

AI4OD

**Embedded Integration von Verfahren der
Künstlichen Intelligenz für eine verbesserte
Objekterkennung im automatisierten Fahren**

Gemeinsamer Schlussbericht

Vorhaben: AI4OD
Förderkennzeichen: 19A20023
Datum: 23.07.2024



Gefördert durch:
 Bundesministerium
für Wirtschaft
und Klimaschutz
aufgrund eines Beschlusses
des Deutschen Bundestages

	<i>Konsortialführer</i>	<i>Ansprechpartner, Kontakt</i>
	Dr. Ing. h.c. F. Porsche AG Porschestraße 911 71287 Weissach	Dr. Sebastian Söhner +49 170 911 3417 sebastian.soechner@porsche.de
	<i>Verbundpartner</i>	<i>Ansprechpartner, Kontakt</i>
	Quality Match GmbH Goldschmidtstraße 5 69115 Heidelberg	Dr. Daniel Kondermann +4915141259855 dk@quality-match.com
	FZI Forschungszentrum Informatik Haid-und-Neu Straße 10-14 76131 Karlsruhe	Prof. Dr.-Ing. Eric Sax Stefan Otten +49 721 9654-156 sax@fzi.de , otten@fzi.de
	Sächsisches Institut für Computational Intelligence und Machine Learning an der Hochschule Mittweida Technikumplatz 17 09648 Mittweida	Prof. Dr. rer. nat. habil. Thomas Villmann +49 3727 58-1134 thomas.villmann@hs-mittweida.de

Inhalt

Abbildungsverzeichnis	v
Einführung	1
Partnervorstellung	3
Porsche AG	3
Unternehmensvorstellung	3
Projektbezogene Kompetenzen	3
Inhaltliche Schwerpunkte	3
Quality Match	3
Unternehmensvorstellung	3
Projektbezogene Kompetenzen	4
Inhaltliche Schwerpunkte	4
FZI Forschungszentrum Informatik	5
Unternehmensvorstellung	5
Projektbezogene Kompetenzen	5
Inhaltliche Schwerpunkte	5
Sächsisches Institut für Computational Intelligence und Machine Learning an der HSMW (SICIM)	6
Unternehmensvorstellung	6
Projektbezogene Kompetenzen	6
Inhaltliche Schwerpunkte	6
Ergebnisse aus den Arbeitspaketen	7
AP 1 Gesamtkonzept: Durchgängige Entwicklung und Prozesssteuerung	7
AP 1.1 Konzeption Gesamtvorgehen	7
AP 1.2 Prozessauslegung A-SPICE	9
AP 1.3 Integration in IT- und Prozesslandschaft	9
AP 1.4 Konzeption Absicherungsstrategie	11
AP 2 Datenmanagement-Tooling für die KI-Funktionsentwicklung	12
AP 2.1 Datenaufzeichnung und -infrastruktur	12
AP 2.2 Datenaufbereitung und -bereitstellung	14
AP 2.3 Datenselektion und Erstellen von Datensätzen	15
AP 2.4 Bewertung der Datensätze	16
AP 2.5 Toolentwicklung	17
AP 3 Funktionsentwicklung: KI Algorithmus	20
AP 3.1 Konzeption und Evaluation verschiedener Ansätze	20
AP 3.2 Toolkette zum Training und Validieren von KI-Funktionen	22
AP 3.3 Benchmarking und Tuning der KI-Funktion	25
AP 3.4 Evaluationsstrategie	26

AP 3.5 Datenlabeling	28
AP 4 Integration und Qualifikation der KI-Funktion	32
AP 4.1 Bereitstellung einer Schnittstelle für die KI-Funktion in die Fahrzeugarchitektur	32
AP 4.2 Integration der KI-Funktion in den Steuergeräte-Verbund.....	32
AP 4.3 Konzeption und Evaluation verschiedener Absicherungsstrategien	32
AP 4.4 Test- und Absicherungsstrategie nach A-SPICE	33
AP 5 Demonstration	35
AP 5.1 Visualisierung der KI-Funktion im Fahrzeug	35
AP 5.2 Konzeption und Durchführung Probandenstudie	35
AP 5.3 Demonstration der Prozess- und Toolinglandschaft für die A-SPICE konforme Entwicklung von KI-Funktionen	35
Literaturverzeichnis	36

Abbildungsverzeichnis

Abbildung 1: Systemschaubild	8
Abbildung 2: Schematischer Aufbau des Software Frameworks im Fahrzeug	13
Abbildung 3: Anforderungsdiagramm für Reflexionsdatensatz	15
Abbildung 4: Vorgehensmodell zur Entwicklung von relevanten Datasplits	16
Abbildung 5: Vergleich von zwei unterschiedlichen Daten-Sampling-Methoden	17
Abbildung 6: Datenstrukturmodell – mit der (mathematischen) Unterscheidung von Features (Merkmale) und Filtern	20
Abbildung 7: Additives Perzeptron-Modell – mathematische Beschreibung	20
Abbildung 8: MOSAIC - Aufmerksamkeit gesteuertes Lernen durch Fokussierung auf helle Strukturen in Bildern/Sequenzen	21
Abbildung 9: One-Class-Classifier	22
Abbildung 10: ProtoTorch	22
Abbildung 11: Toolbeziehungsgraph	23
Abbildung 12: Prozesskette	24
Abbildung 13: Unsichere Klassifikation	25
Abbildung 14: Vergleich Bounding-Box und Pixel-Labeling	26
Abbildung 15: resultierendes Prozessmodell zum KI-Training und Parameter-Tuning bzw. Ergebnisvalidierung	26
Abbildung 16: Überblick der Erkennungszeiten des Menschen	28
Abbildung 17: Darstellung der Annotationspipeline	30
Abbildung 18: Schematischer Ablauf der Transformation eines Bildes in einen Vektor	33

Einführung

Für das autonome Fahren stellt das urbane Umfeld die wohl größte Herausforderung dar. Die Perception des Fahrzeugs wird durch eine Vielzahl an Objekten sowie eine häufige Verdeckung relevanter Verkehrsteilnehmer durch Bauwerke oder parkende Fahrzeuge besonders gefordert. Speziell bei Dunkelheit, wenn kamerabasierte Systeme nur eingeschränkt funktionieren, steigt die Gefahr, Objekte zu übersehen. Dies ist belegt durch eine Studie des ADAC: Obwohl das Verkehrsaufkommen bei Nacht deutlich geringer ist, ereignen sich über 20% der jährlichen Unfälle bei Nacht. Dies entspricht zwar dem Anteil der Nachtfahrten an den Gesamtfahrten, allerdings ist die Gefahr bei einem Nachtunfall schwer verletzt oder getötet zu werden deutlich höher als bei Tagfahrten.

Aktuelle Perceptionssysteme basieren darauf, Objekte dann zu detektieren, wenn sie sich vollständig im sichtbaren Bereich aufhalten. Menschen dagegen sind in der Lage, anhand von anderen Merkmalen auf die Anwesenheit eines relevanten Objekts zu schließen. Ein Beispiel hierfür ist das Erkennen eines Verkehrsteilnehmers anhand der Beobachtung der Lichtkegel seiner Scheinwerfer. Eine derartige Situation ist exemplarisch in Abbildung 1 dargestellt. Menschliche Fahrer nutzen diese Fähigkeit gerade im urbanen Verkehr, in dem Verkehrsteilnehmer sehr häufig durch Bauwerke oder andere Fahrzeuge verdeckt sind. Insbesondere in engen Stadtgebieten mit schlecht einsehbaren Kreuzungen hilft diese Fähigkeit des Menschen, Objekte entscheidende Sekunden früher zu entdecken und somit die richtigen Entscheidungen zu treffen. Um automatisierte Fahrzeuge in Nachtsituationen auf ein menschliches Leistungs- und Sicherheitsniveau zu heben, wird also eine Verbesserung der aktuellen Perzeptionsalgorithmik benötigt. Die Verwendung weiterer Sensoren wie Radar oder Lidar kann die Objektdetektion bei Dunkelheit unterstützen, diese Sensoren sind jedoch auch nicht in der Lage, verdeckte Objekte wahrzunehmen. Dieses Problem kann zusätzlich mit einer Kommunikation zwischen Fahrzeugen (Car2Car) adressiert werden. Allerdings kann das nicht die alleinige Lösung sein, da eine vollständige Ausrüstung aller Fahrzeuge mit Car2Car-Technik noch in weiter Ferne steht und eine verbesserte Perception beispielsweise auch Fahrradfahrer detektieren könnte.

Aktuelle Funktionen der Objekterkennung in Bildern basieren auf Künstlicher Intelligenz (KI) und neuronalen Netzen. Dabei werden die Algorithmen nicht mehr durch den Entwickler vorgegeben, sondern anhand eines möglichst großen und umfassenden Datensatzes von einem neuronalen Netz gelernt. Somit spielen die Daten, mit denen die KI trainiert wurde, eine bedeutende Rolle. Der aktuelle Industriestandard für die Bewertung von Entwicklungsprozessen im Automobilsektor ist Automotive SPICE („Automotive System Process Improvement and Capability Determination“). In diesem Standard ist jedoch die Verwendung einer KI-Funktion nicht vorgesehen. Es ist demnach unklar, inwiefern eine Automotive SPICE-konforme Entwicklung einer KI-basierten Funktion aussehen kann. Da die Performance einer KI-Funktion direkt von den zum Training verwendeten Daten abhängt, müssen diese Daten besonders bedacht werden. Dabei stellen sich Fragen zur Versionierung und Archivierung von Netzen, gemeinsam mit den verwendeten Daten. In Richtung Zertifizierung und Zulassung einer Funktion muss sichergestellt werden, dass im Training und in der Validierung einer Funktion Daten verwendet werden, die auch der Verteilung von Anwendungsszenarien in der realen Welt entsprechen. Zu diesen Fragestellungen existieren bisher weder Prozesse noch Werkzeuge, die mit den Randbedingungen im Kontext einer automobilen Funktionsentwicklung kompatibel sind. Es besteht ein sehr hoher Bedarf, die

aktuell vorherrschende Prozess- und Toollandschaft mit der Entwicklung KI-bezogener Funktionen in Einklang zu bringen.

Im folgenden werden die wichtigsten Arbeitsergebnisse anhand der definierten Arbeitspakete zusammengefasst.

Partnervorstellung

Porsche AG

Unternehmensvorstellung

Porsche ist einer der weltweit führenden Hersteller und Entwickler von Sportwagen und Premium-Fahrzeugen in unterschiedlichen Fahrzeugklassen. Eines der wichtigsten Entwicklungsziele von der Porsche AG ist die Vereinbarkeit von Sportlichkeit und Alltagstauglichkeit. Mit der Entwicklung der Assistenzsysteme, die den Fahrer in der Längsführung oder im effizienten Fahren unterstützen, hat Porsche die Entwicklungskompetenz auf dem Sektor der Vernetzung und Implementierung der Assistenzsysteme demonstriert.

Projektbezogene Kompetenzen

In den vergangenen Jahren hat Porsche vermehrt ML Methoden auf Basis von Lernenden Vektor Quantisieren (LVQ) erforscht und in verschiedenen Fahrzeugsituationen erprobt. Die theoretischen Forschungsergebnisse wurden auf zahlreichen internationalen Konferenzen vorgestellt und international wahrgenommen. Darüber hinaus hat Porsche die fahrzeugspezifische Kompetenz um Funktionen prototypische in Versuchsträgern zu implementieren. Dies beinhaltet zusätzlich die Expertise die dafür notwendigen Fahrzeugmodifikationen (z.B. Verbau zusätzlicher Sensorik) und eine entsprechende Planung der Hardwareressourcen (z.B. Auslegung der Fahrzeugkommunikationsprotokolle und der prototypischen PCs) durchzuführen.

Inhaltliche Schwerpunkte

Im Projekt steht Porsche für ML-basierte Umsetzungen auf Basis der LVQ-Methode, die zeitkritische Klassifizierungsaufgaben adressieren und auf Grund der inhärenten Distanzbasierten-Modellierung interpretierbar sind. Diese beiden Kriterien sind, verglichen mit konventionellen Deep Learning Methoden, um ein Vielfaches attraktiver für die Echtzeitanwendung. Zusätzlich fokussiert sich der Tätigkeitsbereich auf die Inbetriebnahme der entwickelten Funktionen in einem Versuchsträger. Dafür wird Porsche das Know-How einbringen um ein Fahrzeug entsprechen umzurüsten und die notwendigen Deployment-Tools aufzusetzen und bereitzustellen.

Quality Match

Unternehmensvorstellung

Die Quality Match GmbH (Quality Match) ist ein Startup aus Heidelberg, das den Prozess zur Erzeugung qualitativ optimaler Datensätze für KI-Anwendungen perfektioniert.

Gegründet wurde es 2019 von Ex-Apple-Manager und Privatdozent Daniel Kondermann zusammen mit einem erfahrenen und eingespielten Gründerteam. Aktuell beschäftigt Quality Match mehr als 25 Mitarbeiter. Zu den Kunden zählen bisher große Automotive OEMs, Tier-1s, Bosch, Hyundai-Mobis, Mapillary, Zeiss und ein bekanntes soziales Netzwerk.

Quality Match ist nach der Pallas Ludens GmbH das zweite Startup von Daniel Kondermann, das sich ausschließlich mit der Erzeugung von Trainingsdaten und deren Qualitätssicherung beschäftigt. So hat Herr Kondermann an der Entwicklung von Labelingprozessen seit Entstehung der Industrie in den frühen 2000ern maßgeblich beigetragen. Bei Apple hat Herr Kondermann als Manager von 2016-2019 Datensatzqualitätssicherung auf mehreren Anwendungsgebieten betrieben. Akademisch hat er seit 2009 in seiner Forschungsgruppe am „Heidelberg Collaboratory for Image Processing“ (HCI) mit Prof. Bernd Jähne an der Erzeugung realer und synthetischer Datensätze geforscht. Aufgrund der besonderen Situation, dass Herr Kondermann das Promotionsrecht in Heidelberg hält, aber nicht an der Universität angestellt ist und darüber hinaus Doktoranden innerhalb von Quality Match eingestellt werden können, ist ein reibungsfreier Technologietransfer garantiert und die Forschung optimal auf die Bedürfnisse der Anwendungen angepasst.

Dr. Mirko Schmidt (CTO) hat seine Erfahrungen als langjähriger Ingenieur bei Microsoft und Google gesammelt. Der Rest des Teams besteht aus erfahrenen Entwicklern, teilweise mit Promotion in Bildverarbeitung und Machine Learning.

Projektbezogene Kompetenzen

Durch die existierenden Vorarbeiten mit Bosch besteht seit 2020 bereits ein klares Verständnis der Bedürfnisse der Industrie. In Projekten mit anderen Kunden hat Quality Match bereits Produkte zum Labeling und zur Qualitätssicherung von 2D-Bounding-Boxes und Datensatzfiltern entwickelt.

Daniel Kondermann hat bereits sehr früh Arbeiten zu verschiedenen Arten von Datensätzen publiziert und ist dafür in der internationalen Forschung sehr gut vernetzt und auch u.a. durch zahlreiche Publikationen und Workshops bekannt (h-Index 19 und i10-Index 30 bei 1560 Zitaten). Insbesondere hat Herr Kondermann während seiner Zeit am HCI bereits viele Jahre mit Bosch zusammen Datensätze für autonomes Fahren in den Top-Konferenzen (CVPR, ICCV, ECCV) veröffentlicht und ist damit ein optimaler Kandidat für das hier vorgeschlagene Projekt.

