Semantisch durchsuchbares Datenmanagementsystem für die Entwicklung bildbasierter Fahrsysteme

Zur Erlangung des akademischen Grades eines

DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)

von der KIT-Fakultät für
Elektrotechnik und Informationstechnik
des Karlsruher Instituts für Technologie (KIT)

angenommene

DISSERTATION

von

M.Sc. Philipp Rigoll
geb. in Speyer

Tag der mündlichen Prüfung:	14.10.2025
Hauptreferent:	Prof. Dr.-Ing. Eric Sax
Korreferent:	Prof. Dr. Ulrich Schwanecke

Zusammenfassung

Hochautomatisierte Fahrsysteme haben das Potenzial, in Zukunft unsere Mobilität hinsichtlich ökologischer, ökonomischer, sozialer und sicherheitsbezogener Aspekte zu verbessern. Durch den Wegfall des Menschen als Rückfallebene bei der Hochautomatisierung ist während der Entwicklung und beim Testen dieser Systeme besondere Sorgfalt geboten. Fahrsysteme bestehen vermehrt aus bildbasierten, künstlichen neuronalen Netzwerken. Die Bereitstellung von Bildern für die Überprüfung der korrekten Funktionsweise dieser Systeme und insbesondere ihrer Perzeption ist angesichts der Größe der Automotive-Datensätze herausfordernd. Aktuelle Forschungsdatensätze im Bereich der Fahrsysteme umfassen bereits mehrere 100 000 Bilder. Eine manuelle Suche in diesen Datensätzen ist unwirtschaftlich und lässt sich nicht flexibel in den Entwicklungsprozess einbinden.

Diese Dissertation untersucht, wie ein Datenmanagementsystem die Entwicklung und Absicherung kamerabasierter Fahrsysteme ermöglichen kann. Hierfür werden mithilfe des aktuellen Stands der Technik und Wissenschaft Eigenschaften identifiziert, die ein Datenmanagementsystem für die semantische Durchsuchbarkeit aufweisen muss. Ein Konzept für ein Datenmanagementsystem, welches diese Anforderungen adressiert, wird vorgestellt. Das Konzept ist so gestaltet, dass sich das Datenmanagementsystem in den Entwicklungsprozess eines hochautomatisierten Fahrsystems einfügt. Der Kern des Datenmanagementsystems ist die semantische Durchsuchbarkeit und gleichzeitige Unabhängigkeit von manuellen Annotationsaufwänden.

Anhand einer exemplarischen Implementierung des vorgestellten Datenmanagementsystems wird die Einsatzfähigkeit im Entwicklungsprozess mit vier Automotive-Datensätzen erprobt. Der realitätsnahe Einsatz wird zusätzlich im Rahmen einer Nutzerstudie evaluiert. Die Ergebnisse zeigen, dass das entwickelte Datenmanagementsystem bei der Bereitstellung von Bildern für die Entwicklung und Absicherung von hochautomatisierten Fahrsystemen unterstützt.

Diese Dissertation leistet einen Beitrag zur Verbesserung der Mobilität der Zukunft, indem sie die Basis für das Testen von hochautomatisierten, bildbasierten Fahrsystemen schafft. Der Fokus liegt dabei auf einer Lösung, die manuelle Aufwände reduziert und damit wirtschaftlich im Rahmen des Entwicklungsprozesses umsetzbar ist.

Abstract

Highly automated driving systems have the potential to improve our mobility in the future with regard to ecological, economic, social, and safety-related aspects. Due to the elimination of humans as a fallback in highly automated systems, special diligence is required during the development and testing of these systems. Driving systems increasingly consist of image-based, artificial neural networks. Providing images to verify the correct functioning of these systems, especially their perception, is challenging given the size of automotive datasets. Current research datasets in the field of driving systems already include several 100 000 images. Manual searches in these datasets are economically inefficient and cannot be flexibly integrated into the development process.

This dissertation investigates how a data management system can enable the development and validation of camera-based driving systems. For this purpose, the current state of the art and science are used to identify requirements that a data management system must fulfill in order to enable the ability to perform semantic searches. A concept for a data management system that addresses these requirements is presented. The concept is designed in such a way that the data management system fits into the development process of a highly automated driving system. The core of the data management system is the semantic searchability and simultaneous independence from manual annotation efforts.

The presented data management system is implemented exemplarily and the usability in the development process is tested with four automotive datasets. The realistic application of the system is evaluated as part of a user study. The results show that the developed data management system supports the provision of images for the development and validation of highly automated driving systems.

This dissertation contributes to the improvement of future mobility by laying the foundation for testing highly automated, image-based driving systems. The focus is on a solution that reduces manual efforts and can therefore be implemented economically as part of the development process.

Danksagung

Mein herzlichster Dank gilt zunächst meinem Doktorvater Prof. Dr.-Ing. Eric Sax, der mir perfekte Rahmenbedingungen für mein Promotionsvorhaben geschaffen hat und mich stets mit seinem schnellen und konstruktiven Rat unterstützte. Des Weiteren bedanke ich mich bei Prof. Dr. Ulrich Schwanecke für die Übernahme des Korreferats und die anregende Diskussion meines Themas.

Ein großer Dank geht an die gesamte Gruppe von Prof. Dr.-Ing. Eric Sax für viele inspirierende Gespräche, Diskussionen und Einblicke. Dem gesamten Forschungsbereich ESS am FZI Forschungszentrum Informatik möchte ich für das fantastische Arbeitsumfeld und die unermüdliche Unterstützung danken. Durch euch hat mir die Arbeit große Freude bereitet und wahnsinnig viel Spaß gemacht. Im Speziellen möchte ich mich für den liebevoll gestalteten und detailreichen Doktorhut bedanken! Außerdem gilt mein Dank Ulrike Beideck, die mich stets unterstützt und motiviert hat. Für viele langwierige und aufschlussreiche Diskussionen rund um mein Thema gilt mein besonderer Dank Lennart Ries, Jacob Langner, Christian Steinhauser und Patrick Petersen. Im gleichen Atemzug danke ich meinen Berliner Kaffeepausen-Kollegen Jacqueline Henle und Laurenz Adolph für die vielen bereichernden Gespräche.

Besonders möchte ich meiner Familie Kathrin, Fabian, Steffi, Markus, Oskar, Emil, Lena, Paul und vor allem meinen Eltern Michael und Frieda danken. Ihr seid immer ein verlässlicher Anker und eine Unterstützung in jeder Lebenslage. Vielen Dank dafür! Aus tiefstem Herzen danke ich meiner Partnerin Anna Hess für ihr grenzenloses Verständnis, ihre liebevolle Zuneigung und ihre unerschütterliche Unterstützung.

Karlsruhe, im Oktober 2025 *Philipp Rigoll*

Inhaltsverzeichnis

1 Einleitung und Motivation

Die Erforschung und Entwicklung von automatisierten Fahrsystemen intensiviert sich in den letzten Jahren zusehends. Diese Entwicklung lässt sich bei der Betrachtung der wissenschaftlichen Publikationen im Bereich des automatisierten Fahrens in den vergangenen Jahren quantitativ nachvollziehen (siehe Abbildung 1.1). Die Automobilbranche strebt dabei immer höhere Automatisierungsgrade der Fahrsysteme an. Ab der sogenannten Hochautomatisierung der Fahrsysteme steht der

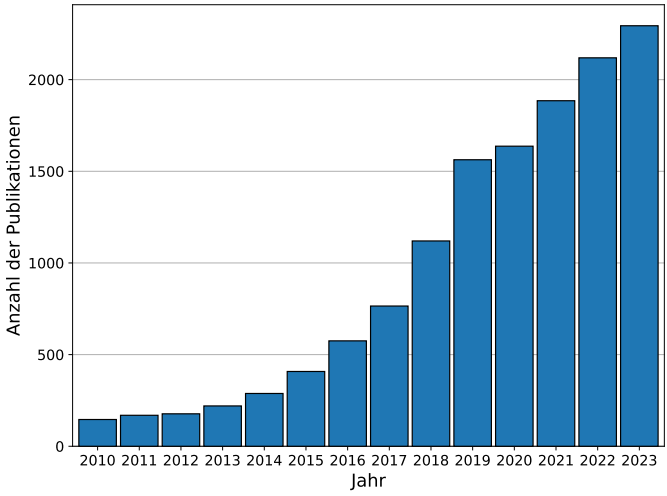


Abbildung 1.1: Entwicklung der Anzahl der veröffentlichten Publikationen im Bereich des automatisierten Fahrens in der Literatur-Datenbank Scopus [1] (zur Erfassung der Daten siehe Abschnitt A.1)

Mensch nicht mehr als Rückfallebene für ein solches System zur Verfügung[1] [2]. Das Fahrsystem ist hierbei so weit automatisiert, dass Objekte und Ereignisse eigenständig erkannt werden müssen. Die auf der Erkennung aufbauenden Klassifizierungen und die daraus folgenden Reaktionen werden dabei nicht mehr von einem Menschen überwacht. Ein manueller Eingriff ist nicht möglich und nicht vorgesehen. Der Fahrer wird demnach vollständig vom Fahrsystem ersetzt, was in der Folge eine Vielzahl von Implikationen nach sich zieht.

1.1 Potenziale von hochautomatisierten Fahrsystemen

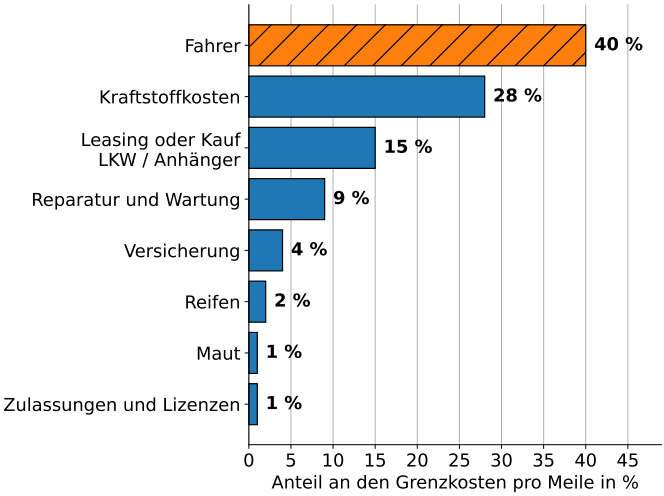


Abbildung 1.2: Zusammensetzung der durchschnittlichen Grenzkosten pro Meile für den LKW-Transport in den Vereinigten Staaten von Amerika in 2022 (nach [3])

[1] Entsprechend der Definition von SAE International J3016 Level 5 [2] (siehe Abschnitt 2.1)

Das Ersetzen des Fahrers durch ein hochautomatisiertes Fahrsystem führt zu Vorteilen auf vielen Ebenen der Mobilität. Bei den LKW-Transport-Dienstleistern wird eine unmittelbare Verringerung der Kosten erwartet. In diesem Wirtschaftszweig machen die Kosten für den Fahrer in den USA einen Anteil von $40\,\%$ an den Kosten pro gefahrener Distanz aus [3] (siehe Abbildung 1.2). Durch die Automatisierung würde ein Großteil dieses Kostenpunktes entfallen. Außerdem fehlten im Jahr 2024 in Deutschland etwa $120\,000$ Berufskraftfahrer [4].

Jedoch versprechen hochautomatisierte Fahrsysteme nicht nur eine ökonomische Verbesserung, sondern es ist auch von einer Steigerung der Sicherheit im Straßenverkehr auszugehen. Laut der National Highway Traffic Safety Administration der Vereinigten Staaten [5] lässt sich ein Anteil von $94\,\%$ der schwerwiegenden Verkehrsunfälle auf menschliche Fehler als Unfallursache zurückführen. Durch den kompletten Wegfall des Menschen im Straßenverkehr wird dieser Unfallgrund obsolet. Somit besteht beim breiten Einsatz von abgesicherten und zuverlässigen hochautomatisierten Fahrsystemen die Hoffnung, dass sich die absolute Zahl der schwerwiegenden Unfälle reduziert.

Zusätzlich gibt es ökologische Argumente, die für eine rasche Entwicklung von hochautomatisierten Fahrsystemen sprechen. Denn mit hochautomatisierten Fahrsystemen ausgerüstete Fahrzeuge können die verfügbare Straßenkapazität um bis zu $80\,\%$ besser ausnutzen als menschliche Fahrer [6]. Für die verbesserte Ausnutzung ist es notwendig, dass ein Straßenabschnitt ausschließlich von hochautomatisierten und vernetzten Fahrzeugen verwendet wird. Das hat zur Folge, dass die Fahrzeuge mit geringerem Abstand und höheren Geschwindigkeiten fahren können. Diese effizientere Fahrweise kann zu einer Verringerung der Umweltbelastung führen [5].

Daneben sind soziale Dimensionen von hochautomatisierten Fahrsystemen von entscheidender Bedeutung. Hochautomatisierte Fahrsysteme haben das Potenzial, die persönliche Mobilität von Menschen, die bislang durch körperliche Einschränkungen oder eine fehlende Fahrerlaubnis vom Individualverkehr ausgeschlossen

waren, zu steigern [5]. Die zu erwartenden Vorteile beim Einsatz von hochautomatisierten Fahrsystemen sind vielfältig und betreffen mehrere Schichten der Mobilität.

1.2 Herausforderungen bei der Entwicklung von hochautomatisierten Fahrsystemen

Jedoch bedeutet die Hochautomatisierung der Fahrsysteme und der Wegfall des Menschen als Rückfallebene auch, dass die Fahrsysteme eigenständig die Situationsvielfalt im Straßenverkehr bewältigen müssen. Um eine verlässliche Funktionsweise zu gewährleisten, sind schon während der Entwicklung Tests der Systeme notwendig. Die Verarbeitung eines Fahrsystems basiert auf der Wahrnehmung der Umgebung. Zum Sicherstellen der Robustheit gegenüber vielfältigen Umgebungsbedingungen wird das Fahrsystem daher mit entsprechenden Eingangsdaten getestet. Zur Erfassung der Umgebung kommen dabei als wesentlicher Sensortyp Kameras zum Einsatz. Daher enthält auch der überwiegende Teil der Forschungsdatensätze zum hochautomatisierten Fahren Bilder [7, 8].

Jedoch ist es nicht möglich, als Eingangsdaten jedes theoretisch mögliche Bild zu testen. Geht man beispielsweise von einer Kamera-Auflösung von $1928 \times 1208\,\mathrm{Pixeln}$ aus, wie sie in einem populären Datensatz [9] der Fahrsystem-Entwicklung verwendet wird. Und nimmt weiter an, dass jedes Pixel 3 Farbkanäle mit jeweils 256 möglichen Farben aufzeichnet. So folgt für die Anzahl aller theoretisch möglichen Bilder:

$$256^{1928 \cdot 1208 \cdot 3} \approx 1{,}1 \cdot 10^{16826546}.$$

Diese Zahl ist größer als eine aktuelle Abschätzung über die Anzahl der Teilchen im beobachtbaren Universum mit einem Wert von $4 \cdot 10^{80}$ [10]. Auch wenn man in der Lage wäre, ein hochautomatisiertes Fahrsystem in jeder Sekunde mit so vielen Bildern zu testen, wie es Teilchen im beobachtbaren Universum gibt, würde der

Abbildung 1.3: Beispiel für eine herausfordernde Situation: überbelichtetes Bild durch die Sonne im Blickfeld der Kamera im Forschungsdatensatz KITTI [11] (zu KITTI siehe Unterabschnitt 5.2.3)

Test für alle theoretisch möglichen Bilder eine Zeit von $8{,}7 \cdot 10^{16826457}$ Jahren beanspruchen.

Obgleich es nicht möglich ist, Fahrsysteme mit jedem theoretisch möglichen Bild zu testen, sind die potenziell herausfordernden Situationen, in denen sie zuverlässig funktionieren müssen, vielfältig. Diese Vielfalt muss sich in den Bildern für die Tests eines hochautomatisierten Fahrsystems widerspiegeln.

Ein Beispiel für eine Herausforderung ist der begrenzte Dynamikumfang einer Kamera bezüglich der Helligkeit. So ist es möglich, dass eine direkte Sonneneinstrahlung in die Kameraoptik zu einer Überbelichtung einzelner Bildbereiche führt (siehe Abbildung 1.3). Gleichzeitig werden dunklere Bildbereiche, wie der Bereich unter einer Brücke, unterbelichtet sein (siehe Abbildung 1.4). Durch diese Einschränkung ist es möglich, dass Informationen aus der Umgebung, wie beispielsweise ein Fahrzeug unter einer Brücke, in den Bilddaten nicht ausreichend repräsentiert sind und von der Fahrfunktion in der Folge nicht wahrnehmbar sind. Jedoch sind nicht nur direkte Lichteinstrahlung in die Kameraoptik und der Dynamikumfang eine Herausforderung. Auch Wettereinflüsse, wie Regentropfen und Nebel (siehe Abbildung 1.5), können dazu führen, dass die Umgebung durch die Fahrfunktion unzureichend wahrgenommen wird.

Nach der Aufnahme eines Bildes mit einer Kamera ist die Wahrnehmung der Umgebung jedoch nicht abgeschlossen. Das aufgenommene Bild muss anschließend im Kontext des hochautomatisierten Fahrsystems verarbeitet und interpretiert werden. Diese korrekte Interpretation eines Bildes erfordert jedoch ein grundlegendes physikalisches und inhaltliches Verständnis der Realität. Ein Beispiel für

Abbildung 1.4: Beispiel für eine herausfordernde Situation: Bild aus dem Forschungsdatensatz KITTI [11] (zu KITTI siehe Unterabschnitt 5.2.3) mit unter- und überbelichteten Bildbereichen

(a) Regentropfen

(b) Nebel

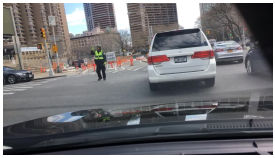
Abbildung 1.5: Beispiele für herausfordernde Wettereinflüsse in Bildern aus dem BDD100K-Datensatz [12] (zu BDD100K siehe Unterabschnitt 5.2.2)

diese Herausforderung sind Verkehrsspiegel. Für die korrekte Interpretation eines Bildes mit einem Verkehrsspiegel muss ein hochautomatisiertes Fahrsystem die physikalische Funktionsweise eines Spiegels verstehen und auf die gegenwärtige Situation anwenden (siehe Abbildung 1.6).

Aber auch einzelne Objekte innerhalb eines Bildes können eine Herausforderung darstellen (siehe Abbildung 1.7). Beispielsweise muss ein Fahrsystem die Überlänge einer Stretch-Limousine beachten. Oder einem Polizisten, der auf der Kreuzung steht und den Verkehr regelt, muss besondere Aufmerksamkeit geschenkt werden. Aber auch temporäre Fahrbahnmarkierungen müssen zuverlässig erkannt werden.

Abbildung 1.6: Beispiel für eine herausfordernde Situation: Bildausschnitt mit einem Verkehrsspiegel aus einem Bild des A2D2-Datensatzes [9]

(a) Stretch-Limousine (BDD100K-Datensatz [12] siehe Unterabschnitt 5.2.2)



(b) Polizist der den Verkehr regelt (BDD100K-Datensatz [12] siehe Unterabschnitt 5.2.2)

(c) Temporäre Fahrbahnmarkierungen (A2D2-Datensatz [9])

Abbildung 1.7: Beispiele für Objekte, die potenziell herausfordernd für hochautomatisierte Fahrsysteme sind

Dies ist nur eine exemplarische Auswahl von potenziell herausfordernden Situationen und Eingabebildern für Fahrsysteme. Im Folgenden wird davon ausgegangen, dass bereits eine Liste mit Beschreibungen von herausfordernden Situationen existiert. Für die Tests müssen jedoch noch entsprechende Bilder als Eingabedaten bereitgestellt werden. Hierfür wird auf Datensätze zurückgegriffen, die aus Bildern bestehen, die im realen Straßenverkehr aufgezeichnet wurden. In diesen Datensätzen müssen die zu den Herausforderungen passenden Bilder identifiziert werden. Ein solcher Datensatz ist zum Beispiel der Forschungsdatensatz BDD100K [12], in welchem alleine schon 100 000 Bilder enthalten sind. Die herausfordernden Situationen in einem solchen Datensatz zu identifizieren, ist mit der semantischen Annotation des Datensatzes vergleichbar. Eine solche manuelle Annotation ist jedoch kostenintensiv (für eine beispielhafte Kostenabschätzung siehe Abschnitt A.2). Geht man dennoch davon aus, dass der gesamte Datensatz hinsichtlich der bekannten Herausforderungen annotiert wurde, ergibt sich ein weiteres Problem. Werden während der weiteren Entwicklung neue, herausfordernde Situationen identifiziert, wäre zur Identifikation dieser eine erneute manuelle Annotation des gesamten Datensatzes notwendig.

1.3 Idee: semantisch durchsuchbares Datenmanagementsystem

Diese Dissertation untersucht, wie ein Datenmanagementsystem zu gestalten ist, um den Problemstellungen bei der Verwaltung und Zusammenstellung von Bilddatensätzen für die Entwicklung und Absicherung kamerabasierter Fahrsysteme zu begegnen. Ein wichtiger Punkt ist die interaktive Einbindung der Lösung in den Entwicklungsprozess. Außerdem muss das Datenmanagementsystem die Semantik der Bilder berücksichtigen. Der Fokus liegt auf einer Lösung, die den manuellen Annotationsaufwand reduziert.

Forschungsfragen

Aus den identifizierten Potenzialen, Herausforderungen und der grundsätzlichen Idee ergeben sich folgende Forschungsfragen:

Forschungsfrage 1

Welche Eigenschaften muss ein Datenmanagementsystem für die Entwicklung von bildbasierten Fahrsystemen besitzen?

Forschungsfrage 2

Wie muss ein Datenmanagementsystem für die Entwicklung von bildbasierten Fahrsystemen konzipiert sein?

Forschungsfrage 3

Wie lässt sich ein Datenmanagementsystem für die Entwicklung von bildbasierten Fahrsystemen realisieren?

1.4 Gliederung der Arbeit

Diese Dissertation hat folgenden Aufbau: Im Anschluss an die Einleitung und Motivation werden die Grundlagen von automatisierten, bildbasierten Fahrsystemen, Kontexten in Automotive-Bilddatensätzen und maschinellen Lernverfahren erläutert (vgl. Kapitel 2). Anschließend wird der aktuelle Stand der Wissenschaft und Technik im Bereich der Bereitstellung von Bildern im Rahmen der Entwicklung von bildbasierten Fahrsystemen diskutiert und aus dieser Analyse werden Anforderungen für ein Datenmanagementsystem abgeleitet (vgl. Kapitel 3). Darauf folgt die Vorstellung eines Konzepts und Entwurfs für ein semantisch durchsuchbares Datenmanagementsystem im Kontext der Entwicklung eines automatisierten, bildbasierten Fahrsystems (vgl. Kapitel 4). Anhand einer prototypischen Realisierung dieses Entwurfs eines Datenmanagementsystems werden Experimente durchgeführt (vgl. Kapitel 5). Diese Experimente dienen als Grundlage für die darauffolgende Evaluation und Diskussion (vgl. Abschnitt 5.8). Zum Schluss werden die gewonnenen Erkenntnisse zusammengefasst und ein Ausblick auf zukünftig notwendige Forschungen gegeben (vgl. Kapitel 6).

2 Grundlagen

2.1 Automatisierte Fahrsysteme

Automatisierte Fahrsysteme unterstützen den Fahrer bei der Durchführung der Fahraufgabe, also der lateralen und longitudinalen Regelung des Fahrzeugs oder übernehmen sie ganz oder teilweise [13]. Der Begriff des automatisierten Fahrsystems schreibt dabei weder den Grad noch die Art der Automatisierung fest. Der Trend geht jedoch zu immer höheren Automatisierungsstufen, bei denen das automatisierte Fahrsystem immer größere Teile der Fahraufgabe übernimmt und den Fahrer entlastet.

2.1.1 Einteilung von Fahrsystemen

Fahrsysteme werden hinsichtlich unterschiedlichster Aspekte kategorisiert. Von besonderem Wert ist die Einteilung hinsichtlich des Automatisierungsgrades. Normiert wurde diese Kategorisierung durch den Verband der Automobilingenieure in den USA[1]. Dieser teilt Fahrsysteme hinsichtlich ihres Automatisierungsgrades in fünf Level auf (siehe Tabelle 2.1). Je höher das Level, desto größer ist der Einfluss des Fahrsystems auf die Fahraufgabe. Auf dem nullten Level existiert keine Automatisierung und der Fahrer steuert das Fahrzeug eigenständig. Bis zum vierten Level muss der menschliche Fahrer auf Anfrage des Fahrsystems als Rückfallebene zur Verfügung stehen. Ab dem vierten Level ist die Automatisierung

[1] SAE International

Tabelle 2.1: Automatisierungslevel nach der Definition des Verbands der Automobilingenieure (SAE International) [2]

Level	Bezeichnung	Beschreibung
0	Keine Automatisierung der Fahraufgabe	Ausführung der gesamten fahrdynamischen Aufgabe durch den Fahrer, auch wenn er durch aktive Sicherheitssysteme unterstützt wird.
1	Fahrerassistenz	Die anhaltende und betriebsbereichsspezifische Ausführung entweder der Quer- oder der Längsführung der dynamischen Fahraufgabe (aber nicht beider gleichzeitig) durch ein Fahrerassistenzsystem in der Erwartung, dass der Fahrer den Rest der dynamischen Fahraufgabe ausführt.
2	Teilautomatisierung der Fahraufgabe	Die dauerhafte und betriebsbereichspezifische Ausführung der dynamischen Fahraufgabe (Quer- und Längsführung des Fahrzeugs) durch ein Fahrerassistenzsystem in der Erwartung, dass der Fahrer die Teilaufgabe Objekt- und Ereignisdetektion und Reaktion ausführt und das Fahrautomatisierungssystem überwacht.
3	Bedingte Automatisierung der Fahraufgabe	Die dauerhafte und betriebsbereichsspezifische Durchführung der gesamten dynamischen Fahraufgabe durch ein automatisiertes Fahrsystem in der Erwartung, dass der Nutzer, der für die dynamische Fahraufgabe als Rückfallebene dient, für die vom automatisierten Fahrsystem ausgegebenen Aufforderungen zum Eingreifen sowie für die Leistung der dynamischen Fahraufgabe relevante Systemausfälle in anderen Fahrzeugsystemen aufnahmebereit ist und entsprechend reagiert.
4	Hochautomatisierte Fahrfunktion	Die dauerhafte und betriebsbereichsspezifische Durchführung der gesamten dynamischen Fahraufgabe und die Bereitstellung einer Rückfallebene für die dynamische Fahraufgabe durch ein automatisiertes Fahrsystem, ohne die Erwartung, dass der Nutzer eingreift.
5	Vollautomatisierte Fahrfunktion	Die dauerhafte und bedingungslose (d. h. nicht betriebsbereichsspezifische) Durchführung der gesamten dynamischen Fahraufgabe und die Bereitstellung einer Rückfallebene für die dynamische Fahraufgabe durch ein automatisiertes Fahrsystem, ohne die Erwartung, dass der Nutzer eingreift.

so umfassend, dass der menschliche Fahrer innerhalb von definierten Rahmenbedingungen nicht mehr als Rückfallebene zur Verfügung stehen muss. Ab diesem Level spricht man von hochautomatisierten Fahrsystemen. Bei einer vollen Automatisierung (Level 5) ist unter keinen Umständen mehr ein menschlicher Fahrer zur Bewältigung der Fahraufgabe notwendig [2].

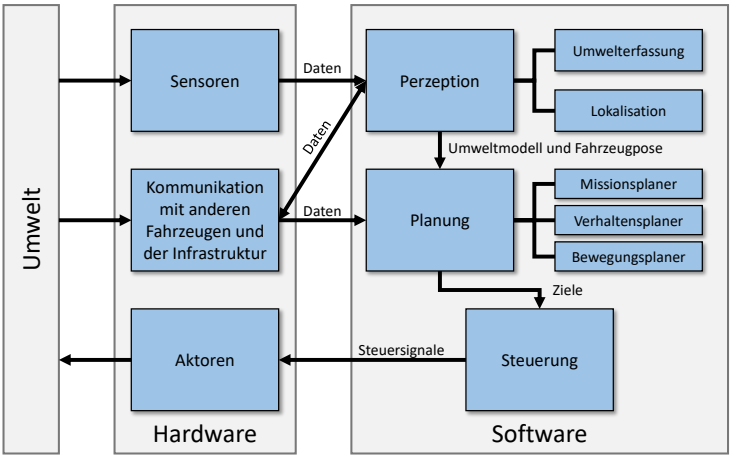


Abbildung 2.1: Teilfunktionen eines automatisierten Fahrsystems (nach [14])

2.1.2 Perzeption, Planung und Steuerung

Entwickler von automatisierten Fahrsystemen zerlegen die Fahraufgabe in einzelne Teilprobleme. Bei der Untersuchung der Funktionsweise von automatisierten Fahrsystemen werden diese daher als einzelne Teilfunktionen betrachtet. Üblich ist eine Gliederung in die Perzeption, die Planung und die Steuerung [14]. Die Perzeption beschreibt die Wahrnehmung der Umgebung und die Eigenlokalisation. Sie bildet somit die Grundlage für den darauf aufbauenden Planungsschritt. Schlussendlich werden die Pläne im Steuerungsschritt in konkrete Handlungen umgewandelt und die Fahrzeugaktoren angesteuert (siehe Abbildung 2.1).

Perzeption

Die Perzeption basiert auf unterschiedlichsten Sensoren und Datenquellen. Als Sensoren und Datenquellen kommen Kameras, Lidar-Sensoren, Radar-Sensoren, globale Navigationssatellitensystem-Sensoren, Inertialsensoren, Odometrie, aber

auch Informationen aus der Kommunikation mit anderen Fahrzeugen und Infrastrukturelementen zum Einsatz [14]. Die durch die Sensoren gewonnenen Messwerte und Daten erzeugen ein Modell der Umgebung. Neben diesem Umgebungsmodell ist die Lokalisation zentraler Bestandteil der Perzeption. Die Lokalisation verortet das Fahrzeug innerhalb des Umgebungsmodells. Das Resultat ist eine Pose, also eine Position und Orientierung, des Fahrzeugs im Umgebungsmodell. Das Problem der simultanen Positionsbestimmung und Umgebungsmodellierung[2] ist ein bekanntes Problem der Robotik [15]. Das Ergebnis des Perzeptionsschritts liefert die Basis für alle weiteren Teilfunktionen. Daher wirkt sich die Robustheit der Perzeption direkt auf die Robustheit des kompletten automatisierten Fahrsystems aus. Das Umgebungsmodell und die Pose des Fahrzeugs werden im Planungsschritt weiterverarbeitet.

Planung

Diese Teilfunktion bestimmt die notwendigen Schritte zur Bewältigung der Fahraufgabe. Die Datengrundlage ist die Pose und das Umgebungsmodell der Perzeption. Die Planung wird hierzu üblicherweise hierarchisch in drei Stufen unterteilt: Missionsplanung, Verhaltensplanung und Bewegungsplanung. Die jeweiligen Planer unterscheiden sich hinsichtlich ihrer Granularität. Der Missionsplaner entscheidet über die Route, also darüber, welche Straßen zur Erreichung eines Ziels befahren werden müssen. Basierend auf diesen groben Entscheidungen agiert der Verhaltensplaner. Dieser trifft zum Beispiel die Wahl der konkreten Fahrbahn und plant die Interaktion mit anderen Verkehrsteilnehmern. Die detailliertesten Entscheidungen übernimmt der Bewegungsplaner, welcher konkrete Aktionen und Trajektorien festlegt [14].

[2] Simultaneous localization and mapping (SLAM)

Steuerung

Im Steuerungsschritt werden die von der Planung festgelegten Entscheidungen umgesetzt. Hierzu wird die Aktorik des Fahrzeugs zur Längs- und Querführung entsprechend angesteuert. Die Fahrzeugsensorik überwacht hierbei die Umsetzung. Der gemessene Umsetzungsfehler wird zurückgeführt und ein Regler passt die Ansteuerung der Aktorik entsprechend an. Der Steuerungsschritt kann demnach als Regelkreis beschrieben werden [14].

