31,3	1,4101	3,8942	5,3043	46,118
	MF 3	13,92	31,1	1,6113	3,6513	5,2626	45,463
	<b>Mittelwert</b>	13,37	31,367	1,5345	3,8137	5,3482	46,605
inria-100x42	MF 1	12,48	31,8	1,4128	3,977	5,3898	47,610
	MF 2	13,44	31,4	1,5409	2,9896	4,5305	39,516
	MF 3	13,31	31,5	1,5211	3,4923	5,0134	43,868
	<b>Mittelwert</b>	13,08	31,567	1,4916	3,4863	4,9779	43,665
caltech-50x21	MF 1	11,01	31,2	1,2704	2,1438	3,4142	29,590
	MF 2	11,13	30,9	1,2967	2,3139	3,6106	30,991
	MF 3	10,68	31,7	1,2129	2,1542	3,3671	29,649
	<b>Mittelwert</b>	10,94	31,267	1,2600	2,2040	3,4640	30,077

## B. Programmablaufpläne

### B.1. Hauptprogramm Notbremssystem



## B.2. Unterprogramm Entscheidungsfindung für Warnung und Bremsbefehl



---

## C. Datenträger

SD-Karte

- DA\_Stampa\_final.pdf
- Skripte
  - Notbrems-System
  - Detektionsmetriken
  - Kameraparameter
  - CAN-Datenbanken
- Messwerte Metriken
  - CSP
  - TINY
  - CALTECH
  - INRIA
- Messwerte Realmessfahrten
  - CSP
  - TINY
  - CALTECH
  - INRIA
- Bilddatensatz