Inhaltliche Schwerpunkte

Im Rahmen der bisherigen Projektverläufe haben die Konsortialpartner festgestellt, dass das Labeling eines Datensatzes zur Früherkennung von Fahrzeugen bei Nacht erhebliche Unklarheiten mit sich bringt. Klassische Objekterkennung beschäftigt sich bis dato mit klar zu definierenden, sichtbaren Objekten. Beispielsweise stellt es für den Menschen kein Problem dar, zu definieren, wo ein Auto im Bild beginnt und wo es aufhört, da sich ein Auto eindeutig vom Hintergrund abhebt. Im Falle von Lichtreflexionen wurde jedoch in Proof of Concept Experimenten herausgefunden, dass die Annotationen unterschiedlicher Annotator*innen stark voneinander abweichen. Dies liegt an der Charakteristik von Lichtreflexionen, welche keine objektiv definierbaren Objektgrenzen besitzen und deren Erscheinung stark vom reflektierten Objekt abhängig ist. Zudem ist in der Forschung bereits bekannt, dass fehlerhafte Annotationen zu erheblichem Fehlverhalten von Machine Learning Algorithmen führen können. Diese Fehlerquelle ist in einem Forschungsprojekt wie im vorliegenden Fall unbedingt auszuschließen, weshalb an dieser Stelle zusätzlicher Forschungsbedarf bezüglich der Sicherstellung von fehlerfreien Annotationen bei schwer zu definierenden Objekten notwendig ist. Gleichzeitig muss dabei sichergestellt werden, dass die Annotationsmethode auf große Datenmengen skalierbar ist.

FZI Forschungszentrum Informatik

Unternehmensvorstellung

Das FZI ist eine gemeinnützige Einrichtung für Informatik Anwendungsforschung und Technologietransfer und fungiert als Innovationsdrehscheibe für Informations- und Kommunikationstechnologien des KIT. Im Bereich Mobilität stehen autonome Fahrzeuge sowie Prozesse und Methoden für die Entwicklung und Absicherung von Fahrzeugfunktionen im Vordergrund der Forschung am FZI, welche im Rahmen mehrerer Verbundprojekte, wie z.B. VVMethoden, KI-Absicherung, SetLevel4To5 sowie in Direktaufträgen mit der Industrie bearbeitet werden. Das FZI forscht und entwickelt gemeinsam mit Partnern aus Wissenschaft, Wirtschaft und Gesellschaft über Anwendungsbereiche hinweg an innovativen Lösungen und anwendungstauglichen Methoden und deren Werkzeugunterstützung. Dabei stehen auch querschnittliche Entwicklungsherausforderungen im Produktlebenszyklus wie Safety (z.B. Projekt SAFE, SysKitHW), Security (z.B. Projekt DEFEnD) und Nachhaltigkeit der Entwicklung sowie rechtliche Aspekte im Fokus. Mit dem House of Living Labs (HoLL) stehen zahlreiche Demonstratoren und Fahrzeuge zur Verfügung, an welchen die entwickelten Methoden erprobt und erlebbar gemacht werden.

Projektbezogene Kompetenzen

Das FZI unterstützt seine Partner aus der Industrie seit vielen Jahren darin, Prozesse und Abläufe zu optimieren und daten- bzw. wissensbasierte Erkenntnisse einfließen zu lassen. Bereits im Jahr 2017 hat das FZI den heute verbreiteten Begriff des "Data-Driven Development" bzw. der datengetriebenen Entwicklung geprägt. Seither wurde an Verfahren und Methoden zur Aufbereitung, Anreicherung und Weiterverwendung von Daten - insbesondere im Bereich der Entwicklung von Fahrerassistenzsystemen - geforscht. Dazu zählen Erkenntnisse zum allgemeinen Aufbau einer Datenpipeline, die Nutzbarmachung von aufgezeichneten Daten für die Bewertung und Analyse von Funktionen sowie die Entwicklung skalierbarer Methoden für die Integration und Qualifikation von Systemen. Durch den Einzug der KI werden zudem verstärkt Forschungsfragen, wie die optimale Repräsentation der Daten für lernende Verfahren, betrachtet.

Am FZI wird zudem seit Jahren an Verifikations- und Validierungsmethoden für Fahrerassistenzmethoden und hochautomatisierende Fahrfunktionen geforscht. Neben zahlreichen Direktbeauftragungen mit den Testabteilungen der OEMs werden auch in großen Verbundprojekten wie VVMethoden und SetLevel4To5 Grundlagen für die Absicherung solcher Funktionen gelegt. Das FZI kann dabei auf Vorwissen zum szenarienbasierten Testen als auch Untersuchungen zur Vollständigkeit und Repräsentativität von Testumfängen zurückgreifen.

Inhaltliche Schwerpunkte

Das FZI hat zwei Schwerpunkte im Projekt AI4OD:

Zum einen sollen die Forschungsarbeiten im Bereich der datengetriebenen Entwicklung angewandt und weiter vertieft werden. Durch den Schwerpunkt auf einer kamera-basierten

Funktion kommen hier neue Forschungsfragen beispielsweise zur Adaption der existierenden Verfahren für Zeitreihen auf Bilder auf.

Zum anderen beschäftigt sich das FZI mit einer gesamtheitlichen Prozessauslegung für die Entwicklung von KI-Funktionen und den Rahmenbedingungen des A-SPICE Referenzprozessmodells. Der Fokus liegt dabei auf notwendigen Anpassungen und Besonderheiten, die aufgrund von lernenden und KI-basierten Funktionen entstehen. Beispielhaft soll in AI4OD eine KI-Funktion in ein seriennahes Fahrzeug integriert und qualifiziert werden.

Sächsisches Institut für Computational Intelligence und Machine Learning an der HSMW (SICIM)

Unternehmensvorstellung

Die HSMW mit dem Hochschulinstitut SICIM ist eine öffentliche Bildungs- und Forschungsstätte im Freistaat Sachsen. Die Ausbildung von Studierenden in ingenieurtechnischen und MINT-Bereichen sind ein wesentlicher Schwerpunkt in der Profilierung der HSMW. Eingebettet in diesen Forschungs- und Bildungsauftrag ist das SICIM als Bündelung der Kompetenzen und Ressourcen im Bereich der Künstlichen Intelligenz und des maschinellen Lernens an der HSMW, welchem Prof. Villmann als Institutsdirektor vorsteht. Forschungsfokus am SICIM sind einerseits die mathematische Verifizierung bzw. mathematisch-theoretisch orientierte Algorithmenentwicklung im Maschinellen Lernen. Andererseits wird am SICIM bezüglich der praktischen Realisierung solcher Algorithmen in technischen Anwendungen und in den Bio-/Lebenswissenschaften geforscht.

Projektbezogene Kompetenzen

Die Forschungsgruppe Computational Intelligence und Methodenentwicklung (CI) am SICIM unter der Leitung von Prof. Dr. rer. nat. habil. Thomas Villmann forscht insbesondere zur Entwicklung mathematisch verifizierter interpretierbarer KI-Modelle und -Algorithmen. Dabei geht es auch um Problemstellungen der Robustheit bzw. Verlässlichkeit (Selbstkonfidenz) solcher Systeme. Insgesamt hat diese Gruppe (derzeit 1 Postdoc, 9 Doktoranden) in den letzten 5 Jahren mehr als 50 Forschungsartikel zu solchen Problemstellungen veröffentlicht. Darüber hinaus wirkt Prof. Villmann in der Arbeitsgruppe zur Erarbeitung einer sächsischen KI-Strategie im Auftrag der sächsischen Staatskanzlei unter Leitung des Staatsministers Schenk mit.

Inhaltliche Schwerpunkte

Der Fokus des SICIM liegt auf der Entwicklung der KI-Modelle zur Erkennung und Klassifizierung der relevanten Szenarien in den Videosequenzen. Dabei stehen Problematiken wie Klassifikationssicherheit, Stabilität und Erfassung der Selbstkonfidenz der KI-Modelle im wissenschaftlichen Problemfeld, welche im engen Bezug zu den allgemeinen Forschungsfoki der CI-Gruppe am SICIM der HSMW stehen. Essentiell ist dabei auch die Interpretierbarkeit des KI-Modells für die ingenieurtechnische Akzeptanz im realen Einsatz.

Ergebnisse aus den Arbeitspaketen

AP 1 Gesamtkonzept: Durchgängige Entwicklung und Prozesssteuerung

AP 1.1 Konzeption Gesamtvorgehen

Grundlage der Gesamtplanung des Vorgehens ist eine Definition des Zielsystems. Alle folgenden Schritte basieren auf einem Verständnis der finalen Funktion und wie sie im Fahrzeug eingesetzt werden soll.

Analyse vorheriger Arbeit

Der PVDN-Datensatz wurde mit Keypoints annotiert, da Lichtartefakte, insbesondere indirekte Artefakte, keine klar definierten Grenzen haben. Diese Keypoints dienen als Basis zur automatischen Generierung von 2D-Bounding-Boxen, da das System auf solche Boxen angewiesen ist. Allerdings bringt diese Methode einige Probleme mit sich: Ungenauigkeit der Bounding-Boxen, Schwierigkeiten bei der Unterscheidung nahe beieinander liegender Instanzen und das Problem des Übersehens annotierter Keypoints.

Um diese Herausforderungen zu bewältigen, wurde ein alternativer Ansatz untersucht, der Keypoints als Seeding-Punkte zur Generierung von Bounding-Boxen verwendet. Trotz der Verbesserungen bleibt das Problem der zusammengeführten Keypoints bestehen. Zudem ist die Auswertung von Bounding-Boxen für indirekte Lichtinstanzen komplex, da automatisch generierte Boxen oft ungenau sind, was das Training von Standard-Objektdetektoren erschwert.

Die Art der Annotation hängt stark von der jeweiligen Aufgabe ab. Für die Objekterkennung müssen Objekte im 2D-Bild oder 3D-Raum verortet werden. Der PVDN-Datensatz formuliert die Erkennung von Lichtreflexionen als Aufgabe der Mengenvorhersage und nutzt Keypoints, um das Intensitätsmaximum einer Reflexionsinstanz zu markieren. Jedoch ist die Unterscheidung verschiedener Reflexionen in komplexen Szenarien subjektiv und problematisch.

Systemdesign und Zielsetzung

Basierend auf der Analyse früherer Arbeiten wurde entschieden, die Erkennung von Lichtartefakten als pixelweise Klassifizierungsaufgabe zu formulieren. Dies bedeutet, Lichtreflexionen als amorphe Regionen (stuff) zu behandeln und nicht als unterscheidbare Instanzen (things). Dies erleichtert die Modellierung und reduziert die Notwendigkeit, klare Grenzen für Lichtreflexionen zu definieren.

Ein vorgeschlagenes ADAS könnte die Matrix-Fernlichtsysteme verwenden, wie in städtischen Szenarien beschrieben. Ein Verkehrswarnassistent könnte in diesen Szenarien den Fahrer vor entgegenkommenden Fahrzeugen warnen, bevor diese in direkter Sichtweite sind. Das System würde verschiedene Sensoren und Fahrzeugbusdaten nutzen, um die Position des entgegenkommenden Fahrzeugs zu schätzen und entsprechende Warnungen auszugeben.

Funktionale Komponenten des Systems

Das Systemdesign umfasst zwei Ansätze: End-to-End und modular. Der End-to-End-Ansatz verwendet alle Eingabedaten direkt zur Schätzung der Fahrzeugposition, ist jedoch schwer zu trainieren und zu debuggen. Der modulare Ansatz, der für diese Arbeit gewählt wurde, ist in drei Module unterteilt:

2D Light Reflection Detection: Erkennung von Lichtreflexionen als Regionen im Bild.

3D-Schätzung: Projektion der erkannten Lichtreflexionen in die 3D-Welt mithilfe von LIDAR-Daten.

Point of Occurrence Estimation: Schätzung des Auftretens des entgegenkommenden Fahrzeugs.

Umsetzung und Herausforderungen

Die Erkennung von Lichtreflexionen wird als Hauptmerkmal für die Erkennung von entgegenkommenden Fahrzeugen bei Nacht verwendet. Diese Aufgabe wird als pixelweise Klassifizierung formuliert, um die Nachteile instanzbasierter Annotationsmethoden zu umgehen. Die 3D-Schätzung erfolgt durch Projektion der LIDAR-Daten, während die Schätzung des Auftretens durch Fusion mit Karteninformationen erfolgen könnte.

Der modulare Ansatz erleichtert das Debugging und die Verwendung regelbasierter Methoden, die besser interpretierbar und validierbar sind. Deshalb wird sich im Verlauf der Arbeit für diesen Ansatz entschieden. Jede Komponente im modularen Design für sich genommen eine komplexe Aufgabe und erfordert eine detaillierte Analyse und Bewertung.

Zum Beispiel ist die Erkennung von Lichtreflexionen eine nicht triviale Aufgabe. Die Lichtreflexionen sind oft sehr schwach und können leicht mit anderen Lichtquellen (z. B. Straßenlaternen, Verkehrsampeln usw.) verwechselt werden. Während [saratrajew2021pvdn-dataset](#) zeigte, dass die Erkennung von Lichtreflexionen anderer Fahrzeuge in ländlichen Szenarien möglich ist, ist unklar, ob dies in komplexeren Szenarien (z. B. in Städten) möglich ist. Daher wird sich die weitere Arbeit in dieser Arbeit auf die Erkennung von Lichtreflexionen in 2D konzentrieren. Der Datensatz soll die Entwicklung und Validierung des 2D-Moduls zur Lichtreflexionserkennung unterstützen. Die 3D-Projektion von detektierten Regionen oder Objekten in das 3D-Koordinatensystem des Fahrzeugs kann, wie bereits dargelegt, mit dem LIDAR-Sensor erfolgen. Die Schätzung des Ortes des Geschehens ist eine nicht triviale Aufgabe und wird für zukünftige Arbeiten vorbehalten bleiben.

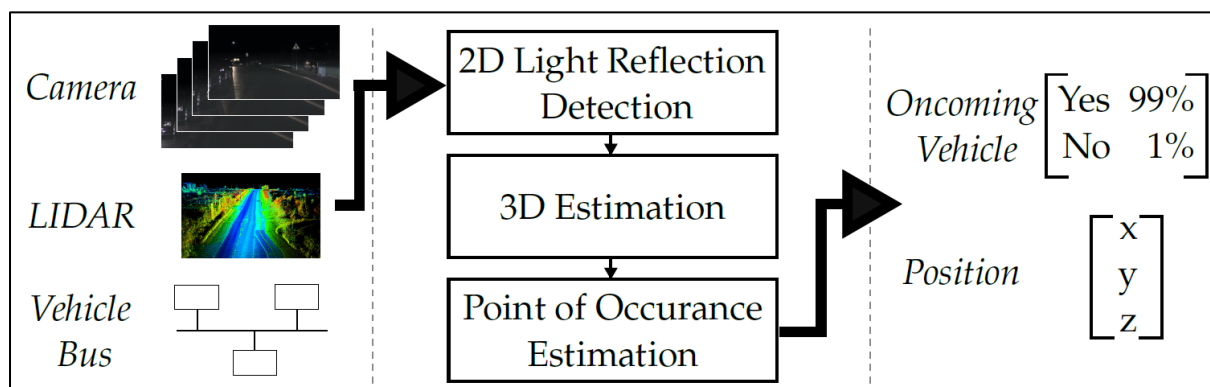


Abbildung 1: Systemschaubild

AP 1.2 Prozessauslegung A-SPICE

Durchführung einer Analyse des A-SPICE-Prozess hinsichtlich der Entwicklung einer KI-Funktion. Eine Literaturrecherche ergab, dass A-SPICE nicht alleine betrachtet werden kann, sondern zusammen mit ISO26262 und ISO/AWI PAS 21448 untersucht werden muss. Die ISO 26262 betrachtet die funktionale Sicherheit und ISO/AWI PAS 21448 ist eine Weiterentwicklung von ISO 26262 speziell für den Automobilbereich, auch bekannt als Safety Of The Intended Functionality (SOTIF).