2.1.3 Pegasus-Ebenen

Automatisierte Fahrsysteme sind äußeren Einflüssen ausgesetzt. Das Forschungsprojekt „Pegasus“[3] [17] hat diese Einflüsse in Ebenen kategorisiert (siehe Tabelle 2.2). Diese Einteilung hat sich zur Beschreibung von Szenarien im Kontext der Entwicklung von automatisierten Fahrsystemen durchgesetzt. Die unterste Pegasus-Ebene „E1 Straßenebene“ enthält Informationen über die Geometrie und Topologie des Straßenverlaufs und die Beschaffenheit und Begrenzungen der Fahrbahn. Die zweite Pegasus-Ebene beschreibt bauliche Barrieren, Beschilderungen und Leiteinrichtungen. Pegasus-Ebene E3 bezieht sich auf zeitlich begrenzte Beeinflussungen der Pegasus-Ebenen E1 und E2. Darunter fallen Einflüsse, die mehr als einen Tag andauern. Ein Beispiel hierfür sind Baustellen. Dynamische und bewegliche Objekte werden von Pegasus-Ebene E4 abgedeckt. Diese Pegasus-Ebene behandelt auch Interaktionen und Manöver. Pegasus-Ebene E5 Umgebungsbedingungen beschreibt weitere Einflüsse der Umgebung, wie das Wetter und die Lichtverhältnisse. Nicht direkt sichtbare Einflüsse in Form von digitalen Informationen werden in der sechsten Pegasus-Ebene beschrieben. Beispiele hierfür sind alle Kommunikationsdaten (V2X), also etwa die Kommunikation zwischen Fahrzeugen (V2V) oder zwischen einem Fahrzeug und Infrastruktur-Einrichtungen (V2I). [17]

[3] Pegasus: **P**rojekt zur **E**tablierung von **g**enerell **a**kzeptierten Gütekriterien, Werkzeugen und Methoden sowie **S**zenarien **u**nd **S**ituationen zur Freigabe hochautomatisierter Fahrfunktionen. [16]

Tabelle 2.2: Übersicht über die Pegasus-Ebenen [17]

Ebene	Konkretisierungen
E1 Straßenebene	Geometrie und Topologie Beschaffenheit und Begrenzung
E2 Leitinfrastruktur	Bauliche Barrieren Schilder und Leiteinrichtungen
E3 Temporäre Beeinflussung E1/E2	Geometrie oder Topologie überlagert zeitlich > 1 Tag
E4 Objekte	Dynamische und bewegliche Objekte Interaktion und Manöver
E5 Umgebungsbedingungen	Wetter Lichtverhältnisse
E6 Digitale Informationen	V2X-Informationen Digitale Karten

Durch das Dekomponieren der Einflüsse in die Pegasus-Ebenen, wird die Beschreibung von Automotive-Daten systematisiert. Beispielsweise kann man auf Pegasus-Ebene E5 festlegen, dass eine Situation Regen umfassen soll. Gleichzeitig muss die Art und Anzahl der dynamischen Objekte (Pegasus-Ebene E4) nicht spezifiziert werden. Diese Einteilung wird im Folgenden verwendet, um gesuchte Situationen zu beschreiben, aber auch um zu bestimmen, auf welche Pegasus-Ebenen eine Suchmethode angewendet werden kann. Da sich die folgenden Ausführungen auf Bilder beschränken, ergibt sich, dass die Pegasus-Ebene E6 Digitale Informationen im Weiteren nicht betrachtet wird.

2.2 Bildbasierte Perzeption

Bilder spielen eine zentrale Rolle in den Datensätzen, die im Zusammenhang mit automatisierten Fahrsystemen veröffentlicht wurden. Nur in den seltensten Fällen kam bei der Datenaufzeichnung keine Kamera zum Einsatz [7, 18, 19]. Ein Grund für den verbreiteten Einsatz von Kameras bei automatisierten Fahrsystemen liegt darin, dass die visuelle Wahrnehmung für den Menschen bei der Fahraufgabe die wichtigste Informationsquelle ist [20]. Ein weiterer Grund ist die Bandbreite der Aufgaben, für welche Kameras die Datengrundlage liefern. Zu diesen Aufgaben gehören die Erkennung räumlicher Merkmale der Umgebung, wie die Größe, Form und Struktur von Objekten, aber auch die Erfassung von Objektklassen und konkreten Objektidentitäten. Eine weitere Aufgabe ist die Wahrnehmung des befahrbaren Bereichs und von Regularien in Form von Straßenschildern und Fahrbahnmarkierungen. Aber auch die Erkennung von Kontexten, Fahrzeuglichtern, Wetterlagen und Fahrsituationen [14, 21].

Funktionsweise einer Kamera

Das Wort Kamera stammt vom neulateinischen Wort *camera obscura*, welches sich mit *dunkle Kammer* übersetzen lässt. Darunter versteht man einen dunklen Raum oder Kasten mit einem Loch. Durch dieses Loch tritt Licht ein und an der dem Loch gegenüberliegenden Fläche entsteht ein Abbild der Umgebung [22]. Auch moderne Kameras arbeiten noch nach einem ähnlichen Prinzip (siehe Abbildung 2.2). Lichtquellen senden Licht in Form von elektromagnetischen Wellen aus, welche auf die Objekte der Umgebung treffen und von diesen gestreut werden. Das Licht, welches in das Objektiv der Kamera gestreut wird, wird von der Kamera verarbeitet. Das Objektiv ist aus Linsen aufgebaut, die das Licht sammeln und auf dem Bildsensor abbilden. Der Bildsensor besteht aus einzelnen Lichtsensoren. In diesen Lichtsensoren wird die elektromagnetische Strahlung durch den inneren photoelektrischen Effekt in elektrische Ladung umgewandelt [23]. Die elektrische Ladung wird anschließend aus den einzelnen Sensorelementen ausgelesen und in Intensitätswerte übersetzt. Zur Unterscheidung von Farben ist

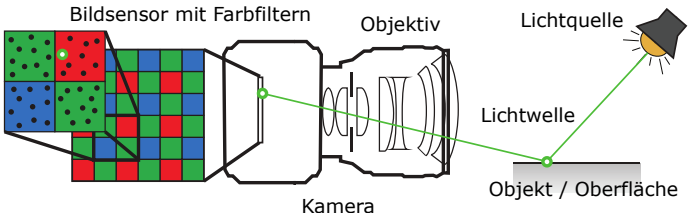


Abbildung 2.2: Strahlengang und prinzipieller Aufbau einer digitalen Kamera (nach [24])

vor den Lichtsensoren ein Gitter aus Farbfiltern angebracht, das jeweils nur eine Farbe passieren lässt. Üblicherweise werden rote, grüne und blaue Farbfilter oder die dazu komplementären Farben Cyan, Magenta und Gelb verwendet [23]. Die Transformation dieser einzelnen Intensitätsmessungen hin zu Farbwerten eines einzelnen Farbpixels erfolgt durch die Interpolation der Intensitätswerte von benachbarten Messungen.

Das resultierende Bild $\mathbf{I} \in \mathbb{N}^{w \times h \times c}$ besteht aus Intensitätswerten. Dabei beschreiben w und h die Breite und Höhe des Bildes in Farbpixeln und c die Anzahl der Farbkanäle. In der Regel werden die Intensitätswerte mit $8\,\mathrm{Bit}$ kodiert, wodurch $2^8 = 256$ mögliche Intensitätswerte pro Farbpixel und Farbkanal bzw. $256^3 = 16\,777\,216$ mögliche Farben pro Pixel repräsentiert werden können [25].

Sensoreffekte

Eine Kamera erzeugt kein exaktes Abbild der Umgebung. Durch die Bauform des Objektives und des Bildsensors kommt es zu Abbildungsfehlern des eintreffenden Lichts. Die durch die optischen Elemente hervorgerufenen Abbildungsfehler werden Aberrationen genannt. Hier unterscheidet man Fehler, die in der physikalischen Optik begründet sind und solche, die durch Toleranzen bei der Fertigung entstehen. Zusätzlich sorgt auch der Bildsensor durch seinen Aufbau und Rauschen zu Abweichungen von einer perfekten Abbildung [25, 26].

Einteilung, Abgrenzung und Limitationen

Kamerasysteme können in drei Klassen eingeteilt werden:

- die einfachste Variante besteht aus einer einzelnen monokularen Kamera,
- für die Abdeckung eines größeren Blinkwinkels werden mehrere monokulare Kameras kombiniert und
- binokulare Kameras werden verwendet, um zusätzlich dreidimensionale Informationen zu erhalten [21].

Die weiteren Ausführungen beschränken sich ausschließlich auf die Daten einzelner monokularer Kameras. Abzugrenzen sind Kameras außerdem von anderen Sensormodalitäten, wie Lidar- und Radar-Sensoren. Gemeinsam haben diese Sensortypen mit Kameras, dass sie elektromagnetische Wellen registrieren. Ein Unterschied ist dabei jedoch das jeweilige Frequenzspektrum. Im Folgenden werden Kameras betrachtet, die im sichtbaren Frequenzspektrum (von $\sim 400\,\mathrm{nm}$ bis $\sim 800\,\mathrm{nm}$; siehe Abbildung 2.3) messen und dabei passiv arbeiten. Passiv bedeutet, dass im Unterschied zu Lidar- und Radar-Sensoren, keine elektromagnetischen Wellen ausgesendet, sondern ausschließlich einfallende elektromagnetische Wellen gemessen werden. Daraus folgt allerdings auch eine starke Abhängigkeit von den Umgebungsbedingungen. So können schlechte Beleuchtungssituationen zu fehlerhaften und ungenauen Bildern führen [27]. Bekannte Situationen, die eine Herausforderung für Kameras darstellen, sind unter anderem: geringe Ausleuchtung, Regen, Staub und Nebel [21]. Zur Bewältigung dieser Situationen wird auf die Fusion mehrerer Sensormodalitäten zurückgegriffen. Dabei sollen die Schwächen von Kameras, Lidar- und Radar-Sensoren, durch jeweils andere Sensormodalitäten kompensiert werden [14, 28].

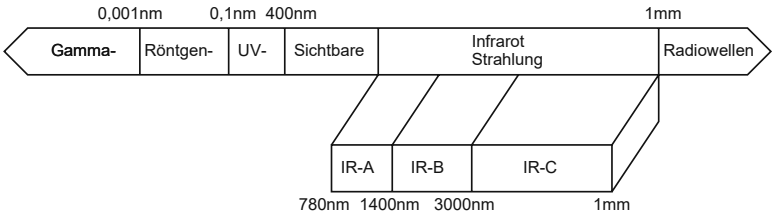


Abbildung 2.3: Elektromagnetisches Frequenzspektrum mit dem sichtbaren Bereich zwischen $\sim$400 nm und $\sim$800 nm [23]

2.3 Kontext

Sensordaten und damit auch Bilder werden nicht isoliert aufgezeichnet. Bereits bei der Aufzeichnung selbst fallen weitere Daten an. Diese Daten, die zusätzliche Informationen über andere Daten bereitstellen, werden Kontexte genannt [29]. Es hängt demnach von der konkreten Problemstellung und der damit verbundenen Sichtweise ab, welche Daten als Kontextdaten gelten. Kontextdaten zeichnen sich durch ihre Strukturiertheit aus, die eine automatisierte Verarbeitung zulässt [30].

Historisch stammt das Konzept der Kontextdaten aus Bibliotheken. Dort werden Bücher katalogisiert und die Durchsuchbarkeit der Bibliothek mit Kontextdaten verbessert [31]. Kontextdaten tragen auch heute noch dazu bei, dass die Daten, auf die sie sich beziehen, besser auffindbar sind. Im Folgenden bezieht sich der Begriff der Kontextdaten auf alle Daten, welche einem Bild, zugeordnet werden können und dieses genauer charakterisieren.

Kontextdaten von Bildern im Rahmen von automatisierten Fahrsystemen stammen aus unterschiedlichen Datenquellen. Ein Teil der Kontextdaten wird bereits bei der Aufzeichnung eines Bildes gespeichert. Das können beispielsweise Fahrzeugzustände, wie die aktuelle Geschwindigkeit, die Fahrzeugposition und die Fahrtrichtung, sein. Es kann sich aber auch um die Informationen anderer Sensorik für die Umgebungserkennung, wie die Regenmenge oder den Abstand zu einem vorausfahrenden Fahrzeug, handeln. Außerdem werden auch nach der Aufzeichnung Kontextdaten ergänzt [32]. Quellen hierfür sind unter anderem Algorithmen

zur Kombination vorhandener Kontextdaten, Wetterdienste, aber auch Kartenanbieter. Daraus lassen sich unter anderem Informationen wie Fahrmanöver [33] bestimmen. Daneben gibt es auch manuelle Annotationen. Diese können während oder nach der Datenaufnahme erfolgen. Es wird beispielsweise annotiert, ob es sich um eine kritische Fahrsituation handelt.

2.3.1 Geografische Daten

Geografische Daten sind eine potenzielle Quelle für Kontexte. Im Folgenden sind geografische Daten, in Form von Punkten, Strecken oder Polygonen, relevant. Das einfachste Objekt ist dabei ein geografischer Punkt. Die Position eines Punktes auf der Erde wird üblicherweise durch ein geografisches Koordinatenpaar bestimmt. Die Koordinaten bestehen dabei aus der geografischen Breite und der geografischen Länge und beziehen sich auf die Oberfläche eines Ellipsoids, welches die Gestalt der Erdoberfläche näherungsweise beschreibt. Die geografische Breite ist vom Äquator ausgehend nach Norden und nach Süden jeweils mit $0°$ bis $90°$ angegeben. Die geografische Länge wird vom Nullmeridian jeweils nach Osten und nach Westen durch $0°$ bis $180°$ beschrieben. Der Nullmeridian verläuft dabei nach Konvention durch die Sternwarte von Greenwich. Zur Unterscheidung von Nord- und Süd- bzw. Ost- und West-Richtung erhalten südliche und westliche Breiten- bzw. Längenwerte ein negatives Vorzeichen. [34]

Um Strecken und Polygone zu repräsentieren, werden mehrere geografische Punkte in einer Liste zusammengefasst. Hierdurch lassen sich etwa Straßen oder Gebäude beschreiben. Neben dieser räumlichen Beschreibung der Daten werden zusätzliche Attribute zu den Punkten, Strecken und Polygonen gespeichert. Die Attribute schreiben den Objekten Eigenschaften zu. Als Beispiel ist es möglich, eine Linie als Straße zu attribuieren. Auch die Charakterisierung der Fahrbahnoberfläche oder die geltende Höchstgeschwindigkeit werden auf diese Weise den Objekten zugeschrieben.

2.4 Maschinelles Lernen

Tom M. Mitchell, einer der Pioniere des maschinellen Lernens, definiert maschinelles Lernen wie folgt:

> „A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.“ [35]

> „Ein Computerprogramm lernt durch Erfahrung E in Bezug auf eine Klasse von Aufgaben T und ein Leistungsmaß P, wenn sich seine Leistung bei Aufgaben in T, gemessen durch P, mit der Erfahrung E verbessert.“

Als Beispiel für eine Aufgabe führt er einen Algorithmus an, der mithilfe von Kameras als Sensoren auf einer vierspurigen Straße fahren soll. Als Leistungsmaß definiert er in diesem Fall die durchschnittlich zurückgelegte Strecke ohne Fehler, wobei die Fehler durch einen Menschen beurteilt werden sollen. Die Erfahrungen, aus denen der Algorithmus lernen soll, sind in diesem Fall aufgezeichnete Bildsequenzen mit Lenkbefehlen von einem menschlichen Fahrer [35].

Die Definition von Mitchell baut auf einer langen Historie in diesem Bereich auf. So hat Karl Steinbuch bereits im Jahr 1961 eine Abhandlung über eine Lernmatrix veröffentlicht [36]. Diese Erfindung legte die Basis für die automatische Zeichenerkennung und automatische Spracherkennung, aber auch die Dekodierung von gestörten Nachrichten.

Heute teilt man Algorithmen des maschinellen Lernens üblicherweise in drei Kategorien ein: das überwachte, das unüberwachte und das bestärkende Lernen. Beim überwachten Lernen trainiert man den Algorithmus mit Paaren aus Ein- und Ausgabedaten. Der Algorithmus lernt dabei den Zusammenhang zwischen der Eingabe und der Ausgabe. Dagegen kreiert ein unüberwacht lernender Algorithmus ein statistisches Modell der Eingabedaten. Auf der Grundlage dieses Modells werden dann Vorhersagen getroffen. Das bestärkende Lernen beschäftigt

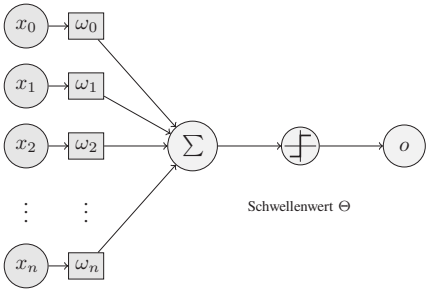


Abbildung 2.4: Einfachste Form eines Perzeptrons mit Eingaben x_i, trainierbaren Gewichten ω_i, Schwellenwert Θ und Ausgabe o (nach [39])

sich mit Agenten, die auf Basis von Versuch und Irrtum lernen. Dafür interagieren die Agenten mit ihrer Umgebung. Das bestärkende Lernen benötigt demnach eine interaktive Umgebung und nicht nur einen starren Datensatz [37].

2.4.1 Künstliche neuronale Netze

Ein bedeutendes Teilgebiet des maschinellen Lernens stellen künstliche neuronale Netze dar. Ihre grundlegende Funktionsweise orientiert sich an den Gehirnen höherer Organismen. Deren elementare Bausteine modellierte der Psychologe Frank Rosenblatt mit dem Perzeptron-Modell [38] (siehe Abbildung 2.4).

Ein Perzeptron bildet einen Eingabe-Vektor auf einen Ausgabe-Vektor ab. In der einfachsten Form besteht das Perzeptron aus einem einzelnen künstlichen Neuron. Dieses Neuron erhält n Werte x_i als Eingabe, welche jeweils mit trainierbaren Gewichten ω_i multipliziert und anschließend addiert werden. Überschreitet diese Summe einen Schwellenwert Θ, so ist die Ausgabe o des Neurons 1 und ansonsten 0:

$$o = \begin{cases} 1 & \text{wenn } \sum_{i=0}^{n} \omega_i x_i > \Theta \\ 0 & \text{sonst} \end{cases} . \tag{2.1}$$

Das Perzeptron wird überwacht trainiert, sodass beim Training für einen Eingabevektor die gewünschte Ausgabe bekannt ist und die Gewichte ω_i entsprechend

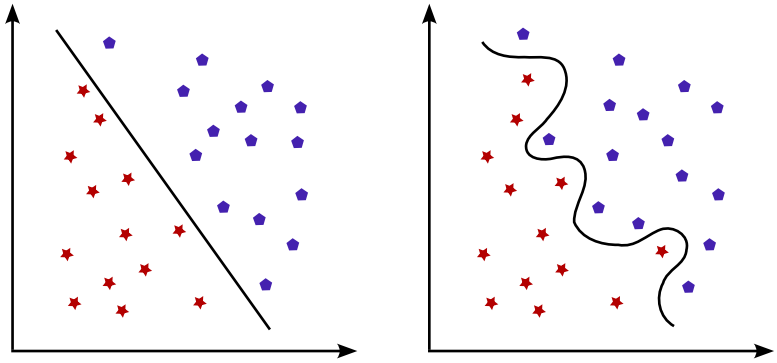


Abbildung 2.5: Jeweils zwei Mengen in $\mathbb{R}^2$: links linear separierbar, rechts nicht linear separierbar

aktualisiert werden. Durch die Anordnung mehrerer künstlicher Neuronen nebeneinander in einer Schicht ist als Ausgabe auch ein Vektor, statt eines Skalars möglich. Ein trainiertes Perzeptron kann auf diese Weise Klassifikationen von Vektoren vornehmen. Dabei werden auch Vektoren richtig klassifiziert, welche leicht von den Trainingsvektoren abweichen. Dieses Verhalten wird Generalisieren genannt [40].

Ein Nachteil des Perzeptrons mit nur einer Schicht ist die Tatsache, dass eine Klassifizierung von Vektoren nur dann möglich ist, wenn die Klassen linear separierbar (siehe Abbildung 2.5) sind. Zur Beseitigung dieser Einschränkung wird das Perzeptronkonzept verbessert. In einem ersten Schritt werden die Neuronen in mehreren Schichten angeordnet, wobei die Ausgabe einer Schicht als Eingabe für die nächste Schicht dient. Außerdem erhält jedes Neuron eine nicht lineare Aktivierungsfunktion f, welche den Aktivierungsgrad des Neurons bestimmt:

$$o = f\left(\sum_{i=0}^{n} \omega_i x_i\right). \tag{2.2}$$

Durch diese Modifikationen können auch nicht linear separierbare Klassen richtig klassifiziert werden [38].

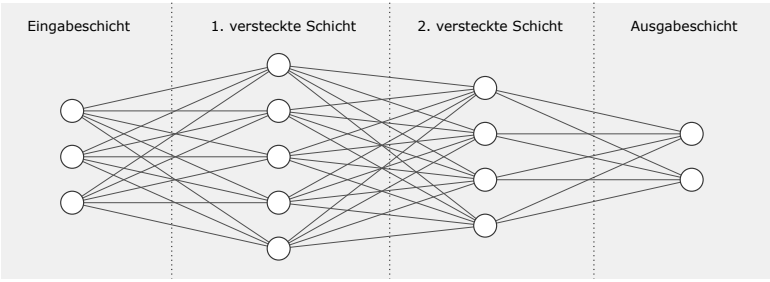


Abbildung 2.6: Künstliches neuronales Netzwerk mit drei Eingangsneuronen, zwei Ausgangsneuronen und zwei versteckten Schichten

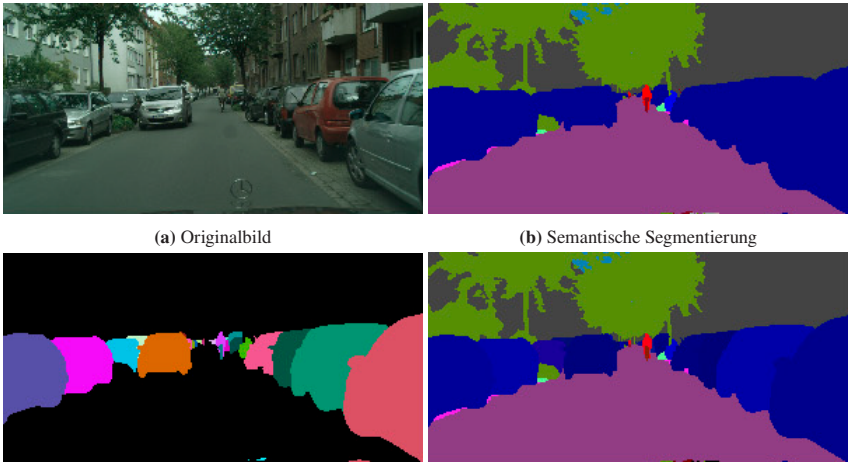
Das grundlegende Konzept der Anordnung von einzelnen künstlichen Neuronen in Schichten bildet die Basis für die Architekturen von künstlichen neuronalen Netzen (siehe Abbildung 2.6). Die Schichten, die weder zum Eingang noch zum Ausgang gehören, werden versteckte Schichten genannt. Eine konkrete Anzahl an versteckten Schichten, ab denen eine Architektur zu den tiefen künstlichen neuronalen Netzen[4] gezählt wird, existiert nicht [40].

Das Training, also die Anpassung der Gewichte des Netzes, ist Hauptgegenstand der Forschung an künstlichen neuronalen Netzen. Das grundlegende Vorgehen beruht darauf, Trainingsdaten durch das Netzwerk zu propagieren und das Ergebnis mit dem erwarteten Ergebnis abzugleichen. Der daraus resultierende Fehler wird zurück durch das Netzwerk propagiert. Auf Grundlage dieses Fehlers und der damit verbundenen Aktivierungsabweichungen werden die Gewichte der Neuronen optimiert [41].

2.5 Panoptische Segmentierung

Künstliche neuronale Netze zur semantischen Segmentierung [43] weisen jedem Pixel innerhalb eines Bildes eine Klasse zu (siehe Abbildung 2.7b). Das Resultat

[4] Deep Learning



(a) Originalbild **(b)** Semantische Segmentierung

(c) Instanzsegmentierung **(d)** Panoptische Segmentierung

Abbildung 2.7: Vergleich von semantischer Segmentierung, Instanzsegmentierung und panoptischer Segmentierung [42]

einer semantischen Segmentierung ist demnach ein Bild mit identischer Auflösung wie das Eingangsbild. Statt der Ursprungsfarben ist in der Ausgabe für jedes Pixel kodiert, welcher Klasse das Pixel im Eingangsbild entspricht. Die möglichen Klassen werden beim Training festgelegt und durch manuelles Labeling zugeordnet. Beispiele für Klassen sind unter anderem „Himmel“ und „Kraftfahrzeuge“. [43]

Im Unterschied zur semantischen Segmentierung unterscheidet die Instanzsegmentierung [44] (siehe Abbildung 2.7c) die Instanzen von zählbaren Objektklassen in einem Bild. Sind in einem Bild zum Beispiel drei Kraftfahrzeuge sichtbar, so erhalten diese bei der Instanzsegmentierung in der Ausgabe unterschiedliche Kodierungen. Durch die Farbkodierung wird weiterhin spezifiziert, dass ein Pixel der Klasse „Kraftfahrzeuge“ zugeordnet ist, aber gleichzeitig sind die einzelnen Kraftfahrzeuge unterscheidbar. [44]

Eine Segmentierung, die sowohl Instanzen derselben zählbaren Objektklasse unterscheidet als auch nicht zählbare Objekte detektiert, wird panoptische Segmentierung [45] (siehe Abbildung 2.7d) genannt. Nicht zählbare Objektklassen, wie

der Himmel, erhalten bei der panoptischen Segmentierung eine Kodierung, die keine Instanzen unterscheidet. Im Gegensatz dazu sind zählbare Objektklassen mithilfe ihrer Kodierung, wie bei der Instanzsegmentierung, hinsichtlich ihrer Instanzen unterscheidbar. [45]

2.6 Multimodale Vektorrepräsentation mit CLIP

Im Folgenden werden Repräsentationen von Bildern und Texten benötigt, die miteinander vergleichbar sind. So soll etwa die Ähnlichkeit zwischen dem Text „Vegetation“ und einem Bild von einem Baum messbar sein. Die Methode Contrastive Language Image Pre-training (CLIP) [46] überführt zur Lösung dieses Problems sowohl Texte als auch Bilder in eine Vektorrepräsentation. Das Vorgehen von CLIP beginnt mit dem Zusammenstellen eines Datensatzes aus Bild-Text-Paaren. In der ursprünglichen Veröffentlichung von CLIP [46] wurde mit 400 Millionen Bild-Text-Paaren trainiert. Zum Sammeln der Daten wurde das Internet nach HTML-Bilder-Tags durchsucht. Die Bilder wurden heruntergeladen und der Alternativtext der Bilder, welcher zur Sicherstellung der Barrierefreiheit für Menschen mit Sehbehinderung dient, wurde als textuelle Beschreibung verwendet. In einem Filterschritt wurden diejenigen Paare verworfen, deren Beschreibung keinen Teil einer vordefinierten Liste von $500\,000$ Begriffen enthielt. Auf diese Weise wurde sichergestellt, dass die gesammelten Texte natürliche Sprache enthalten.

Mit diesem Datensatz aus Bild-Text-Paaren (I_i, T_i) werden bei der CLIP-Methode anschließend zwei Encoder trainiert. Ein Encoder überführt Texte $\mathcal{E}_{\text{Text}}$ und der andere Bilder $\mathcal{E}_{\text{Bild}}$ in Vektoren:

$$\mathcal{E}_{\text{Bild}}(I_i) \in \mathbb{R}^n, \tag{2.3}$$

$$\mathcal{E}_{\text{Text}}(T_i) \in \mathbb{R}^n. \tag{2.4}$$

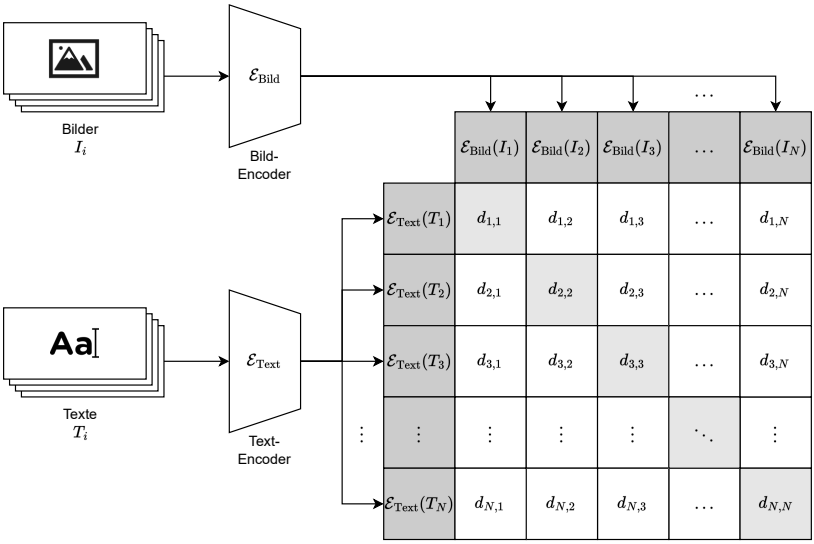


Abbildung 2.8: Training von CLIP mit Contrastive Learning; Die Winkelabstände der zusammengehörigen Bild-Text-Paare (Diagonale) werden beim Training maximiert und die restlichen Winkelabstände minimiert (nach [46])

Die im Folgenden verwendete Implementierung von CLIP erzeugt $n = 512$ dimensionale Vektoren. Die Encoder werden dabei so trainiert, dass die resultierenden Vektoren für ähnliche Bilder $\mathcal{E}_{\text{Bild}}(I_i)$ und Texte $\mathcal{E}_{\text{Text}}(T_i)$ kleine Winkelabstände $d_{i,j}$ zueinander haben. Das dabei angewendete Verfahren heißt Contrastive Learning [47]. Ein Trainingsschritt enthält dabei mehrere (N) Bild-Text-Paare. Beim Training werden die Winkelabstände zwischen allen Vektorrepräsentationen der Bilder und Texte eines Trainingsschrittes berechnet. Die Gewichte der beiden Encoder werden aktualisiert, sodass die Winkelabstände der zueinander gehörigen Bild-Text-Paare minimiert und die aller anderen maximiert werden (siehe Abbildung 2.8).

Das Ergebnis des Trainings von CLIP sind Encoder für Bilder und Texte, die eine multimodale Vektorrepräsentation erlauben. Semantische Ähnlichkeiten sind über Winkelabstände im Vektorraum kodiert. Dadurch, dass CLIP mit Daten aus dem Internet trainiert wurde, ist nicht bekannt, in welchen Domänen CLIP eingesetzt

werden kann [46]. Die Eignung für die Automotive-Domäne wird im Rahmen der Evaluation des vorgestellten Verfahrens untersucht.

2.6.1 Relationale und Vektordatenbanken

Für das im Weiteren vorgestellte Konzept ist es notwendig, aus einer Menge von Vektoren die ähnlichsten Vektoren zu identifizieren. Um diese Suche zu optimieren, muss ein geeigneter Datenbanktyp für die Verwaltung gewählt werden. Ein verbreiteter Datenbanktyp ist die relationale Datenbank [48]. Die Daten sind in relationalen Datenbanken in Tabellen organisiert, wobei jede Zeile einer Tabelle einem Eintrag in der Datenbank entspricht. Die Einträge in der Datenbank können aufeinander verweisen und dadurch in Relation gesetzt werden. Einer der Vorteile von relationalen Datenbanken ist, dass sie Konsistenzgarantien für die Einträge in der Datenbank zusichern können. Die Interaktion mit relationalen Datenbanken basiert in der Regel auf der strukturierten Abfragesprache SQL [49, 50]. Diese ermöglicht die effiziente Kombination mehrerer Suchkriterien bei der Datenabfrage. [51]

Für das effiziente Speichern und Vergleichen von Vektoren sind relationale Datenbanken nicht ausgelegt. Im Gegensatz dazu sind Vektordatenbanken für die Suche nach ähnlichen Vektoren optimiert [52, 53]. Durch diese Optimierung wird die Suche auch bei mehreren Millionen Einträgen noch innerhalb von Sekunden ausgeführt [54]. So führte der Vektordatenbank-Anbieter Milvus [53] beispielsweise auf einem Server mit 8 virtuellen Computer-Prozessoren und $32\,\mathrm{GB}$ Hauptspeicher ein Experiment auf einem Datensatz mit $9\,990\,000$ Vektoren durch. Die durchsuchten Vektoren hatten 96 Dimensionen und als Distanzmaß wurde das innere Produkt verwendet. Milvus berichtet für die Suche eine mittlere Latenzzeit von unter $5\,\mathrm{ms}$. [54] Daraus lässt sich schließen, dass die Ähnlichkeitssuche von Vektoren in Vektordatenbanken schnell genug ist, um sie auch bei mehreren Millionen Einträgen in einer interaktiven Suche mit einem Nutzer einzusetzen.

3 Stand der Wissenschaft und Technik zur Bereitstellung von Bildern für Perzeptionstests

3.1 Einordnung von Perzeptionstests: Umfeld und Randbedingungen

Klassischerweise basiert die Entwicklung von automatisierten Fahrsystemen in der Automobilindustrie auf dem V-Modell [55]. Beim V-Modell handelt es sich ursprünglich um ein Vorgehensmodell zur Softwareentwicklung, welches initial 1992 durch das Bundesministerium des Innern für alle Bundesbehörden empfohlen wurde. Einer der Hauptbestandteile in diesem Modell ist der an das Wasserfallmodell [56] angelehnte Ablauf der Systemerstellung. Dieser Systemerstellungsablauf ist V-förmig und gliedert sich in zwei Äste (siehe Abbildung 3.1). Angefangen bei der Erhebung der Anforderungen an das Gesamtsystem, über dessen Entwurf, wird die Entwicklung hin zum Komponentenentwurf und zur Implementierung im linken Ast immer detaillierter. Hierdurch wird die Systemkomplexität in kleinere Elemente zerlegt. Dieses Vorgehen ist mit dem Teile-und-herrsche-Paradigma vergleichbar. Anschließend erfolgt im rechten Ast die schrittweise Integration der einzelnen Komponenten. Die Integration ist begleitet von Tests, die der Verifikation dienen. [55, 57]

> **Definition 3.1 (Verifikation)** *Bei der Verifikation wird nachgewiesen, dass das System die zuvor aufgestellten Anforderungen erfüllt. [58]*

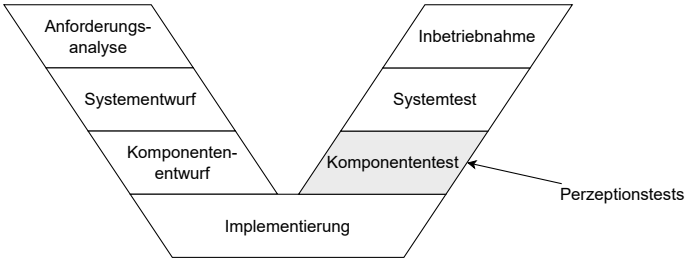


Abbildung 3.1: V-Modell: Ablauf der Systemerstellung mit Verortung der Perzeptionstests (nach [57] und [13])

Die horizontalen Ebenen innerhalb des V-Modells beziehen sich auf die jeweils gleichen Abstraktionsgrade. Am Ende des V-Modells wird in einem letzten Schritt die Validierung durchgeführt.

> **Definition 3.2 (Validierung)** *Bei der Validierung wird überprüft, ob das System für den geplanten Verwendungszweck geeignet ist. [58]*

Im Folgenden wird der Fokus auf die Perzeption mittels Kamerabildern innerhalb dieses Entwicklungsprozesses gelegt. Die Perzeption ist eine Komponente des Gesamtsystems, daher muss auch die Perzeption verifiziert und folglich getestet werden.

Um die Anforderungen an Kamera-Perzeptions-Tests nachzuvollziehen, ist es notwendig, sich die Eingangsdaten der bildbasierten Perzeption zu vergegenwärtigen. Eine für Automotive-Anwendungen übliche Kamera kann rechnerisch mehr als $10^{16826546}$ unterschiedliche Bilder ausgeben (vgl. Abschnitt 1.2). Zur Verarbeitung dieser Eingabedaten werden bei automatisierten Fahrsystemen zunehmend maschinelle Lernverfahren eingesetzt. Neben der verwendeten Algorithmik hängt das resultierende System damit auch von den verwendeten Trainingsdaten ab. Anforderungen richten sich daher nicht mehr so wie bisher nur an das System, sondern zusätzlich an die zur Entwicklung verwendeten Daten [59]. Diesem Umstand begegnet der Data-Driven Engineering Prozess (DDE) [59]. Ähnlich wie

von Reisgys et al. [60] vorgeschlagen, wird bei diesem Prozess der streng lineare Charakter des V-Modells aufgebrochen und um iterative Schleifen ergänzt. Grundlage für dieses Vorgehen ist die Operational Design Domain (ODD).

> **Definition 3.3 (Operational Design Domain)** *Die Operational Design Domain beschreibt die Betriebsbedingungen, unter denen ein bestimmtes Fahrautomatisierungssystem oder ein bestimmtes Merkmal davon funktionieren soll, einschließlich, aber nicht beschränkt auf umgebungsbedingte, geografische und tageszeitliche Einschränkungen und/oder das erforderliche Vorhandensein oder Nichtvorhandensein bestimmter Verkehrs- oder Fahrbahnmerkmale. [61]*

Aus der ODD leiten sich anschließend die Datenanforderungen ab.

> **Definition 3.4 (Datenanforderungen)** *Die Datenanforderungen beschreiben die Daten, mit welchen das automatisierte Fahrsystem funktionieren muss, auf semantischer Ebene. (in Anlehnung an [59])*

Das Ziel ist eine Sammlung von Datenanforderungen, die gemeinsam eine Vollständigkeit der Daten, wie sie das Datenqualitätsmodell des internationalen Standards ISO 25012 [62] fordert, sicherstellen. Ein Beispiel für eine Datenanforderung ist die textuelle Beschreibung über das Vorhandensein einer Objektklasse mit bestimmten Eigenschaften in den Daten [59]. Konkret könnte eine Datenanforderung beispielsweise verlangen, dass in den Daten ein Fahrzeug mit einer bestimmten Farbe vorhanden ist. Es ist aber zum Beispiel auch möglich, dass eine Datenanforderung eine regennasse Straße fordert. Darüber hinaus sind Datenanforderungen auch in der Lage, Kombinationen einzelner Objekte und Sachverhalte zu beschreiben. Ein Beispiel für eine solche Datenanforderung ist ein Fahrzeug mit einer bestimmten Farbe auf einer regennassen Straße. Datenanforderungen ermöglichen auch die Charakterisierung von abstrakteren Konzepten. Als Beispiel sind hier überbelichtete Bilder zu nennen.

Die Verifikation der Perzeptionskomponente erfolgt dann basierend auf den Datenanforderungen. [59] Beim Testen der Datenanforderungen kommt es vor, dass

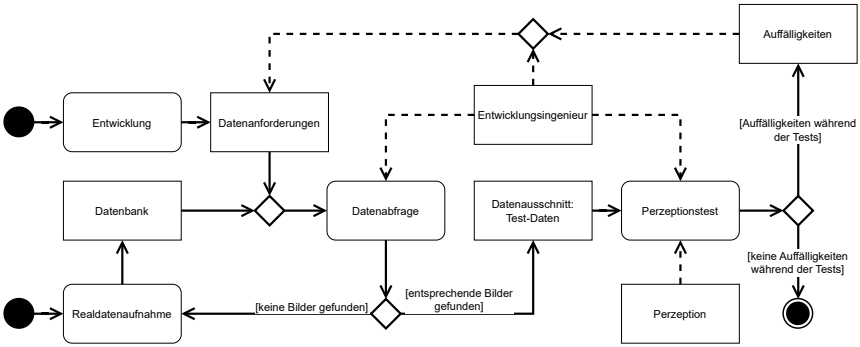


Abbildung 3.2: Umfeld von Perzeptionstests mit der sich iterativ vergrößernden Abdeckung der Tests

Auffälligkeiten auftreten, die nicht durch Datenanforderungen abgedeckt sind, für den geplanten Verwendungszweck des Systems aber durch Datenanforderungen erfasst sein sollten. In diesem Fall müssen Datenanforderungen angepasst oder ergänzt werden. Dieser Vorgang ist dann Teil der Validierung. Das Vorgehen erhält dadurch einen iterativen Charakter, bei dem sich die Abdeckung der Datenanforderungen fortwährend steigert. [59] Diesem iterativen Anteil muss durch eine interaktive Nutzbarkeit begegnet werden (siehe Abbildung 3.2).

> **Definition 3.5 (Interaktive Nutzbarkeit)** *Der Benutzer hat die Möglichkeit, in Echtzeit mit den Algorithmen, Modellen und Daten zu interagieren und alle relevanten Parameter zu manipulieren und zu kontrollieren. [63]*

Das primäre Ziel ist im Folgenden jedoch, die den Datenanforderungen entsprechenden Bilder für Perzeptionstests bereitzustellen und auf diese Weise bei der Verifikation der Perzeptionskomponente zu unterstützen.

3.2 Untersuchung möglicher Bildquellen

Generell kommen mehrere Bildquellen für die Verifikation der Perzeption infrage. Sie unterscheiden sich hinsichtlich ihrer Durchsuchbarkeit, aber auch anhand ihres

Realitätsgrades. Es werden drei grundsätzliche Datenquellen für die Entwicklung von automatisierten Fahrsystemen unterschieden: die Bildsynthese mit Methoden der Computergrafik, die Bildsynthese mit künstlichen neuronalen Netzen und die Realdatenaufnahme.

Auf der einen Seite stehen die Simulation und Bildsynthese mit Methoden der Computergrafik. In diesem Fall werden das Fahrzeug mit der Sensorik und die Umgebung in einem Modell abgebildet und die Prozesse sowie die Messung mit dem Kamerasensor simuliert [64]. Das erlaubt die freie Variation der generierten Szenen. Hierdurch können bereits bei der Generation die Modellzustände als Metadaten gespeichert werden, um eine Durchsuchbarkeit anhand der Datenanforderungen zu ermöglichen. Oder es werden den Datenanforderungen entsprechende Bilder ad hoc erzeugt. Dona et al. [65] präsentieren in einem Überblick, welche Möglichkeiten es für den Einsatz von Simulationen beim virtuellen Testen gibt. Jedoch basieren die resultierenden Bilder bei der Simulation vollständig auf den Modellen für die Umgebung und die Sensorik. Somit ist auch die Realitätstreue abhängig von den zugrunde liegenden Modellen. Modelle stellen jedoch nur ein vereinfachtes Abbild der Realität dar [66], weshalb sich die gerenderten Bilder von real aufgenommenen Bildern unterscheiden.

Einen ähnlichen Ansatz verfolgt die Bildsynthese mit künstlichen neuronalen Netzen. Hier werden künstliche neuronale Netze mit real aufgenommenen Bildern trainiert. Als Techniken kommen vorwiegend Transformer-Architekturen [67] oder Generative Adversarial Networks [68] zum Einsatz. Diese künstlichen neuronalen Netze sind anschließend in der Lage, Bilder zu generieren, die aussehen, als würden sie aus der gleichen Grundgesamtheit stammen, wie die real aufgenommenen Bilder [69–71]. Zusätzlich zur Generierung existieren auch Methoden zur Veränderung existierender Bilder mit künstlichen neuronalen Netzen [72–75]. Auch in diesem Fall können Parameter, mit denen ein Bild generiert oder verändert wird, gespeichert werden, um ein späteres Auffinden zu ermöglichen. Ebenfalls ist eine gezielte Generierung oder Veränderung der Bilder entsprechend der Datenanforderungen umsetzbar. Es ist nicht notwendig, manuell ein explizites Modell der Umgebung aufzubauen und auch die Notwendigkeit für ein Sensormodell entfällt.

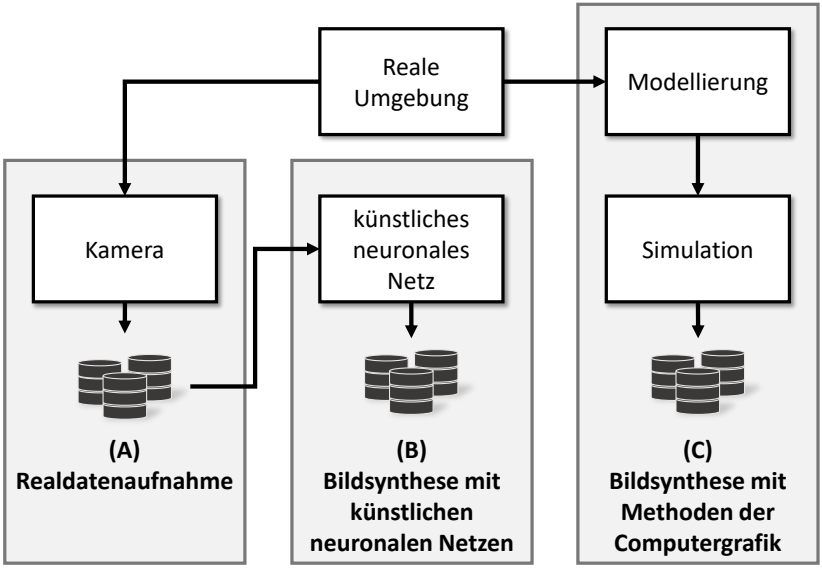


Abbildung 3.3: Vergleich von Datenquellen hinsichtlich ihrer Beziehung zur realen Umgebung

Sensoreffekte werden von künstlichen neuronalen Netzwerken implizit übernommen, da diese bereits in den Trainingsdaten enthalten sind. Allerdings entsteht auch bei der Bildsynthese mit künstlichen neuronalen Netzen indirekt während des Trainings mit den real aufgenommenen Bildern ein Modell der Umgebung.

Durch die Modellbildung existiert eine Lücke zwischen den real aufgenommenen (A) und den synthetisch erzeugten Bildern (B, C) [76] (siehe Abbildung 3.3). Dabei ist es unerheblich, ob das Modell, wie bei der Bildsynthese mit Methoden der Computergrafik (C), händisch oder implizit, beim Training eines künstlichen neuronalen Netzes (B), erzeugt wird. Ob diese Lücke für die Perzeption in allen Fällen klein genug ist, um dennoch verlässliche Testergebnisse bei der Verifikation zu erhalten, ist gegenwärtig bisher nicht gesichert. Im Folgenden wird daher ausschließlich der Umgang mit real aufgenommenen Bildern betrachtet. Diese haben im Vergleich zu den synthetisch erzeugten Bildern jedoch den Nachteil, dass für sie nicht automatisch Annotationen existieren. Das Auffinden von real

aufgenommenen Bildern anhand der Datenanforderungen für Verifikationstests ist daher noch eine offene Fragestellung, die mit den folgenden Ausführungen geschlossen wird.

3.3 Stand der Wissenschaft und Technik bei der Bildsuche

Die Bildsuche (engl. Image Retrieval) beschäftigt sich mit dem Auffinden von bestimmten Bildern in Datensätzen [77]. Das Ziel ist es, einen Nutzer in die Lage zu versetzen, Bilder aufzufinden, ohne dass er einen Datensatz manuell durchsuchen muss. Die hierzu existierenden Methoden werden in der Literatur verschiedenen Kategorien zugeordnet. Diese sind jedoch nicht immer trennscharf definiert. Es lassen sich folgende Kategorien identifizieren [77–80]:

- textbasierte Methoden,
- kontextbasierte Methoden,
- inhaltsbasierte Methoden und
- hybride Methoden.

3.3.1 Textbasierte Methoden

Im einfachsten Fall existieren textuelle Annotationen zu den Bildern, anhand derer ein Datensatz mit Methoden der klassischen Textsuche durchsucht wird [77]. Textuelle Annotationen in diesem Sinne sind von Menschen manuell verfasste Bildbeschreibungen für jedes Bild eines Datensatzes. Es ist jedoch zur Automatisierung der Annotation möglich, auf Methoden zur automatischen Untertitelung von Bildern zurückzugreifen [81, 82]. Die textbasierte Suche ist auf die Aspekte eines Bildes beschränkt, welche bei der Annotation erfasst wurden. Iyer et al. [83] nutzen textuelle Annotationen zur Suche mit einem Referenzbild. Der Nutzer gibt

ein Referenzbild vor, für welches automatisiert eine textuelle Beschreibung generiert wird. Mit dieser Beschreibung suchen die Autoren anschließend in einer Bilddatenbank nach ähnlichen textuellen Beschreibungen.

3.3.2 Kontextbasierte Methoden

Eine Alternative zur Suche nach textuellen Annotationen ist die Suche nach Kontextdaten. Die Kontextdaten stammen dabei von zusätzlichen Sensoren, wie Temperatursensoren, oder aus anderen Quellen, wie zum Beispiel geografische Daten [PR7]. Im Vergleich zu den textuellen Annotationen werden Kontextdaten strukturiert abgelegt, um die Maschinenlesbarkeit zu gewährleisten. Naito et al. [84] nutzen unter anderem die Geschwindigkeit, die geografische Position, den Lenkwinkel und die Beschleunigungswerte des Aufnahmefahrzeugs zum Auffinden von aufgezeichneten Fahrzeugdaten. Ebenso reichern Klitzke et al. [85] Fahrzeugdaten mit Kontextdaten an, um diese anschließend zu durchsuchen. Sie setzen dabei beispielsweise auf Wetterdaten, die Straßenklasse und die Anzahl der Fahrspuren. Die Relevanz von Kontextdaten bei Automotive-Bilddatensätzen wurde von Heidecker et al. [86] untersucht. Hierzu verglichen sie die Leistungsfähigkeit von künstlichen neuronalen Netzwerken für die Objektdetektion in Abhängigkeit von Kontextdaten, wie zum Beispiel der Tageszeit, des Bewölkungsgrades und der Beleuchtungssituation. Elspas et al. [87] kodieren Kontextdaten als Zeichenfolgen und durchsuchen Fahrzeugdaten anschließend mit regulären Ausdrücken.

3.3.3 Inhaltsbasierte Methoden

Der Fokus der inhaltsbasierten Suchmethoden liegt auf den Bilddaten.

> **Definition 3.6 (Bilddaten)** *Bilddaten bezeichnen die rohen und uninterpretierten Pixelwerte (vgl. Abschnitt 2.2) eines Bildes.*

Die inhaltsbasierten Suchmethoden extrahieren aus den Bilddaten Merkmale. Hierbei werden Merkmale auf niedriger und höherer Ebene unterschieden [77].

Die Merkmale niedriger Ebene beziehen sich auf Farben, Texturen, einfache Formen und Ähnliches. Deselaers et al. [88] untersuchen unterschiedliche Merkmale der niedrigeren Ebene für die Suche nach Bildern. Korn et al. [89] nutzen eine Mustersuche, um in der medizinischen Anwendung ähnliche Tumoren zu finden und diese dann mit Kontextdaten zu korrelieren. Dagegen erfassen höhere Merkmale abstraktere semantische Konzepte [90]. Das Ziel dabei ist es, die semantische Lücke zwischen einer textuellen Beschreibung und dem semantischen Inhalt der Bilder zu schließen [91]. Inhaltliche Suchmethoden, die auf Merkmalen höherer Ebene basieren, erlauben demnach die Bildsuche hinsichtlich der Semantik, wohingegen der Fokus bei Merkmalen der niedrigeren Ebene auf dem Erscheinungsbild liegt.

Eine spezielle Form der Inhaltsanalyse ist die Anomaliedetektion. Die Anomaliedetektion bezieht sich ebenfalls auf inhaltsbasierte Merkmale der niedrigeren oder höheren Ebene. Anomaliedetektionen können demnach auch semantische Inhalte erfassen. Hier wird jedoch nicht gezielt nach bestimmten Ausprägungen gesucht, sondern es werden Ausreißer in einem Datensatz identifiziert [92, 93]. Zum Beispiel fokussiert sich Shoeb et al. [94] auf unbekannte Hindernisse auf der Straße. Die Hindernisse werden als unbekannt erkannt und von einer panoptischen Segmentierung (vgl. Abschnitt 2.5) maskiert. Anschließend ist die Suche nach Hindernissen mit natürlicher Sprache möglich. Der Fokus liegt dabei auf Objekten, die Ausreißer im Datensatz darstellen. Zhang et al. [95] geben einen Überblick über Methoden, die sich speziell auf das Finden von kritischen Situationen spezialisiert haben. Die Anomaliedetektion bzw. die Detektion von kritischen Szenarien erlaubt keine gezielte, allgemeine Suche.

3.3.4 Hybride Methoden

Zusätzlich zu den vorgestellten Suchverfahren existieren auch hybride Ansätze, die verschiedene der vorgestellten Methoden kombinieren. Neben der Suche im Anschluss an das Aufzeichnen und Abspeichern der Bilder ist es auch möglich, die

Daten schon während der Fahrt durch einen Menschen annotieren zu lassen. Hierbei wird beispielsweise die aktuelle Wettersituation erfasst oder herausfordernde Fahrsituationen werden in den Daten markiert.

3.3.5 Fokus: Generische semantische Suche

Die Analyse der bereits existierenden Methodenkategorien führt zu folgenden Entscheidungen in Bezug auf das zu entwickelnde Datenmanagementsystem:

- Das Datenmanagementsystem muss, im Gegensatz zur Anomalieerkennung, eine gezielte Suche ermöglichen.
- Werden Bilder anhand von Bilddaten (inhaltsbasierte Methoden) durchsucht, ist es möglich, dass die Bilder, bei denen die Interpretation der Bilddaten eine Herausforderung darstellt, nicht gefunden werden. Diese herausfordernden Bilder sind jedoch von Interesse für die Entwicklung von Fahrsystemen. Bei entsprechend verfügbarer Datengrundlage muss daher die Suche unabhängig von Bilddaten durchgeführt werden.
- Bei Bildeigenschaften, bei denen eine Suche ohne die Einbeziehung von Bilddaten nicht möglich ist, muss eine generische semantische Suche stattfinden. Das entspricht einer inhaltsbasierten Methode, welche höhere Merkmale nutzt.

Basierend auf diesen Erkenntnissen erfolgt eine Untersuchung der Arbeiten, die sich mit der generischen semantischen Suche beschäftigen. Die Methode von Nguyen et al. [96] extrahiert für die Suche Objekte aus Bildern einer Datenbank. Die Suche erfolgt anhand eines Anfragebildes. Eine generische Suche ohne entsprechendes Anfragebild ist nicht möglich. Dagegen bietet Merantix [97] eine kommerzielle Softwarelösung an, die die semantische Suche mit natürlicher Sprache, basierend auf CLIP-Vektorrepräsentationen (vgl. Abschnitt 2.6), ermöglicht. Die Suche bezieht sich dabei immer auf das gesamte Bild. Eine spezifische Suche nach Eigenschaften einzelner Objekte innerhalb eines Bildes ist nicht

möglich. Hess et al. [98] übertragen die Suche mit CLIP-Vektorrepräsentationen von Bildern auf Lidar-Punktwolken. Sai et al. [99] untersuchen die Erzeugung von Kontextdaten im Bereich von automatisierten Fahrsystemen mittels CLIP-Vektorrepräsentationen. Sie analysieren die Eignung explizit auch auf Datensätzen, die pro Bild nur ein einzelnes Objekt beinhalten. Stage et al. [100] nutzen die CLIP-Vektorrepräsentationen zur Verkleinerung eines Bilddatensatzes, mit dem Ziel, ähnliche Bilder zu löschen. Zhou et al. [101] verbessern die Detektionsperformance von CLIP durch das Hinzunehmen einiger gelabelter Beispieldaten. Für diese Verbesserung benötigen sie annotierte Beispiele der zu suchenden Klasse. Li et al. [102] und Luddecke et al. [103] segmentieren Bilder anhand einer textuellen Eingabe. Für ein Eingabebild können mit natürlicher Sprache Objekte spezifiziert werden, die in diesem Bild lokalisiert und segmentiert werden. Aufbauend auf diesen Methoden lässt sich wahrscheinlich auch eine Suchfunktion für einen gesamten Datensatz auf Objektebene konstruieren. Das wäre auch mit BLIP-2 [104] möglich. Bei BLIP-2 beantwortet ein Large Language Model textuell formulierte Fragen, die zu einem Eingabebild gestellt werden. Allerdings müsste bei allen drei Methoden für jede Suchanfrage ein künstliches neuronales Netz auf jedem Bild des Datensatzes ausgeführt werden. Dieser Umstand wirkt sich negativ auf die Skalierbarkeit und damit auf die interaktive Nutzung als Suchfunktion aus.

3.4 Ableitungen aus dem Stand der Wissenschaft und Technik

Die Herausforderung bei der Suche nach real aufgenommenen Bildern ist die semantische Lücke. Diese Lücke bezieht sich auf die semantisch beschränkte Ausdrucksfähigkeit bei einer Suchanfrage. Einige Arbeiten verfolgen das Ziel, diese Lücke zu schließen und es zu ermöglichen, eine Bilddatenbank auf semantischer Ebene zu durchsuchen. Ein Beispiel ist die Suche nach Bildern mit verschneiter Fahrbahn. Die klassischen Methoden benötigen hierzu jedoch zusätzliche Kontexte zu den Bildern. Im Beispiel wären das die zugehörigen Wetterdaten. Die

Suche anhand von Kontextdaten hat den Vorteil, dass die Ergebnisse nachvollziehbar und unabhängig von Bilddaten sind. Wenn auf eine manuelle Annotation der Daten verzichtet werden soll, ist eine Voraussetzung hierfür, dass Quellen oder Methoden zur Erzeugung der Kontexte existieren. Außerdem sind initiale Kontexte notwendig, um die Kontextdaten abzufragen und zu berechnen. Im Beispiel der verschneiten Fahrbahn sind das etwa die Position des Fahrzeugs und die Uhrzeit der Aufnahme. Das bedeutet, dass bereits bei der Datenaufnahme ein initialer Satz an Kontexten aufgezeichnet werden muss. Dabei bietet es sich an, sich auf Kontextdaten zu beschränken, die üblicherweise ohnehin im Fahrzeug zur Verfügung stehen. Beispiele hierfür sind Informationen über die Uhrzeit, die Fahrzeugposition oder die Fahrtrichtung [7].

Neuere Verfahren bieten die Möglichkeit, die semantische Lücke bei der Suche ohne externe Datenquellen zu schließen. Dabei kommen künstliche neuronale Netze zum Einsatz. Diese erkennen dann zum Beispiel den Schnee auf der Fahrbahn direkt in den Bilddaten. Entscheidend ist dabei, dass diese Methoden ohne händische Annotationen und weitere Kontexte auskommen. Außerdem ist die Flexibilität bei der Suche auf semantischer Ebene ein Vorteil. Allerdings ist die Nachvollziehbarkeit des Zustandekommens der Ergebnisse bei der Suche mit künstlichen neuronalen Netzen eingeschränkt.

Eine Forschungslücke, der sich im Folgenden gewidmet wird, ist die semantische Suche nach Bildern mit natürlicher Sprache auf Objektebene für die Entwicklung von automatisierten Fahrsystemen. Mit Objekten sind an dieser Stelle nicht nur die Objekte aus Pegasus-Ebene 4 [16] (vgl. Unterabschnitt 2.1.3) gemeint, sondern auch die Beschaffenheit der Objekte in den Pegasus-Ebenen 1 bis 3.

Im aktuellen Stand der Technik und Wissenschaft werden bisher nur eingeschränkt Methoden für die Suche auf verschiedenen Pegasus-Ebenen kombiniert. Die im Weiteren vorgeschlagene Kombination von Suchmethoden führt zum Schließen der semantischen Lücke auf mehreren Pegasus-Ebenen. Das ermöglicht eine durchgängige Nutzbarkeit der kombinierten Methoden im Rahmen der Entwicklung von automatisierten Fahrsystemen.

Tabelle 3.1: Übersicht über verwandte Arbeiten und ihre Eigenschaften; Für Arbeiten aus dem Automotive-Kontext ist die Durchsuchbarkeit der Pegasus-Ebenen [16] angegeben (vgl. Unterabschnitt 2.1.3)

	Automotive				Nicht Automotive				
	Klitzke et al. [85]	**Elspas et al. [87]**	**Naito et al. [84]**	**Merantix [97]**	**Nguyen et al. [96]**	**Iyer et al. [83]**	**Li et al. [102]**	**Luddecke et al. [103]**	**Li et al. [104]**
Durchsuchbarkeit der Pegasus-Ebenen	❍	❍	❍	❍	Nicht Automotive				
Interaktive Nutzbarkeit hinsichtlich Laufzeit	✔	✔	✔	✔	✔	✔	✘	✘	✘
Keine manuelle Annotation notwendig	✔	✔	❍	✔	✔	✔	✔	✔	✔
Keine initialen Kontexte notwendig	✘	✘	✘	✔	✔	✔	✔	✔	✔
Unabhängig von Bilddaten	✔	✔	✔	✘	✘	✘	✘	✘	✘
Suche mit natürlicher Sprache	✘	✘	✘	✔	✘	✘	✔	✔	✔

Legende: ✔ gegeben, ❍ eingeschränkt, ✘ nicht gegeben

In den Fällen, in denen es ohne händische Annotationen möglich ist, wird im folgenden Konzept die Suche mit klassischen Methoden, basierend auf Kontexten und daher ohne Einbeziehung von Bilddaten, durchgeführt. Ist das nicht möglich, wird die semantische Lücke bei der Suche mittels künstlicher neuronaler Netze geschlossen. Zur Gewährleistung maximaler Ausdrucksstärke hinsichtlich der Semantik unterstützt das Datenmanagementsystem die Suche mit natürlicher Sprache. Hierdurch werden auch nur aufwendig parametrierbare Bildeigenschaften flexibel auffindbar.

Die interaktive Nutzbarkeit des Datenmanagementsystems während der Entwicklung von automatisierten Fahrsystemen ist ein zentrales Merkmal des Konzepts. Dadurch wird dem iterativen Charakter des Entwicklungsprozesses Rechnung getragen. Bei der Analyse des Stands der Wissenschaft und Technik hat sich gezeigt,

dass nicht alle Verfahren Laufzeiten für eine interaktive Nutzung gewährleisten (siehe Tabelle 3.1). Daher werden diese Ansätze nicht weiterverfolgt und die Ausführung von künstlichen neuronalen Netzen auf jedem Bild des Datensatzes erfolgt nur während vorbereitender Schritte für die Suche.

3.4.1 Anforderungen an ein Datenmanagementsystem

Für die sinnvolle, entwicklungsbegleitende Verwendung des Datenmanagementsystems muss dieses entsprechenden Anforderungen genügen. Diese Anforderungen ergeben sich aus der Analyse des gegenwärtigen Stands der Technik und Wissenschaft (siehe Tabelle 3.1) und aus der Betrachtung des aktuellen Vorgehens bei der Entwicklung von automatisierten Fahrsystemen.

Anforderung A1 Real aufgenommene Bilder müssen semantisch auf den Pegasus-Ebenen [17] durchsuchbar sein.

Anforderung A2 Das Datenmanagementsystem muss sich hinsichtlich der Laufzeiten interaktiv in den Entwicklungsprozess einfügen.

Anforderung A3 Das Datenmanagementsystem muss ohne manuelle Annotation der Daten funktionieren.

Anforderung A4 Als initiale Kontexte dürfen nur automatisch aufgezeichnete Informationen (Zeitstempel, Fahrzeugposition und Fahrtrichtung) verwendet werden.

Anforderung A5 Es sollen Suchmethoden verwendet werden, die unabhängig von Bilddaten funktionieren.

4 Konzept und Entwurf des semantisch durchsuchbaren Datenmanagementsystems Damast

Die identifizierten Anforderungen (vgl. Unterabschnitt 3.4.1) legen die Rahmenbedingungen für den Aufbau und die Funktionsweise des im Folgenden neu entwickelten, semantisch durchsuchbaren **Da**ten**ma**nagementsy**st**ems (Damast) fest. Das System ist zweigeteilt (siehe Abbildung 4.1): Im ersten Teil (1) durchlaufen die Daten einen Anreicherungsprozess. In diesem Prozess werden die Daten in Damast hinzugefügt und um zusätzliche Informationen ergänzt. Nach dieser initialen Vorverarbeitung sind sie über Damast auffindbar. Der zweite Teil (2) von Damast besteht aus den Methoden zum Durchsuchen der angereicherten Informationen.

4.1 Anreicherungsprozess

Die Anreicherung (1) der Daten (siehe Abbildung 4.2) ist der erste Teil von Damast. Bei diesem Prozess handelt es sich um einen Vorverarbeitungsschritt. Alle Daten, die mit Damast auffindbar sein sollen, werden hier verarbeitet. Der Anreicherungsprozess verarbeitet jedes Datum nur einmal. Neu hinzugekommene Daten durchlaufen die Anreicherung und werden dem System hinzugefügt, ohne dass die bereits prozessierten Daten erneut bearbeitet werden müssen.

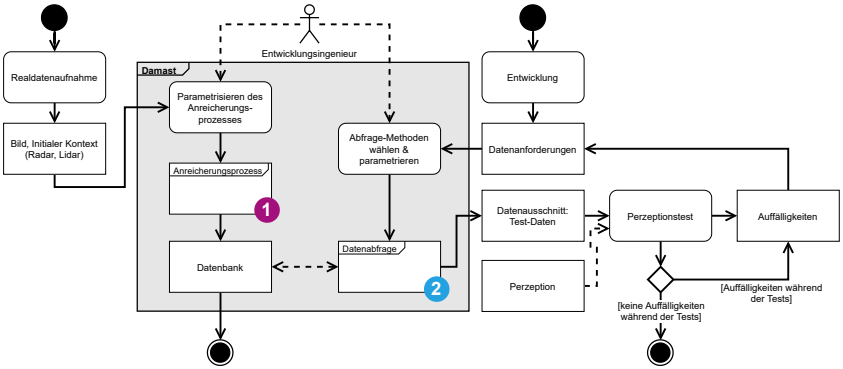


Abbildung 4.1: Übersicht über den Ablauf von Damast eingebettet in die Entwicklung einer Perzeption

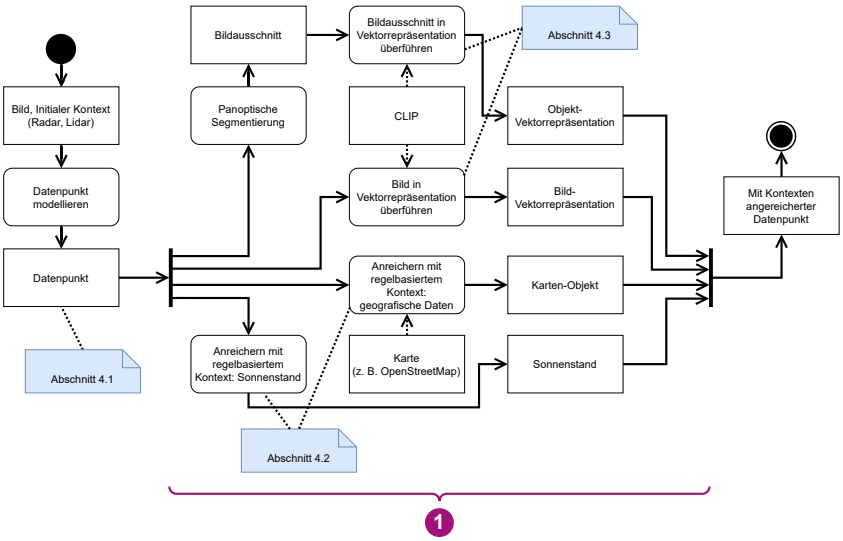


Abbildung 4.2: Ablaufdiagramm des Anreicherungsprozesses 1 von Damast

Der Anreicherungsprozess beginnt mit den in der Realaufnahme aufgezeichneten Daten. Diese Rohdaten werden als Datenpunkte (vgl. Unterabschnitt 4.1.2) modelliert. Das System erweitert die Datenpunkte anschließend um Kontextdaten.