Im Rahmen von A-SPICE ergeben sich nach einer Analyse Anpassungen an der Software Engineering Process Group (SWE). Eine Unstimmigkeit ist zum Beispiel die Datengenerierung und -haltung, die bei der klassischen Entwicklung nicht benötigt werden, aber ein essenzieller Bestandteil bei der Entwicklung von Machine Learning Software sind. Als Anpassung wird ein W-Modell vorgeschlagen. Dieses Entwicklungsmodell besteht aus zwei V-Modellen: einem für die datengetriebene Entwicklung und einem für die KI-Entwicklung.

AP 1.3 Integration in IT- und Prozesslandschaft

Konzepte zum Tracking von Anforderungen und Testcases (Traceability)

Ein zentrales Element von ASPICE ist die Traceability, die die Rückverfolgbarkeit von Anforderungen bis hin zu den entsprechenden Testfällen fordert. Codebeamer, ein leistungsfähiges Application Lifecycle Management (ALM) Tool, unterstützt die Erfüllung dieser Anforderungen durch umfassende Funktionen, die eine durchgehende Traceability ermöglichen.

1. Anforderungsmanagement

Codebeamer bietet eine robuste Plattform für das Anforderungsmanagement. Anforderungen können detailliert erfasst, organisiert und verwaltet werden. Die hierarchische Strukturierung von Anforderungen erleichtert die Übersicht und Verwaltung komplexer Projekte. Jede Anforderung wird mit einer eindeutigen ID versehen und kann mit zusätzlichen Informationen wie Priorität und Status versehen werden.

2. Verknüpfung von Anforderungen mit anderen Artefakten

Ein Schlüsselement der Traceability ist die Verknüpfung von Anforderungen mit anderen Entwicklungsartefakten. In Codebeamer können Anforderungen direkt mit Implementierungseinheiten, Designdokumenten und insbesondere mit Testfällen verknüpft werden. Diese bidirektionalen Links ermöglichen es, von jeder Anforderung aus die zugehörigen Implementierungs- und Testartefakte zu verfolgen und umgekehrt.

3. Änderungsmanagement

Das Änderungsmanagement-Modul von Codebeamer stellt sicher, dass alle Änderungen an Anforderungen und deren Auswirkungen vollständig dokumentiert und nachverfolgt werden. Jede Änderung wird versioniert, und Change Requests ermöglichen eine formelle und nachvollziehbare Handhabung von Änderungen. Die Auswirkungen von Änderungen auf andere Artefakte werden analysiert und dokumentiert, um die Konsistenz und Integrität des Systems zu gewährleisten.

4. Testmanagement

Codebeamer unterstützt das gesamte Testmanagement, von der Erstellung bis zur Ausführung und Protokollierung von Testfällen. Testfälle werden detailliert beschrieben und spezifischen Anforderungen zugeordnet. Dies stellt sicher, dass jede Anforderung durch entsprechende Testfälle abgedeckt wird. Die Ergebnisse der Testausführung werden in Codebeamer protokolliert und mit den entsprechenden Anforderungen verknüpft, was eine lückenlose Dokumentation der Testabdeckung ermöglicht.

5. Reporting und Dashboards

Um die Einhaltung der Traceability-Anforderungen zu überwachen, bietet Codebeamer leistungsstarke Reporting- und Dashboard-Funktionen. Traceability-Matrizen visualisieren die Verknüpfungen zwischen Anforderungen, Implementierungen und Tests und zeigen auf einen Blick, ob alle Anforderungen durch Testfälle abgedeckt sind. Anpassbare Berichte und Dashboards ermöglichen es, den aktuellen Status der Traceability und andere wichtige Metriken zu überwachen.

6. Integration und Kollaboration

Codebeamer integriert sich nahtlos in bestehende Entwicklungsumgebungen und unterstützt die Zusammenarbeit zwischen verschiedenen Teams und Tools. Durch Integrationen mit Werkzeugen wie JIRA, Jenkins und Git wird eine durchgehende Traceability auch in komplexen Tool-Landschaften gewährleistet. Funktionen wie Aufgabenmanagement, Diskussionsforen und Wikis fördern die Kommunikation und Zusammenarbeit im Team.

Umsetzung im Projekt

In stark forschungslastigen Projekten ist die Definition von konkreten Anforderungen an die Funktion in einer frühen Phase nur schwer möglich. Da im vorliegenden Projekt nur die PoC-Phase in der Funktionsentwicklung durchlaufen wurde, wurde auf die Integration einer komplexen Toolchain wie oben beschrieben verzichtet, da dies keinen Mehrwert gebracht hätte.

Datenspeicher und -austausch

Bei der Entwicklung einer Bildverarbeitungsfunktion entstehen hohe Datenmengen, die entweder in der Cloud oder On-Premise gespeichert und ausgetauscht werden können. Cloud-Speicherung bietet flexible Kosten, basierend auf Nutzung, während On-Premise eine initiale Investition erfordert, aber langfristig niedrigere Betriebskosten haben kann. Die Cloud ermöglicht eine einfache Skalierbarkeit ohne zusätzliche Hardware, während On-Premise eine sorgfältige Planung erfordert und zu längeren Bereitstellungszeiten führen kann. Der Datenaustausch in der Cloud ist einfach dank standardisierter APIs, während er in On-Premise-Umgebungen komplexer sein kann und benutzerdefinierte Integrationen erfordert. Die Entscheidung hängt von den spezifischen Anforderungen, dem Budget und der Flexibilität eines Unternehmens ab, wobei die Cloud Flexibilität und einfache Integration bietet, während On-Premise mehr Kontrolle und Sicherheit bietet, aber mit höheren Anfangsinvestitionen und begrenzter Skalierbarkeit verbunden sein kann. Es ist wichtig, die Vor- und Nachteile abzuwägen und die Lösung zu wählen, die am besten zu den langfristigen Zielen und Anforderungen passt. Aufgrund der besseren Skalierbarkeit und Datenaustauschmöglichkeiten wurde sich im Projekt auf einen Datenspeicher auf AWS entschieden.

AP 1.4 Konzeption Absicherungsstrategie

Zunächst wurde die ISO 21448 herangezogen, die auch unter dem Namen "Safety of the intended functionality" (SOTIF) bekannt ist. Im Anhang D dieser Norm wird eine Verifizierungs- und Validierungsstrategie für automobiler Wahrnehmungssysteme beschrieben. Es wurde festgestellt, dass die Testfahrten nicht alle möglichen befahrbaren Straßen abdecken können, was auch für die im Projektkontext durchgeführten Testfahrten gilt. Bei der Durchsicht des Beispiels wurde deutlich, dass KI-Methoden einen deutlich erhöhten Testaufwand erfordern, da nicht nur der Programmcode getestet werden muss, sondern auch eine detaillierte Betrachtung der zur Entwicklung verwendeten Daten durchgeführt werden muss. Es wurden eine mögliche datengetriebene Teststrategie sowie eine 6-schrittige Anpassung für A-SPICE entwickelt, die im partnerspezifischen Abschlussbericht des FZI beschrieben werden.

AP 2 Datenmanagement-Tooling für die KI-Funktionsentwicklung

AP 2.1 Datenaufzeichnung und -infrastruktur

Um eine datengetriebene Entwicklung grundlegend möglich zu machen, müssen zunächst Daten aufgezeichnet werden. Dafür muss neben der Durchführung von Erprobungsfahrten durch bspw. seriennahe Entwicklungsfahrzeuge oder Fahrzeugflotten eine Infrastruktur für die Aufzeichnung, Übertragung und Speicherung der Daten geschaffen werden. Als Versuchsfahrzeug wurde hierbei ein Porsche Cayenne verwendet.

Hardware

Als Rechenplattform wurde ein leistungsfähiger Car-PC gewählt. Dieser hat zwei Intel Xeon E5-2667v4 CPUs und eine NVIDIA Tesla V100 GPU. Diese Kombination aus CPU und GPU Power ermöglicht das Prototypen von Funktionen in einem frühen Entwicklungsstadium, in dem man nicht durch Rechenleistung eingeschränkt sein möchte. Im späteren Entwicklungsbetrieb gibt es dann die Möglichkeit, der Funktion weniger Ressourcen zur Verfügung zu stellen. 10GB Ethernet Anschlüsse ermöglichen den Datenaustausch zu weiteren Schnittstellen.

Software

Das System basiert auf einem Ubuntu Betriebssystem. Der Kern ist das ROS2 Kommunikationssystem, welches ein Publisher/Subscriber Modell zur Verfügung stellt. Einzelne Funktionen - sogenannte Nodes - können somit Daten dem System zur Verfügung stellen oder Daten, welche von anderen Nodes bereitgestellt werden, empfangen.

Datenaufzeichnung in ROS2

In ROS2 werden Daten hauptsächlich über sogenannte "Bags" aufgezeichnet. Diese Bags sind Dateien, in denen Nachrichten, die zwischen ROS2-Knoten ausgetauscht werden, gespeichert werden. Sie ermöglichen das Aufzeichnen und spätere Abspielen von Nachrichten, was insbesondere für das Debugging, die Datenanalyse und die Reproduktion von Szenarien hilfreich ist. Dieses Vorgehen bietet folgende Vorteile:

- **Standardisierung:** ROS2 bietet eine standardisierte Methode zur Datenaufzeichnung, die in vielen Projekten wiederverwendet werden kann.
- **Integrierte Werkzeuge:** Die Tools zur Aufzeichnung und Wiedergabe von Daten sind gut integriert und einfach zu bedienen.
- **Unterstützung für verschiedene Formate:** ROS2 unterstützt verschiedene Nachrichtenformate und Datentypen, was die Flexibilität bei der Aufzeichnung erhöht.
- **Zeitstempel und Synchronisation:** Jede Nachricht in einer Bag-Datei ist mit einem Zeitstempel versehen, was die Synchronisation und Analyse erleichtert.
- **Reproduzierbarkeit:** Durch das Abspielen der aufgezeichneten Daten kann man Tests und Experimente reproduzieren, was für die Entwicklung und das Debugging von Systemen entscheidend ist.
- **Analysen und Debugging:** Mit den aufgezeichneten Daten können Entwickler umfangreiche Analysen durchführen und Fehler besser identifizieren.

Zusammengefasst eignet sich ROS2 zur Datenaufzeichnung aufgrund seiner standardisierten, gut integrierten Werkzeuge und der Flexibilität bei der Handhabung verschiedener Datentypen und Nachrichtenformate. Die Möglichkeit, Daten mit Zeitstempeln aufzuzeichnen und wiederzugeben, erleichtert die Analyse und das Debugging erheblich.

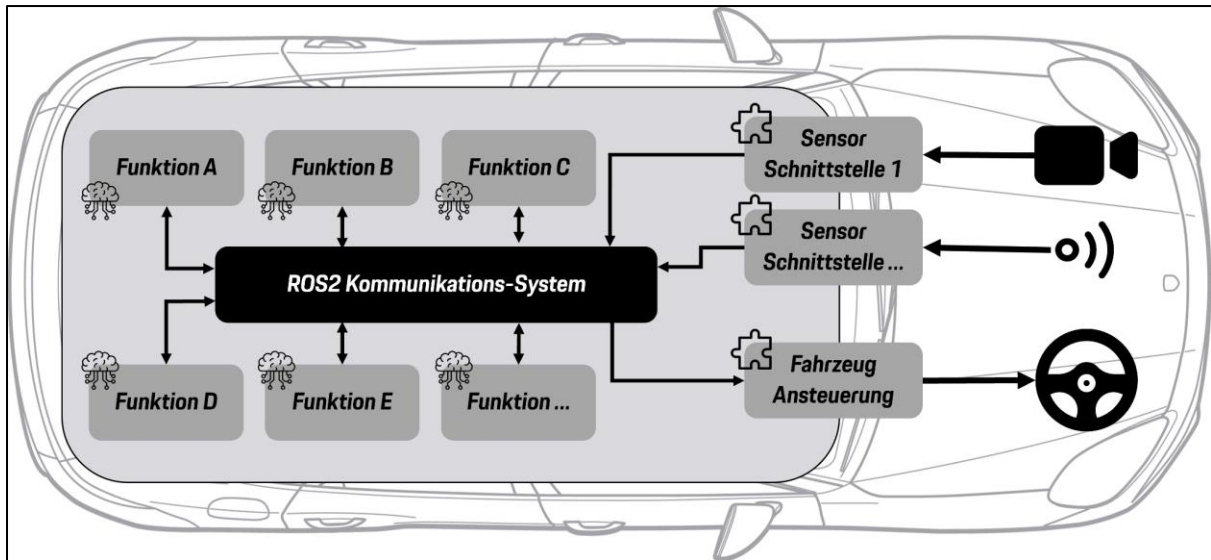


Abbildung 2: Schematischer Aufbau des Software Frameworks im Fahrzeug

Sensorset

Der Aufbau besteht aus vier Kameras: einer nach links vorne gerichteten Weitwinkelkamera, einer nach rechts vorne gerichteten Weitwinkelkamera, einer nach vorne-mittig gerichteten Weitwinkelkamera und einer nach vorne-mittig gerichteten Schmalwinkelkamera.

Die drei Weitwinkelkameras ermöglichen es, einen großen Nahbereich vor dem Ego-Fahrzeug bis zu fast 180 Grad zu erfassen.

Die Engwinkelkamera dient zur Erfassung der Szene vor dem Ego-Fahrzeug in weiter Entfernung.

Die Bilder werden mit einer Auflösung von 3840 mal 2160 Pixeln bei neun Bildern pro Sekunde aufgenommen.

Datenaufzeichnung

Die Daten wurden in den Monaten April und Mai in zwei mittelgroßen Städten in Deutschland (Leonberg und Pforzheim) erhoben. Die Daten wurden in der Nacht und damit bei Dunkelheit erhoben. Die Daten zeigen städtische Fahrscenarien mit oft engen Straßen oder Verdeckungen an Kreuzungen durch parkende Fahrzeuge oder Kurven. Daraus ergaben sich 140 Szenen und insgesamt 14688 Bilder.

Eine Szene enthält nur ein entgegenkommendes Fahrzeug, und die vom entgegenkommenden Fahrzeug verursachten Lichtreflexionen sind in den Bildern deutlich sichtbar. Die Szenen beginnen in der Regel damit, dass weder ein entgegenkommendes Fahrzeug noch die vom entgegenkommenden Fahrzeug verursachten Lichtreflexe in den Bildern sichtbar sind. Dann werden die vom entgegenkommenden Fahrzeug verursachten Lichtreflexe sichtbar, aber das Fahrzeug selbst ist noch nicht zu sehen. Schließlich wird das entgegenkommende Fahrzeug sichtbar. Die Szene endet ein paar Bilder nach dem Sichtbarwerden des entgegenkommenden Fahrzeugs. Damit enthält der Datensatz Frames aus drei verschiedenen Phasen:

- Phase (a): Frames ohne Lichtreflexionen und direkt sichtbare entgegenkommende Fahrzeuge.
- Phase (b): Frames mit Lichtreflexionen, aber ohne direkt sichtbare entgegenkommende Fahrzeuge.
- Phase (c): Frames, bei denen sowohl Lichtreflexe als auch das entgegenkommende Fahrzeug sichtbar sind.

Damit kann das Verhalten von Algorithmen in verschiedenen Phasen der Entwicklung einer Szene bewertet werden, wie sie in realen Szenarien auftreten.

Eine Szene enthält zwischen 44 und 127 Bilder, mit einem Durchschnitt von 105 Bildern pro Szene.

AP 2.2 Datenaufbereitung und -bereitstellung

Die Aufbereitung und Bereitstellung der Daten wurden über die Datenmanagement-Tools sichergestellt, die im Abschnitt zu AP 2.5 näher beschrieben werden. In AP 2.2 wurden die Anforderungen dafür evaluiert und definiert. Im Ergebnis erforderte das Forschungsvorhaben eine Datenmanagementlösung, innerhalb derer das Speichern, Analysieren und Löschen der Daten effizient umgesetzt ist. Die für diesen Zweck entwickelten und weiterentwickelten Tools, insbesondere die Plattform HARI und das zugehörige SDK stellen dies sicher. Das SDK ermöglicht den Projektpartnern den selbstständigen Upload der Daten, bei dem die Rohdaten bereits automatisch anonymisiert werden. Danach können die Daten einfach in die Plattform HARI zur Visualisierung, Filterung und Analyse importiert und Projekten zur Annotation und Qualitätssicherung zugeführt werden. Beim Import werden automatisch Attribute in HARI erstellt, nach denen die Daten gefiltert und sortiert werden können. Diese Attribute werden basierend auf in den Daten enthaltenen grundlegenden Informationen, wie z.B. Größe und Helligkeit der Bilder oder Anzahl und Geometrie der bereits innerhalb der Bilder markierten Objekte beim Import erstellt. Weitere Attribute werden basierend auf den Ergebnissen der Annotationen zum Datenlabelling und zur Qualitätssicherung hinzugefügt. Für die Annotationsprojekte können direkt innerhalb HARI Subsets erstellt werden, auf denen annotiert werden soll. Dies ist einerseits basierend auf angewandten Filtern möglich, andererseits können z.B. zum Zwecke der Qualitätssicherung Subsets aus zufälligen Stichproben erstellt werden. Beim Erstellen der Subsets hat der Anwender außerdem die Wahl, ob die Annotationen auf den gesamten Bildern oder nur Teilausschnitten um relevante, zu überprüfende Objekte durchgeführt werden. Die Ergebnisse der Annotationen werden nach Abschluss der Projekte automatisch aggregiert und auf der Plattform HARI zur Verfügung gestellt. Für auf Annotationsprojekten basierende Attribute werden automatisch statistische Metriken aus den Verteilungen der Antworten der einzelnen Annotatoren berechnet und bereitgestellt, nach denen die Daten ebenfalls sortiert und gefiltert werden können. Dadurch ist es möglich, bereits vor der Durchführung weiterer Qualitätssicherungsschritte einen Eindruck von der Qualität der Daten und der durchgeführten Annotationen zu erhalten. Die Metriken können darüber hinaus für Methoden des Aktiven Lernens verwendet werden. Die fertig annotierten Daten können anschließend ebenfalls selbstständig vom Anwender im JSON Format exportiert werden.

Durch diesen Aufbau der Datenmanagement Pipeline können Daten nach Bedarf und Verfügbarkeit bereitgestellt, annotiert und analysiert werden. Dies ermöglicht die asynchrone Arbeit an mehreren Subsets, wodurch sich das Datenmanagement innerhalb des Projekts flexibler und effizienter gestalten lässt. Die Löschung der Datensätze ist ebenfalls innerhalb von HARI implementiert und wird über AWS S3 bewerkstelligt.

AP 2.3 Datenselektion und Erstellen von Datensätzen

Im Rahmen dieses Arbeitspakets wurde an einem Systems Engineering-Ansatz zur Beschreibung von Datensätzen gearbeitet. Unser Ziel war es, eine eindeutige semantische Beschreibung für Bilder zu erhalten, um diese zur Generierung von Datensätzen zu verwenden.

Im Detail wurde am Aufbau eines Requirements-Diagramms gearbeitet, das bisher zur Beschreibung von Anforderungen an Software eingesetzt wird, jedoch noch nicht für Anforderungen an Datensätze. Mit einem solchen Requirements-Diagramm wäre es möglich, maschinell zu untersuchen, ob die Daten innerhalb oder außerhalb der definierten Operational Design Domain liegen. Dies ermöglicht automatisierte Prüfungen und Garantien der Eignung der eingesetzten Daten bezüglich eines zu definierenden Zwecks bzw. definierter Anforderungen.

Zuerst wurden über eine Literaturrecherche verschiedene existierende Ontologien identifiziert und deren Anwendbarkeit untersucht. Danach erfolgte eine Übersetzung in ein Requirements-Diagramm, um automatisiert die Anforderungsabdeckung von Datensätzen testen zu können. Durch die Verwendung eines Requirements-Diagramms kann sichergestellt werden, dass die Datensätze vollständig und korrekt sind und den Anforderungen entsprechen.

Aufgrund des Ausstiegs von siasearch und der daraus resultierenden Verzögerungen beim Labeling und der Anonymisierung der Daten musste eine neue Lösung gesucht werden. Zwischen der Porsche AG und dem FZI wurde im Berichtszeitraum ein Joint-Controller-Agreement verhandelt, um in Zukunft auch nicht anonymisierte Daten austauschen zu können. Um die Verzögerungen möglichst gering zu halten, wurde mit der Erarbeitung der Methoden auf öffentlich verfügbaren Datensätzen begonnen. Entstanden ist dabei ein Vorgehensmodell zur Entwicklung von relevanten Datasplits, wie in der folgenden Abbildung dargestellt:

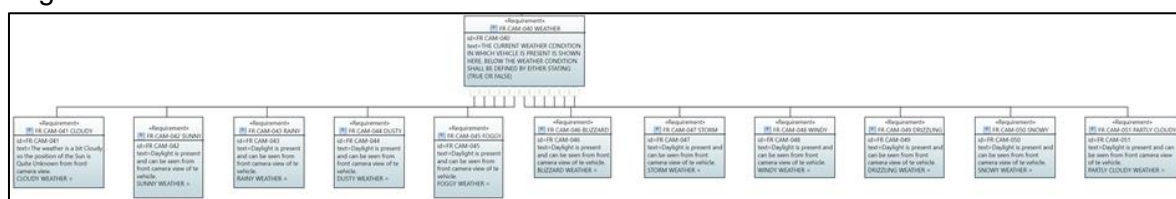


Abbildung 3: Anforderungsdiagramm für Reflexionsdatensatz

Die zentrale Idee war, dass alle Daten (CAN, Bilder, Radar, etc.) in einem Datalake gespeichert werden. Anschließend können diese Daten mit weiteren Datenquellen angereichert werden, beispielsweise mit OpenStreetMap, um die Straßenkategorie zu ermitteln. Diese Informationen können dann genutzt werden, um mit einem geeigneten Datasplit-Verfahren die Daten aufzuteilen. Ein Beispiel ist Stratified Sampling. Dieses Verfahren wählt Stichproben, die die Originaldaten am besten repräsentieren. Allerdings können diese Stichprobenverfahren jedoch immer noch zu Data Leaks führen, wenn ihnen keine besondere Aufmerksamkeit geschenkt wird. Zum Beispiel kann ein Leak auftreten, wenn verschiedene Bilder aus der gleichen Sequenz für Training und Test ausgewählt werden. Aus diesem Grund sieht unser Verfahren den Schritt der Data Evaluation I vor, die sich der Leakage Analyse widmet. Einige Beispiele zu Splits für Trainings- und Testdatensätze werden in folgender Abbildung gezeigt:

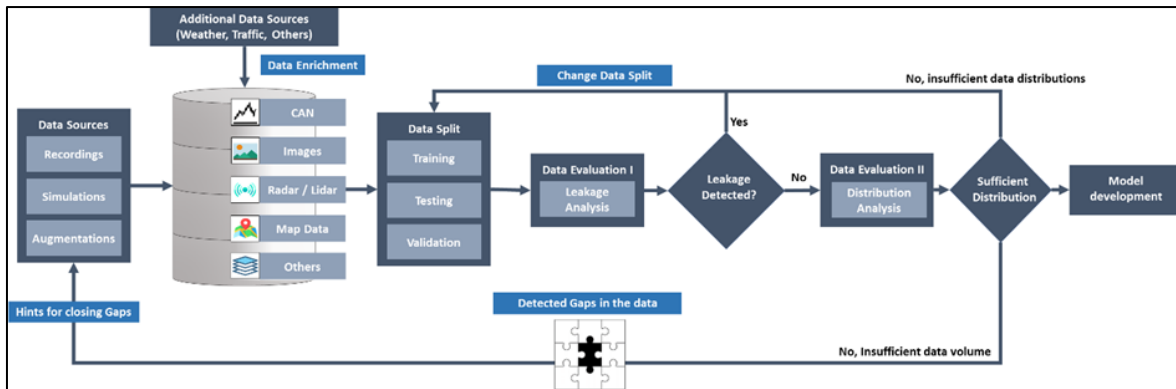


Abbildung 4: Vorgehensmodell zur Entwicklung von relevanten Datasplits

AP 2.4 Bewertung der Datensätze

Da in dem Projekt lange Zeit keine hinreichende Datenlage vorhanden war, konnte kein Referenzdatensatz zur Verfügung gestellt werden. Um eine durchgehende Verwendung von Daten während der Entwicklung und Absicherung zu gewährleisten, wurde eine Literatur- und Internetrecherche zu potentiell bereits vorhandenen Lösungen durchgeführt. Dabei wurde ein Datenversionierungs-Tool (DVC¹) von iterative² als potenziell relevant eingestuft. Dieses soll beim Management von Datensätzen unterstützen. DVC könnte dabei unter anderem sicherstellen, dass die in AP2.3 erstellten Datensätze und Datensplits bei allen Partnern einheitlich verwaltet, versioniert und eingesetzt werden.

Mitte 2022 wurden die Verhandlungen zum Joint-Controller-Agreement abgeschlossen. Mit diesem Agreement war es möglich, dass der Projektpartner Porsche die eingefahrenen Daten zur Verfügung stellen konnte. Die bereitgestellten Daten waren anonymisierte Rohdaten, in diesem Fall Video- bzw. Bilddaten, die zunächst keine Labels enthielten. Für die weiteren Arbeiten im Projekt, insbesondere die Zusammenstellung eines Referenzdatensatzes, sind Labels jedoch von großer Bedeutung. Zusammen mit allen Projektpartnern und durch ein von Porsche entwickeltes Labeling-Tool wurden die Bilder daher manuell gelabelt.

Die manuell erstellten Labels durch verschiedene Labeler wiesen große Unterschiede auf, wodurch die direkte Verwendung der Daten nur eingeschränkt möglich war. Die Lessons Learned zum Labeling von Lichtreflexionen können jedoch durch Quality Match aufgenommen und verarbeitet werden. Für das Labeling diffuser und nicht eindeutig zuordenbarer Lichtreflexionen im Allgemeinen wurden Anforderungen und Guidelines diskutiert und erfasst. Durch die eingeschränkte Anwendbarkeit der im Projekt erfassten und gelabelten Daten wurde für die Methodenentwicklung zum Erstellen von Referenzdatensätzen der BDD100k-Datensatz³ verwendet. Es wurden insbesondere die dort vorhandenen Metalabels verwendet, um damit unterschiedliche Daten-Splits zu erstellen und anschließend zu evaluieren.

Die entwickelte Methode wurde anschließend mit einer umfangreichen Studie evaluiert. Dabei ist der beschränkte Umfang des Datensatzes zu beachten, dessen potentieller Einfluss auf die Ergebnisse der Evaluation nicht ausgeschlossen werden kann. Diese Herausforderung gilt jedoch bei jedem (öffentlichen) Datensatz.

Für die Evaluation wurden mehr als 250 Modelle trainiert. In der folgenden Abbildung sieht man den Vergleich von zwei unterschiedlichen Daten-Sampling-Methoden. Man erkennt

¹ (DVC, 2024)

² (DVC AI, 2024)

³ (Yu, et al., 2020)

eindeutig, dass die unterschiedlichen Sampling-Methoden nur einen geringfügigen Einfluss auf die Performance des Netzes haben. Das bedeutet, dass die Bild-Domäne nicht den erwarteten großen Einfluss auf die Modellperformance hat. Dennoch konnten mit einer gezielten stratifizierten Auswahl der Bilder vergleichbare Ergebnisse erreicht werden, wie mit einer zufallsbasierten Auswahl der Bilder. Die dadurch gewonnene Nachvollziehbarkeit ohne Performanceverluste der KI-Modelle ist für eine allgemeine A-SPICE-Absicherungsstrategie zuträglich.

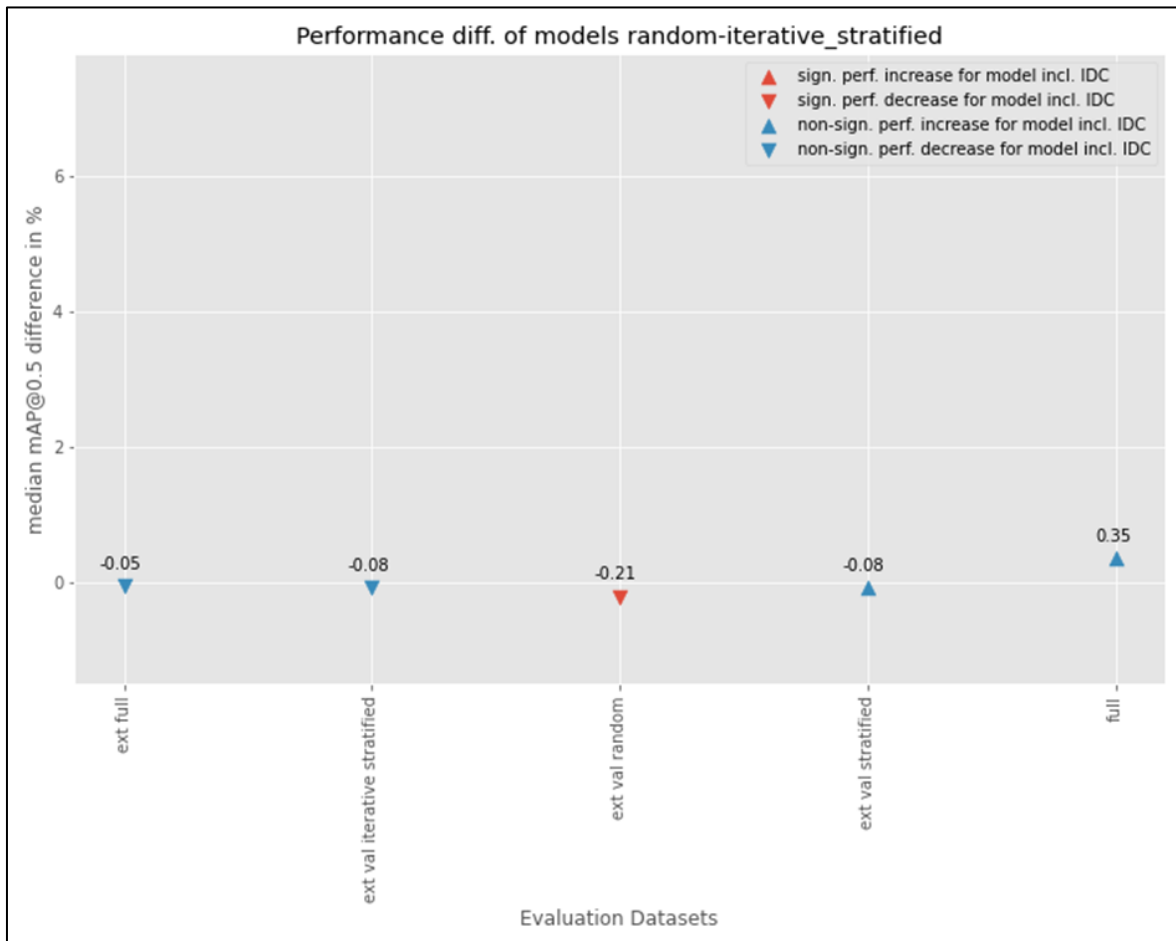


Abbildung 5: Vergleich von zwei unterschiedlichen Daten-Sampling-Methoden

AP 2.5 Toolentwicklung

HARI und SDK

Die Plattform **HARI** (Highly Actionable Realtime Insights) bildet das Kernstück der in AP 2.2 entwickelten Pipeline zur Datenaufbereitung und -bereitstellung. Um den Anforderungen des Forschungsvorhabens zu entsprechen, wurden an der Plattform, die von der Quality Match GmbH vor Beginn des Projekts ursprünglich als internes Tool entwickelt wurde, grundlegende Änderungen und Erweiterungen vorgenommen. Zur Bereitstellung und Integration der Plattform in die Entwicklungsumgebung des Projektes wurde ein **SDK** entwickelt, das den Projektpartnern die einfache und selbstständige Registrierung ihrer Rohdaten erlaubt. Durch das SDK werden die Daten in einem Arbeitsschritt validiert, wobei sie auf die Korrektheit des

Datenformats für den Upload hin geprüft werden und bei erfolgreicher Validierung hochgeladen werden. Schlägt die Validierung fehl, so erhält der Nutzer Informationen darüber, in welchen Aspekten der Datensatz nicht dem nötigen Uploadformat entspricht. Danach können die erfolgreich hochgeladenen Daten über die graphische Nutzeroberfläche in HARI importiert werden. Dies ermöglicht es den Projektpartnern, ihre Daten selbständig und ohne zusätzlichen Entwicklungsaufwand der Plattform zuzuführen.