4.1.1 Ausgangsbasis: Realdatenaufnahme

Die im Folgenden betrachteten Realdaten entstehen durch Sensoraufzeichnungen während realer Fahrten. Diese Aufnahmefahrten sind Erprobungsfahrten während der Fahrzeugentwicklung oder Fahrten von Kunden, deren Daten aus dem Feld zurück in die Entwicklung übertragen werden. Abzugrenzen ist die reale Aufnahmefahrt von der synthetischen Bildgeneration mittels Simulation [64] oder der Veränderung existierender Datensätze mittels neuronaler Netze [PR1, 105]. Bei diesen Verfahren existiert ein Modell, welches als Grundlage für die Bildgeneration genutzt wird (siehe Abbildung 3.3). Eine einfachere Durchsuchbarkeit ist in diesem Fall bereits durch dieses Modellwissen realisierbar. Auch wenn es im Weiteren nicht explizit betrachtet wird, ist es dennoch möglich, das hier vorgestellte Konzept auch auf synthetisch erzeugte Bilder anzuwenden. Das gilt für die klassischen Kontexte jedoch nur unter der Voraussetzung, dass die synthetischen Daten über realistische initiale Kontexte verfügen (vgl. „initiale Kontexte“ unten in diesem Abschnitt).

Die Realdaten können aus unterschiedlichen Quellen stammen, es ist für Damast unerheblich, von welchem Fahrzeug die Daten aufgezeichnet wurden. Es muss auch nicht unterschieden werden, ob es sich um Daten aus Erprobungsfahrten mit alten Softwareständen handelt. All diese Daten, aus unterschiedlichen Quellen, werden in einem gemeinsamen Datenmanagementsystem verwaltet. Die Informationen über die Aufnahmeumstände sind dabei ein zusätzliches Kontextdatum. Im Folgenden werden Informationen, die sich auf eine gesamte Aufnahme beziehen, unter dem Begriff Metadaten zusammengefasst. Metadaten können neben Hinweisen auf die Aufnahmequelle auch weitere Informationen wie zum Beispiel Versionsstände enthalten. Der genaue Inhalt der Metadaten sollte auf das zu entwickelnde automatisierte Fahrsystem abgestimmt werden.

Die Aufzeichnung der Daten geschieht mit unterschiedlichen Sensoren. Neben Kameras kommen zum Beispiel auch Radare, Lidare und Ultraschallsensorik bei Datenaufzeichnungen zum Einsatz. Diese zusätzlichen Sensordaten sind zur Anwendung des Konzepts nicht notwendig. Jedoch ist zumindest teilweise eine Übertragung der diskutierten Methoden für die Durchsuchbarkeit auf andere Sensormodalitäten denkbar, solange für jeden aufnehmenden Sensor die Einbauposition im Fahrzeug erfasst wird. Der Fokus liegt allerdings auf Bildern, sodass eine weitere Beurteilung der Durchsuchbarkeit der zusätzlichen Sensormodalitäten nicht erfolgt.

Entscheidend für die Anwendbarkeit des Konzepts ist das Vorhandensein eines initialen Satzes von Kontextdaten. Diese umfassen:

- Aufnahmezeitpunkt (Zeitstempel),
- Aufnahmeort (geografische Koordinaten: Längen- und Breitengrad) und
- Fahrtrichtung (Himmelsrichtung in Grad).

Im einfachsten Fall werden diese Informationen von der Fahrzeugsensorik aufgezeichnet. Ist ein Mitschneiden der Fahrzeugdaten nicht möglich, werden die Daten über zusätzliche, externe Sensorik, wie einen Sensor für ein globales Navigationssatellitensystem und eine inertiale Messeinheit, erfasst. Diese Informationen werden im Folgenden dieses Konzeptes als initiale Kontexte vorausgesetzt. Daten ohne vollständigen Satz von initialen Kontexten werden dennoch weiterverarbeitet. Schritte, in denen ein Metadatum notwendig ist, welches nicht vorhanden ist, werden für diese Daten übersprungen. Die Konsequenz für diese Daten ist dann, dass diese hinsichtlich dieser angereicherten Kontexte nicht auffindbar sind. Die Durchsuchbarkeit bezüglich der weiteren Kontexte bleibt davon unberührt.

Alle nach der Realdatenaufnahme zur Verfügung stehenden Daten werden als Klassen modelliert (siehe Abbildung 4.3). Die Daten stehen anfangs nicht explizit, sondern nur indirekt über die Zeitstempel in Zusammenhang zueinander. Die Metadaten beziehen sich immer auf einen ganzen Satz an aufgenommenen Daten. Explizite Zusammenhänge werden im nächsten Schritt modelliert und erzeugt.

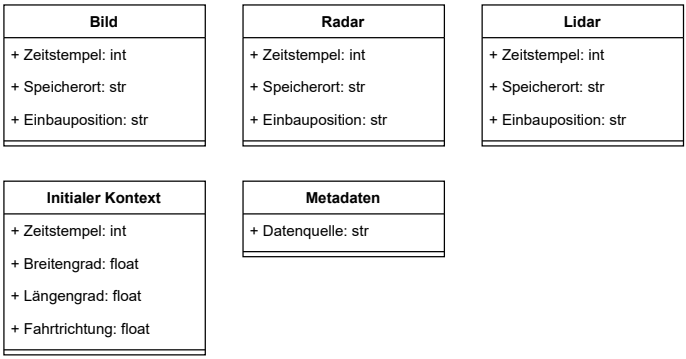


Abbildung 4.3: Klassendiagramme aller nach der Realdatenaufnahme zur Verfügung stehenden Daten; noch ohne explizite Verknüpfungen im Vergleich zur weiteren Verknüpfung über den Datenpunkt (vgl. Abbildung 4.4)

4.1.2 Grundbaustein: Datenpunkt

Die Daten werden für die weitere Verarbeitung als Datenpunkte modelliert (siehe Abbildung 4.4). Grundlage für diesen Verarbeitungsschritt sind die Zeitstempel der Daten. Das System synchronisiert die Daten anhand des Zeitstempels. Diese Synchronisation erfolgt für jede Einzelaufnahme separat. Hierzu werden die Zeitstempel der Bilder einer Kamera als Referenz verwendet. Der Nutzer kann die Referenzkamera frei wählen und speichert die Wahl zur späteren Nachvollziehbarkeit in den Metadaten. Jeder Zeitstempel der Referenzkamera ist der Ausgangspunkt für die Konstruktion eines Datenpunktes. Im nächsten Schritt ordnet das System die Daten der restlichen Sensoren den entsprechenden Datenpunkten zu, indem es für jedes Metadatum den jeweils zeitlich nächstgelegenen Datenpunkt wählt. Die Metadaten werden entsprechend der zugehörigen Aufnahmequelle assoziiert. Beim Synchronisieren ist darauf zu achten, dass die akzeptablen Zeitdifferenzen von dem zu entwickelnden, automatisierten Fahrsystem abhängig sind. Nach dem Auffinden der Daten mit Damast stehen jedoch wieder die Rohdaten mit ihrer ursprünglichen Messfrequenz zur Entwicklung zur Verfügung.

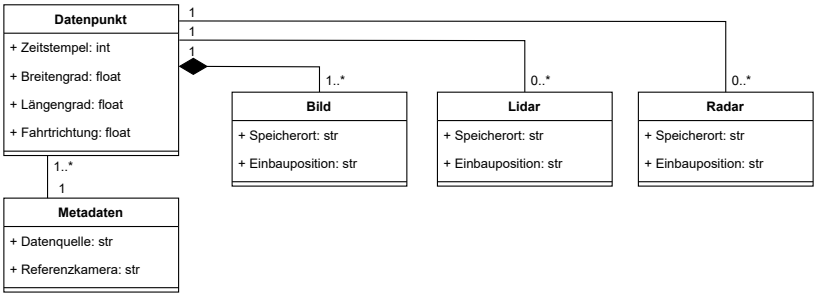


Abbildung 4.4: Der Datenpunkt ist das zentrale Element von Damast. Mit ihm sind die Sensordaten, Metadaten und im Folgenden auch mittelbar die Kontexte assoziiert.

Datenpunkt: Der Datenpunkt ist das zentrale Element von Damast. Ein Datenpunkt beinhaltet die initialen Kontextdaten (Aufnahmezeitpunkt, Aufnahmeort und Fahrtrichtung) und den Zeitstempel, mit welchem die Daten synchronisiert wurden.

Bild: Der Fokus von Damast liegt auf Bildern. Jeder Datenpunkt ist mit mindestens einem Bild assoziiert, von welchem auch der Zeitstempel übernommen wird. Der Speicherort der Bilder muss nicht verändert werden, sondern es wird in Damast auf den Speicherpfad der Bilder verwiesen. Neben der Einbauposition des Kamerasensors können weitere Informationen, wie zum Beispiel die Hard- bzw. Software-Version, an dieser Stelle abgelegt werden.

Lidar, Radar: Diese Klassen sind größtenteils identisch mit der Klasse der Bilder. Sie stehen stellvertretend für weitere Sensormodalitäten. Weitere Daten aus anderen Sensoren können auf die gleiche Weise in der Modellierung abgebildet werden. Auch hier werden zusätzliche Informationen wie die Einbauposition des Sensors gespeichert.

Metadaten: Unter Metadaten fallen alle Informationen, die sich auf eine gesamte Aufnahme aus einer Quelle beziehen. Daher ist es möglich, ein Metadatum auch mehreren Datenpunkten zuzuordnen. Ein Beispiel für ein Metadatum ist die Fahrzeugklasse des aufzeichnenden Fahrzeugs. Diese Daten

müssen manuell eingepflegt werden und ihre Zusammensetzung orientiert sich an der Relevanz für das zu entwickelnde, automatisierte Fahrsystem.

Der Datenpunkt und seine assoziierten Klassen sind die Grundlage für die weitere Verarbeitung in Damast. Aufbauend auf dieser Vorverarbeitung erfolgt die Anreicherung mit Kontexten. Hierzu speichert das System die Daten anhand ihrer Modellierung in einer relationalen Datenbank (vgl. Unterabschnitt 2.6.1). Eine relationale Datenbank gibt durch ihr festes Datenbankschema eine Struktur vor, bietet Konsistenzgarantien und erlaubt die Kombination mehrerer Abfragen.

4.1.3 Allgemeines zur Kontextanreicherung

Die Durchsuchbarkeit ist mit den Informationen eines Datenpunkts und seiner assoziierten Klassen an dieser Stelle noch eingeschränkt. Kontexte (vgl. Abschnitt 2.3) bieten weiterführende Informationen und verbessern die Durchsuchbarkeit hinsichtlich zusätzlicher Aspekte. Ausgehend von den Informationen, die in einem Datenpunkt gespeichert sind, gewinnt das System die Kontexte aus externen Quellen oder durch die Untersuchung der zugehörigen Bilder. Die resultierenden Kontextinformationen werden anschließend mit den bereits in Damast gespeicherten Daten assoziiert.

Die Berechnung der hier vorgestellten Kontexte erfolgt für jeden Datenpunkt unabhängig (siehe Abbildung 4.2). Das erlaubt die Parallelisierung der Berechnung hinsichtlich der Datenpunkte. Zusätzlich können Kontexte, die nicht aufeinander aufbauen, parallel berechnet werden. Auf diese Weise wird dem Berechnungsaufwand für den Anreicherungsprozess bei steigender Größe des Datensatzes begegnet.

Im Folgenden wird zwischen klassischen Kontexten und Kontexten, die auf Vektorrepräsentationen von Bildern basieren, unterschieden (siehe Tabelle 4.1). Diese Kontexttypen unterscheiden sich anhand der Methoden, mit denen sie durchsucht werden können. Klassische Kontexte werden als Sammlung von Booleans, Ganzzahlen, Gleitkommazahlen oder Zeichenfolgen in einer Datenbank gespeichert.

Tabelle 4.1: Kontexttypen und ihre Unterteilung in klassische Kontexte und Kontexte basierend auf Vektorrepräsentationen

Unterscheidungsmerkmale von Kontexten		Klassischer Kontext	Kontext basierend auf Vektorrepräsentationen
Methoden zum Durchsuchen	Methoden relationaler Datenbanken	x	
	Ähnlichkeitssuche in Vektorräumen		x
Ursprung	Initiale Kontexte	x	
	Extraktion aus Bilddaten	x	x
Datentypen	Booleans, Ganzzahlen, Gleitkommazahlen und Zeichenfolgen	x	
	Vektoren		x

Die Suche nach diesen Kontexten erfolgt daher mit den Methoden relationaler Datenbanken (vgl. Unterabschnitt 2.6.1). Im Gegensatz dazu erfolgt die Suche bei Kontexten, die auf Vektorrepräsentationen von Bildern basieren, mit einer Ähnlichkeitssuche in Vektorräumen. Ein weiteres Unterscheidungsmerkmal der Kontexte ist die Quelle, aus der die Informationen stammen. Es werden Kontexte betrachtet, die als Ausgangspunkt die initialen Kontexte nutzen, und Kontexte, die aus den Bilddaten extrahiert werden.

4.2 Anreicherung mit klassischen Kontexten

Den Ausgangspunkt für klassische Kontexte bilden die initialen Kontextdaten (Aufnahmezeitpunkt, Aufnahmeort und Fahrtrichtung). Auf deren Grundlage ermittelt Damast ergänzende Informationen aus externen Datenbanken oder berechnet diese mithilfe von Modellen. Ein klassischer Kontext bezieht sich immer direkt auf ein einzelnes Bild und die Ausrichtung der entsprechenden Kamera und ist daher nur mittelbar mit einem Datenpunkt assoziiert (siehe Abbildung 4.5). Für die

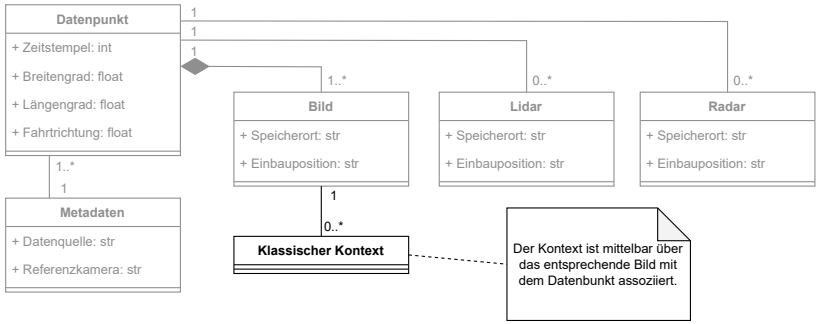


Abbildung 4.5: Klassische Kontexte werden mit dem entsprechenden Bild assoziiert. (Erweiterung von Abbildung 4.4)

Berechnung eines klassischen Kontextes ist die Ausrichtung der Kamera und damit der Bildausschnitt entscheidend. Ohne Beschränkung der Allgemeinheit wird im Folgenden von einer Kamera ausgegangen, deren Ausrichtung hinsichtlich der Horizontalachse mit der Fahrtrichtung identisch ist. Für Kameras, die nicht in Fahrtrichtung montiert sind, gilt das Folgende nach einer Rotation entsprechend der Einbauposition relativ zur Fahrtrichtung analog.

In Übereinstimmung mit Anforderung A1 (siehe Unterabschnitt 3.4.1) sind die folgenden klassischen Kontexte so gewählt, dass sie möglichst viele Pegasus-Ebenen abdecken. Die Auswahl erhebt dabei keinen Anspruch auf Vollständigkeit. Weitere Kontexte können zu einer höheren Genauigkeit bei der Suche oder einer größeren Abdeckung der Pegasus-Ebenen (vgl. Unterabschnitt 2.1.3) führen. Analog zu dem hier vorgestellten Prozess zur Integration der klassischen Kontexte kann für weitere klassische Kontexte vorgegangen werden. Beispielhaft werden hier der Sonnenstand und geografische Daten als klassische Kontexte vorgestellt, da diese die Pegasus-Ebenen vielfältig abdecken.

4.2.1 Kontext: Sonnenstand

Als Beispiel für einen ersten klassischen Kontext wird der Sonnenstand betrachtet. Der Sonnenstand hat einen entscheidenden Einfluss auf die Belichtung der im

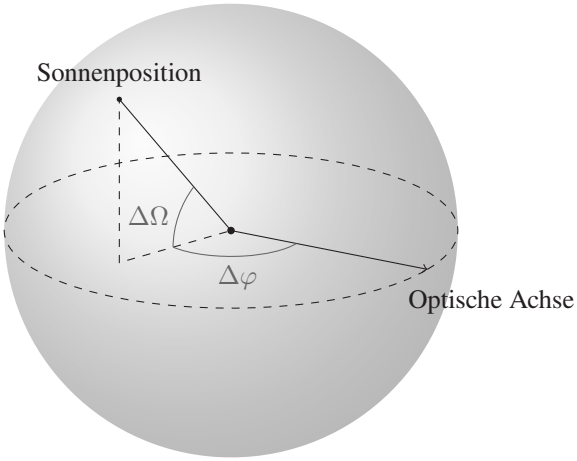


Abbildung 4.6: Sonnenposition relativ zur optischen Achse der aufzeichnenden Kamera mit relativem Höhenwinkel $\Delta\Omega$ und relativem Azimut $\Delta\varphi$ (nach [PR3])

Freien aufgenommenen Bilder im Automotive-Umfeld.[1] Der Stand der Sonne beeinflusst nicht nur die Ausleuchtung und den Schattenwurf, sondern er kann auch zu optischen Abbildungsfehlern in der Kamera führen.

Die Berechnung des Sonnenstandes basiert auf der geografischen Position und dem Aufnahmezeitpunkt. Letztlich entscheidend für die Lichtverhältnisse ist der Sonnenstand relativ zur aufnehmenden Kamera. Im Folgenden wird mit $\Delta\Omega \in [-90°, 90°)$ der relative Höhenwinkel und mit $\Delta\varphi \in [-180°, 180°)$ der relative Azimut zur optischen Achse der aufzeichnenden Kamera bezeichnet (siehe Abbildung 4.6). Zur Berechnung des relativen Sonnenstandes ist zusätzlich auch noch die Information über die Fahrtrichtung $\alpha \in [0°, 360°)$ notwendig.

Eine durch die Einbauposition oder Steigungen bedingte Rotation β der Kamera um die Querachse des Fahrzeugs wird im Folgenden nicht im Detail betrachtet. Jedoch führt eine Addition der Rotation zum Endergebnis zu einer entsprechenden

[1] Der Kontext Sonnenstand ist veröffentlicht in [PR3].

Korrektur $\Delta\Omega' = \Delta\Omega + \beta$, solange der resultierende relative Höhenwinkel $\Delta\Omega'$ im Bereich von $-90°$ und $90°$ liegt.

Der Sonnenstand wird im Horizontsystem angegeben, das von einem Beobachter auf der Erdoberfläche ausgeht. Die Position der Sonne wird in diesem System über den Höhenwinkel $\Omega \in [-90°, 90°)$ und den Azimut $\varphi \in [0°, 360°)$ angegeben. Eine entsprechende Berechnungsvorschrift wird zum Beispiel von Reda et al. [106] vorgeschlagen. Der relative Azimut ergibt sich anschließend wie folgt:

$$\Delta\varphi = ((\varphi - \alpha + 180°) \bmod 360°) - 180°. \tag{4.1}$$

4.2.2 Kontext: Geografische Daten

In Karten (vgl. Unterabschnitt 2.3.1) sind Objekte verzeichnet, die in Kamerabildern sichtbar sein können und die sich potenziell auf die Leistungsfähigkeit von automatisierten Fahrsystemen auswirken.[2] Ein Beispiel hierfür sind Brücken, die durch ihren Schattenwurf zu einem hohen Dynamikumfang im Bild führen und dadurch eine Über- bzw. Unterbelichtung von Bildern begünstigen. Weitere Beispiele sind Verkehrsspiegel, Zebrastreifen, Ampeln, Windräder und Fahrradwege. Das Ziel dieser Kontextanreicherung ist es, die Datenpunkte hinsichtlich der Sichtbarkeit von Karten-Objekten durchsuchbar zu machen.

Die Sichtbarkeit eines Objekts im Kamerabild hängt vom Aufnahmeort, der Fahrtrichtung und dem Öffnungswinkel der Kamera ab. Die Kartendaten umfassen in diesem Fall die geografische Position des Objekts zusammen mit der Bezeichnung des Objekttyps. Für ausgedehnte Objekte, wie Brücken, sind mehrere geografische Punkte durch Kanten zu einem Objekt verbunden. Die Kartendaten müssen für den jeweiligen Aufnahmezeitpunkt des Bildes vorliegen.

Für die Bestimmung der Sichtbarkeit eines Objekttyps im Kamerabild extrahiert das System zunächst alle Objekte dieses Typs in der Umgebung des Fahrzeugs

[2] Der Kontext geografische Daten ist veröffentlicht in [PR3].

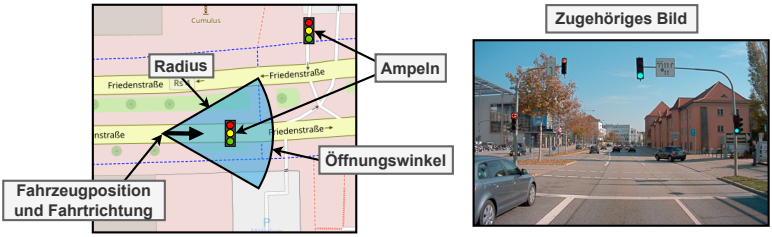


Abbildung 4.7: Schematische Darstellung der Kontextbestimmung für die Sichtbarkeit von geografischen Objekten am Beispiel von Ampeln aus OpenStreetMap-Kartendaten [107] mit dem zugehörigen Bild aus dem A2D2-Datensatz [9] (nach [PR8])

aus den Kartendaten. Der Radius, in dem die Objekte extrahiert werden, hängt vom Objekttyp ab und muss manuell als Parameter vom Nutzer festgelegt werden. Windräder sind beispielsweise aufgrund ihrer Größe und Höhe aus größerer Entfernung sichtbar als ein Zebrastreifen. Ein größerer Radius erhöht die Anzahl der falsch-positiven und ein zu kleiner Radius erhöht die Anzahl der falsch-negativen Ergebnisse.

Nach der Extraktion der Objekte des entsprechenden Typs aus den Kartendaten prüft das System, ob mindestens eines der Objekte im Sichtfeld der Kamera liegt. Mithilfe der Fahrzeugposition, der Fahrtrichtung, dem horizontalen Öffnungswinkel der Kamera und dem zuvor festgelegten Radius wird ein Kreissegment definiert (siehe Abbildung 4.7). Fällt mindestens eines der Objekte in dieses Kreissegment, so wird die Sichtbarkeit des Objekttyps im Kamerabild angenommen.

Zusammen mit der Objektklasse wird gespeichert, welcher Radius und welcher Öffnungswinkel vom Nutzer gewählt wurden. Ein Beispiel ist die Suche nach Ampeln in einem Radius von $30\,\mathrm{m}$ und einem Öffnungswinkel von $30°$ vor der Kamera. Die mit der Methode ermittelte Sichtbarkeit wird als Boolean gespeichert.

4.2.3 Klassische Kontexte in Damast

Jeder klassische Kontext wird als eigenes Objekt in Damast abgebildet. Klassische Kontexte sind immer direkt mit einem Bild assoziiert (siehe Abbildung 4.8). Dabei können auch mehrere Kontexte des gleichen Kontexttyps einem Bild zugeordnet sein. Das ist beispielsweise bei Karten-Objekten von unterschiedlichem Typ der Fall.

Hier wurde eine exemplarische Auswahl von klassischen Kontexten vorgestellt. Analog können andere Kontexte mit einem Datenpunkt verknüpft und in Damast abgebildet werden. Infrage kommen alle Informationen, die sich ausgehend von den initialen Kontexten bestimmen lassen. Weitere klassische Kontexte können zum Beispiel die folgenden sein:

- Wetterdaten,
- Verkehrs- und Stau-Informationen,
- infrastrukturelle Daten (z. B. Ampelschaltungen),
- und Höheninformationen.

Diese Informationen werden, wie die bereits vorgestellten klassischen Kontexte, in Damast mit den restlichen Daten assoziiert und gespeichert (vgl. Abbildung 4.5).

4.3 Anreicherung mit Kontexten basierend auf Vektorrepräsentationen

Bei den klassischen Kontexten legt der Nutzer im Vorhinein durch die Wahl der Anreicherungsmethoden und der Parameter fest, nach welchen Bildeigenschaften eine Suche möglich sein wird. So können zum Beispiel nur die Karten-Objekte gefunden werden, die in der Anreicherung betrachtet wurden. Eine Suche nach beliebigen Bildeigenschaften ist mit klassischen Kontexten nicht umsetzbar. Eine

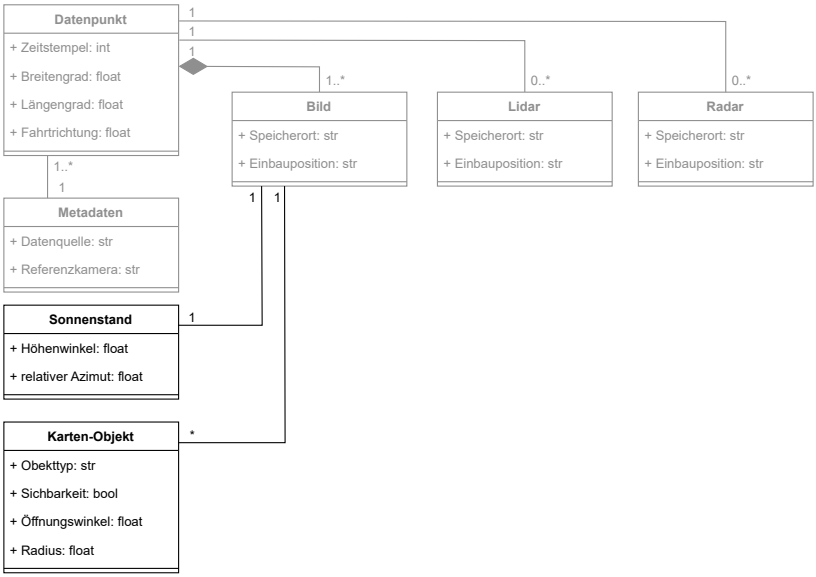


Abbildung 4.8: Datenpunkt angereichert mit den klassischen Kontexten Sonnenstand und Karten-Objekt (Erweiterung von Abbildung 4.4)

generische Suche wird jedoch mit den Kontexten, welche auf Vektorrepräsentationen (vgl. Abschnitt 2.6) basieren, möglich.[3] Mit dieser Methode kann semantisch nach beliebigen Bildeigenschaften gesucht werden, ohne dass eine explizite Anreicherung notwendig ist.

Hierfür überführt der CLIP-Encoder $\mathcal{E}_{\text{Bild}}$ die Bilder $\{I_{i,\text{Gesamtbild}}\}_{i=1}^{n}$ des Datensatzes in Vektorrepräsentationen (vgl. Abschnitt 2.6). Diese Repräsentationen werden anschließend genutzt, um die Bilder in Damast zu durchsuchen. Dieser Vorverarbeitungsschritt ist für jedes Bild nur einmal während der Anreicherung notwendig.

[3] Das folgende Vorgehen ist veröffentlicht in [PR4].

(a) Gesamtbild **(b)** Isoliertes Fahrzeug

Abbildung 4.9: Im Gesamtbild ist die Stretchlimousine nur ein Fahrzeug unter mehreren. Wird das Fahrzeug isoliert, schlagen sich die Objekteigenschaften deutlicher in der Vektorrepräsentation nieder. (Bildquelle: [12])

4.3.1 Berechnung der Vektorrepräsentationen

Zunächst wird die Vektorrepräsentation jedes Bildes berechnet:

$$\mathcal{E}_{\text{Bild}}(I_{i,\text{Gesamtbild}}) = \vec{e}_{i,\text{Gesamtbild}}. \tag{4.2}$$

Dadurch wird die semantische Suche nach Eigenschaften, die das gesamte Bild umfassen, ermöglicht. Ein Beispiel für eine solche Eigenschaft ist die Überbelichtung eines Bildes.

Nun soll die semantische Durchsuchbarkeit am Ende nicht nur auf Gesamtbildebene möglich sein. Nimmt ein Aspekt des Bildes, wie eine Überbelichtung oder ein Objekt, nur einen Ausschnitt des Bildes ein und ist im Gesamtbild nicht dominant, so ist davon auszugehen, dass dieser Teil durch den Vektor nur unzureichend repräsentiert und folglich nicht aufzufinden ist. Beispielsweise ist es möglich, dass die Suche nach einer Stretchlimousine fehlschlägt, da sie auf einem Bild nur ein Fahrzeug unter mehreren ist (siehe Abbildung 4.9). Im Bereich der Bildsuche wird dieses Problem, bei dem nur ein Teil des Bildes von Interesse ist und der Rest ignoriert werden soll, als *Distraction Challenge* bezeichnet [108]. Die Isolation des Fahrzeugs aus dem Gesamtbild sorgt in diesem Fall dafür, dass die Objekteigenschaften für den CLIP-Encoder erfassbar werden. Daher berechnet

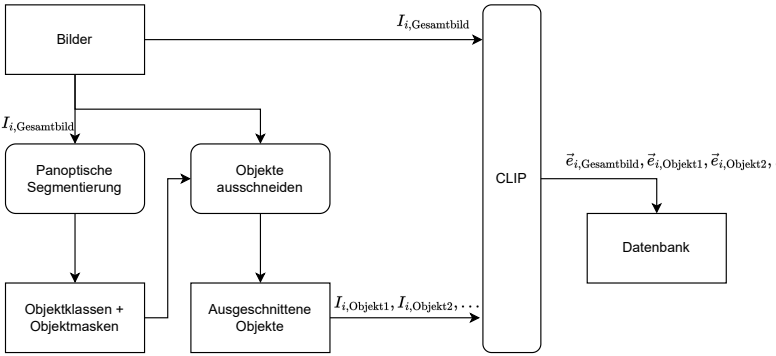


Abbildung 4.10: Übersicht über die Anreicherung mit Vektorrepräsentationen auf Gesamtbild- und Objektebene

das System bei der Suche nach Objekteigenschaften die Vektorrepräsentationen auf Objektebene (siehe Abbildung 4.10).[4]

Hierzu wird ein Bild zunächst in seine Bestandteile aufgeteilt. Diese Zerlegung erfolgt durch eine panoptische Segmentierung f_{panoptic} [109] (vgl. Abschnitt 2.5), die das Bild $I_{i,\text{Gesamtbild}}$ hinsichtlich der Objektklassen und Objektinstanzen zerlegt:

$$f_{\text{panoptic}}(I_{i,\text{Gesamtbild}}) = (\vec{o}_{i,1}, \ldots, \vec{o}_{i,n(i)}). \tag{4.3}$$

Dabei steht $n(i)$ für die Anzahl der detektierten Objekte im Bild. Die resultierenden Objekte $\vec{o}_{i,j} = (I_{i,j}, c_{i,j})$ bestehen aus dem Bildausschnitt des Objekts $I_{i,j}$ und der zugehörigen von der panoptischen Segmentierung prädizierten Objektklasse $c_{i,j} \in \mathcal{C}$. Der Bildausschnitt wird erzeugt, indem die Maske der panoptischen Segmentierung genutzt wird, um den nicht zum Objekt gehörenden Hintergrund zu schwärzen und das Objekt an dem die Objektmaske einschließenden Begrenzungsrechteck auszuschneiden (siehe Abbildung 4.11). $\mathcal{C}$ beschreibt die Menge aller von der panoptischen Segmentierung prädizierbaren Objektklassen.

[4] Das folgende Vorgehen ist veröffentlicht in [PR5].