HARI wurde ursprünglich entwickelt, um Kunden die Ergebnisse von Annotationsprojekten zu zweidimensionalen Bilddaten in Form von Visualisierungen zur Verfügung zu stellen. Die Einbindung von Daten in Form von Videosequenzen in HARI war daher ein wichtiges Arbeitsergebnis, um die weitere Annotation und Analyse der Trainingsdaten für das Forschungsvorhaben zu gewährleisten. Durch die Visualisierung von Videosequenzen innerhalb der Toolkette ist es möglich, interessante Ausschnitte aus den aufgenommenen Szenen für die Annotation zu identifizieren. So kann sich die Annotation auf relevante Ausschnitte beschränken. Dies war notwendig, um die für das Vorhaben notwendigen Annotationen kosteneffizient und innerhalb der verfügbaren Budget- und Zeitgrenzen fertigzustellen.

Nach den vorgenommenen Veränderungen an HARI können Mitglieder des Konsortiums unter geringem Aufwand umfangreiche Video-Datensätze hochladen und importieren. Die importierten Videos können sodann kuratiert und Subsets zugeordnet werden. In der Nutzeroberfläche von HARI können Videos nun intuitiv abgespielt, angehalten und durchsucht werden. Auch die Auswahl und Visualisierung einzelner Objekte in Form von Bounding Boxes innerhalb des Bildmaterials musste für das Vorhaben angepasst und optimiert werden. Da es für das Projekt nötig war, Bild- und Videodateien mit mehreren verschiedenen Objektklassen zu visualisieren, wurde unter anderem die Anzeige des Klassennamen des ausgewählten Objekts integriert. Die einzelnen Objektklassen können darüber hinaus mit unterschiedlichen Farben gekennzeichnet werden, was sich für die Analyse von Datensätzen mit vielen verschiedenen Klassen unerlässlich herausstellte. Auch die Filteroptionen und Funktionen zur Erstellung von Subsets mussten für die Integration von Videodateien umfassend angepasst werden.

Im Verlauf des Forschungsvorhabens wurde offensichtlich, dass die Verwaltung und Kuratierung von Attributen einen Kernaspekt des erfolgreichen Datenmanagements für das Vorhaben darstellen. Als Attribut wird in HARI eine Fragestellung zu einem Bild oder Objekt bezeichnet, welche im Rahmen eines Annotationsprojekts geprüft wurde. Eine wichtige Neuerung waren hier die Anpassungen von HARI, die es ermöglichten, Attribute umzubenennen. Diese Funktionalität war entscheidend für die Organisation der annotierten Objekte und die Qualitätssicherung der Annotationsprojekte. Standardmäßig werden Attribute in HARI mit der Frage bezeichnet, die während des Annotationsprojekts gestellt wurde. In der Verwendung als internes Tool, war dies ausreichend. Diese Umsetzung stellte sich jedoch innerhalb des Projekts als Hindernis dar, da zur Annotation und Qualitätssicherung die gleiche Frage oft mehrmals auf den gleichen Daten gestellt werden musste. Die Möglichkeit der Umbenennung erleichterte hier die Zuordnung und Organisation der Attribute innerhalb bestimmter Arbeitsschritte erheblich. Darüber hinaus wurde die Funktionalität des Vergebens von Tags in HARI auf Videodateien ausgeweitet. Dies ermöglicht es, die Daten zusätzlich mit individuell definierten Metadaten zur Filterung und Sortierung anzureichern.

Da HARI vor Beginn des Forschungsvorhabens als internes Tool verwendet wurde und nur die Anzeige der Visualisierungen für unternehmens-externes Personal ermöglichte, waren auch bei den Zugangsberechtigungen für weitergehende Funktionalitäten Anpassungen nötig. Seiten für die Anmeldung, das Einloggen und Vergeben bzw. Zurücksetzen von Passwörtern wurden erstellt. Darüber hinaus wird es den Teilhabern des Konsortiums ermöglicht, Zugangsberechtigungen und Nutzerprofile direkt innerhalb des Tools zu beantragen und einzurichten. Durch die Steuerung der Zugriffsberechtigung innerhalb HARI wird vermieden, dass verschiedene Benutzerkonten auf Datensätze anderer Projekte zugreifen können. Dies gewährleistet Sicherheit und Datenschutz auf der Plattform. In seiner Gesamtheit wurde HARI von einem ursprünglich internen Tool zu einer Softwarelösung weiterentwickelt, die externen Nutzern umfangreiche Funktionalität zur Visualisierung und Qualitätskontrolle von Datenannotationen bereitstellt.

Pipeline Builder

Nachdem die Subsets in HARI erstellt wurden, auf denen Annotationen durchgeführt werden sollen, wird im nächsten Schritt die Annotationspipeline erstellt und gestartet. Dazu steht der von der Quality Match GmbH entwickelte Pipeline Builder als Tool zur Verfügung. Auch dieser musste für die Projektanforderungen, insbesondere für die Einbindung von Videodateien, erweitert werden. Das Erstellen von Annotationspipelines auf Grundlage von Videosequenzen wird nun vom Pipeline Builder unterstützt. Darüber hinaus wurden neue Nutzeroberflächen für die Annotation von Videos erstellt und getestet. Hervorzuheben ist hier die neuartige SliderUI, die von der Quality Match GmbH im Verlauf des Projekts entwickelt wurde. Sie ermöglicht es, innerhalb einer Videosequenz mittels eines Schiebereglers einen bestimmten Zeitpunkt exakt zu markieren. Für die Annotationen der Lichtreflexionen wurde diese Nutzeroberfläche benötigt, um den genauen Zeitpunkt zu kennzeichnen, an dem die Lichtreflexionen erstmals sichtbar werden.

Anbindung von Gaming Crowds

Um die Durchführbarkeit der Annotationen im Rahmen des Projekts durch Gaming Crowds zu untersuchen, mussten diese in die Toolkette mit eingebunden werden. Zu diesem Zweck wurde eine Schnittstelle entwickelt, die es erlaubt, Projekte in den Gaming Crowds zu initiieren und verwalten. Da im Gegensatz zu professionellen Annotatoren die Verfügbarkeit von Annotatoren in den Gaming Crowds nicht vollständig planbar ist, wurde die Projektselektion innerhalb dieser Crowds automatisiert, um die Auslastung der Crowds zu optimieren und die Annotationsprojekte zeiteffizient durchzuführen. Zusätzlich wurde eine Prioritäten-Zuweisung innerhalb der Gaming Crowd implementiert, die es ermöglicht, je nach Projekt und Zeitfenster bestimmten Aufgaben Vorrang zu geben. Diese Priorisierung sorgt dafür, dass dringende Projekte oder solche mit besonderer Bedeutung für die Pipeline schneller bearbeitet werden. Um die Anzahl potenzieller Laienannotatoren zu erweitern, wurden die Benutzeroberflächen in 10 Sprachen übersetzt. Dies vereinfacht die Nutzung der Plattform für Annotatoren aus verschiedenen Sprachräumen. Die Ergebnisse aus verschiedenen Sprachen und Projekten werden im Anschluss automatisch konsolidiert. Diese Erweiterungen erhöhen die Reichweite der Plattform und steigern folglich die Effizienz der Annotationsprozesse weiter, da so ein breiter Pool an möglichen Annotatoren verfügbar ist.

AP 3 Funktionsentwicklung: KI Algorithmus

AP 3.1 Konzeption und Evaluation verschiedener Ansätze

Für die Implementierung der KI-Algorithmen wurde zunächst in Abstimmung mit den Partnern das Open-Source-Framework PyTorch installiert und auf Funktionalität hinsichtlich von KI-Funktionen getestet. Dabei ist die Ausnutzung von parallelen Hardwarestrukturen (GPU) wichtig. Für die zu entwickelnden KI-Modelle wurde ein modularer Ansatz verfolgt, der heterogene Datenstrukturen zulässt, aber auch die Interpretierbarkeit des Modells erhält.

Es wurde eine mathematisch notwendige Unterscheidung von Merkmalen (Features) zu Filtern herausgearbeitet. Letztere erfüllen die Bedingung der Additivität, welche durch KI-Modelle ausgenutzt werden könnte. Das resultierende Datenstrukturmodell (siehe Abb. Datenstrukturmodell) wurde als grundlegende KI-Datenstruktur etabliert

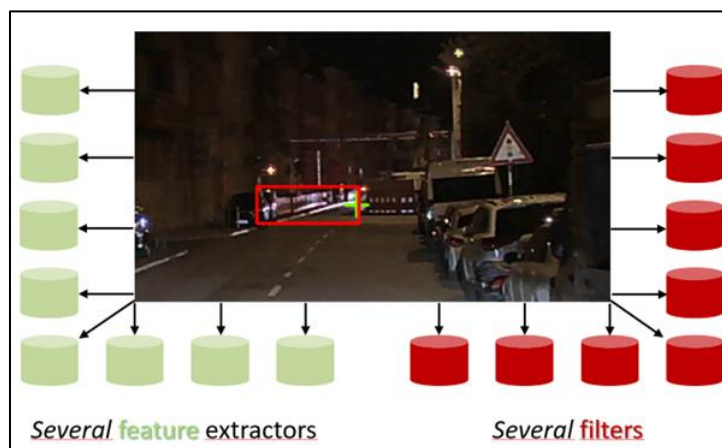


Abbildung 6: Datenstrukturmodell – mit der (mathematischen) Unterscheidung von Features (Merkmalen) und Filtern

Daraus abgeleitet wurde ein additives Neuronen-Modell (Perzeptron) für die gewichtete Feature-/Filter-Kombination als adaptive Verarbeitungsmöglichkeit von Bild- und Videodaten, siehe Abb. Additives Perzeptron-Modell.

Neural additive model – super filter generation

- Local data scene of an image/frame \mathbf{x}
- **Several feature extractors** f_k with response vectors $\phi_k(\mathbf{x}) \in \mathbb{R}^{n_k} \longrightarrow$ prototypes $\varphi_k \in \mathbb{R}^{n_k}$

 dissimilarity measure $d_k(\phi_k, \varphi_k)$ maybe depending on local parameters Ω_k
- **Several filters** g_l with response vectors $\theta_l(\mathbf{x}) \in \mathbb{R}^m$
Super filter $\theta(\mathbf{x}) = \sum_{l=1}^L \beta_l \cdot \theta_l(\mathbf{x}) \in \mathbb{R}^m \longrightarrow$ prototypes $\vartheta \in \mathbb{R}^m$
 dissimilarity measure $D(\theta, \vartheta)$ maybe depending on parameters Ω

- Overall response vector $\rho(\mathbf{x}) = (\phi_1, \dots, \phi_K, \theta) \in \mathbb{R}^{n_1} \times \dots \times \mathbb{R}^{n_K} \times \mathbb{R}^m$
- Overall prototype $\mathbf{w} = (\varphi_1, \dots, \varphi_K, \vartheta) \in \mathbb{R}^{n_1} \times \dots \times \mathbb{R}^{n_K} \times \mathbb{R}^m$

Perceptron model $y = \sigma(\langle \rho(\mathbf{x}), \mathbf{w} \rangle + b)$

R. Agarwal et al. (Hinton):
 Neural Additive Models: Interpretable Machine Learning with Neural Nets,
<https://arxiv.org/abs/2004.13912v1>

Abbildung 7: Additives Perzeptron-Modell – mathematische Beschreibung

Um nachvollziehbare KI-Entscheidungen zu erhalten, wurde der Schwerpunkt der Entwicklung auf die Verwendung interpretierbarer KI-Modelle gelegt. Im Fokus standen dabei Varianten des Classification-by-Components-Modells (CbC) sowie der sogenannten Lernenden Vektor Quantisierer (LVQ). Beide Algorithmen sind für Klassifikationsaufgaben entwickelt worden und wurden hier im Kontext der Szenen-/Lichtreflexionsklassifikation angewendet. Dabei stellte sich das CbC-Modell als numerisch zu instabil heraus und wurde nach intensivem Testen und numerischen Experimenten verworfen.

Damit wurde die KI-Entwicklung vornehmlich mit LVQ-Modellen durchgeführt und mit komplexen Deep-Modellen (nicht interpretierbar) verglichen. Dabei wurden verschiedene Lernstrategien entwickelt und getestet (active learning - Aufmerksamkeit gesteuertes Lernen, reject options - Zurückweisungsstrategien) sowie robuste Lernszenarien für reproduzierbare Lernergebnisse evaluiert. Beim im Projekt entwickelten Aufmerksamkeit gesteuerten MOSAIC-Lernen wird in komplexen Bildern zunächst ein Fokus (Lokalisationsspot) gesetzt (hier helle Konturbereiche = eventuelle Lichtreflexionen), die dann im KI-Modell ausgewertet werden (siehe Abbildung MOSAIC)⁴.

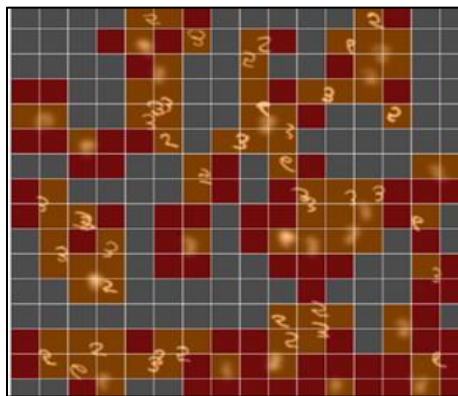


Abbildung 8: MOSAIC - Aufmerksamkeit gesteuertes Lernen durch Fokussierung auf helle Strukturen in Bildern/Sequenzen

Zusätzlich wurden Szenarien mit unsicherem Labeling modelliert, um (ungewollt) fehlerhafte Trainingsbeispiele adäquat im Training zu behandeln. Die getesteten Varianten waren in ihrer Performanz unbefriedigend bzw. haben zu stark restringierte mathematische Annahmen, welche in Realsituationen oft nicht zu erfüllen sind. Stabilere Ergebnisse wurden teilweise mit probabilistischen LVQ-Klassifizierungsmodellen erhalten, da damit besser Unsicherheiten abgebildet werden können⁵.

Zudem wurde eine hierarchische Klassifikationsstrategie verfolgt, da vergleichsweise wenige Reflexionsbilder (bzw. Bildbereiche) relativ zur Gesamtdatenmenge. Entsprechend wurde ein LVQ-basierter sogenannter One-Class-Classifier (OCC) zur Detektion von Reflexionen entwickelt^{6,7} (siehe Abbildung One-Class-Classifier). Diesem OCC nachgeschaltet wurde dann die Reflexionsklassifikation mittels Standard-LVQ zur Identifikation des Reflexionstyps.

⁴ (Engelsberger & Villmann, 2023)

⁵ (Bakhtiari & Villmann, 2022)

⁶ (Staps, et al., 2022)

⁷ (Staps, Villmann T, & Paaßen, 2024)

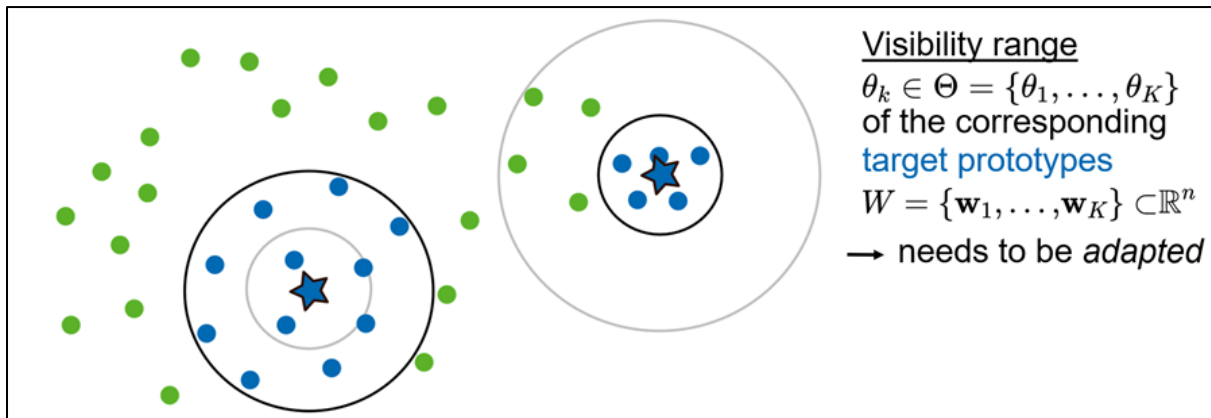


Abbildung 9: One-Class-Classifier

Nur die blaue Klasse (Zielklasse - im Projekt: Lichtreflexion) wird detektiert. Eine weitere Unterscheidung innerhalb der Zielklasse wird nicht vorgenommen, sondern erst im nachgeschalteten Typ-Klassifizierer für Reflexionsarten bestimmt

Weiterhin wurden konkrete Features/Filters für Bilder und Bildsequenzen implementiert und im Kontext der urbanen Straßenszenen-Bewertung validiert (Texturen, SURF, Segmentierungen, optischer Fluss), die dann als Input für die zu trainierenden LVQ-Klassifizierer dienen. Diese können in heterogenen LVQ-Modellen zusammengefasst gelernt werden, wie in ⁸ gezeigt wurde.

Die gleichzeitige Verwendung hochkomplexer Features führt jedoch zu unverhältnismäßig hoher Rechenkomplexität der Algorithmen sowohl in der Trainingsphase als auch im Arbeitsmodus und ist daher bei limitierten Rechenressourcen im Fahrzeug nicht oder nur eingeschränkt einsetzbar.

AP 3.2 Toolkette zum Training und Validieren von KI-Funktionen

Die KI-Modelle und Algorithmen bzw. Tools zur Feature-Extraktion bzw. Filterung wurden in der Pytorch-Umgebung als eigenständiges Tool ProtoTorch entwickelt (siehe Abbildung ProtoTorch)

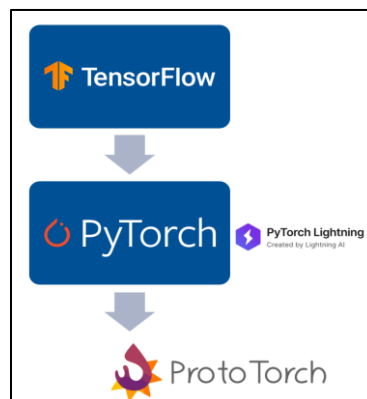


Abbildung 10: ProtoTorch

⁸ (Villmann, et al., 2022)

Ausgehend von der Umgebung Tensorflow wurden alle KI-Algorithmen in die Pytorch-Umgebung portiert bzw. darin entwickelt.

ProtoTorch ist als Open Source verfügbar (github.com/si-cim/prototorch). Die KI-Modelle sind ebenfalls unter dem Github-Eintrag github.com/si-cim/prototorch_models allgemein veröffentlicht.

Das Zusammenwirken der Modelle und Komponenten wurde in einem komplexen Toolbeziehungsgraphen (TBG) etabliert, sodass eine semi-automatische Abarbeitung erfolgen kann (siehe Abbildung Toolbeziehungsgraph). Das ist hinsichtlich vieler zu testender Parameterkonfigurationen für unterschiedliche Modellansätze und Szenarien von essenzieller Wichtigkeit. Insbesondere wird damit die Reproduzierbarkeit und exakte Protokollierung der Ergebnisse gewährleistet.

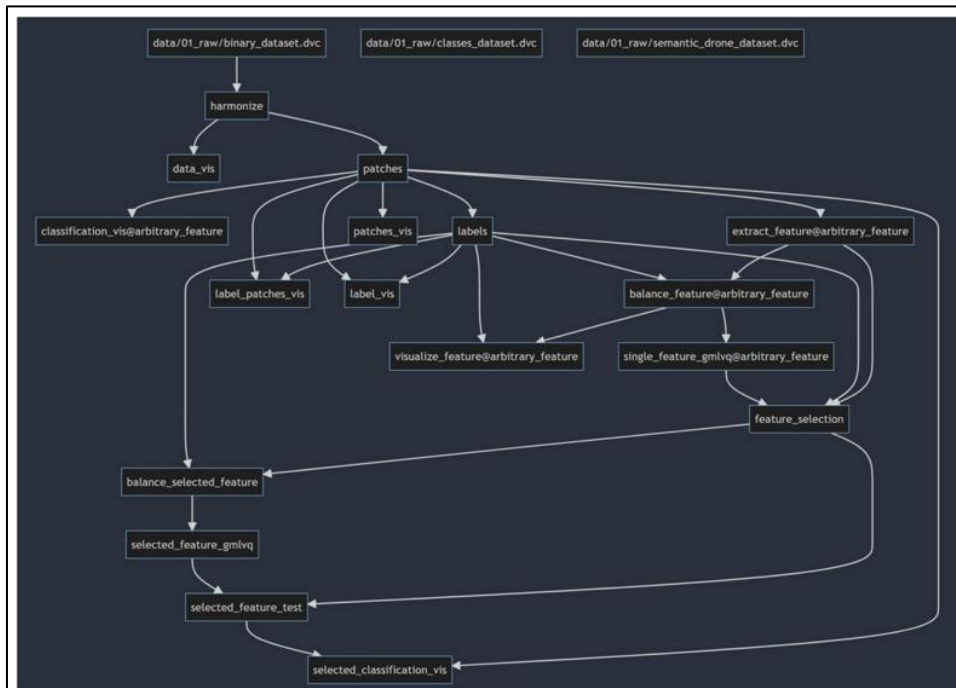


Abbildung 11: Toolbeziehungsgraph

Ausgehend von der Umgebung Tensorflow wurden alle KI-Algorithmen in die Pytorch-Umgebung portiert bzw. darin entwickelt. Der Toolbeziehungsgraph stellt das mögliche Zusammenwirken der Module und Tools im KI-Lernprozess dar.

Ausgehend vom TBG können dann Prozessketten definiert werden, die ein komplexes Lernszenario abbilden, standardisiert bearbeiten und die Ergebnisse in einem Protokoll ablegen (Abbildung Prozesskette).

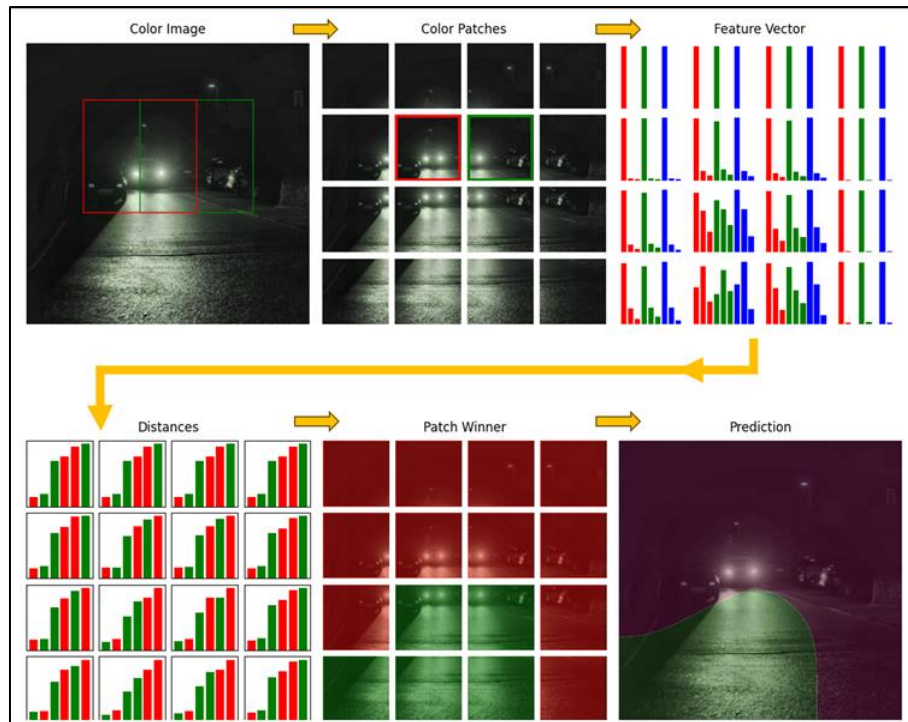


Abbildung 12: Prozesskette

Schematische Darstellung einer Prozesskette basierend auf dem Toolbeziehungsgraphen. Das Ergebnisprotokoll dient zur Evaluierung des KI-Lernerfolgs. Dabei erlaubt der TBG die Einbindung verschiedener KI-Modelle und Datenvorverarbeitungen, resultierend in unterschiedlichen Varianten einer Prozesskette.

Der Lernerfolg wird für Trainingsdatensätze mittels der erreichten Klassifizierungsgenauigkeit bewertet. Hier sind jedoch sowohl die Unsicherheiten der Labelgenauigkeit der Trainingsdaten als auch die Wahrscheinlichkeitsbewertungen der Klassifikation durch das KI-Modell einzubeziehen (Abbildung Unsichere Klassifikation).

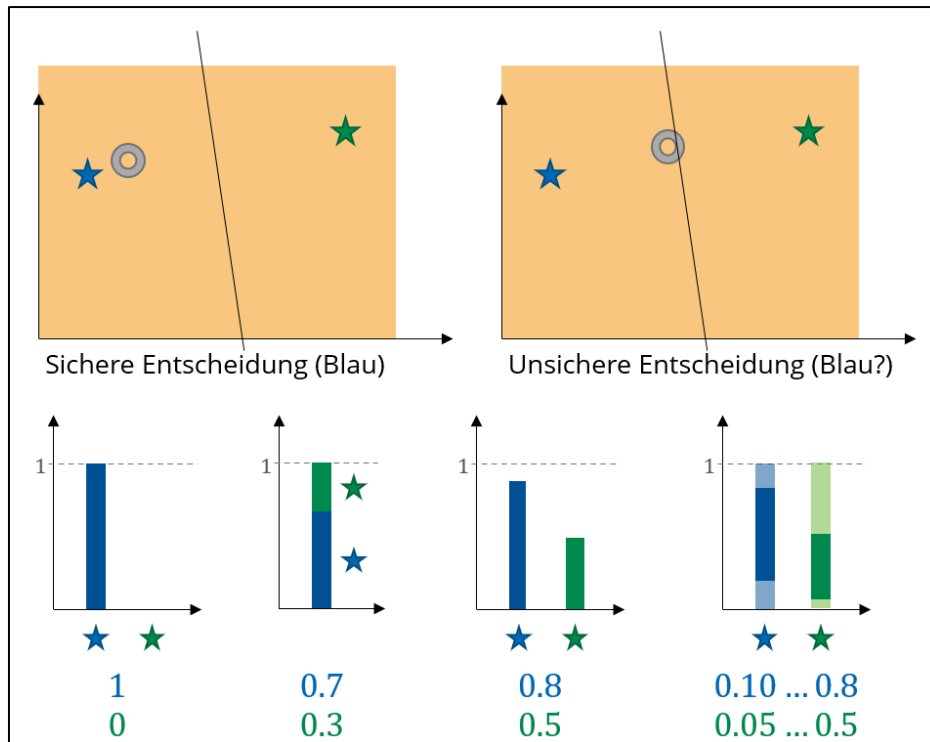


Abbildung 13: Unsichere Klassifikation

Schematische Darstellung einer unsicheren Klassifikation in einem Prototyp-basierten probabilistischen Klassifizierer.

Deshalb werden die mittels einer konkreten Prozesskette erhaltenen probabilistischen Klassifizierungen der Szenen über die Evaluation der resultierenden Konfusionsmatrix bewertet. Beispielsweise eignet sich das sogenannte F_1 -Maß, welches die precision/Positiv-Vorhersagewert und recall/Sensitivität gewichtet, besonders im hier vorliegenden Fall von stark unbalancierter Daten (wenige Reflexionsbeispiele vs. viele Bilder ohne spezifische Reflexionen).

AP 3.3 Benchmarking und Tuning der KI-Funktion

Die etablierten Prozessketten wurden sowohl für Benchmarking-Datensätze als auch schwierige Realsituationen getestet, um eine optimale Funktionalität der KI-Funktionen zu erreichen. Das resultiert in aufwendigen Schemata zur Parameteroptimierung der einzelnen KI-Lernmethoden, welche benutzergesteuert teil-automatisiert wurden (user guided automatic search).

Die Benchmarkinguntersuchungen führten dazu, dass das zuerst im Projekt favorisierte Labeling-Konzept mittels Bounding-Box-Verfahren (rechteckige Bildausschnitte in den Frames für Aufmerksamkeitsfokussierung und entsprechende Label) nach Diskussion mit allen Projektpartnern verworfen und durch eine (pixelweise) Bildsegmentierung mit pixelweisem Labeling der Licht- bzw. Lichtreflexionsbereiche unter Verwendung spezieller Zusatz-Attribute ersetzt wurde (Abbildung Vergleich Bounding-Box und Pixel-Labeling).



Abbildung 14: Vergleich Bounding-Box und Pixel-Labeling

Vergleich der Labelingverfahren basierend auf Bounding-boxes (links) und pixelweisem Labeling (rechts).

Beim pixelweisen Segmentieren/Labeln entstehen jedoch geometrisch komplexere Gebilde (verbunden mit erhöhten Rechenaufwand für das KI-Training), die aber die Lichtverhältnisse besser widerspiegeln. Deshalb erzielen mit Bounding-Box trainierte KI-Modelle schlechtere Ergebnisse als solche basierend auf Segmentierungen.

Um die numerische Effizienz der KI-Lernalgorithmen zu erhöhen, wurden aufwendige Optimierungen der Implementierungen insbesondere unter dem Aspekt des Zusammenwirkens der verschiedenen Hardware-Software-Komponenten (Storage-Server, GPU-Server, Standard-Framework) vorgenommen.