(a) Ausgangsbild mit einschließendem Begrenzungsrechteck des entsprechenden Objekts

(b) Panoptische Segmentierung überlagert mit dem Ausgangsbild

(c) Ausgeschnittenes Objekt mit geschwärztem Hintergrund

Abbildung 4.11: Beispiel für das Ausschneiden eines Objekts mittels panoptischer Segmentierung (Bildquelle: [9])

Für das Gesamtbild $I_{i,\text{Gesamtbild}}$ und für jeden Bildausschnitt $I_{i,j}$ werden anschließend die Vektorrepräsentationen berechnet:

$$\mathcal{E}_{\text{Bild}}(I_{i,\text{Gesamtbild}}) = \vec{e}_{i,\text{Gesamtbild}}, \tag{4.4}$$

$$\mathcal{E}_{\text{Bild}}(I_{i,j}) = \vec{e}_{i,j}. \tag{4.5}$$

Zu jedem Bild existiert nach der Verarbeitung neben der Vektorrepräsentation des Gesamtbildes auch eine Vektorrepräsentation jedes erkannten Objekts im Bild inklusive der entsprechenden Objektklasse (siehe Tabelle 4.2).

Zusätzlich zu den Vektorrepräsentationen werden auch die einschließenden Begrenzungsrechtecke der Objekte gespeichert. Diese Information ermöglicht zusätzlich die Suche nach dem Vorhandensein, der Größe und der Position von Objekten.

Tabelle 4.2: Beispiel für die Daten, die bei der Anreicherung mit Vektorrepräsentationen auf Objektebene entstehen

#Bild	Objektklasse	Vektorrepräsentation
1	`Gesamtbild`	$\vec{e}_{\text{Bild1,Gesamtbild}}$
1	`Auto`	$\vec{e}_{\text{Bild1,Objekt1}}$
1	`Person`	$\vec{e}_{\text{Bild1,Objekt2}}$
1	`Auto`	$\vec{e}_{\text{Bild1,Objekt3}}$
...	...	...
2	`Gesamtbild`	$\vec{e}_{\text{Bild2,Gesamtbild}}$
2	`Person`	$\vec{e}_{\text{Bild2,Objekt1}}$
...	...	...

4.3.2 CLIP-Vektorrepräsentationen in Damast

Der Kontext basierend auf Vektorrepräsentationen wird zusammen mit den klassischen Kontexten in Damast verwaltet (siehe Abbildung 4.12). Die Vektorrepräsentation (vgl. Abschnitt 2.6) des Gesamtbildes wird direkt mit dem entsprechenden Bild assoziiert. Das System verknüpft die Bildausschnitte der Objekte mit dem zugehörigen Bild. Mit dem jeweiligen Bildausschnitt wird dann die Vektorrepräsentation des Objekts verbunden. Dadurch entsteht zwischen Vektorrepräsentationen und Bild bzw. Bildausschnitt jeweils eine Eins-zu-eins-Beziehung. Das ermöglicht eine Auslagerung der Vektorrepräsentationen. Dieser Schritt ist notwendig, da sich die Vektorrepräsentationen nicht ausreichend effizient mit den Methoden klassischer relationaler Datenbanken vergleichen lassen. Die Vektoren, die der Bild-Encoder der verwendeten CLIP-Implementierung generiert, haben 512 Dimensionen: $\vec{e} \in \mathbb{R}^{512}$. Die Vektoren repräsentieren durch das Training von CLIP die Semantik des jeweiligen Bildausschnitts. Die Ablage in einer Vektordatenbank [53] ermöglicht die effiziente Abfrage der Vektoren (vgl. Unterabschnitt 2.6.1).

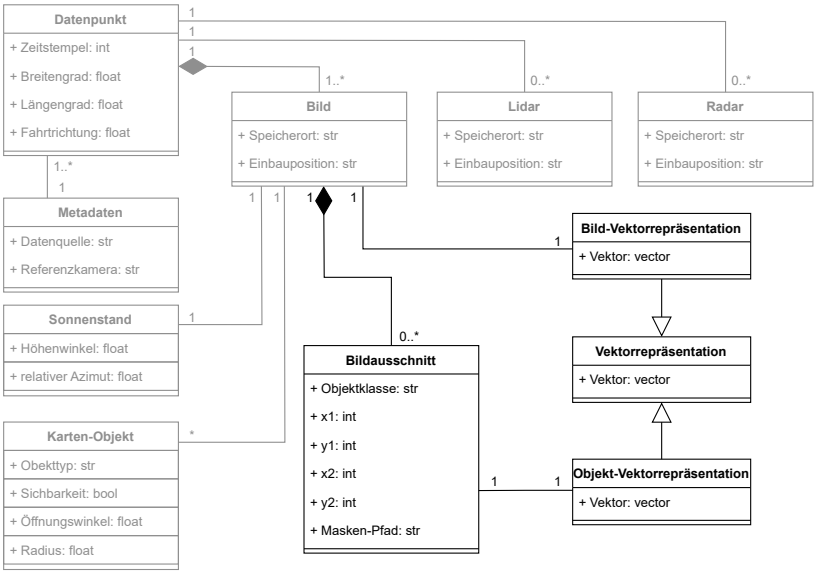


Abbildung 4.12: Übersicht über das Datenmodell von Damast am Ende der Anreicherung (Erweiterung von Abbildung 4.8)

4.4 Datenabfrage in Damast

Im Anschluss an die Anreicherung der Datenpunkte in Damast sind diese anhand der Kontexte auffindbar. Die Datenanforderungen legen fest, nach welchen Kontexten der Nutzer sucht. Aus ihnen folgt daher auch die Wahl der Suchmethode. Der Nutzer bestimmt außerdem auf Basis der Datenanforderungen die Parametrisierung der Methoden zum Durchsuchen von Damast.

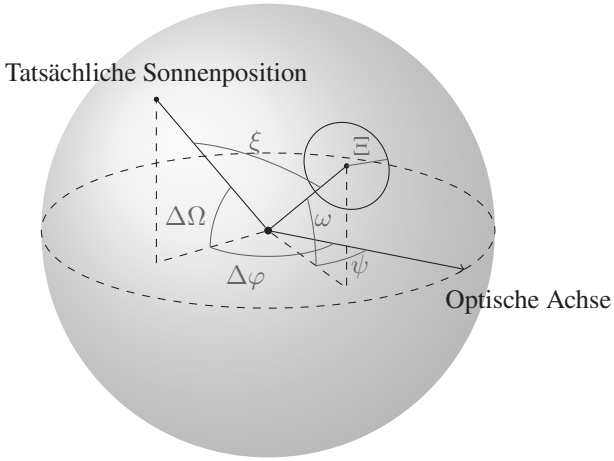


Abbildung 4.13: Schematische Darstellung der Abfrage einer relativen Sonnenposition mit Höhenwinkel ω, Azimut ψ und maximal zulässiger Winkelabweichung Ξ und der tatsächlichen relativen Sonnenposition eines Datenbankeintrags mit Höhenwinkel $\Delta\Omega$ und Azimut $\Delta\varphi$. Die berechnete Winkelabweichung ist mit ξ bezeichnet. [PR3]

4.4.1 Suche nach Sonnenpositionen

Bei der Formulierung der Suche nach bestimmten Sonnenständen[5] in den Daten gibt es zwei Möglichkeiten. Zum einen können feste Unter- und Obergrenzen für den relativen Höhenwinkel $\Delta\Omega$ und Azimut $\Delta\varphi$ gefordert werden. In diesem Fall beschränkt sich die Abfrage in der Datenbank auf entsprechende Filter für die jeweiligen Spalten. Zum anderen hat der Nutzer auch die Möglichkeit, eine relative Sonnenposition mit Höhenwinkel ω, Azimut ψ und maximal zulässiger Winkelabweichung Ξ vorzugeben (siehe Abbildung 4.13). Anschließend berechnet das System für jeden Eintrag in der Datenbank die Winkelabweichung zur angefragten

[5] Der Kontext Sonnenstand ist veröffentlicht in [PR3].

relativen Sonnenposition. Hierfür wird der tatsächliche relative Sonnenstand eines Eintrags als Vektor dargestellt:

$$\vec{s}_{\text{Eintrag}} = \begin{pmatrix} \sin(\Delta\Omega) \cdot \cos(\Delta\varphi) \\ \sin(\Delta\Omega) \cdot \sin(\Delta\varphi) \\ \cos(\Delta\Omega) \end{pmatrix}. \tag{4.6}$$

Auch der angefragte relative Sonnenstand wird als Vektor repräsentiert:

$$\vec{s}_{\text{Anfrage}} = \begin{pmatrix} \sin(\omega) \cdot \cos(\psi) \\ \sin(\omega) \cdot \sin(\psi) \\ \cos(\omega) \end{pmatrix}. \tag{4.7}$$

Für die Winkelabweichung des Eintrags ergibt sich dann:

$$\begin{aligned} \xi &= \arccos\left(\frac{\vec{s}_{\text{Anfrage}} \cdot \vec{s}_{\text{Eintrag}}}{\|\vec{s}_{\text{Anfrage}}\|\|\vec{s}_{\text{Eintrag}}\|}\right) \\ &= \arccos\left(\sin(\Delta\Omega)\sin(\omega)\cos(\Delta\varphi - \psi) + \right. \\ &\quad \left. \cos(\Delta\Omega)\cos(\omega)\right). \end{aligned} \tag{4.8}$$

Die Einträge werden anschließend anhand der maximal zulässigen Winkelabweichung gefiltert: $\xi < \Xi$. Es ist zusätzlich auch eine Sortierung hinsichtlich der Winkelabweichung ξ möglich, sodass das erste Ergebnis den zur Anfrage ähnlichsten relativen Sonnenstand besitzt.

4.4.2 Suche nach Kartenobjekten

Für die Suche nach einem Kartenobjekt muss der gesuchte Objekttyp spezifiziert sein. Die Suche ist dabei auf Objekttypen begrenzt, die der Nutzer vor der Anreicherung festgelegt hat. Wurden für einen Objekttyp mehrere Parameter für Radius und Öffnungswinkel gewählt, so sind auch diese bei der Suche zu spezifizieren.

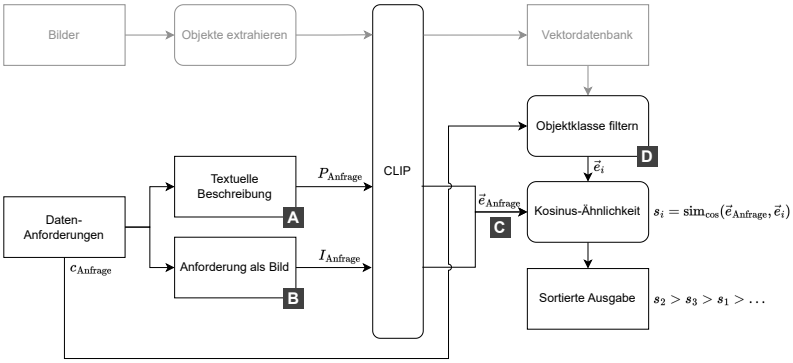


Abbildung 4.14: Übersicht über die Suche mit Vektorrepräsentationen (in Anlehnung an [PR4])

Initial wird die Datenbank nach den Objekttypen, dem Radius und dem Öffnungswinkel gefiltert. Die anschließende Abfrage beschränkt sich auf das Filtern nach der Sichtbarkeit des Objekts. Dabei ist es möglich, entweder das Vorhandensein eines Objekts zu fordern oder das Vorhandensein explizit auszuschließen.

4.4.3 Suche nach Vektorrepräsentationen

Die Suche mit Vektorrepräsentationen basiert auf der Suche nach ähnlichen Vektoren (siehe Abbildung 4.14 im Folgenden mit den Referenzen **A**, **B**, **C**, **D**). CLIP überführt neben Bildern auch Texte in eine Vektorrepräsentation (vgl. Abschnitt 2.6). Die Vektorrepräsentationen von Texten und Bildern beziehen sich dabei auf den gleichen latenten Raum, was bedeutet, dass ähnlichen Texten und Bildern ähnliche Vektorrepräsentationen zugeordnet werden [46]. Im Folgenden wird diese Eigenschaft ausgenutzt, um textuelle Anfragen zu stellen.

CLIP wurde mit englischen Bild-Text-Paaren trainiert [46], sodass auch die hier gestellten Anfragen in englischer Sprache verfasst sein müssen. Die sich aus den Datenanforderungen ergebenden Anfragen werden in natürlicher Sprache formuliert. Sucht der Nutzer zum Beispiel mit P_{Anfrage} =„overexposed“ nach

einem überbelichteten Bild **A**, so überführt der CLIP-Encoder die Anfrage in die Vektorrepräsentation **C**:

$$\mathcal{E}_{\text{Text}}(P_{\text{Anfrage}}) = \vec{e}_{\text{Anfrage}}. \tag{4.9}$$

Ist eine Datenanforderung durch ein Bild I_{Anfrage} repräsentiert, so ist es auch möglich, dieses Bild für die Anfrage zu nutzen **B**. In diesem Fall wird der Anfrage-Vektor mit dem CLIP-Encoder für Bilder berechnet **C**:

$$\mathcal{E}_{\text{Bild}}(I_{\text{Anfrage}}) = \vec{e}_{\text{Anfrage}}. \tag{4.10}$$

Die Bilder in der Datenbank werden anhand der Ähnlichkeit des zugehörigen Vektors zum Abfragevektor $\vec{e}_{\text{Abfrage}}$ sortiert. Hierzu wird für jede Vektorrepräsentation in der Datenbank die Kosinus-Ähnlichkeit zum Abfragevektor berechnet. Die Kosinus-Ähnlichkeit [110] von Vektoren ist definiert als:

$$\text{sim}_{\cos}(\vec{x}, \vec{y}) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}, \tag{4.11}$$

mit $\vec{x} \in \mathbb{R}^n$, $\vec{y} \in \mathbb{R}^n$ und $n \in \mathbb{N}$. Die Kosinus-Ähnlichkeit ist ein Skalar und ein Maß dafür, inwieweit die beiden Vektoren in die gleiche Richtung zeigen. Zeigen die Vektoren in die gleiche Richtung, hat sie den Wert 1, zeigen sie in entgegengesetzte Richtung, hat sie den Wert -1. Die Kosinus-Ähnlichkeit des Abfragevektors mit einem Vektor $\vec{e}_i$ aus der Datenbank wird mit $s_i = \text{sim}_{\cos}(\vec{e}_{\text{Anfrage}}, \vec{e}_i)$ bezeichnet. Anhand der Kosinus-Ähnlichkeiten s_i erfolgt die Sortierung der Bilder. Je größer der Wert der Kosinus-Ähnlichkeit, desto ähnlicher sind sich die Vektoren und desto ähnlicher schätzt CLIP die Semantik der Anfrage und des Datenbankeintrags ein.

Wenn es Bildeigenschaften gibt, die explizit nicht gefunden werden sollen, formuliert der Nutzer einen zweiten Anfrageteil $P_{\text{Anfrage,negativ}}$. Analog zum bereits

Beschriebenen berechnet das System auch zu diesem Anfrageteil die Vektorrepräsentation und anschließend die Kosinus-Ähnlichkeit zu jedem Datenbankeintrag $s_{\text{negativ},i}$. Zusammen mit den Kosinus-Ähnlichkeiten der ursprünglichen Anfrage $s_{\text{positiv},i}$ ergibt sich dann die resultierende Ähnlichkeit:

$$\begin{aligned} s_{\text{resultierend},i} &= \text{sim}_{\cos}(\vec{e}_{\text{positiv}}, \vec{e}_i) - \text{sim}_{\cos}(\vec{e}_{\text{negativ}}, \vec{e}_i) \\ &= s_{\text{positiv},i} - s_{\text{negativ},i}, \end{aligned} \tag{4.12}$$

nach der die Datenbank sortiert wird.

Wenn sich eine Datenanforderung auf die Eigenschaften einzelner Objekte bezieht, so ist die Suche auf Objektebene der Suche auf Gesamtbildebene vorzuziehen. Ein Beispiel ist die Suche nach einer Stretchlimousine (vgl. Abbildung 4.9), die ein Fahrzeug unter mehreren in einem Gesamtbild ist. Bei der Suche mit Vektorrepräsentationen auf Objektebene besteht die Anfrage aus zwei Komponenten: $(c_{\text{Anfrage}}, P_{\text{Anfrage}})$. c_{Anfrage} spezifiziert die angefragte Objektklasse: $c_{\text{Anfrage}} \in \mathcal{C}$. Die möglichen Objektklassen $\mathcal{C}$ sind durch die verwendete panoptische Segmentierung festgelegt. Der Nutzer beschreibt das Gesuchte, ebenso wie bei der Suche auf Gesamtbildebene, textuell: P_{Anfrage} **B**. Bei der Suche nach einer Stretchlimousine sähe die Anfrage demnach beispielsweise folgendermaßen aus: („car“, „stretch limousine“).

Das System berechnet den Anfragevektor analog zur Suche auf Gesamtbildebene (vgl. Gleichung 4.9 und **C**):

$$\mathcal{E}_{\text{Text}}(P_{\text{Anfrage}}) = \vec{e}_{\text{Anfrage}}. \tag{4.13}$$

Wird das Gesuchte mit einem Bild beschrieben, so erfolgt die Berechnung des Anfragevektors wie folgt (vgl. Gleichung 4.10 und **C**):

$$\mathcal{E}_{\text{Bild}}(I_{\text{Anfrage}}) = \vec{e}_{\text{Anfrage}}. \tag{4.14}$$

Der entscheidende Unterschied bei der Suche auf Objektebene ist, dass die Bildausschnitte zunächst nach Einträgen mit der entsprechenden Objektklasse

(a) Suche nach einem Bild ohne motorisierte Fahrzeuge, mit mindestens einem Gebäude und mehr als zehn Personen

(b) Suche nach einem Bild mit einem Bus auf der linken Bildhälfte und einem Lastkraftwagen auf der rechten Bildhälft

(c) Suche nach einem Bild mit einer Person, deren Begrenzungsrechteck mindestens 20 % der Bildfläche einnimmt

Abbildung 4.15: Beispielergebnisse für die Suche nach Objekten (Bildquellen: [12])

c_{Anfrage} gefiltert werden **D**. Die zu den verbleibenden Bildausschnitten gehörenden Objekt-Vektorrepräsentationen werden anschließend anhand von $\vec{e}_{\text{Anfrage}}$ sortiert. Die Sortierung erfolgt analog zu der Suche auf Gesamtbildebene anhand der Kosinus-Ähnlichkeit. Es ist möglich, dass in einem Bild eine Objektklasse mehrfach vertreten ist. Um zu verhindern, dass ein Bild mehrfach in den Ergebnissen auftaucht, wird für jedes Bild nur das Resultat mit der höchsten Kosinus-Ähnlichkeit in die Ergebnisse aufgenommen.

4.4.4 Suche nach Objekten

Bei der Konstruktion der Objekt-Vektorrepräsentationen werden auch die in einem Bild vorhandenen Objekte und deren Positionen als Bildausschnitte gespeichert (vgl. Unterabschnitt 4.3.1). Die Eigenschaften dieser Bildausschnitte werden als klassische Kontexte behandelt. Das erlaubt dem Nutzer die Suche nach der Anzahl von Objekten einer bestimmten Klasse, nach der Position und der Größe von Objekten im Bild.

Bei der Suche nach einer bestimmten Anzahl von Objekten legt der Nutzer für jede Objektklasse $c \in \mathcal{C}$ eine Mindest- und Maximalzahl fest. Das System wählt die Bilder mit der passenden Anzahl von assoziierten Bildausschnitten mit entsprechender Objektklasse c aus. Ein Beispiel ist die Suche nach einem Bild ohne motorisierte Fahrzeuge, mit mindestens einem Gebäude und mehr als zehn Personen (siehe Abbildung 4.15a).

Zusätzlich ist es möglich, das Vorhandensein eines Objekts einer ausgewählten Objektklasse an einer bestimmten Position zu fordern. Das wird entweder über Minimal- und Maximalwerte für das einschließende Begrenzungsrechteck des Objekts realisiert oder die Ergebnisse werden anhand des Abstands des Schwerpunkts des einschließenden Begrenzungsrechtecks zu einer geforderten Position im Bild sortiert. Ein Beispiel ist die Suche nach Fahrzeugen in bestimmten Bildbereichen. Ein Nutzer hat zum Beispiel die Möglichkeit, nach einem Bild zu suchen, in dem in der linken Bildhälfte ein Bus und in der rechten Bildhälfte ein Lastkraftwagen zu sehen ist (siehe Abbildung 4.15b). Diese Suche lässt sich realisieren, indem gefordert wird, dass die rechte Kante des Begrenzungsrechtecks des Busses in der linken Bildhälfte und die linke Kante des Begrenzungsrechtecks des Lastkraftwagens in der rechten Bildhälfte ist.

Außerdem kann ein Nutzer nach Objekten mit bestimmter Größe suchen. Die Größe bezieht sich dabei auf die Größe, mit der das Objekt im Bild erscheint, und sie wird bestimmt, indem die vom Begrenzungsrechteck oder von der Objektmaske eingeschlossene Pixelanzahl berechnet wird. Für die Unabhängigkeit von der Auflösung des Bildes wird diese Pixelzahl durch die Gesamtpixelzahl des Bildes

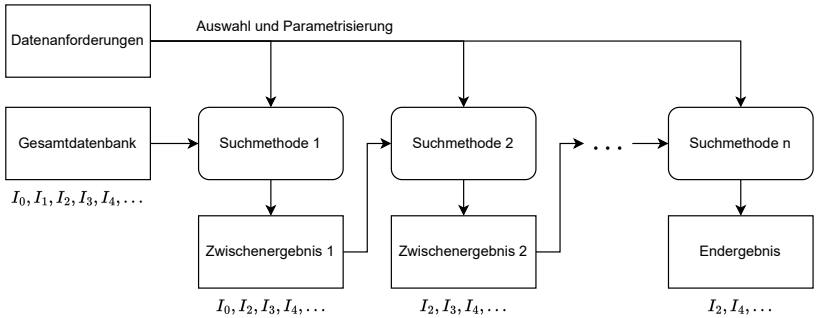


Abbildung 4.16: Kaskadierung von Suchmethoden

geteilt. Daraus resultiert die relative Größe des Objekts im Bild. Der Nutzer gibt bei seiner Suche eine Ober- und Untergrenze für die relative Größe einer Objektklasse an. Zu beachten ist, dass die Fläche eines Begrenzungsrechtecks immer nur eine Obergrenze für die tatsächliche Fläche des Objekts ist. Das tatsächliche Objekt hat also in fast allen Fällen eine kleinere Fläche (vgl. Abbildung 4.9b). Der Nutzer findet mit dieser Methode zum Beispiel Personen, die sich nah an der aufnehmenden Kamera befinden (siehe Abbildung 4.15c).

4.4.5 Kombination von Suchmethoden

Der Nutzer kann die vorgestellten Suchmethoden kaskadieren. Eine Abfrage besteht dann aus mehreren Einzelabfragen, deren Bearbeitung nacheinander erfolgt. Die erste Suchmethode wird weiterhin auf die Gesamtdatenbank angewendet. Die folgenden Methoden beziehen sich jeweils auf die vorhergehenden Zwischenergebnisse. Die Zwischenergebnisse der einzelnen Suchen bestehen aus einer Sammlung von Bild-Objekten. Das System filtert die Bild-Objekte, wie zuvor beschrieben, mit den entsprechenden Methoden (siehe Abbildung 4.16).

Wird eine Methode basierend auf Vektorrepräsentationen verwendet, so werden die Bilder lediglich sortiert. Sollen mehrere dieser Methoden kaskadiert werden,

so ist es notwendig, Ergebnisse, deren Kosinus-Ähnlichkeit kleiner als ein Grenzwert ist, zu verwerfen. Auf diese Weise sind nur diejenigen Bilder im Zwischenergebnis enthalten, die die entsprechende Suchanfrage erfüllen. Ein allgemeingültiger Grenzwert für die Kosinus-Ähnlichkeit existiert nicht. Ein sinnvoller Wert kann jedoch aufgrund der Verteilung der Werte der Kosinus-Ähnlichkeit geschätzt werden.[6]

4.4.6 Beispiel für die Nutzung von Damast

Der Nutzer hat beim Anreicherungsschritt und bei der Datenanfrage von Damast die Möglichkeit, das System zu parametrisieren und mit ihm zu interagieren (siehe Abbildung 4.1). Als Beispiel wird im Folgenden eine Datenanforderung betrachtet, die eine Situation fordert, in der ein Zebrastreifen aufgrund von Überbelichtung nur unzureichend erkennbar ist. Für das Beispiel wird der KITTI-Datensatz [9] verwendet.

Parametrisierung des Anreicherungsprozesses

Zu Beginn muss der Anreicherungsprozess einmalig durch den Nutzer parametrisiert werden. Die Parametrisierung ist abhängig von allen bekannten Datenanforderungen. Es müssen folgende Entscheidungen getroffen werden:

1. Der Nutzer muss entscheiden, welche klassischen Kontexte hinzugefügt werden, und diese Methoden entsprechend implementieren. (Beispiel: Geografische Daten als klassischen Kontext)
2. Die Methoden der klassischen Kontexte müssen parametrisiert werden. (Beispiel: Als geografische Objekte sollen Zebrastreifen betrachtet werden.

[6] Eine entsprechende Methode ist in [PR6] veröffentlicht.

Die Farbkamera des A2D2-Datensatzes [9] hat einen horizontalen Öffnungswinkel von $90°$ und es ist davon auszugehen, dass ein Zebrastreifen mit dieser Kamera bis zu einer Entfernung von $20\,\mathrm{m}$ erkennbar ist.)

3. Der Nutzer wählt eine vortrainierte panoptische Segmentierung und legt damit fest, welche Objektklassen mit Damast auffindbar sind. (Beispiel: Wahl einer panoptischen Segmentierung, die mit den Klassen vortrainiert wurde, nach denen gesucht werden soll. Das konkrete Beispiel stellt keine Anforderungen an die Klassen.)

Im Anschluss läuft der Anreicherungsprozess ohne weitere Interaktion mit dem Nutzer ab.

Datenabfrage durch den Nutzer

Nach dem Anreicherungsschritt, der Teil der Vorverarbeitung ist und nur einmal für jeden Datenpunkt in der Datenbank durchgeführt werden muss, können Datenabfragen durchgeführt werden. Bei der Datenabfrage hat ein Nutzer folgende Entscheidungsmöglichkeiten:

1. Der Nutzer wählt zur Datenanforderung passende Abfragemethoden.
2. Der Nutzer parametrisiert die gewählten Abfragemethoden und bringt sie in eine zur Datenanforderung passende Reihenfolge.

Im Beispiel werden zwei Suchmethoden kombiniert:

1. Suchmethode: Geografische Daten
   - Parametrisierung: Sichtbarkeit von Zebrastreifen mit einem Öffnungswinkel von $90°$ und in einer Entfernung von maximal $20\,\mathrm{m}$
2. Suchmethode: Vektorrepräsentation auf Gesamtbildebene
   - Parametrisierung: Anfrage nach überbelichteten Bildern mit dem Anfragetext $P_{\text{Anfrage}} =$ „overexposed“

Abbildung 4.17: Beispiel für die Suche mit Damast; erstes Ergebnis einer Suche nach einer Situation, in der ein Zebrastreifen aufgrund von Überbelichtung nur unzureichend erkennbar ist (Bildquelle: [11])

Beim Verarbeiten der Anfrage durch Damast wird für jede Suchmethode die Bedingung abgearbeitet und das Zwischenergebnis als Ausgangsbasis an die folgende Methode weitergegeben. Das Endergebnis enthält die sortierten Bild-Objekte, wobei das erste Ergebnis die Datenanforderung am besten erfüllt (siehe Abbildung 4.17).

5 Evaluation von Damast anhand einer prototypischen Realisierung

Für die qualitative und quantitative Bewertung wird Damast prototypisch implementiert. Auf Grundlage dieser Implementierung werden Experimente durchgeführt, um die generelle Funktionsweise von Damast zu evaluieren. Ferner erfolgt eine systematische Evaluation hinsichtlich des Erfüllungsgrads der Anforderungen an das Datenmanagementsystem (vgl. Unterabschnitt 3.4.1).

5.1 Auswahl der Hardware- und Softwarekomponenten

Hardware

Die Umsetzung erfolgt auf einem dedizierten Computer, welcher mit dem Ubuntu-Linux-Betriebssystem[1] in der Version 20.04 LTS betrieben wird. Der Hauptprozessor des Computers ist ein Intel Core i9-9900K CPU @ 3,60 GHz. Es stehen 16 GB Hauptspeicher zur Verfügung. Als Grafikkarte ist eine GeForce RTX 2080 Ti mit 11 GB Grafikkartenspeicher verbaut.

[1] `https://ubuntu.com/` (abgerufen am 12.11.2024)

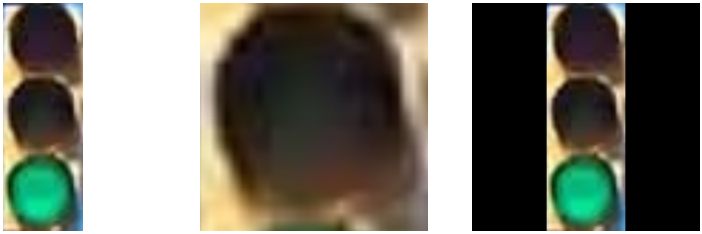
Programmiersprache und Datenbanksysteme

Die Implementierung von Damast ist in der Programmiersprache Python[2] in der Version 3.12.5 realisiert. Als relationales Datenbankmanagementsystem wird PostgreSQL[3] in der Version 15.3 verwendet. Als Erweiterung des Datenbankmanagementsystems wird pgvector[4] in der Version 0.4.4 eingesetzt. Pgvector erweitert PostgreSQL um die Funktionalitäten von Vektordatenbanken und ermöglicht die effiziente Verwaltung von Vektorrepräsentationen (vgl. Unterabschnitt 2.6.1).

CLIP-Implementierung

Als CLIP-Implementierung[5] (vgl. Abschnitt 2.6) wird die von OpenAI trainierte Version ViT-B/32 verwendet [111]. Dieses Modell wurde von OpenAI mit 400 Millionen Bild-Text-Paaren, die aus öffentlich zugänglichen Internetquellen stammen, trainiert. Diese CLIP-Version wurde mit quadratischen Bildern mit einer Kantenlänge von 224 Pixeln trainiert. Die OpenAI-Implementierung skaliert daher jedes Bild, sodass die kürzere Kante die Länge von 224 Pixeln hat. Anschließend wird das Bild mittig ausgeschnitten, sodass es quadratisch ist. Dadurch gehen bei Bildern, die nicht quadratisch sind, an den Rändern Informationen über die Bildinhalte verloren. Damast berechnet jedoch auch für Bilder und Bildausschnitte, die nicht quadratisch sind, die Vektorrepräsentationen. Um in diesen Fällen keine Bildinhalte für die Berechnung der Vektorrepräsentationen zu verlieren, ist die Implementierung von Damast so ausgeführt, dass die Bildränder mit schwarzen Pixeln aufgefüllt werden, sodass die resultierenden Bilder quadratisch sind. Anschließend überträgt Damast die quadratischen Bilder an die Vorverarbeitung der OpenAI-Implementierung. Durch dieses Vorgehen werden die Bilder weder verzerrt noch werden Bildinformationen abgeschnitten. Es entsteht lediglich an den unteren und oberen oder an den seitlichen Rändern ein schwarzer

[2] `https://www.python.org/` (abgerufen am 12.11.2024)
[3] `https://www.postgresql.org/docs/release/15.3/` (abgerufen am 12.11.2024)
[4] `https://github.com/pgvector/pgvector` (abgerufen am 12.11.2024)
[5] `https://github.com/openai/CLIP` (abgerufen am 12.11.2024)



(a) Originalbildausschnitt **(b)** Vorverarbeitung der OpenAI-Implementierung [46] **(c)** Vorverarbeitung in Damast

Abbildung 5.1: Vorverarbeitung eines Bildausschnitts für die Berechnung der Vektorrepräsentation; durch die Vorverarbeitung in Damast stehen zum Zeitpunkt der Vektorrepräsentationsberechnung die Informationen über alle Lichter der Ampel zur Verfügung (Bildquelle [12])

Tabelle 5.1: Objektklassen im Cityscapes-Datensatz; Klassen, die aufgrund ihres geringen Auftretens nicht trainiert werden konnten, sind nicht aufgeführt; Objekte, bei denen Mask2Former Instanzen unterscheidet, sind mit † markiert[6] [112]

Objektgruppe	**Objektklasse**
Flächen	road, sidewalk
Menschen	person†, rider†
Fahrzeuge	car†, truck†, bus†, train†, motorcycle†, bicycle†
Bauten	building, wall, fence
Objekte	pole, traffic sign, traffic light
Natur	vegetation, terrain
Himmel	sky

Rand. Damast beachtet auf diese Weise beispielsweise bei dem Bildausschnitt einer Ampel alle Lichter in der Vektorrepräsentation durch CLIP. Im Falle der reinen OpenAI-Implementierung stünde bei der Vektorrepräsentationsberechnung nur die Information über das mittlere Licht zur Verfügung (siehe Abbildung 5.1).