Das resultierende und implementierte Prozessmodell ist schematisch als lineare Pipeline dargestellt:

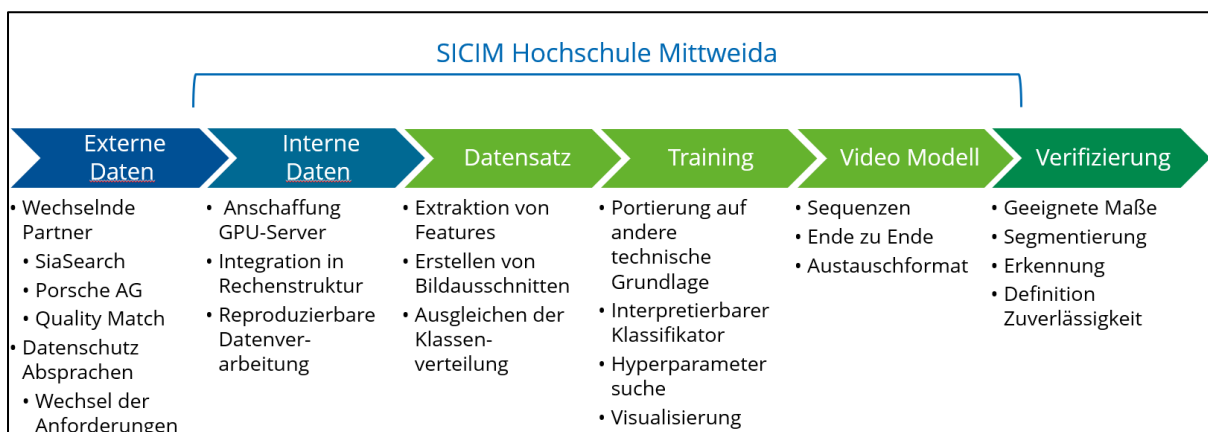


Abbildung 15: resultierendes Prozessmodell zum KI-Training und Parameter-Tuning bzw. Ergebnisvalidierung

AP 3.4 Evaluationsstrategie

Die Evaluation von KI-Verfahren bedarf einer geeigneten Evaluationsstrategie. Für die funktionale Evaluation muss die Funktion dabei umfassend, insbesondere in schwierigen, aber möglichst realistischen Situationen, getestet werden. Dies kann entweder durch Tests in der realen Welt oder mittels simulativer Ansätze erfolgen. Simulationsumgebungen wie

CARLA (Car Learning to Act) bieten eine leistungsstarke Plattform zur Entwicklung und zum Testen von Bildverarbeitungsfunktionen im automobilen Kontext.

Simulatives Testing von Wahrnehmungsfunktionen bietet zahlreiche Vorteile. Es ermöglicht das Testen gefährlicher oder extrem seltener Szenarien ohne Risiko für Menschen oder reale Fahrzeuge, was die Sicherheit erheblich steigert. Zudem sind Tests in Simulationen oft deutlich günstiger, da keine physischen Ressourcen wie Fahrzeuge, Kraftstoff oder Testgelände benötigt werden. Ein weiterer Vorteil ist die Wiederholbarkeit: Szenarien können exakt reproduziert werden, was konsistente Bedingungen für Vergleichstests und Regressionstests sicherstellt. Simulationen bieten auch eine hohe Skalierbarkeit, da eine große Anzahl von Tests schnell und parallel durchgeführt werden kann, was die Testabdeckung erheblich erhöht. Darüber hinaus sind Simulationen flexibel und ermöglichen die einfache Anpassung und Variation von Testbedingungen, um verschiedene Szenarien und Umgebungen zu evaluieren.

Allerdings gibt es auch Nachteile. Simulationen können die Komplexität und Unvorhersehbarkeit der realen Welt nicht vollständig nachbilden, was dazu führen kann, dass bestimmte Probleme oder Verhalten erst im realen Einsatz sichtbar werden. Zudem erfordert die Entwicklung und Pflege realistischer Simulationsmodelle und Umgebungen erhebliche Ressourcen und Expertise. Es besteht auch die Gefahr, dass die Ergebnisse durch ungenaue oder vereinfachte Modelle verfälscht werden, was die Validität der Tests beeinträchtigen kann. Schließlich kann die hohe Abstraktionsebene in Simulationen dazu führen, dass sie nicht alle Aspekte der realen Welt, wie etwa unerwartete Interaktionen oder Umgebungsbedingungen, adäquat erfassen. Speziell im Bereich der Wahrnehmungsfunktionen ist eine realitätsgetreue Nachbildung der Bilder, wie sie durch die Kamera im Fahrzeug aufgenommen werden, essenziell. Leider können aktuelle Simulationsumgebungen dieses Problem noch nicht vollständig lösen.

Die Nutzung von CARLA für das Projekt wurde ausführlich analysiert. Details dazu sind unter "AP 4.4 Test- und Absicherungsstrategie nach A-SPICE" aufgeführt. Zusammenfassend lässt sich sagen, dass CARLA nicht effizient für die Simulation von Lichtreflexionen durch entgegenkommende Fahrzeuge genutzt werden kann. Deshalb ist das simulative Testen hier nicht möglich, und es muss auf reale Daten zurückgegriffen werden. Diese realen Daten stammen aus dem PVDN-Urban Datensatz, welcher in "AP 3.5 Datenlabeling" eingeführt wird.

Performance des Menschen

Um die Wahrnehmungs- und Performancegrenzen des Menschen zu erfassen, wurde eine Probandenstudie durchgeführt. Diese erforschte, wie schnell der Mensch entgegenkommende Fahrzeuge bei Nacht im urbanen Raum erkennen kann, und ob dies vor der direkten Sichtbarkeit der Fahrzeuge möglich ist. Die komplette Studie inklusive aller Details sind der Veröffentlichung zu entnehmen (Ewecker et al.). Zusammenfassend lässt sich folgendes sagen:

In Fällen, in denen entgegenkommende Fahrzeuge keine sichtbaren Lichtreflexionen erzeugen, sind Menschen in der Lage, entgegenkommende Fahrzeuge bei Nacht in städtischen Szenarien mit einer mittleren Verzögerung von 0.62s zu erkennen, nachdem das Fahrzeug direkt sichtbar wurde, wenn sie aufgefordert werden, das Fahrzeug so schnell wie möglich zu erkennen. In Fällen, in denen Lichtreflexionen, die durch das entgegenkommende

Fahrzeug verursacht werden, vor der direkten Sicht sichtbar sind, sind Menschen in der Lage, das Fahrzeug mit einer mittleren Verzögerung von 0.17s zu erkennen, nachdem das Fahrzeug direkt sichtbar wurde. Während die mittlere Erkennungszeit auf eine allgemeine Erkennung des entgegenkommenden Fahrzeugs nach direkter Sicht hinweist, wurden viele Fälle aufgezeichnet, in denen eine Erkennung vor direkter Sicht möglich war.

Die starke Korrelation zwischen der Dauer der Sichtbarkeit von Lichtreflexionen, die durch das entgegenkommende Fahrzeug verursacht werden, und der menschlichen Erkennungszeit sowie die Eye-Tracking-Daten zeigen deutlich, dass Lichtreflexionen das Hauptmerkmal sind, das von Menschen verwendet wird, um entgegenkommende Fahrzeuge vor der direkten Sicht zu erkennen. Sie sollten daher bei der Entwicklung von Datensätzen und Algorithmen berücksichtigt werden.

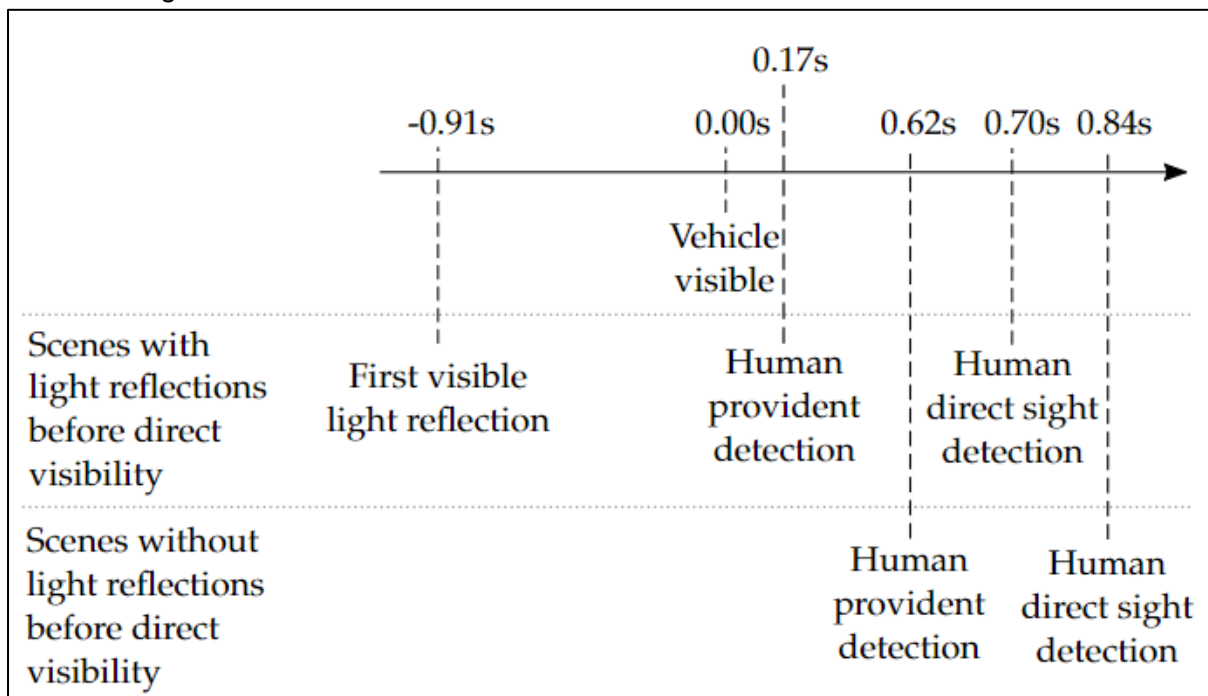


Abbildung 16: Überblick der Erkennungszeiten des Menschen

AP 3.5 Datenlabeling

Zur Annotation der Lichtreflexionen wurden mehrere Ansätze getestet, insbesondere um kostensparende Methoden zu evaluieren, da Datenannotationen im Allgemeinen sehr kostenintensiv sind. Um die Anzahl der zu annotierenden Bilder auf den relevanten Teil zu reduzieren, wurde eigens für das Forschungsvorhaben eine SiderUI entwickelt, mit Hilfe derer die Annotatoren den Anfang und das Ende von Lichtreflexionen entgegenkommender Fahrzeuge markieren konnten. Dadurch sollten die Videoausschnitte selektiert werden, die tatsächlich entgegenkommende Fahrzeuge enthalten. Die tatsächliche Annotation der Lichtreflexionen wurde schlussendlich mit dem unten beschriebenen Verfahren bewerkstelligt. Annotationspipelines für Lichtreflexionen wurden sowohl für falsche positive als auch falsch negative Annotationen in mehreren Iterationen entworfen und eingehend getestet.

Des Weiteren wurde die Annotation mit Laienannotatoren in Form von Gaming Crowds, d.h. Spielern von Online-Browserspielen als kostengünstige Alternative zu professionellen Annotatoren erprobt und evaluiert. Zur Evaluation wurden drei verschiedene kategorische

Annotationsaufgaben zur Qualitätssicherung von bestehenden Lichtreflexionsannotationen in beiden Gruppen von Annotatoren (Laien und professionell) durchgeführt und analysiert:

- Kategorisierung des Lichttyps der markierten Lichtreflexion (Fahrzeugscheinwerfer, Straßenbeleuchtung, andere)
- Kategorisierung von Fahrzeugtypen (PKW, LKW, Bus, Fahrrad/Motorrad)
- Kategorisierung von parkenden und fahrenden Fahrzeugen

Die Annotationen wurden jeweils mit 5 Annotatoren pro Gruppe realisiert. Der Vergleich der Ergebnisse dieser Arten von Annotationsprojekten zeigt, dass sich einfache Fragen, wie die Kategorisierung von Fahrzeugtypen oder parkenden und fahrenden Fahrzeugen mit vergleichbarer Qualität auch durch Laienannotatoren bewerkstelligen lassen. Dies ist ein wichtiges Ergebnis, da die Annotation in den Gaming Crowds um ein Vielfaches günstiger ist, als die Annotation durch professionelle Annotatoren. Es zeigte sich jedoch auch, dass die komplizierteste Frage nach dem Lichttyp in der Annotation durch die Gaming Crowds deutlich schlechtere Ergebnisse liefert als sie durch die professionellen Annotatoren erzielt werden. Daher lässt sich schließen, dass, während Laienannotatoren für einfache Annotationsaufgabe eine kosteneffiziente Alternative darstellen, sie für die Annotation von komplexeren Fragen ungeeignet sind.

Zur Annotation der Fahrzeuge selbst bei Nacht wurde in einem iterativen Prozess ein vortrainiertes Modell zur Fahrzeugerkennung evaluiert und verbessert. Das Vorgehen kombiniert automatische Annotation durch Maschinelles Lernen mit manueller Qualitätsprüfung durch Menschen. Ein Modell, das für die Erkennung von Fahrzeugen bei Tag trainiert wurde, wurde auf die Nachtbilder angewendet. In mehreren Annotationsprojekten wurden die Vorhersagen des Modells sodann durch menschliche Annotatoren überprüft und die Fehler wurden korrigiert. Dazu wurden eigens Annotationspipelines zur Detektion von falsch positiven und falsch negativen Fahrzeugannotationen entworfen, evaluiert und eingesetzt. Nach Abschluss der Annotationen wurde das Modell jeweils erneut auf den verbesserten Daten trainiert. Der gesamte Prozess wurde dreimal an Subsets des Datensatzes wiederholt. Bereits nach der zweiten Iteration erreichte der Recall für die Klasse "car" einen Wert von fast 100%, sowie einen mittleren Wert von 86%, wenn alle Modellvorhersagen berücksichtigt werden. Für einen hohen Konfidenz-Threshold des Modells lässt sich ebenfalls die Precision für alle Klassen auf 100% steigern. Nach Abschluss der drei Iterationen stellte sich für die Klassen "car" und "truck" heraus, dass das iterativ verbesserte Modell insgesamt sowohl weniger falsch positive als auch falsch negative Vorhersagen produziert. Die Verbesserung eines Modells durch Einbeziehung von menschlichen Annotationen kann daher als erfolgreich betrachtet werden.

Zu den anfänglichen Strategien, die für die Annotation von Lichtreflexionen untersucht wurden, gehörten Ansätze für die Beschriftung durch Nicht-Experten unter Verwendung von Spiel-Crowds. Diese Methoden erwiesen sich jedoch als ungeeignet für diese Aufgabe. Stattdessen wurde ein keypoint-basierter Ansatz gewählt, der sich durch seine Effizienz und Genauigkeit auszeichnet, wie in den vorangegangenen Abschnitten der Studie beschrieben.

Annotationsprozess

Experten-Annotationen: Die Annotation wurde von drei Experten durchgeführt. Jedes Bild wurde zunächst von einem Experten annotiert, wobei die Annotationen anschließend von den

beiden anderen überprüft wurden. Dieser Ansatz erwies sich als effizienter, als wenn mehrere Annotatoren dasselbe Bild von Grund auf neu beschriften mussten.

Tooling: Das Annotationstool CVAT (Computer Vision Annotation Tool) wurde für die Aufgabe verwendet. Dieses Tool wurde bereits in früheren Annotationsstudien eingesetzt und hat seine Wirksamkeit und Zuverlässigkeit bewiesen.

Methode: Die Annotation von binären Masken ist eine der kostspieligsten Annotationsformen. Deshalb wurde eine Methode entwickelt, durch welche der Annotator lediglich Keypoints auf eine Reflektion setzen muss, und die dazugehörige Maske automatisch erzeugt wird.

- Vorverarbeitung: Das RGB-Bild wird in Graustufen umgewandelt und dann unscharf gemacht, um das Rauschen zu reduzieren.
- Erzeugung einer binären Maske: Für jeden beschrifteten Keypoint wird ein Flood-Fill-Algorithmus verwendet. Dieser Algorithmus füllt den Bereich um den Keypoint mit einer binären Maske, die die Lichtreflexion hervorhebt.
- Nachbearbeitung: Die binären Masken werden mit Standardverfahren verfeinert, um sicherzustellen, dass kleine Lücken gefüllt werden und die Masken die Lichtreflexionen genau darstellen.
- Kombination der Masken: Abschließend werden alle einzelnen binären Masken zu einer einzigen umfassenden binären Maske für jedes Bild kombiniert, die alle kommentierten Lichtreflexe abdeckt.

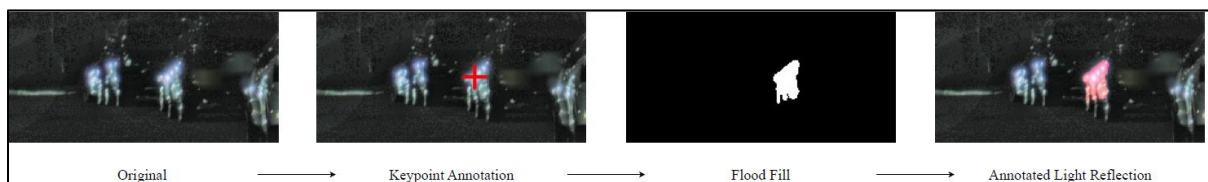


Abbildung 17: Darstellung der Annotationspipeline

Wesentliche Merkmale der Methode:

- Effizienz: Die Keypoint-Annotation ist effizient und erfordert im Vergleich zu komplexeren Annotationsmethoden wie Polygonen nur minimalen Aufwand.
- Integration: Im Gegensatz zu früheren Methoden, bei denen die Keypoints nach der Annotation verwendet wurden, um Saliency Maps zu erstellen, dienen die Keypoints hier direkt als Annotationstool.
- Qualitätskontrolle: Die Annotatoren haben die Möglichkeit, die generierten binären Masken zu überprüfen und anzupassen und eventuelle Ungenauigkeiten zu korrigieren, um qualitativ hochwertige Annotationen zu gewährleisten.

In einer Annotationsstudie wurde der keypoint-basierte Ansatz mit Bounding-Box- und Polygon-Methoden verglichen. Sechs Annotatoren wurden beauftragt, 30 aufeinanderfolgende Bilder mit allen drei Ansätzen zu annotieren. Diese Bilder enthielten verschiedene Arten von Lichtreflexionen von entgegenkommenden Fahrzeugen.

Die Keypoint-Annotation erwies sich als die schnellste Methode, während die Polygon-Annotation am längsten dauerte. Die Übereinstimmung zwischen den Annotatoren, gemessen

durch Intersection over Union (IoU), zeigte eine vergleichbare Genauigkeit bei Keypoints, Bounding Boxes und Polygonen.

Dieser Ansatz unterstreicht die Praktikabilität der Verwendung von Keypoints für die Erstellung binärer Masken und bietet ein gutes Gleichgewicht zwischen Effizienz und Genauigkeit bei der Annotation von Lichtreflexionen in nächtlichen Stadtszenen.

Der doppelte Fokus des PVDN-Urban-Datensatzes auf Lichtreflexionen und Fahrzeugbeschriftungen bietet eine umfassende Grundlage für verschiedene Forschungs- und Anwendungsbereiche, darunter:

- Vorausschauende Fahrzeugdetektion: Das primäre Ziel ist die Nutzung von Lichtreflexionen zur Vorhersage der Anwesenheit entgegenkommender Fahrzeuge.
- Zeitvorhersage: Verwendung von Anmerkungen zur Schätzung der Zeit, bis ein entgegenkommendes Fahrzeug direkt sichtbar wird, was für fortschrittliche Fahrerassistenzsysteme (ADAS) entscheidend ist.
- Erkennung dichter Objekte: Verbesserung von Algorithmen zur Erkennung von nahe gelegenen Objekten bei schlechten Lichtverhältnissen.
- Transfer-Lernen: Bietet eine wertvolle Ressource zum Testen und Verbessern bestehender Algorithmen in städtischen Nachtszenarien.

Der sorgfältige Annotationsprozess des Datensatzes gewährleistet qualitativ hochwertige Daten, die Fortschritte in diesen und anderen verwandten Bereichen unterstützen.

AP 4 Integration und Qualifikation der KI-Funktion

AP 4.1 Bereitstellung einer Schnittstelle für die KI-Funktion in die Fahrzeugarchitektur

Für die Realwelt-Tests im Fahrzeug muss die KI-Funktion in das Fahrzeug integriert werden. Dazu müssen im seriennahen Entwicklungsfahrzeug ein Steuergerät bereitgestellt und Schnittstellen geschaffen werden. Die Bilddaten müssen der Funktion über den Bus bereitgestellt und diese muss entsprechende Ausgaben auf den Bus senden können. Dafür wurden die entsprechenden Schnittstellen zum Fahrzeug BUS vom Car-PC via ROS2 Nodes umgesetzt. Über diese Schnittstellen kann das Fahrzeug sowohl quer als auch längs gesteuert werden.

AP 4.2 Integration der KI-Funktion in den Steuergeräte-Verbund

Um die Funktion später ins Fahrzeug integrieren zu können, wurde hier viele Schnittstellen mit anderen Wirkkettenpartnern im Fahrzeug abgestimmt. Die finale Integration ins Fahrzeug erfolgte jedoch nicht aufgrund des Verzugs in der Funktionsentwicklung.

AP 4.3 Konzeption und Evaluation verschiedener Absicherungsstrategien

Da die Datenanalyse nicht mehr von siasearch durchgeführt wurde, wurde damit begonnen, eine eigene Anwendung für die Visual-Analytics-Methode zu entwickeln. Die Herausforderung bestand darin, ein geeignetes Datenbankschema zu entwickeln, in dem die Metadaten zu den Bildern gespeichert werden und das mit verschiedenen öffentlich verfügbaren Datensätzen funktioniert. Aktuell fehlt noch die Schnittstelle zur Bewertung trainierter Modelle.

Die Visual-Analytics-Methode nutzt die Daten zur Bewertung von Funktionen. In der Entwicklung und Absicherung von SAE Level 3+ Funktionen geht der aktuelle Trend in der Automobilindustrie hin zum szenariobasierten Testen. Szenarien sind dabei insbesondere auf das Verhalten des System Under Test (SUT) und dessen Interaktion mit der Systemumwelt ausgelegt. Dieser Ansatz lässt sich jedoch nicht nahtlos auf bildbasierte Perzeptionsfunktionen übertragen. Stattdessen muss hier mit den in Arbeitspaket 2.3 erarbeiteten Datasplits gearbeitet werden. Es wurden verschiedene Saliency-Map-Methoden entwickelt, die im KI-Kontext kontrovers diskutiert werden, jedoch für AI4OD als anwendbar erschienen. Diese Methoden stammen aus der Computer Vision und wurden angepasst, um zu zeigen, auf welche Bildbereiche eine KI bei der Entscheidungsfindung schaut. Aktuell liegt der Fokus auf der Implementierung und dem Training erster KI-Modelle. Allerdings sind Saliency-Maps aufgrund ihrer Anfälligkeit für Modelländerungen nicht für eine dauerhafte Absicherungsstrategie geeignet. Sie können jedoch sinnvoll eingesetzt werden, wenn das Modell stabil ist und in Kombination mit zusätzlichen Informationen und Methoden, wie SiL-Tests, betrachtet wird. Diese SiL-Tests helfen, Fehler in KI-Modellen zu finden, die dann mit Saliency-Maps gezielt behoben werden können.

Außerdem hat das FZI in Zusammenarbeit mit Porsche, eine Methode zu entwickelt, die es ermöglichen soll, relevante Ereignisse während der Erprobungsfahrten zu erkennen. Aktuell finden dazu Vorarbeiten statt. Das FZI baut dafür eine "Vector Search Engine" auf, die auf der frei und kostenlos verfügbaren Software qdrant basiert. Diese Software hilft dabei, Vektoren

von Bildern einfach zu speichern und in der Datenbank zur Suche ähnlicher Bilder zu verwenden. Diese Eigenschaft der Bildvektoren soll im Projekt genutzt bzw. adaptiert werden, um den Informationsgehalt eines Bildes bzw. den Anteil neuer Informationen eines Bildes im Vergleich zu bereits in der Datenbank vorhandenen Bildern zu bewerten. Die Eignung des Verfahrens zur selektiven, datensparsamen Datenaufnahme soll evaluiert werden.

In der folgenden Abbildung sieht man einen schematischen Ablauf, wie ein Bild in einen Vektor transformiert wird. Aktuell nutzen wir dafür ein ResNet und schneiden den letzten Layer ab, um so einen Vektor mit 512 Dimensionen zu erhalten. ResNet ist ein tiefes neuronales Netzwerk, das zur Klassifikation von Bildern verwendet wird.

Diese Vector-Methode wurde in einem wissenschaftlichen Paper erfolgreich auf der ITSC 2023 eingereicht. In diesem Paper wird ein Konzept vorgestellt, das systematisch Datensätze und deren Einfluss auf die Güte einer trainierten KI-Funktion evaluiert.

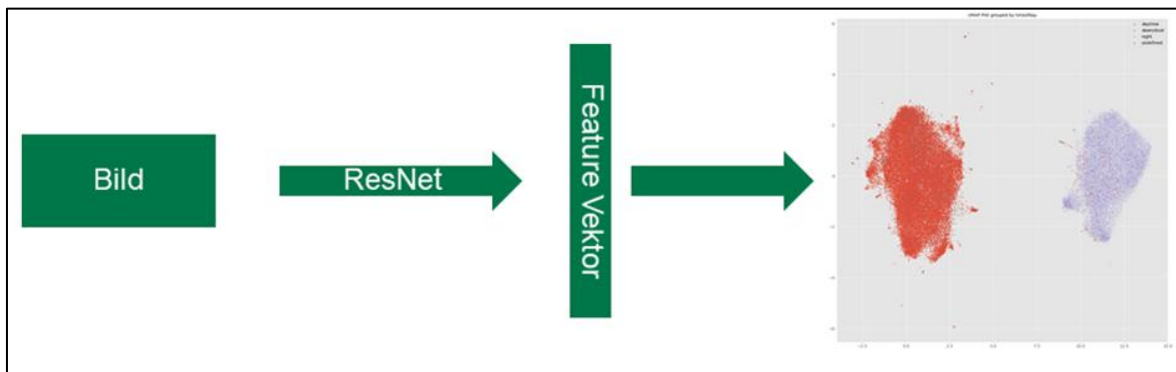


Abbildung 18: Schematischer Ablauf der Transformation eines Bildes in einen Vektor

AP 4.4 Test- und Absicherungsstrategie nach A-SPICE

Aufgrund der aktuell fehlenden, gelabelten Realdaten wurden Methoden zur Datengenerierung und automatischem Labeling in CARLA getestet, um Daten zu generieren. CARLA ist eine frei verfügbare High Fidelity Simulationsumgebung basierend auf Unreal, die sich als Software-in-the-Loop (SiL) Testumgebung für eine zu testende (KI-) Funktion eignet. Dies entspricht dem klassischen Entwicklungsprozess von Funktionen in der Automobilindustrie und lässt sich in den A-SPICE integrieren. Der Plan sah vor, die Simulationsumgebung im SWE.6 Software Qualification Test zu nutzen. Zusätzlich sollte das in AP 2.3 entwickelte Anforderungsdiagramm verwendet werden, um nicht nur die Anforderungen des Systems, sondern auch die Datenanforderungen in der SYS.2 System Requirements Analysis zu ermitteln und festzuhalten. Aufgrund der Verzögerungen war die Umsetzung jedoch nicht mehr vollumfänglich möglich.

Deshalb wurde im Rahmen des AP 4.4 zuerst versucht, anhand der im CARLA Simulator (Version 0.9.14) aufgenommenen Bilder einen gelabelten Datensatz anzufertigen. Dazu wurden einzelne Szenen in CARLA mit unterschiedlichen Lichtverhältnissen aufgezeichnet. Zudem besteht in Carla die Möglichkeit, Instance Segmentation und Semantic Segmentation auf Fotos anzuwenden, da durch die Game Engine schon bekannt ist, wo welche Oberflächen zugeordnet werden müssen und somit eine 100%ig richtige Zuordnung möglich ist. CARLA ist für die gewünschte Bildaufzeichnung aufgrund von Software-Design-Entscheidungen, wie sich herausgestellt hat, nur bedingt geeignet:

- Die asynchrone Kommunikation zwischen Client (Fotofunktion) und Server (Welt) verhindert die genaue Synchronisierung der Lichtquellen. Dazu wären längere Pausen zwischen einzelnen Zeitschritten nötig, was zu Serverabstürzen führen kann.
- Kameraaufzeichnungen sind nur einmal pro Zeitschritt möglich. Statt die Simulation anzuhalten, müssen hierfür wegen der Abstürze aber alle Fahrzeuge angehalten werden.
- Durch die Asynchronität der Kommunikation kann nicht sichergestellt werden, dass vor einer Fotoaufnahme auch wirklich alle Lichtquellen so eingestellt sind, wie benötigt, was zu falschen automatischen Labelling führt.

Deshalb wurde das Labeling per Hand durchgeführt und auf reale Daten von Porsche zurückgegriffen, um ein Netz zu trainieren und zu bewerten.

AP 5 Demonstration

Aufgrund der erheblichen Komplexität im Datenlabeling und dem damit verbundenen Verzug der Funktionsentwicklung konnte die Demonstration im Fahrzeug nicht mehr erfolgen. Die geplante Kapazität wurde hierbei von den Demonstrationsaktivitäten in "AP 3.5 Datenlabeling" verschoben.

AP 5.1 Visualisierung der KI-Funktion im Fahrzeug

Nicht erfolgt, siehe oben.

AP 5.2 Konzeption und Durchführung Probandenstudie

Nicht erfolgt, siehe oben.

AP 5.3 Demonstration der Prozess- und Toolinglandschaft für die A-SPIICE konforme Entwicklung von KI-Funktionen

Durch den Wegfall von siasearch konnten einige der vorangehenden Arbeiten nur lückenhaft oder theoretisch umgesetzt werden, was sich bis zum Demonstrator durchgezogen hat.

Literaturverzeichnis

- Bakhtiari, M., & Villmann, T. (2022). Modification of the Classification-by-Component Predictor Using Dempster-Shafer-Theory. *WSOM*.
- DVC. (2024). Von <https://dvc.org/> abgerufen
- DVC AI. (2024). Von <https://iterative.ai/> abgerufen
- Engelsberger, A., & Villmann, T. (2023). *MOSAIC – ein Algorithmus zur automatischen Fokustrierung bei der Bildklassifikation*. Hochschule Mittweida: interner Bericht.
- Staps, D., Schubert, R., Kaden, M., Lampe, A., Hermann, W., & Villmann, T. (2022). Prototype-based one-class-classification learning using local representations. *IJCNN*.
- Staps, D., Villmann T., & Paaßen, B. (2024). K minimum enclosing balls for outlier detection. *WSOM*.
- Villmann, T., Staps, D., Ravichandran, J., Saralajew, S., Biehl, M., & Kaden, M. (2022). A learning vector quantization architecture for transfer learning based classification in case of multiple sources by means of null-space evaluation. *IDA*.
- Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., . . . Darrell, T. (2020). In *BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning* (S. 2636-2645). Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.