[6] `https://www.cityscapes-dataset.com/dataset-overview/` (abgerufen am 5.11.2024)

Panoptische Segmentierung

Die panoptische Segmentierung (vgl. Abschnitt 2.5) wird in Damast durch Mask2Former[7] von Facebook realisiert [109]. Konkret wird das panoptische Modell mit einem Swin-L-Backbone [113], der auf dem Datensatz ImageNet-22K [114] vortrainiert wurde, verwendet. Das Finetuning der Mask2Former-Architektur erfolgte durch Facebook mit dem Cityscapes-Datensatz [112]. Mask2-Former ermöglicht im Folgenden die Segmentierung unterschiedlicher Objektklassen (siehe Tabelle 5.1). Bei den Objektgruppen Menschen und Fahrzeuge unterscheidet Mask2Former zusätzlich die Instanzen einer Klasse. Da die panoptische Segmentierung mit dem Cityscapes-Datensatz trainiert wurde, wird dieser Datensatz für die folgende Evaluation nicht verwendet. Auf diese Weise wird eine klare Trennung zwischen Trainings- und Testdaten sichergestellt.

5.2 Verwendete Datensätze

Die Experimente werden auf öffentlich zugänglichen Datensätzen mit realen Kameraaufnahmen durchgeführt. Für die Evaluation von Damast und seiner einzelnen Komponenten müssen die Datensätze bestimmte Eigenschaften aufweisen. Die Auswahl der Datensätze erfolgt daher auf Grundlage folgender Kriterien:

- Für die Evaluation von Damast auf der Gesamtbildebene: Es sind Annotationen, die jeweils den Gesamteindruck des Bildes umfassen, für den Datensatz verfügbar.
- Für die Evaluation von Damast auf der Objektebene: Der Datensatz enthält detaillierte Annotationen von Objekten im Automotive-Kontext.
- Zur Evaluation der Durchsuchbarkeit der Pegasus-Ebenen und zum Zusammenstellen von Teildatensätzen: Der Datensatz besteht aus abwechslungsreichen Bildern mit Variationen innerhalb der Pegasus-Ebenen.

[7] `https://github.com/facebookresearch/Mask2Former/` (abgerufen am 12.11.2024)

Abbildung 5.2: Bild bei Schnee mit zugehörigem Vergleichsbild bei Tag mit klarem Himmel aus dem ACDC-Datensatz [116]

- Zur Evaluation der klassischen Kontexte in Damast: Der Datensatz enthält für jedes Bild initiale Kontexte: geografische Koordinaten und Fahrtrichtung.

Dabei erfüllt keiner der untersuchten Datensätze [7, 8, 18, 115] alle Kriterien. Jedoch sind für die einzelnen Teile der Evaluation jeweils nur ausgewählte Kriterien relevant. Daher werden mehrere Datensätze verwendet, die jeweils die passenden Eigenschaften für den entsprechenden Evaluationsteil besitzen. Die Verwendung mehrerer Datensätze in der Evaluation untersucht außerdem, ob Damast für das Datenmanagement mehrerer Bilddatensätze gleichzeitig geeignet ist.

5.2.1 ACDC-Datensatz

Der ACDC-Datensatz [116] wurde bei vielfältigen Umgebungsbedingungen aufgezeichnet. Er enthält Bilder bei Nebel, Nacht, Regen und Schnee. Für jedes Bild in einem dieser Umgebungszustände existiert außerdem ein Bild aus einer vergleichbaren Perspektive bei Tag mit klarem Himmel (siehe Abbildung 5.2). Diese Vergleichsbilder tragen im ACDC-Datensatz die Bezeichnung „normal“. Der Datensatz umfasst insgesamt 8012 Bilder (siehe Tabelle 5.2) und für jedes dieser Bilder ist die Umgebungsbedingung annotiert.

Tabelle 5.2: Verteilung der Umgebungsbedingungen im ACDC-Datensatzes [116]

Umgebungsbedingung	Gesamtanzahl der Bilder	Anzahl der Bilder im Testteil
normal	4006	2000
Nebel	1000	500
Nacht	1006	500
Regen	1000	500
Schnee	1000	500

5.2.2 BDD100K-Datensatz

Der Fokus des BDD100K-Datensatzes [12] sind abwechslungsreiche Bilder hinsichtlich der Geografie, der Umgebungsbedingungen und des Wetters. Die Daten wurden per Crowdsourcing[8] gewonnen und bestehen aus 100 000 Videos mit einer Länge von jeweils 40 Sekunden. Der BDD100K-Bilddatensatz wurde aus diesen Daten extrahiert, indem jeweils der zehnte Frame jedes Videos als Bild verwendet wird. Die Daten wurden vorwiegend in bevölkerungsreichen Gebieten, wie New York, Berkeley und San Francisco, aufgezeichnet. Wodurch laut der Annotationen des Datensatzes 1 021 857 Fahrzeuge, 343 777 Verkehrszeichen und 129 262 Personen für die Evaluation von Damast zur Verfügung stehen [12]. Im Folgenden wird ausschließlich der extrahierte Bilddatensatz verwendet und mit BDD100K bezeichnet.

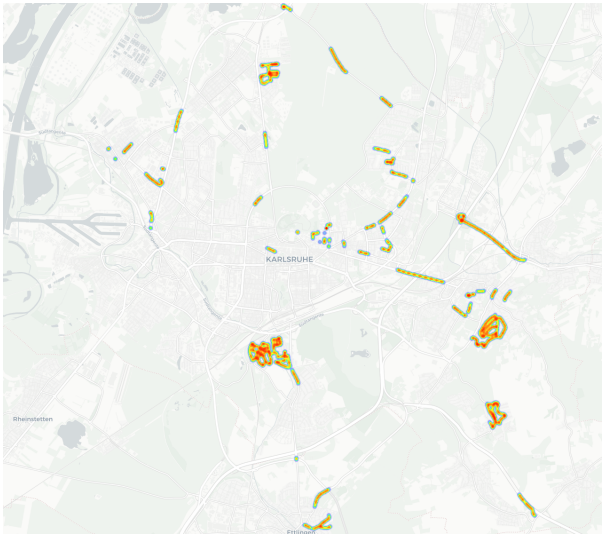


Abbildung 5.3: Heatmap über die geografischen Positionen während der Aufzeichnung des KITTI-Datensatzes [11] (Quelle der Karte: [107])

5.2.3 KITTI-Datensatz

Die Bilder des KITTI-Datensatzes [11] wurden in Karlsruhe (Deutschland) aufgenommen (siehe Abbildung 5.3). Der Datensatz enthält für jeden Frame Informationen über die Uhrzeit, das Datum, die geografische Position und die Fahrtrichtung. Diese Informationen wurden von einem globalen Navigationssatelliten- und einem Trägheitsnavigationssystem gemessen. Dadurch eignet sich der KITTI-Datensatz für die Evaluation der Teile von Damast, die auf geografischen Koordinaten und der Fahrtrichtung basieren.

Abbildung 5.4: Beispielbilder mit der zugehörigen Fahrzeugmodellbezeichnung aus dem Stanford-Cars-Datensatz [117]

5.2.4 Stanford-Cars-Datensatz

Der Stanford-Cars-Datensatz [117] besteht aus 16 185 Bildern einzelner Fahrzeuge. Die Bilder wurden in zwei disjunkte Teildatensätze aufgeteilt, um die Leistungsfähigkeit unterschiedlicher Methoden miteinander vergleichen zu können. Hierbei stehen 8144 Bilder für das Training und 8041 Bilder für das Testen von Methoden bereit. Die Bilder des Datensatzes wurden nicht während der Fahrt aufgezeichnet, sondern umfassen meist stehende Fahrzeuge aus unterschiedlichen Winkeln. Teilweise handelt es sich um Werbebilder, die Texte enthalten. Für die Fahrzeuge existieren jeweils Modellbezeichnungen, bestehend aus der Fahrzeugmarke und der Modellbezeichnung inklusive des Einführungsjahres (siehe Abbildung 5.4). Insgesamt beinhaltet der Datensatz 196 unterschiedliche Fahrzeugmodelle.

[8] Mehr als 10 000 Fahrer stellten Videoaufnahmen ihrer Fahrten zur Erstellung des BDD100K-Datensatzes bereit [12].

5.2.5 Kartendaten

Als Quelle für die Kartendaten verwendet Damast OpenStreetMap [107]. Die Geoinformationen, die das OpenStreetMap-Projekt zur Verfügung stellt, werden durch die Nutzer des Projekts aggregiert. Die Abdeckung, der Detailreichtum und die Qualität der Daten sind daher abhängig von der betrachteten Lokalität. OpenStreetMap stellt die Daten in Form von geografischen Punkten, Linien oder Polygonen zur Verfügung (vgl. Unterabschnitt 2.3.1). Die geografischen Punkte sind mit in Freitext verfassten Schlüssel-Wert-Paaren versehen. Diese Annotationen unterliegen Konventionen[9], die das OpenStreetMap-Projekt festgelegt hat, und geben Aufschluss über den Typ und die Eigenschaften der geografischen Objekte.

5.3 Anwendbarkeit von Damast im Automobilkontext

In Damast sollen Bilder durch eine semantische, textuelle Beschreibung auffindbar sein. Um diese Fähigkeit zu bewerten, wird zunächst evaluiert, inwiefern Damast Bilder aus dem Automotive-Kontext diskriminiert.[10] Für die Verknüpfung zwischen semantischer, textueller Beschreibung und der Semantik der Bilder wird in Damast CLIP verwendet. Für die Evaluation der Diskriminierungsfähigkeit werden der ACDC-Datensatz (vgl. Unterabschnitt 5.2.1) und der Stanford-Cars-Datensatz (vgl. Unterabschnitt 5.2.4) verwendet. Der ACDC-Datensatz enthält Annotationen zu den Umgebungsbedingungen, daher dient er der Evaluation der Anwendbarkeit von Damast auf die Gesamtbildebene im Automobilkontext. Mit dem Stanford-Cars-Datensatz, der Bilder einzelner Fahrzeuge und Annotationen zu diesen enthält, wird die Diskriminationsfähigkeit von Damast in Bezug auf die Objekteigenschaften im Automobilkontext untersucht.

[9] `https://wiki.openstreetmap.org/` (abgerufen am 26.11.2024)
[10] Ähnliche Auswertungen wurden in den Veröffentlichungen [PR4] und [PR5] durchgeführt.

Tabelle 5.3: Klassen im ACDC-Datensatz [116] und englische, textuelle Beschreibung für die Klassifikation

Klasse	Textuelle Beschreibung (englisch)
Nebel	`fog`
Nacht	`night`
Regen	`rain`
Schnee	`snow`
normal	`clear sky`

5.3.1 Gesamtbildebene

Im ersten Evaluationsteil wird die Diskriminationsfähigkeit hinsichtlich unterschiedlicher Bildeigenschaften von Damast auf Gesamtbildebene im Automobilkontext untersucht. Hierfür soll die CLIP-Komponente von Damast (vgl. Abschnitt 2.6) die Test-Bilder des ACDC-Datensatzes hinsichtlich ihrer Beschreibung der Umgebungsbedingungen klassifizieren. Der ACDC-Datensatz enthält insgesamt 5 unterschiedliche Umgebungsbedingungen: Nebel, Nacht, Regen, Schnee und „normal“. Diese Umgebungsbedingungen werden im Folgenden als Klassen aufgefasst. Jeder Klasse wird eine textuelle Beschreibung zugeordnet (vgl. Tabelle 5.3). Da die verwendete CLIP-Implementierung in Damast mit englischen Daten trainiert wurde, sind auch die textuellen Beschreibungen der Klassen auf Englisch. Die Evaluation erfolgt auf den 4000 Bildern $\{I_i\}_{i=1}^{n}$ aus dem Testteil des Datensatzes.

Der CLIP-Text-Encoder $\mathcal{E}_{\text{Text}}$ überführt zunächst die textuellen Klassenbeschreibungen $\mathcal{C} = \{c_1, c_2, \ldots\}$ (vgl. Abschnitt 2.6) in ihre Vektorrepräsentation:

$$\mathcal{E}_{\text{Text}}(c_j) = \vec{e}_{\text{Klasse},j}. \tag{5.1}$$

Anschließend überführt der CLIP-Bild-Encoder $\mathcal{E}_{\text{Bild}}$ jedes Bild des ACDC-Testdatensatzes in seine Vektorrepräsentation:

$$\mathcal{E}_{\text{Bild}}(I_i) = \vec{e}_{\text{Bild},i}. \tag{5.2}$$

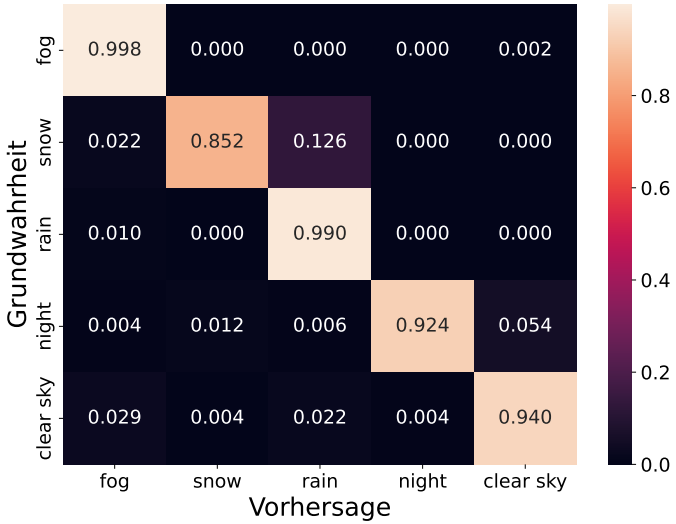


Abbildung 5.5: Zeilenweise normierte Konfusionsmatrix der Klassifikation des ACDC-Datensatzes [117] hinsichtlich der Umgebungsbedingungen

Das Bild I_i wird der Klasse $\hat{c}_i$ zugeordnet, bei der die Kosinus-Ähnlichkeit zwischen der Bild-Vektorrepräsentation und der Klassen-Vektorrepräsentation maximal ist (vgl. Unterabschnitt 4.4.3):

$$\hat{j}_i = \arg\max_j \left(\mathrm{sim}_{\cos}(\vec{e}_{\mathrm{Klasse},j}, \vec{e}_{\mathrm{Bild},i})\right), \tag{5.3}$$

$$\hat{c}_i = c_{\hat{j}_i}. \tag{5.4}$$

Mit diesem Vorgehen wird eine Klassifikationsgenauigkeit von $94{,}1\,\%$ erzielt. Die Bilder der Klasse „Schnee“ werden am häufigsten ($12{,}6\,\%$ der Bilder der Klasse „Schnee“) der falschen Klasse „Regen“ zugeordnet (siehe Abbildung 5.5). Betrachtet man die falsch klassifizierten Bilder, könnte ein Grund für diesen Fehler der geschmolzene Schnee auf der Windschutzscheibe sein, der als Regentropfen interpretiert wurde. Wird das nach der Klassengröße gewichtete F1-Maß berechnet, erhält man einen Wert von 0.94 (siehe Tabelle 5.4). In 3762 von 4000 Fällen

Tabelle 5.4: Ergebnisse der Klassifikation des ACDC-Datensatzes [117] hinsichtlich der Umgebungsbedingungen

Umgebungsbedingung	F1-Maß	Genauigkeit	Sensitivität	Anzahl Instanzen
normal	0.96	0.99	0.94	2000
Nacht	0.95	0.98	0.92	500
Nebel	0.93	0.87	1.00	500
Schnee	0.91	0.97	0.85	500
Regen	0.90	0.82	0.99	500

werden die Bilder hinsichtlich der Umgebungsbedingungen korrekt semantisch diskriminiert.

5.3.2 Objektebene

Mit dem ACDC-Datensatz wird die Anwendbarkeit von Damast auf die Semantik auf Gesamtbildebene untersucht. Damast soll jedoch auch auf der Objektebene semantisch diskriminieren können. Für diese Untersuchung werden die 8041 Testbilder des Stanford-Cars-Datensatzes (vgl. Unterabschnitt 5.2.4) verwendet. Die Annotationen in Form von Fahrzeugmodellbezeichnungen im Datensatz werden im Folgenden als Klassen aufgefasst.

Bei der Klassifikation werden die 8041 Bilder jeweils einer der 196 Fahrzeugmodellbezeichnungen zugeordnet. Dabei wird eine Genauigkeit von $58{,}2\,\%$ erreicht. Das bedeutet, dass Damast über die Hälfte der Fahrzeug-Bilder der korrekten Fahrzeugmodellbezeichnung zuordnet. Allerdings erfolgt bei 3 der 196 Fahrzeugmodellen für jede Instanz eine falsche Zuordnung des Fahrzeugmodells (Auswertung auf Klassenebene: Abschnitt A.3). Das führt zu einem nach Klassengrößen gewichteten Mittel des F1-Maßes von 0.563. Werden zur Klassifizierung nicht nur die Klassen mit der größten Kosinus-Ähnlichkeit in Betracht gezogen, sondern die k ähnlichsten Klassen, so erhöht sich die Genauigkeit (siehe Tabelle 5.5). Die verwendete CLIP-Implementierung schließt bei diesen Automotive-Daten die

Tabelle 5.5: Genauigkeit der Klassifikation des Stanford-Cars-Datensatzes bei Beachtung der k Klassen mit der größten Kosinus-Ähnlichkeit

k	**Genauigkeit**
1	58,2 %
2	74,7 %
3	81,9 %
4	86,7 %
5	89,7 %
6	91,6 %
7	92,8 %
8	94,1 %
9	95,1 %
10	95,9 %

semantische Lücke zwischen den Objekten in Bildern und der textuellen Beschreibung in Form des Fahrzeugmodells.

Der vorgestellte Klassifikationsansatz wird Zero-Shot-Learning [118] genannt, da für die Klassifikation kein Training mit den Daten aus dem Stanford-Cars-Datensatz erfolgte. Modelle, bei denen ein explizites Training mit den Bildern aus dem Stanford-Cars-Datensatz durchgeführt wird, erreichen auf dem Testteil des Datensatzes Genauigkeiten von bis zu 97,3 % [119].

5.4 Durchsuchbarkeit der Pegasus-Ebenen

Anhand einer Nutzerstudie wird die Durchsuchbarkeit der Pegasus-Ebenen (vgl. Unterabschnitt 2.1.3) mit Damast, wie sie in Anforderung A1 (vgl. Unterabschnitt 3.4.1) gefordert ist, evaluiert. Die Nutzer müssen in der Studie mit dem auf Vektorrepräsentationen basierenden Teil von Damast in 17 Aufgaben nach festgelegten Situationen suchen (siehe Tabelle 5.6). Die Situationen decken die gesuchten Pegasus-Ebenen 1 bis 5 ab. Sie orientieren sich am geplanten Einsatzzweck, der Unterstützung bei der Entwicklung von automatisierten Fahrsystemen.

Tabelle 5.6: Aufgaben, deren Formulierungen in der Nutzerstudie und deren entsprechenden Pegasus-Ebenen (vgl. Unterabschnitt 2.1.3)

Aufgabentitel	**Aufgabenformulierung**	**Pegasus-Ebene**
Pflastersteine	*Finde ein Bild, auf dem eine Straße mit Pflastersteinen zu sehen ist.*	E1
Zebrastreifen	*Finde ein Bild, auf dem ein Zebrastreifen zu sehen ist.*	E1
Busspur	*Finde ein Bild, auf dem eine Busspur zu sehen ist.*	E1
Bremsschwelle	*Finde ein Bild, auf dem eine Bremsschwelle (künstliche Erhebung in der Straße zur Begrenzung der Geschwindigkeit) zu sehen ist.*	E2
Kreisverkehr	*Finde ein Bild, auf dem ein Kreisverkehr zu sehen ist.*	E2
Palme	*Finde ein Bild, auf dem eine Palme (Palmengewächs) zu sehen ist.*	E2
temporäre Fahrbahnmarkierung	*Finde ein Bild, auf dem eine temporäre Fahrbahnmarkierung zu sehen ist.*	E3
Verkehrsleitkegel	*Finde ein Bild, auf dem ein Verkehrsleitkegel zu sehen ist.*	E3
Baustellenschild	*Finde ein Bild, auf dem ein Baustellenschild (Verkehrszeichen, das vor einer Baustelle warnt) zu sehen ist.*	E3
Straßenbauer	*Finde ein Bild, auf dem ein Straßenbauer (Person, die Straßen instand hält) zu sehen ist.*	E4
Baumaschine	*Finde ein Bild, auf dem eine Baumaschine zu sehen ist.*	E4

Aufgabentitel	**Aufgabenformulierung**	**Pegasus-Ebene**
abgelenkte Person	*Finde ein Bild, auf dem eine abgelenkte Person zu sehen ist.*	E4
Fahrzeug mit Werbung	*Finde ein Bild, auf dem ein Fahrzeug mit Werbung zu sehen ist.*	E4
Verkehrsstau	*Finde ein Bild, auf dem ein Verkehrsstau zu sehen ist.*	E4
Regentropfen	*Finde ein Bild, auf dem Regentropfen auf der Windschutzscheibe zu sehen sind.*	E5
Schnee	*Finde ein Bild, auf dem Schnee zu sehen ist.*	E5
blendende Sonne	*Finde ein Bild, auf dem die Sonne blendet.*	E5

Damast durchsucht die Datensätze ACDC (vgl. Unterabschnitt 5.2.1), BDD100K (vgl. Unterabschnitt 5.2.2) und KITTI (vgl. Unterabschnitt 5.2.3). Insgesamt stehen auf diese Weise 176 581 Bilder aus Datensätzen mit abwechslungsreichen Orten, Wettersituationen und Objekten zur Verfügung. Durch die kombinierte Suche in drei Datensätzen wird darüber hinaus untersucht, ob Damast unterschiedliche Datensätze gleichzeitig verwalten kann.

Der Nutzer führt seine Suche mit Damast in einer Streamlit-basierten[11] Benutzeroberfläche durch (siehe Abbildung 5.6). Auf dieser wird dem Nutzer die jeweils aktuelle Aufgabe angezeigt und er parametrisiert die Suchanfrage. Hierfür formuliert er in natürlicher Sprache seine Suchanfrage. Außerdem wählt der Nutzer, ob er für die Suche das Gesamtbild oder nur eine bestimmte Objektklasse berücksichtigen will. Anschließend präsentiert Damast dem Nutzer die 10 passendsten Ergebnisse der Suche (siehe Abbildung 5.7). Der Nutzer entscheidet dann, ob eines der 10 Bilder die aktuelle Suchanfrage erfüllt oder keines der Bilder passend

[11] `https://github.com/streamlit/streamlit` (abgerufen am 12.11.2024)

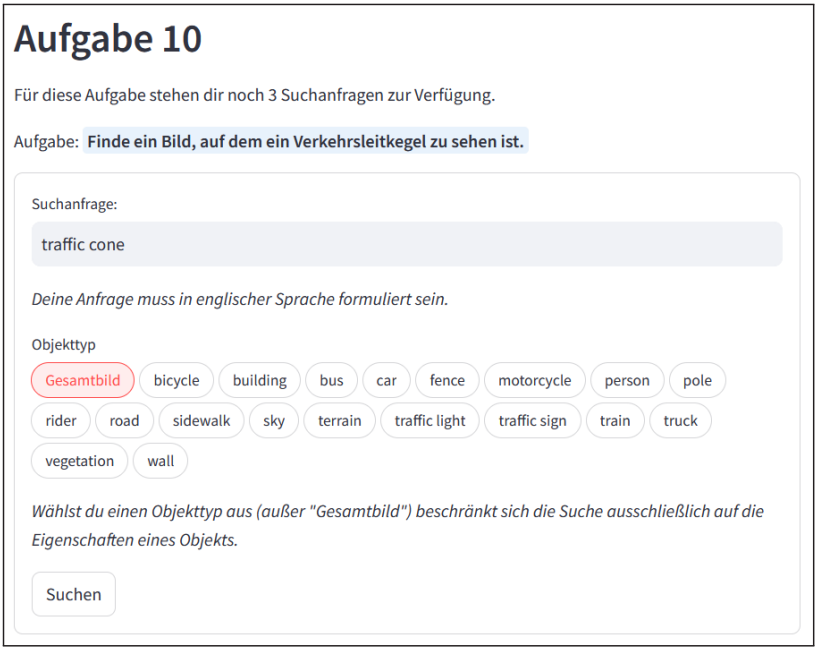


Abbildung 5.6: Benutzeroberfläche für die Damast-Nutzerstudie am Beispiel der Aufgabe „Verkehrsleitkegel“: Aufgabenstellung und Parametrisierung einer Suchanfrage

ist. Der Nutzer soll dabei das erste Bild auswählen, das ihm für die Aufgabe passend erscheint. Wird vom Nutzer keines der Bilder als passend bewertet, startet eine neue Suchanfrage, die vom Nutzer parametrisiert wird. Für eine Aufgabe stehen jedem Nutzer 3 Suchanfragen zur Verfügung. Führt keine der 3 Suchanfragen zum Erfolg, wird die Aufgabe als „nicht gefunden“ markiert. Um die Nutzer mit Damast und der Benutzeroberfläche vertraut zu machen, startet die Nutzerstudie mit zwei Übungsaufgaben, die die Parametrisierung von Beispielsuchanfragen vorgeben.

Für die folgende Auswertung speichert das Framework der Nutzerstudie die Interaktionen der Nutzer. Die gesammelten Informationen umfassen die Parametrisierung der Suchanfragen, ausgewählte Bilder und Zeitstempel. Es nahmen 15 Personen an der Nutzerstudie teil. Insgesamt wurden 255 Aufgaben gestellt. Die

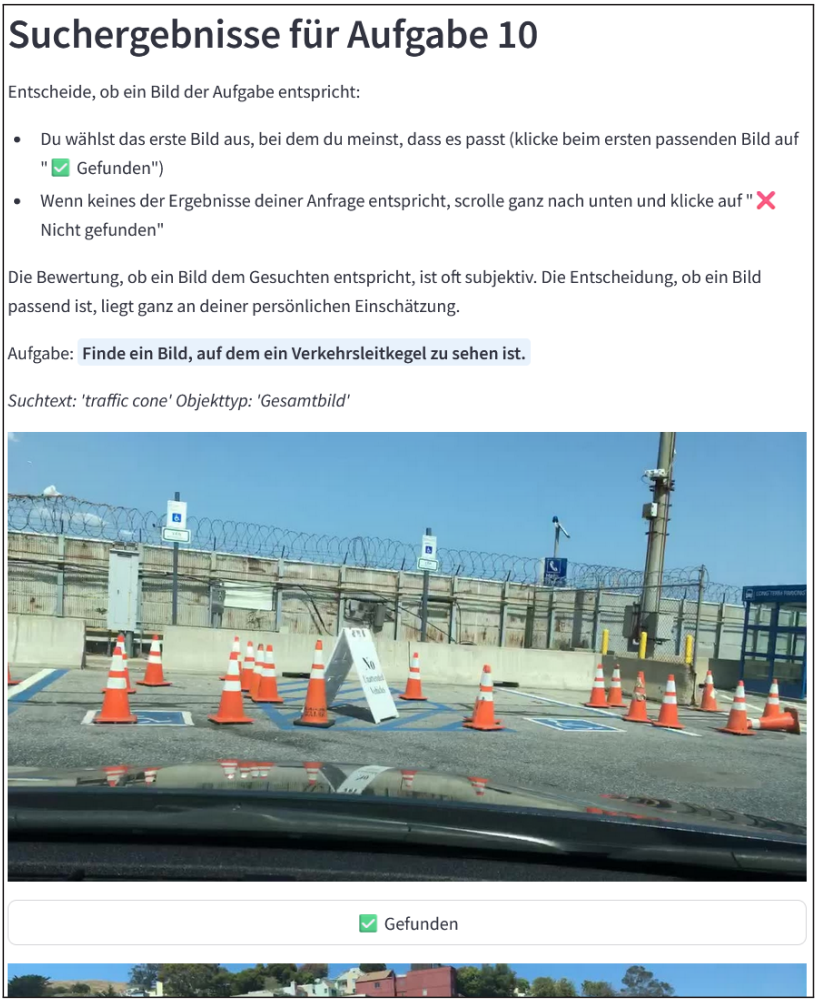


Abbildung 5.7: Benutzeroberfläche für die Damast-Nutzerstudie am Beispiel der Aufgabe „Verkehrsleitkegel“: Ausgabe und Bewertung der Ergebnisse einer Suchanfrage (hier nur 1 von 10 Ergebnisbildern dargestellt)

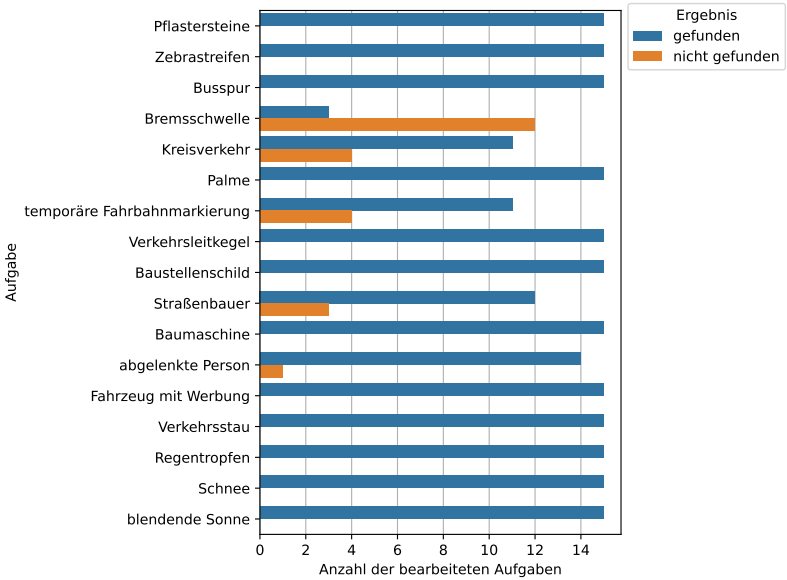


Abbildung 5.8: Übersicht über die Aufgaben und Ergebnisse der Nutzerstudie

Nutzer fanden in 231 Fällen (90,6 %) ein passendes Bild zur Aufgabe (verwendete Suchbegriffe und Objektklassen: Abschnitt A.4). Bei 12 Aufgaben wurde von jedem Teilnehmer ein passendes Bild gefunden (siehe Abbildung 5.8). Lediglich bei der Aufgabe „Bremsschwelle“ fand die Mehrheit der Nutzer (12 Teilnehmer) kein passendes Bild.

Durchschnittlich führten die Nutzer für das erfolgreiche Auffinden eines Bildes 1,15 Suchanfragen durch (siehe Abbildung 5.9). Bei 4 Aufgaben musste kein Nutzer mehr als eine Suchanfrage durchführen, um ein passendes Bild zu finden. Die Bearbeitungszeit für eine Aufgabe lag im Durchschnitt bei 34,4 s (siehe Abbildung 5.10). Hier fallen hauptsächlich die erfolglosen Suchen, mit einer Durchschnittsbearbeitungszeit von 95,2 s, ins Gewicht. In den 90,6 % der erfolgreichen Suchen betrug die Durchschnittsbearbeitungszeit 27,7 s.

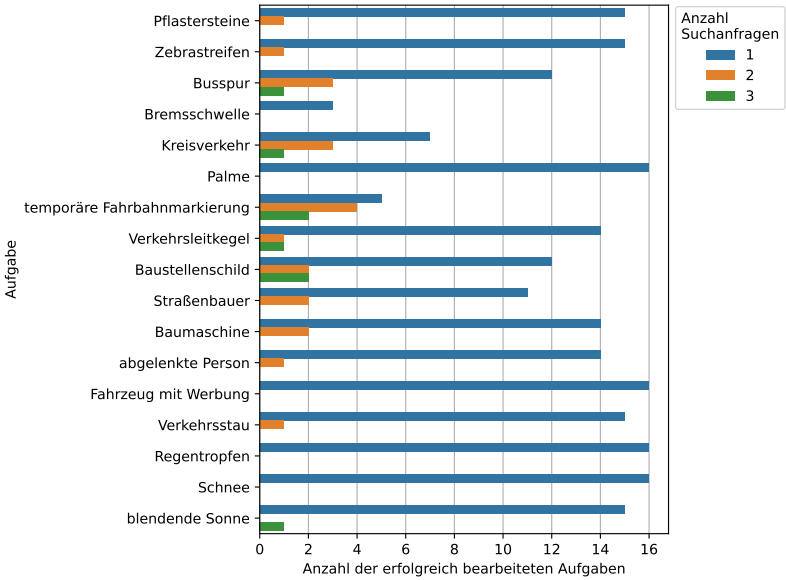


Abbildung 5.9: Anzahl der Suchanfragen je erfolgreich bearbeiteter Aufgabe

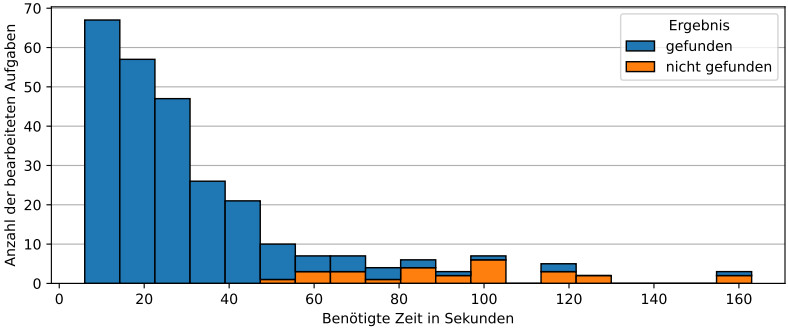


Abbildung 5.10: Bearbeitungszeit pro Aufgabe in der Nutzerstudie

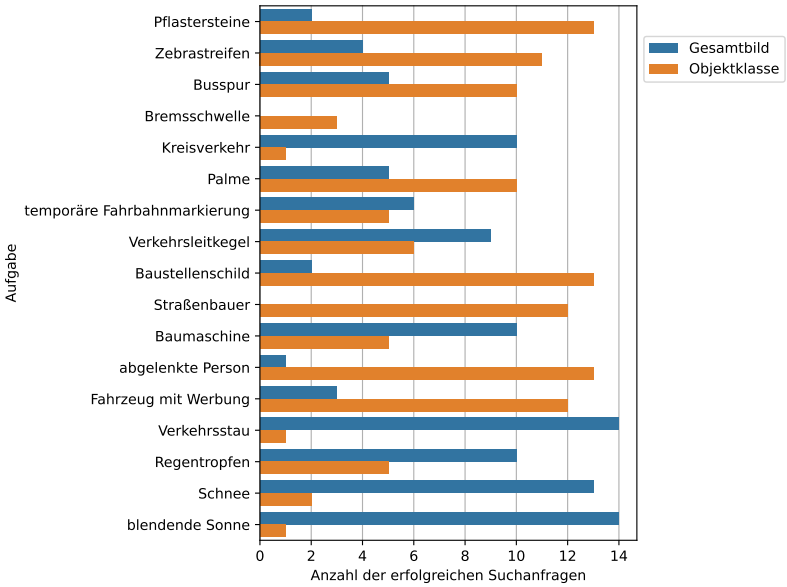


Abbildung 5.11: Parametrisierung der erfolgreichen Suchanfragen: Suche auf dem Gesamtbild oder der Objektebene

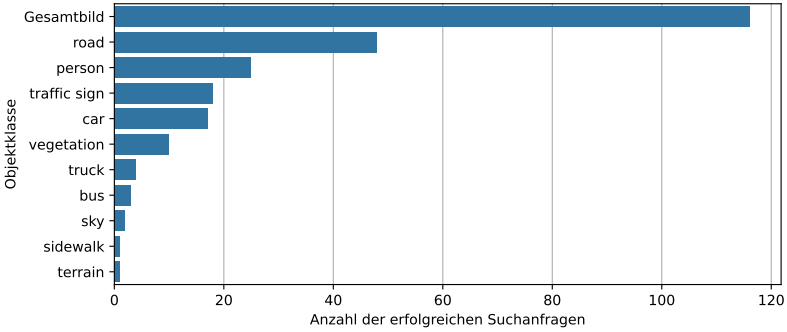


Abbildung 5.12: Übersicht über die Auswahl der Objektklassen bei erfolgreichen Suchanfragen

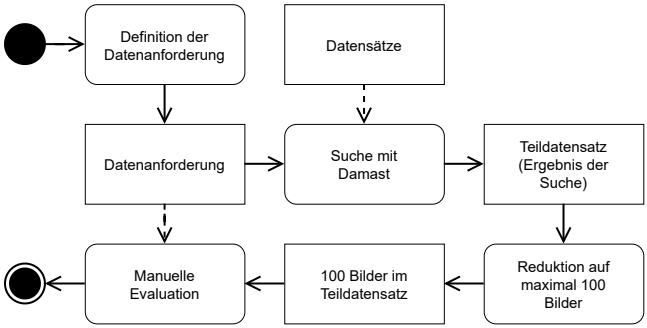


Abbildung 5.13: Vorgehen zur Evaluation der Fähigkeit von Damast Teildatensätze zusammenzustellen

Bei der Parametrisierung der Suchanfragen entschied der Nutzer, ob Damast auf dem Gesamtbild oder auf der Objektebene suchen soll (vgl. Unterabschnitt 4.4.3). In $52{,}7\,\%$ der erfolgreichen Suchanfragen nutzen die Probanden die Möglichkeit zur Suche auf der Objektebene (siehe Abbildung 5.11). Die Ausnahme bildete die Pegasus-Ebene E5 (Aufgaben: „Regentropfen“, „Schnee“, „blendende Sonne“). Die von dieser Ebene abgedeckten Umgebungsbedingungen betreffen in den meisten Fällen die Erscheinung des gesamten Bildes und daher suchten die Nutzer in diesen Fällen häufiger auf dem Gesamtbild als auf der Objektebene des Bildes. Am häufigsten wählten die Nutzer bei erfolgreichen Suchanfragen die Objektklasse „road“ aus (sieht Abbildung 5.12). Diese Auswahl wurde vor allem auf Pegasus-Ebene E1 getroffen, da sich diese auf die Straßenebene bezieht (vgl. Abschnitt A.4). Insgesamt konnten die Nutzer die drei Datensätze mittels des auf Vektorrepräsentationen basierenden Teils von Damast bezüglich der Pegasus-Ebenen durchsuchen.

5.5 Zusammenstellen von Teildatensätzen

Das Zusammenstellen von Teildatensätzen anhand von Datenanforderungen für die Entwicklung von Fahrsystemen ist die zentrale Aufgabe von Damast. Zur Evaluation dieser Fähigkeit (siehe Abbildung 5.13) werden zunächst rudimentäre

Tabelle 5.7: Datenanforderungen zur Evaluation des auf Vektorrepräsentationen basierenden Teils von Damast hinsichtlich des Auffindens mehrerer Bilder in den Pegasus-Ebenen (vgl. Unterabschnitt 2.1.3) und die zugehörigen Parametrisierungen der Damast-Suche

Datenanforderung	Suchbegriff	Objektklasse	Pegasus-Ebene
Zebrastreifen	zebra crossing	road	E1
Stoppschild	stop	traffic sign	E2
Verkehrsleitkegel	traffic cone	Gesamtbild	E3
Personenkraftwagen der Marke Mercedes-Benz	mercedes	car	E4
Person mit rotem Oberteil	red top	person	E4
Polizist	police man	person	E4
verschneiter Personenkraftwagen	snow	car	E4
Stretch-Limousine	stretch limousine	car	E4
überbelichtetes Bild	overexposed	Gesamtbild	E5

Datenanforderungen in Form von Schlagworten definiert, die für die Entwicklung von automatisierten Fahrsystemen von Interesse sind. Anschließend werden mit Damast die entsprechenden Suchanfragen durchgeführt, um die Datenanforderungen zu erfüllen. Die Ergebnisse der Suchen werden manuell mit den zugehörigen Datenanforderungen abgeglichen. Wenn ein Bild die Datenanforderung erfüllt, wird es als passend markiert. Liefert Damast mehr als 100 Bilder als Ergebnis zurück, werden ausschließlich die ersten 100 Bilder betrachtet.

Vektorrepräsentation

Bei der auf Vektorrepräsentationen basierenden Suche werden die Datenanforderungen so definiert, dass die Pegasus-Ebenen E1 bis E5 (vgl. Unterabschnitt 2.1.3)

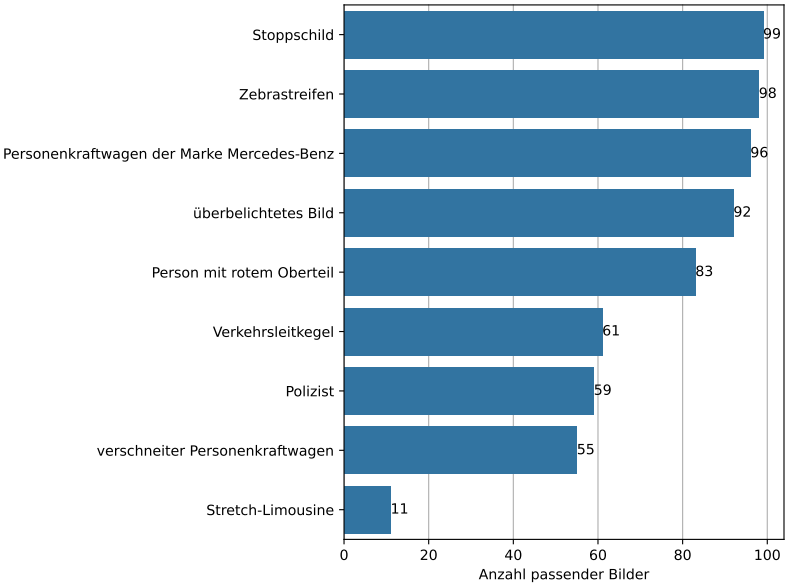


Abbildung 5.14: Anzahl der zur jeweiligen Datenanforderung passenden Ergebnisse, welche mit dem auf Vektorrepräsentationen basierenden Teil von Damast gefunden wurden

abgedeckt sind[12]. Der Fokus liegt jedoch auf den beweglichen Objekten in Ebene 4 (siehe Tabelle 5.7), da diese von keiner anderen Suchmethode erfasst werden. Durchsucht werden, wie im vorherigen Abschnitt, die Datensätze ACDC (vgl. Unterabschnitt 5.2.1), BDD100K (vgl. Unterabschnitt 5.2.2) und KITTI (vgl. Unterabschnitt 5.2.3) mit insgesamt 176 581 Bildern. Die Suchanfragen werden entsprechend den Datenanforderungen parametrisiert (siehe Tabelle 5.7).

Die Evaluation zeigt, dass die Anzahl der zu einer Datenanforderung passenden Ergebnisse in den ersten 100 Bildern von der jeweiligen Datenanforderung abhängt (siehe Abbildung 5.14). Fast alle ($\geq$ 92 %) der evaluierten Ergebnisse zu den Datenanforderungen „Stoppschild“, „Zebrastreifen“, „Personenkraftwagen

[12] Eine ähnliche Auswertung wurde in den Veröffentlichungen [PR5] durchgeführt.

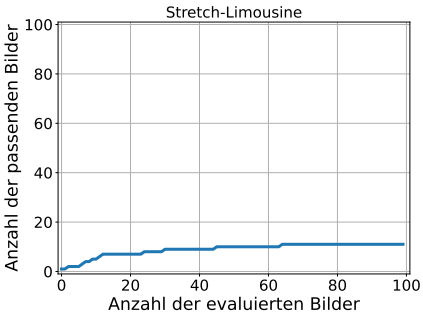


Abbildung 5.15: Kumulierte Verteilung der zur Datenanfrage „Stretch-Limousine“ (vgl. Tabelle 5.7) passenden Bilder in den ersten 100 Ergebnissen des auf Vektorrepräsentationen basierenden Teils von Damast

der Marke Mercedes-Benz“ und „überbelichtetes Bild“ entsprechen den Datenanforderungen. Im Gegensatz dazu sind nur 11 der 100 evaluierten Bilder zur Suchanfrage für Stretch-Limousinen tatsächlich Stretch-Limousinen. Der Grund hierfür ist, dass entweder nicht mehr Bilder in den Datensätzen mit Stretch-Limousinen enthalten sind oder Damast nicht in der Lage ist, diese zu identifizieren. Der Großteil der gefundenen Stretch-Limousinen ist in den ersten Ergebnissen von Damast enthalten (siehe Abbildung 5.15). Ein Nutzer, der einen Datensatz mit Bildern von Stretch-Limousinen zusammenstellen möchte, findet in den ersten 50 Ergebnissen von Damast 10 Bilder mit dem Gesuchten. Für alle 9 Datenanfragen sind in den ersten 100 Damast-Ergebnissen passende Bilder enthalten (Verteilung der passenden Bilder in den Ergebnissen: Abschnitt A.5).

Geografische Daten

Der Datensatz KITTI (vgl. Unterabschnitt 5.2.3) enthält für jeden Frame Informationen über die Uhrzeit, das Datum, die geografische Position und die Fahrtrichtung. Daher wird anhand dieses Datensatzes die Fähigkeit von Damast zur Zusammenstellung von Teildatensätzen, die auf den geografischen Daten basieren, evaluiert. Die Datenanforderungen bei der Evaluation von Damast basierend auf

Tabelle 5.8: Datenanforderungen zur Evaluation des auf geografischen Daten basierenden Teils von Damast hinsichtlich des Auffindens mehrerer Bilder in den Pegasus-Ebenen [17] und die zugehörigen Parametrisierungen der Damast-Suchen für geografische Daten anhand von Openstreetmap-Schlüssel-Wert-Paaren (OSM-Schlüssel und OSM-Wert)

Datenanforderung	OSM-Schlüssel	OSM-Wert	Pegasus-Ebene
Straßenbahngleise	railway	tram	E1
Zebrastreifen	crossing_ref	zebra	E1
Bahnübergang	railway	level_crossing	E1
Autobahn	highway	motorway	E1
Ampel	highway	traffic_signals	E2
Verkehrsinsel	area:highway	traffic_island	E2
Brücke	man_made	bridge	E2
Bushaltestelle	highway	bus_stop	E2

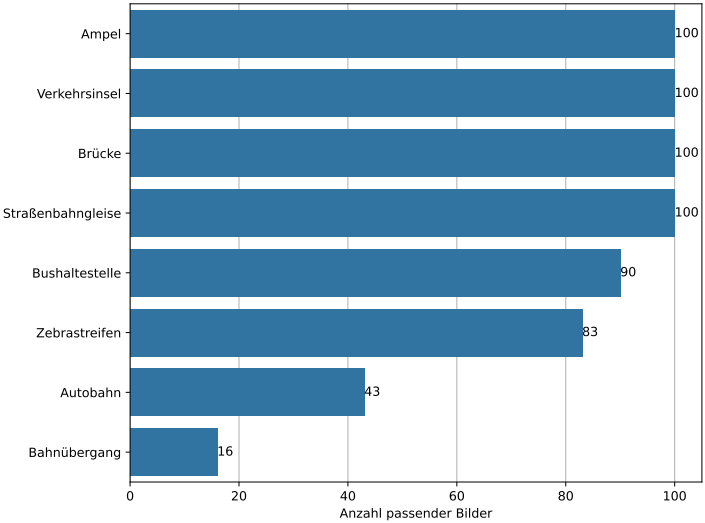


Abbildung 5.16: Anzahl der zur jeweiligen Datenanforderung passenden Ergebnisse, welche in den geografischen Daten mit Damast gefunden wurden

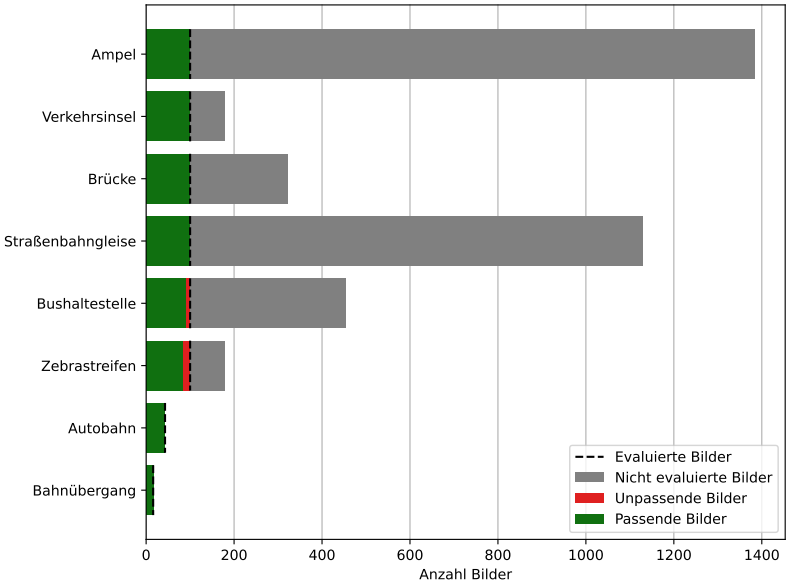


Abbildung 5.17: Gesamtzahl der mit Damast in den geografischen Daten gefundenen Bilder

geografischen Daten beziehen sich auf die ersten beiden Pegasus-Ebenen, da geografische Daten für diese beiden Ebenen Informationen bereitstellen (siehe Tabelle 5.8). Für alle Anfragen werden ein Öffnungswinkel von $30°$ und ein Radius von $30\,\mathrm{m}$ verwendet (siehe Unterabschnitt 4.2.2). Die Datenanforderungen werden in entsprechende Schlüssel-Wert-Paare nach den Openstreetmap-Konventionen (vgl. Unterabschnitt 5.2.5) übersetzt (siehe Tabelle 5.8).

Damast gibt für jede dieser Parametrisierungen alle entsprechenden Bilder zurück. Die Reihenfolge, in der Damast die Bilder zurückgibt, trägt in diesem Fall keine Information. Für die jeweils ersten 100 Bilder wird überprüft, ob sie die jeweilige Datenanfrage erfüllen. Im Falle der Datenanforderungen „Ampel“, „Verkehrsinsel“, „Brücke“ und „Straßenbahngleise“ entsprechen alle 100 Bilder der Anfrage (siehe Abbildung 5.16). Bei den Datenanforderungen „Bushaltestelle“ sind 90 und bei „Zebrastreifen“ sind 83 Bilder der Ergebnisse von Damast passend. Nur bei

den Datenanforderungen „Autobahn“ und „Bahnübergang“ werden von Damast weniger als 100 Bilder identifiziert, jedoch entsprechen alle den jeweiligen Datenanforderungen (siehe Abbildung 5.17). Bei allen anderen Datenanforderungen werden durch Damast mehr als 100 Bilder gefunden, weshalb potenziell noch mehr Bilder für die Datenanforderung zur Verfügung stehen würden. Ob diese Bilder tatsächlich zu den Datenanforderungen passen, wird jedoch nicht evaluiert.

Sonnenstand

Untersucht wird die Funktionalität von Damast zum Zusammenstellen von Teildatensätzen bezüglich des relativen Sonnenstands (vgl. Unterabschnitt 4.2.1). Hierzu wird ebenfalls der KITTI-Datensatz (vgl. Unterabschnitt 5.2.3) verwendet. Neben der Verfügbarkeit der geografischen Position und der Fahrtrichtung für jeden Frame hat dieser Datensatz den Vorteil, dass er hauptsächlich aus Bildern besteht, bei denen das Wetter die Evaluation des relativen Sonnenstands anhand der im Bild sichtbaren Sonne oder Schattenwürfe erlaubt. Ferner sind im Datensatz vielfältige relative Sonnenstände enthalten (siehe Abbildung 5.18).

Die durch den Sonnenstand ermittelte Lichtsituation bezieht sich auf Pegasus-Ebene 5. In allen Fällen werden ein relativer Höhenwinkel $\omega = 20°$ und eine maximale Winkelabweichung $\Xi = 10°$ angefragt. Da eine exakte Bestimmung des relativen Sonnenstands aus den Bildern nicht möglich ist, wird für die Evaluation nach 4 markanten relativen Azimuten ψ gesucht: Sonne von links $\psi = -90°$, Sonne von vorn $\psi = 0°$, Sonne von rechts $\psi = 90°$ und Sonne von hinten $\psi = 180°$. Von insgesamt 400 evaluierten Bildern ist der Sonnenstand bei 3 nicht bestimmbar, da das Aufnahmefahrzeug in einer Garage steht. In allen anderen Fällen ist der Sonnenstand evaluierbar und entspricht der angefragten Situation (siehe Abbildung 5.19). Für jeden angefragten Sonnenstand werden zwar jeweils nur 100 Ergebnisbilder evaluiert, jedoch findet Damast für jede Anfrage mehr als 350 Bilder im KITTI-Datensatz.

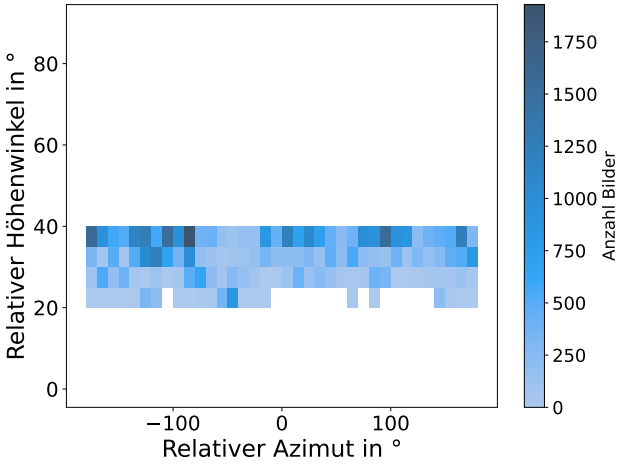


Abbildung 5.18: Relative Sonnenstände im KITTI-Datensatz

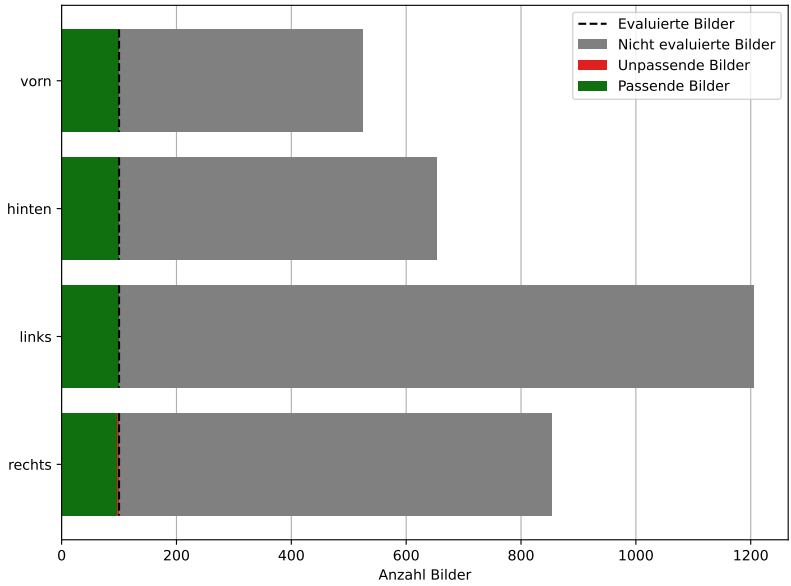


Abbildung 5.19: Anzahl der von Damast gefundenen Sonnenstände und Evaluationsergebnis

5.6 Kombination von Suchmethoden

Nach separaten Experimenten mit den Suchmethoden von Damast wird zuletzt die Kombination von Suchmethoden exemplarisch evaluiert. Wenn die Suche Abfragen von geografischen Daten oder dem Sonnenstand enthält, wird der KITTI-Datensatz verwendet (vgl. Unterabschnitt 5.2.3). Bei den anderen Suchen werden zusätzlich der ACDC- (vgl. Unterabschnitt 5.2.1) und BDD100K-Datensatz (vgl. Unterabschnitt 5.2.2) durchsucht. Es wird jeweils nur das erste Ergebnis von Damast evaluiert. Die exemplarischen Suchen und Ergebnisse sind im Folgenden dargestellt.

Unterbelichtetes Fahrzeug durch Sonne von vorn

Gesucht wird ein Bild, bei dem ein Fahrzeug aufgrund der blendenden Sonne unterbelichtet ist. Die Damast-Suche wird wie folgt parametrisiert:

- Abfragemethode: Sonnenstand (Sonne von vorn)
  - Höhenwinkel $\omega = 20°$
  - Relativer Azimut $\psi = 0°$
  - Maximale Winkelabweichung $\Xi = 10°$
- Abfragemethode: Vektorrepräsentation (unterbelichtetes Fahrzeug)
  - Objektklasse: car
  - Suchbegriff: underexposed

Das erste Ergebnis von Damast bei dieser Suche ist ein Bild, bei dem die Sonne von vorn in die Kamera scheint, der Himmel überbelichtet ist und Fahrzeuge unter einer Brücke in der Folge unterbelichtet sind (siehe Abbildung 5.20).

Abbildung 5.20: Erstes Damast-Ergebnis für die Suche nach einem unterbelichteten Fahrzeug durch Sonne von vorn (Bilddatensatz: [11])

Angestrahlte und überbelichtete Hausfassade

Es wird ein Bild gesucht, bei welchem eine Hausfassade durch Sonnenlicht von hinten überbelichtet ist. Die Damast-Suche wird wie folgt parametrisiert:

- Abfragemethode: Sonnenstand (Sonne von hinten)
  - Höhenwinkel $\omega = 20°$
  - Relativer Azimut $\psi = 180°$
  - Maximale Winkelabweichung $\Xi = 10°$
- Abfragemethode: Vektorrepräsentation (überbelichtete Hausfassade)
  - Objektklasse: building
  - Suchbegriff: dazzle

Das erste Ergebnis von Damast ist ein Bild, bei dem die Sonne von hinten auf eine Hausfassade scheint und diese daher überbelichtet ist (siehe Abbildung 5.21).

Abbildung 5.21: Erstes Damast-Ergebnis für die Suche einer überbelichteten Hausfassade durch Sonnenlicht von hinten (Bilddatensatz: [11])

Bremsendes Fahrzeug vor Zebrastreifen

Es wird ein Bild gesucht, bei welchem ein Fahrzeug vor einem Zebrastreifen bremst. Die Damast-Suche wird wie folgt parametrisiert:

- Abfragemethode: Geografische Daten (Zebrastreifen)
  - OSM-Schlüssel: crossing_ref
  - OSM-Wert: zebra
- Abfragemethode: Vektorrepräsentation (bremsendes Fahrzeug)
  - Objektklasse: car
  - Suchbegriff: breaking car

Das erste Ergebnis von Damast bei dieser Suche ist ein Bild, auf welchem ein Fahrzeug vor einem Zebrastreifen bremst (siehe Abbildung 5.22).

Abbildung 5.22: Erstes Damast-Ergebnis für die Suche nach einem bremsenden Fahrzeug vor einem Zebrastreifen (Bilddatensatz: [11])

Kind an einer Bushaltestelle

Es wird ein Bild gesucht, auf welchem ein Kind an einer Bushaltestelle steht. Die Damast-Suche wird wie folgt parametrisiert:

- Abfragemethode: Geografische Daten (Bushaltestelle)
  - OSM-Schlüssel: highway
  - OSM-Wert: bus_stop
- Abfragemethode: Vektorrepräsentation (Kind)
  - Objektklasse: person
  - Suchbegriff: child

Das erste Ergebnis von Damast bei dieser Suche zeigt eine Bushaltestelle und ein Kind, das neben dem Unterstand der Bushaltestelle steht (siehe Abbildung 5.23).

Abbildung 5.23: Erstes Damast-Ergebnis für die Suche nach einem Kind an einer Bushaltestelle (Bilddatensatz: [11])

Schulbus in der rechten Bildhälfte

Es wird ein Bild gesucht, bei dem auf der rechten Bildhälfte ein Schulbus sichtbar ist. Der Schulbus soll nah am aufzeichnenden Fahrzeug sein. Die Damast-Suche wird wie folgt parametrisiert:

- Abfragemethode: Vektorrepräsentation (Schulbus)
  - Objektklasse: bus
  - Suchbegriff: schoolbus
- Abfragemethode: Objektposition (Schulbus rechts im Bild)
  - Linke Kante des Begrenzungsrechtecks des Schulbusses auf der rechten Seite des Bildes
- Abfragemethode: Objektgröße (Schulbus nahe am aufzeichnenden Fahrzeug)
  - Fläche des Begrenzungsrechtecks des Schulbusses ist größer als $30\,\%$ der Bildfläche

Das erste Ergebnis von Damast ist ein Bild, auf welchem das aufzeichnende Fahrzeug links an einem Schulbus vorbeifährt. Das aufzeichnende Fahrzeug befindet sich dabei direkt neben dem Schulbus (siehe Abbildung 5.24).

Abbildung 5.24: Erstes Damast-Ergebnis für die Suche nach einem nahen Schulbus in der rechten Bildhälfte (Bilddatensatz: [12])

Verkehrspolizist vor dem aufzeichnenden Fahrzeug

Es wird ein Bild gesucht, bei dem ein Verkehrspolizist vor dem aufzeichnenden Fahrzeug zu sehen ist. Die Damast-Suche wird wie folgt parametrisiert:

- Abfragemethode: Vektorrepräsentation (Verkehrspolizist)
  - Objektklasse: person
  - Suchbegriff: traffic policeman
- Abfragemethode: Objektposition (Verkehrspolizist vor dem aufzeichnenden Fahrzeug)
  - Die linke Kante des Begrenzungsrechtecks des Verkehrspolizisten ist rechts vom linken Drittel des Bildes.
  - Die rechte Kante des Begrenzungsrechtecks des Verkehrspolizisten ist links vom rechten Drittel des Bildes.

Das erste Ergebnis von Damast ist ein Bild, auf welchem ein Verkehrspolizist vor dem aufzeichnenden Fahrzeug auf der Straße steht (siehe Abbildung 5.25).

Abbildung 5.25: Erstes Damast-Ergebnis für die Suche nach einem Verkehrspolizisten vor dem aufzeichnenden Fahrzeug (Bilddatensatz: [12])

Durch die Kombination der Suchmethoden wird der Nutzer in die Lage versetzt, Suchanfragen, die sich auf mehrere Pegasus-Ebenen gleichzeitig beziehen, umzusetzen. Die exemplarische Kombination der Suchmethoden von Damast findet in jedem der evaluierten Fälle mit dem ersten Ergebnis das Gesuchte in der Bilddatenbank.

5.7 Laufzeiten von Damast

Anforderung A2 (vgl. Unterabschnitt 3.4.1) fordert von Damast eine interaktive Nutzbarkeit im Entwicklungsprozess. Diese ist nur gegeben, wenn Damast entsprechende Laufzeiten erreicht. Die Implementierung von Damast erfolgt ohne Parallelisierung, und beim Aufruf von künstlichen neuronalen Netzen wird eine Batch-Größe von 1 gewählt. In Bezug auf die Implementierung dienen die folgenden Messwerte daher mit Blick auf diese Optimierungsoptionen als obere Schranke. Die Laufzeitmessungen erfolgen in der bereits beschriebenen Hard- und Software-Umgebung (vgl. Abschnitt 5.1).

Tabelle 5.9: Gemessene Laufzeiten pro Bild, jeweils gemittelt über die Anreicherung von 1000 Bildern oder Datenabfragen

	Anreicherungsschritt	**Datenabfrage**
geografische Daten	0,001 s	0,119 s
Sonnenstand	0,001 s	0,022 s
Vektorrepräsentationen	2,05 s	0,35 s

5.7.1 Laufzeiten bei klassischen Kontexten

Die Laufzeiten für die klassischen Kontexte werden separat für den Anreicherungsschritt und die Datenabfrage bestimmt (siehe Tabelle 5.9). Zur Bestimmung der Laufzeit des Anreicherungsschritts werden die Kontexte jeweils zu 1000 Bildern hinzugefügt und die Dauer anhand der Anzahl der Bilder gemittelt.

Bei den geografischen Kontextdaten (vgl. Unterabschnitt 4.2.2) wird der Anreicherungsschritt so parametrisiert, dass Ampeln in einem Abstand von 30 m und mit einem Öffnungswinkel von 30° sichtbar sind. Die Anreicherung mit den geografischen Daten erfolgt im Durchschnitt in 0,001 s pro Bild. Die Sonnenstände (vgl. Unterabschnitt 4.2.1) werden ebenfalls im Durchschnitt in 0,001 s pro Bild in Damast abgelegt.

Für die Laufzeitmessung bei der Datenabfrage werden zunächst 47 962 Bilder inklusive entsprechender Kontexte in Damast abgelegt. Es werden 1000 Anfragen durchgeführt und das Ergebnis gemittelt. Im Durchschnitt dauert eine Anfrage der geografischen Kontextdaten 0,119 s und die Anfragen des Sonnenstands 0,022 s.

5.7.2 Laufzeiten bei Kontexten basierend auf Vektorrepräsentationen

Auch für die Laufzeituntersuchung der auf Vektorrepräsentationen basierenden Kontexte (vgl. Abschnitt 4.3) werden die Vorverarbeitung und die anschließende

Datenabfrage separat betrachtet (siehe Tabelle 5.9). Zur Vorverarbeitung werden hier die panoptische Segmentierung, die Berechnung der Vektorrepräsentation des Gesamtbildes und die Berechnungen der Vektorrepräsentationen aller Objekte im Bild gezählt. Die Laufzeit der Vorverarbeitung[13] wird über 1000 Bilder gemittelt und beträgt im Mittel pro Bild 2,05 s. Für die Bestimmung der Datenanfrage-Laufzeit werden zunächst 100 000 Bilder in Damast vorverarbeitet. Anschließend werden 1000 zufällige Anfragen an Damast gestellt. Die Anfragen werden im Durchschnitt in 0,35 s beantwortet.

[13] Diese Laufzeitmessung wurde bereits in der Veröffentlichung [PR5] durchgeführt.

5.8 Evaluation von Damast und Diskussion

5.8.1 Bestimmung der Anforderungserfüllung anhand der Ergebnisse der Experimente

Die Evaluation untersucht, inwiefern Damast bei der Entwicklung von bildbasierten Fahrsystemen unterstützt. Hierfür wird die Erfüllung der identifizierten Anforderungen an ein Datenmanagementsystem für die Entwicklung von bildbasierten Fahrsystemen (vgl. Unterabschnitt 3.4.1) anhand der Ergebnisse aus den Experimenten mit der Damast-Implementierung (vgl. Abschnitte 5.3, 5.4, 5.5, 5.6 und 5.7) überprüft.

Anforderung A1: Real aufgenommene Bilder müssen semantisch auf den Pegasus-Ebenen durchsuchbar sein.

Die generelle Funktion des auf Vektorrepräsentationen basierenden Teils von Damast mit real aufgenommenen Bildern wird durch zwei Experimente gezeigt (vgl. Abschnitt 5.3). Die beiden Experimente beziehen sich jeweils auf die Gesamtbildebene beziehungsweise auf die Objektebene. Die Durchsuchbarkeit der Pegasus-Ebenen (vgl. Unterabschnitt 2.1.3) mit allen Damast-Methoden wird explizit in mehreren Experimenten (vgl. Abschnitt 5.4 und Abschnitt 5.5) untersucht. Die Datenanforderungen in diesen Experimenten sind dabei so definiert, dass sie für jede Damast-Methode jeweils die Durchsuchbarkeit aller potenziell infrage kommenden Pegasus-Ebenen überprüfen. In den Experimenten werden von Damast zu jeder Datenanforderung passende Bilder gefunden. Daraus wird gefolgert, dass Damast in der Lage ist, die Pegasus-Ebenen E1 bis E5 zielgerichtet, semantisch zu durchsuchen (siehe Tabelle 2.2).

Die Ausnahme bildet Pegasus Ebene E6 „Digitale Informationen“, welche sich auf nicht visuell wahrnehmbare Einflüsse bezieht und daher nicht mit Damast durchsuchbar ist. Datenanforderungen zu allen anderen Pegasus-Ebenen werden mit Damast erfolgreich erfüllt. Die Pegasus-Ebenen E1 „Straßenebene“ und E2

Tabelle 5.10: Ergebnisse aus den Experimenten: Durchsuchbarkeit der Pegasus-Ebenen (vgl. Unterabschnitt 2.1.3) mit Damast

Ebene	Damast-Methode		
	Vektorrepräsentation	Geografische Daten	Sonnenstand
E1 Straßenebene	✔	✔	✘
E2 Leitinfrastruktur	✔	✔	✘
E3 Temporäre Beeinflussung E1/E2	✔	✘	✘
E4 Objekte	✔	✘	✘
E5 Umgebungsbedingungen	✔	✘	✔
E6 Digitale Informationen	✘	✘	✘

Legende: ✔ durchsuchbar, ✘ nicht durchsuchbar

„Leitinfrastruktur“ sind zusätzlich zu der auf Vektorrepräsentationen basierenden Methode über die geografischen Daten durchsuchbar. Bilder zu Datenanforderungen in der Pegasus-Ebene E5 „Umgebungsbedingungen“ werden mit der Sonnenstandmethode und der auf Vektorrepräsentationen basierenden Methode aufgefunden. Aufgrund ihres temporären Charakters sind die Pegasus-Ebenen E3 „temporäre Beeinflussung E1/E2“ und E4 „Objekte“ ausschließlich mit der auf Vektorrepräsentationen basierenden Methode durchsuchbar.

Pegasus-Ebenen-übergreifende Datenanforderungen werden in den Experimenten zur Kombination von Suchmethoden (vgl. Abschnitt 5.6) untersucht. Die Ergebnisse dieser Experimente zeigen, dass Damast auch in diesem Fall zu jeder Datenanforderung ein passendes Bild liefert.

Anforderung A2: Das Datenmanagementsystem muss sich hinsichtlich der Laufzeiten interaktiv in den Entwicklungsprozess einfügen.

Die Bedingung für den Einsatz von Damast im Entwicklungsprozess von bildbasierten Fahrsystemen ist eine ausreichend kurze Laufzeit, sodass eine interaktive Benutzung möglich ist. In den Experimenten zur Laufzeit (vgl. Abschnitt 5.7) beantwortet Damast die Datenabfragen in maximal $0{,}35\,\mathrm{s}$. Die Laufzeit ist kurz genug, dass ein Nutzer während des Entwicklungsprozesses interaktiv mit Damast interagieren kann.

Ferner zeigen die Ergebnisse der Nutzerstudie (vgl. Abschnitt 5.4), dass die Nutzer anhand von nur zwei Beispielanfragen in der Lage sind, die Parametrisierung von Damast zu lernen. Anschließend finden die Nutzer zu $90{,}6\,\%$ der Datenanforderungen passende Bilder. Für die erfolgreiche Bearbeitung einer Datenanforderung benötigen sie dabei im Durchschnitt weniger als eine halbe Minute. Während des Entwicklungsprozesses von Fahrsystemen ist Damast daher in der Lage, interaktiv zu den Datenanforderungen passende Bilder bereitzustellen.

Anforderung A3: Das Datenmanagementsystem muss ohne manuelle Annotation der Daten funktionieren.

Für die Nutzung von Damast ist keine manuelle Annotation notwendig. Die Damast-Methoden basieren im Fall der klassischen Kontexte (vgl. Abschnitt 4.2) auf externen Datenquellen und initialen Kontextdaten. Das Sammeln der initialen Kontexte (Aufnahmezeitpunkt, geografische Koordinaten und Fahrtrichtung) bedarf keines manuellen Eingriffs, sondern die initialen Kontexte werden durch entsprechende Sensorik während der Fahrt automatisch für jedes Bild aufgezeichnet.

Der Kontext, der Vektorrepräsentationen nutzt, basiert auf einer vortrainierten panoptischen Segmentierung und einer vortrainierten CLIP-Implementierung

(vgl. Abschnitt 5.1). Im Fall der panoptischen Segmentierung war für das Training ein explizites Annotieren von Bildern notwendig. Die Annotation und das Training wurden nicht im Rahmen der Entwicklung von Damast durchgeführt, sondern sind bereits durch den Anbieter der panoptischen Segmentierung erfolgt. Für diesen Teil ist eine manuelle Annotation notwendig, jedoch erfolgt keine gesonderte Annotation für die Nutzung im Rahmen von Damast. Es wird auf eine bereits trainierte panoptische Segmentierung zurückgegriffen, denn nach dem Training der panoptischen Segmentierung ist keine weitere manuelle Annotation notwendig.

CLIP (vgl. Abschnitt 2.6), welches die Vektorrepräsentationen berechnet, wurde von der Organisation, die es zur Verfügung stellt, mit Bild-Text-Paaren aus dem Internet trainiert. Die notwendigen Annotationen wurden daher in diesem Fall bereits implizit durch die Ersteller der Internetseiten durchgeführt.

Für die Nutzung von Damast sind ausschließlich folgende manuelle Aufwendungen notwendig: Auswahl einer vortrainierten panoptischen Segmentierung und CLIP-Implementierung. Selektion einer passenden Quelle für Kartendaten und gewünschte geografische Objekte (vgl. Abschnitt 4.2). Die weitere Vorverarbeitung (vgl. Abschnitt 4.1) erfolgt anschließend automatisch.

Anforderung A4: Als initiale Kontexte dürfen nur automatisch aufgezeichnete Informationen (Zeitstempel, Fahrzeugposition und Fahrtrichtung) verwendet werden.

Als initiale Kontexte (vgl. Abschnitt 4.2) benötigt Damast lediglich Aufnahmezeitpunkt, geografische Koordinaten und Fahrtrichtung. Diese sind jedoch nur für die klassischen Kontexte notwendig. Die natürlichsprachliche Suche mit Damast verwendet ausschließlich die Bilddaten. Liegen die initialen Kontexte nicht vor, können die Bilder dennoch mit Damast verwaltet werden. In diesem Fall steht in Damast nur die natürlichsprachliche Suche zur Verfügung.

Anforderung A5: Es sollen Suchmethoden verwendet werden, die unabhängig von Bilddaten funktionieren.

Die klassischen Kontexte von Damast (vgl. Abschnitt 4.2) greifen nicht auf die Bilddaten zu. Die Suche nach geografischen Objekten oder dem Sonnenstand ist daher unabhängig von den Bilddaten. Die natürlichsprachliche Suche basiert jedoch auf den Bilddaten. Die Verwendung der Bilddaten führt zu folgenden Schwierigkeiten: Fordert eine Datenanforderung eine Situation mit herausfordernden Bilddaten für die Perzeption eines Fahrsystems, ist es möglich, dass es auch für Damast herausfordernd ist, diese Situation in den Bilddaten zu erkennen und die entsprechenden Bilder bereitzustellen. Diese Schwierigkeit wird jedoch aufgelöst, da der Vorverarbeitungsschritt von Damast im Vergleich zu den Komponenten eines Fahrsystems keine Echtzeitanforderungen erfüllen muss. Den künstlichen neuronalen Netzen, welche in Damast zum Einsatz kommen, steht daher mehr Zeit zum Erkennen der gewünschten Situationen in den Bilddaten zur Verfügung und es können aufwendigere Architekturen der künstlichen neuronalen Netze zum Einsatz kommen. So benötigt der Anreicherungsschritt von Damast in den Experimenten mehr als $2\,\mathrm{s}$ pro Bild (vgl. Abschnitt 5.7). Eine solch lange Laufzeit wäre für ein Fahrsystem, das sich in dieser Zeit bei einer Geschwindigkeit von beispielsweise $50\,\mathrm{km\,h^{-1}}$ $27{,}8\,\mathrm{m}$ weit bewegt, nicht hinnehmbar. Darüber hinaus unterliegen die Hardware-Komponenten für ein Fahrsystem Limitationen hinsichtlich ihres Stromverbrauchs und damit ihrer Leistungsfähigkeit [120]. Diese Einschränkungen bestehen für die Vorverarbeitung von Damast nicht.

Außerdem ist mit Damast auch eine indirekte Suche nach Situationen möglich. Ein Beispiel findet sich in der Nutzerstudie, in der eine Datenanforderung Bilder mit „temporären Fahrbahnmarkierungen“ verlangte. Statt diese direkt zu suchen, suchten einige Nutzer allgemeiner nach Baustellen und Straßenbauarbeiten. Für die erfolgreiche Suche wird dabei auf andere Bildbereiche zurückgegriffen, die das Vorhandensein des gesuchten Objekts nahelegen.

5.8.2 Diskussion und Limitationen von Damast

Damast hat alle an ein Datenmanagementsystem für die Entwicklung von bildbasierten Fahrsystemen gestellten Anforderungen erfüllt. Die prototypische Implementierung von Damast findet in den Experimenten für jede Datenanforderung passende Bilder. Daher ist davon auszugehen, dass Damast die Entwicklung von bildbasierten Fahrsystemen durch die Bereitstellung von Bildern unterstützt. Allerdings hat Damast auch Eigenschaften und Limitationen, die bei der Nutzung zu beachten sind.

Anwendungsbereiche im Entwicklungsprozess

Das Konzept von Damast wurde mit dem Ziel entwickelt, Bilder für den Test von bildbasierten Perzeptionskomponenten bereitzustellen (vgl. Abschnitt 3.1). Zusätzlich können auch Datenanforderungen für den Test anderer Komponenten oder des gesamten Fahrsystems erfüllt werden. In den Experimenten wurde gezeigt, dass Damast in der Lage ist, Situationen wie Stau, ausscherende Schulbusse, bremsende Fahrzeuge und den Verkehr regelnde Polizisten zu identifizieren. Diese Fähigkeit, spezifische Situationen in Datensätzen zu finden, ist beim Testen von Komponenten der Planung oder des gesamten Fahrsystems (vgl. Unterabschnitt 2.1.2) nützlich. Mit Damast können, durch die Verknüpfung der Informationen (vgl. Unterabschnitt 4.1.2) die Daten aller Sensoren bereitgestellt werden.

Verwaltung mehrerer Datensätze

Die Experimente wurden mit drei Automotive-Datensätzen, die während der Fahrt aufgezeichnet wurden, durchgeführt (vgl. Abschnitt 5.2). Hiermit wird gezeigt, dass Damast in der Lage ist, gleichzeitig mehrere Datensätze zu verwalten. Zur Unterscheidung der Bilder hinsichtlich ihres Ursprungsdatensatzes wird ein Attribut, welches den Datensatznamen beinhaltet, zu den Metadaten in Damast (vgl. Unterabschnitt 4.1.2) hinzugefügt.

Nachvollziehbarkeit

Die Nachvollziehbarkeit von Ergebnissen ist bei der Verwendung von künstlichen neuronalen Netzwerken eingeschränkt. Das gilt auch für den auf Vektorrepräsentationen basierenden Kontext in Damast. Das Zustandekommen der panoptischen Segmentierung und der Vektorrepräsentationen ist von einem Nutzer nicht direkt nachvollziehbar. Diese Einschränkung ist bei den klassischen Kontexten von Damast nicht gegeben, da diese auf nachvollziehbaren Regeln und Parametern basieren.

Durch die Größe des Trainingsdatensatzes von CLIP[14] ist nicht nachvollziehbar, welche Objekte und Konzepte (z. B. Wettersituationen) von CLIP in der Vektorrepräsentation berücksichtigt werden. Die Genauigkeit von Damast ist dadurch abhängig von der jeweiligen Suche und a priori nicht vorhersagbar. Die Experimente zeigen jedoch, dass die Datenanforderungen mit Damast durch die interaktive Nutzung und gegebenenfalls mehrfache Suchen dennoch erfüllbar sind.

Laufzeitoptimierung

Die untersuchte Damast-Implementierung nutzt für den Anreicherungsschritt (vgl. Abschnitt 4.1) keine Parallelisierung und benötigt pro Bild $2{,}05\,\mathrm{s}$ (vgl. Abschnitt 5.7). Zur Beschleunigung dieses Vorgangs ist eine Parallelisierung hinsichtlich der Bilder und hinsichtlich der Kontexte möglich.

[14] Die CLIP-Implementierung von OpenAI wurde mit 400 Millionen Bild-Text-Paaren trainiert. Außerdem hat OpenAI den für das Training ihrer CLIP-Implementierung verwendeten Datensatz nicht veröffentlicht. [111] (vgl. Abschnitt 2.6

Vielfalt der Teildatensätze

Steht ein Fahrzeug während der Datenaufzeichnung und verändert sich die Umgebung nicht, wird eine Reihe ähnlicher Bilder aufgezeichnet. Bei der Zusammenstellung von Teildatensätzen mit Damast ist es daher möglich, dass ein Teildatensatz ähnliche Bilder enthält. Um ähnliche Bilder in Teildatensätzen zu verhindern, ist daher als zusätzlicher Schritt nach der Teildatensatz-Zusammenstellung durch Damast ein Filtern mit den Bedingungen einer minimalen zeitlichen und/oder geografischen Distanz der Aufnahmeorte möglich.

Genauigkeit künstlicher neuronaler Netze

Damast liefert in den durchgeführten Experimenten zu jeder Datenanforderung passende Bilder und stellt entsprechende Teildatensätze zusammen. Es existieren jedoch Methoden, die bei der Identifikation einzelner Datenanforderungen eine höhere Genauigkeit erzielen. Das sind unter anderem künstliche neuronale Netze, die überwacht trainiert (vgl. Abschnitt 2.4) werden. Beispielsweise erreicht eine solche Methode auf dem Test-Teil des Stanford-Cars-Datensatzes (vgl. Unterabschnitt 5.2.4) eine Klassifikationsgenauigkeit von $96{,}9\,\%$ [121]. Jedoch sind diese Methoden auf einzelne Datenanforderungen spezialisiert. In diesem Fall zum Beispiel die Identifikation von Fahrzeugmodellen. Daher erlauben diese Methoden keine generalisierte Zusammenstellung von Bildern anhand von Datenanforderungen.

Objekte, die zum Zeitpunkt des Trainings der verwendeten künstlichen neuronalen Netzwerke nicht existierten, können mit dem auf Vektorrepräsentationen basierenden Teil von Damast nicht identifiziert werden. Ein Beispiel für eine solche Objektklasse sind die in den vergangenen Jahren aufgekommenen Elektro-Tretroller.

Genauigkeit klassischer Kontexte

Die klassischen Kontexte in Damast (vgl. Abschnitt 4.2) haben den Vorteil, dass das Gesuchte nicht in den Bilddaten identifiziert werden muss. Das ist der Fall, da die Informationen für die Suche aus von den Bilddaten unabhängigen Quellen stammen. Diese Eigenschaft hat allerdings auch Nachteile. Werden etwa Kartendaten verwendet, die zeitlich nicht zu den Bildern passen, ist es möglich, dass Damast falsch positive Ergebnisse zurückgibt oder passende Bilder übersieht. Zusätzlich schwankt die Abdeckung der in der Implementierung verwendeten OpenStreetMap-Daten. Dadurch ist es möglich, dass der Methode Bilder bei der Suche entgehen, da die gesuchten geografischen Objekte nicht in den Kartendaten enthalten sind.

Die mit den klassischen Kontexten gesuchten geografischen Objekte oder die Sonne können zudem durch andere Objekte verdeckt und daher im Bild nicht sichtbar sein. Darüber hinaus wird mit dem beschriebenen Vorgehen nicht unterschieden, ob ein Objekt in gleicher Höhe wie das aufzeichnende Fahrzeug ist. Es lässt sich daher nicht unterscheiden, ob ein Bild auf oder unter einer Brücke aufgenommen wurde.[15]

In den OpenStreetMap-Daten sind Straßen als Strecken zwischen geografischen Punkten dargestellt (vgl. Unterabschnitt 5.2.5). Wurde ein Bild auf einem Verbindungsstück zwischen zwei Punkten aufgezeichnet und der nächste geografische Punkt dieser Straße ist für die gewählte Such-Parametrisierung (Öffnungswinkel, Radius) zu weit entfernt (vgl. Unterabschnitt 4.2.2), übersieht eine Suche nach Straßeneigenschaften das zugehörige Bild. Der Grund hierfür ist, dass die vorgestellte Methode die Eigenschaften der geografischen Punkte und nicht der Linien analysiert (siehe Abbildung 5.26). Ein Beispiel für diese Limitation ist die Suche nach Bildern auf der Autobahn, bei der Bilder auf längeren Strecken nicht gefunden werden.

[15] Eine Methode, die Bilder, die unter einer Brücke entstanden sind, anhand von geografischen Daten identifiziert, ist in [PR3] veröffentlicht.

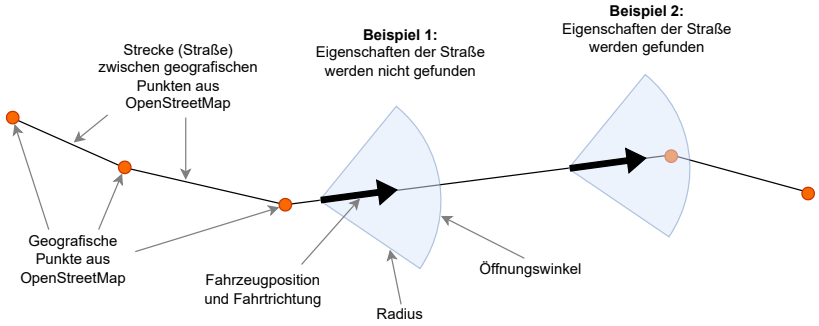


Abbildung 5.26: Beispiel für eine Limitation bei der Suche nach Straßeneigenschaften

Bei der Suche mit Damast nach bestimmten Sonnenständen (vgl. Unterabschnitt 4.2.1) wird angenommen, dass das Fahrzeug waagerecht auf der Erdoberfläche steht. Diese Bedingung ist jedoch nicht immer erfüllt, zum Beispiel wenn das Fahrzeug einen Berg hoch- oder herunterfährt. Um den dadurch entstehenden Fehler zu korrigieren, ist es möglich, den Roll- und Nickwinkel des Fahrzeugs bei der Fahrt aufzuzeichnen und die Berechnung des relativen Sonnenstandes um diese Freiheitsgrade zu erweitern.

6 Zusammenfassung und Ausblick

6.1 Beiträge dieser Dissertation

Bildbasierte, hochautomatisierte Fahrsysteme müssen während der Entwicklung getestet werden, um eine robuste und verlässliche Funktionsweise sicherzustellen. Für diese Tests ist es notwendig, Bilder bereitzustellen. Die manuelle Bereitstellung von Bildern aus Datensätzen anhand von semantischen Kriterien ist jedoch kostenintensiv (vgl. Abschnitt A.2).

Diese Dissertation befasst sich zur Lösung dieser Herausforderung daher mit folgenden zentralen Fragestellungen (siehe Forschungsfragen Abschnitt 1.3):

- Welche Eigenschaften muss ein Datenmanagementsystem für die Entwicklung von bildbasierten Fahrsystemen besitzen?
- Wie muss ein Datenmanagementsystem für die Entwicklung von bildbasierten Fahrsystemen konzipiert sein?
- Wie lässt sich ein Datenmanagementsystem für die Entwicklung von bildbasierten Fahrsystemen realisieren?

Notwendige Eigenschaften eines Datenmanagementsystems für die Entwicklung von bildbasierten Fahrsystemen

In dieser Dissertation wurden am aktuellen Stand der Wissenschaft und Technik Eigenschaften identifiziert (vgl. Unterabschnitt 3.4.1), die ein Datenmanagementsystem besitzen muss, um bei der Entwicklung von bildbasierten, hochautomatisierten Fahrsystemen zu unterstützen. Ein Datenmanagementsystem muss in der Lage sein, real aufgenommene Bilder zielgerichtet und semantisch zu durchsuchen. Die Suche muss sich dabei auf alle Pegasus-Ebenen erstrecken. Es ist entscheidend, dass das Datenmanagementsystem interaktiv in den Entwicklungsprozess eingebettet ist. Um die Größe der Datensätze zu bewältigen und die Kosten zu limitieren, sind manuelle Annotationen der Daten ausgeschlossen. Zur Reduktion des Aufwandes bei der Datenaufzeichnung sind die zusätzlich zu speichernden Informationen auf den Zeitstempel, die Fahrzeugposition und die Fahrtrichtung zu beschränken. Um auch Bildeigenschaften zu finden, deren Identifikation in den Bilddaten eine Herausforderung darstellt, müssen die Suchmethoden, soweit möglich, unabhängig von den Bilddaten funktionieren.

Konzept eines Datenmanagementsystems für die Entwicklung von bildbasierten Fahrsystemen

Das auf den identifizierten Eigenschaften basierende Konzept für das Datenmanagementsystem Damast ist in den Anreicherungsprozess und die Datenabfrage aufgeteilt (vgl. Kapitel 4). Das zentrale Artefakt in Damast ist der Datenpunkt (vgl. Unterabschnitt 4.1.2), welcher Sensormessungen, Kontext- und Metadaten zusammenfasst. Jeder Datenpunkt durchläuft den Anreicherungsprozess von Damast ein einziges Mal beim Hinzufügen in das Datenmanagementsystem. Anschließend ist der Datenpunkt in Damast auffindbar. Damast unterscheidet zwischen klassischen, regelbasierten Kontexten (geografische Objekte und Sonnenstand) und einem Kontext, der auf Bilddaten basiert. Diese Kontexte können von einem Nutzer beim Zusammenstellen von Teildatensätzen kombiniert werden und decken

die Eigenschaften in den Pegasus-Ebenen 1 bis 5 ab. Hierbei realisiert Damast auch die Identifikation von Eigenschaften auf der Ebene einzelner Objekte.

Umsetzung des Datenmanagementsystems für die Entwicklung von bildbasierten Fahrsystemen

Das Datenmanagementsystem Damast ist prototypisch umgesetzt (vgl. Kapitel 5). Mit dieser Implementierung werden mehrere Automotive-Datensätze verwaltet. Basierend auf dieser Umsetzung wurden Experimente durchgeführt und die Leistungsfähigkeit von Damast untersucht. Damast wurde evaluiert und die Ergebnisse der Experimente wurden mit den identifizierten Eigenschaften abgeglichen.

Zusammenfassung der wissenschaftlichen Beiträge dieser Dissertation

Zusammenfassend sind die wesentlichen wissenschaftlichen Beiträge dieser Dissertation:

- Es wurden notwendige Eigenschaften eines Datenmanagementsystems für die Entwicklung von bildbasierten Fahrsystemen identifiziert.
- Das Datenmanagementsystem Damast wurde für die Entwicklung von bildbasierten Fahrsystemen konzipiert.
- Damast ist interaktiv in den Entwicklungsprozess von bildbasierten Fahrsystemen eingebettet.
- Damast beinhaltet kombinierbare Methoden für das semantische Zusammenstellen von Teildatensätzen.
- Die Methoden in Damast basieren auf Bilddaten oder auf externen Quellen. Sie erlauben die Identifikation von Bild- und Objekteigenschaften auf den für bildbasierte Fahrsysteme relevanten semantischen Ebenen.

- Das in dieser Dissertation vorgestellte Datenmanagementsystem Damast wurde prototypisch umgesetzt.
- Die Funktionsweise von Damast wurde anhand mehrerer Automotive-Datensätze demonstriert und evaluiert.

6.2 Ausblick

Ausgehend von den Beiträgen der vorliegenden Dissertation ergeben sich anknüpfende Forschungsideen. Damast verwaltet Datensätze, die bereits aufgezeichnet wurden. Eine Adaption von Damast könnte die Daten, die tatsächlich aufgezeichnet, gespeichert und verwaltet werden müssen, reduzieren. Bereits während der Aufzeichnung würde ein solches System entscheiden, ob das aktuelle Kamerabild eine Datenanforderung erfüllt und gespeichert werden sollte oder verworfen wird.

Das Ziel der Entwicklung von Damast lag in der Erfüllung von Datenanforderungen, indem zu den Datenanforderungen passende Bilder bereitgestellt werden. Ein interessanter Ansatz wäre, die Methoden von Damast explorativ einzusetzen und in Datensätzen unbekannte Konzepte und Objekte zu identifizieren. Ein systematisiertes Vorgehen könnte zur Generierung von Datenanforderungen genutzt werden. Eine weitere Forschungsfrage wäre, ob eine Erweiterung von Damast dazu genutzt werden könnte, zu entscheiden, ob Datenanforderungen hinsichtlich eines vorliegenden Datensatzes vollständig sind oder anhand des Datensatzes ergänzt werden müssen.

Ein Anknüpfungspunkt für weitere Forschungen ist die Übertragung von Damast aus der Entwicklung im Automobilbereich in weitere Domänen. Hier kommen alle Anwendungsfelder infrage, in welchen Bilderdatensätze semantisch durchsucht werden sollen und eine manuelle Annotation nicht sinnvoll durchführbar ist.

Der Fokus von Damast liegt auf dem Durchsuchen von Bilddatensätzen. Andere Daten werden in Damast zwar auch verwaltet und sind daher indirekt auffindbar,

allerdings ist die direkte semantische Durchsuchbarkeit anderer Datentypen, also beispielsweise Lidar-, Radar- und Zeitreihendaten, bisher nicht untersucht.

Die weitere Forschung sollte sich mit Datenanforderungen, die aufgrund der Datenlage nicht erfüllt werden können, befassen. An den Stellen, an denen Bilder nicht auffindbar sind, da ein Datensatz sie nicht beinhaltet, sollte der Einsatz generativer Modelle zur Erzeugung der fehlenden Bilder untersucht werden.

A Anhang

A.1 Methode: Anzahl der Publikationen im Bereich automatisiertes Fahren

Zur Bestimmung der Anzahl der Publikationen im Bereich automatisiertes Fahren über die Zeit wurde die Literatur-Datenbank Scopus [1] abgefragt. Zur Abfrage der Daten wurden alle Publikationen, in denen die Worte „automated“ und „driving“ im Titel, in der Zusammenfassung oder in den Schlüsselwörtern vorkommen, berücksichtigt. Der Publikationszeitraum wurde auf die Jahre 2010 bis 2023 eingegrenzt.

A.2 Beispielhafte Kostenabschätzung für die Annotation

Exemplarisch werden die Kosten des Annotationsdienstes von Amazon Web Services[1] für die Abschätzung verwendet. Zur Kostenabschätzung für die Annotation eines Datensatzes mit herausfordernden Situationen, die sich auch auf Eigenschaften einzelner Objekte beziehen können, werden folgende Annahmen getroffen:

- Es sollen insgesamt 100 000 Bilder innerhalb eines Monats annotiert werden.

[1] `https://aws.amazon.com/de/` (abgerufen am 12.02.2025)

- Der Aufwand bzw. die Kosten zur Annotation mit herausfordernden Situationen einschließlich der Objektebene sind mit einer semantischen Segmentierung eines Bildes vergleichbar.
- Zur Sicherstellung einer ausreichenden Annotationsgenauigkeit muss die Annotation pro Bild durch drei Personen durchgeführt werden. (Diese Annahme deckt sich mit einer Beispielrechnung von Amazon Web Services.[2])

Die bei Amazon Web Services angefragten Kosten teilen sich in die Kosten für die Plattformen Amazon SageMaker[3] und Amazon Mechanical Turk[4] auf. Die am 12.02.2025 angefragten Kosten[5] betragen insgesamt 258 000 USD.

Eine weitere Orientierung bietet außerdem ein Erfahrungswert aus der Beauftragung eines Annotationsunternehmens zur Instanzsegmentierung von Bildern aus dem Jahr 2021. Die extrapolierten Kosten für die Instanzsegmentierung von 100 000 Bildern betrugen dort ca. 350 000 €.

A.3 Klassifikation des Stanford-Datensatzes: Auswertung auf Klassenebene

Tabelle A.1: Ergebnisse der Klassifikation des Stanford-Cars-Datensatzes [117] nach dem Fahrzeugsmodell

Fahrzeugmodell	F1-Maß	Präzision	Trefferquote	Anzahl Instanzen
smart fortwo Convertible 2012	0.97	1.00	0.95	40
Nissan Juke Hatchback 2012	0.97	0.98	0.95	44
Tesla Model S Sedan 2012	0.96	0.95	0.97	38
Audi R8 Coupe 2012	0.95	0.93	0.98	43
Lamborghini Reventon Coupe 2008	0.95	0.92	0.97	36
FIAT 500 Abarth 2012	0.95	0.93	0.96	27

[2] `https://aws.amazon.com/de/sagemaker-ai/groundtruth/pricing/` (abgerufen am 12.02.2025)
[3] `https://aws.amazon.com/de/sagemaker/` (abgerufen am 12.02.2025)
[4] `https://www.mturk.com/` (abgerufen am 12.02.2025)
[5] `https://calculator.aws/#/createCalculator/SageMakerGroundTruth` (abgerufen am 12.02.2025)

Fahrzeugmodell	F1-Maß	Präzision	Trefferquote	Anzahl Instanzen
Cadillac CTS-V Sedan 2012	0.95	0.90	1.00	43
Volkswagen Beetle Hatchback 2012	0.93	0.89	0.98	42
McLaren MP4-12C Coupe 2012	0.93	0.86	1.00	44
Dodge Challenger SRT8 2011	0.93	0.90	0.95	39
Lamborghini Diablo Coupe 2001	0.92	0.91	0.93	44
Porsche Panamera Sedan 2012	0.91	0.97	0.86	43
Cadillac SRX SUV 2012	0.91	0.87	0.95	41
Ford Mustang Convertible 2007	0.90	0.86	0.95	44
Volkswagen Golf Hatchback 2012	0.90	0.87	0.93	43
Bugatti Veyron 16.4 Coupe 2009	0.90	0.87	0.93	43
Chrysler PT Cruiser Convertible 2008	0.89	0.95	0.84	45
Lincoln Town Car Sedan 2011	0.89	0.84	0.95	39
Volkswagen Golf Hatchback 1991	0.89	0.84	0.93	46
Ferrari California Convertible 2012	0.88	0.81	0.97	39
Nissan Leaf Hatchback 2012	0.88	0.83	0.93	42
Ford Fiesta Sedan 2012	0.88	0.92	0.83	42
Ford GT Coupe 2006	0.87	0.84	0.91	45
Hyundai Veloster Hatchback 2012	0.87	0.86	0.88	41
Fisker Karma Sedan 2012	0.86	0.86	0.86	43
Mercedes-Benz 300-Class Convertible 1993	0.85	0.84	0.85	48
Volvo C30 Hatchback 2012	0.84	0.79	0.90	41
BMW X6 SUV 2012	0.84	0.87	0.81	42
Geo Metro Convertible 1993	0.83	0.71	1.00	44
Volvo 240 Sedan 1993	0.83	0.92	0.76	45
Dodge Charger Sedan 2012	0.82	0.85	0.80	41
Bentley Arnage Sedan 2009	0.82	0.76	0.90	39
Dodge Journey SUV 2012	0.82	0.76	0.89	44
Ferrari FF Coupe 2012	0.82	0.89	0.76	42
Chrysler Crossfire Convertible 2008	0.82	0.76	0.88	43
Toyota 4Runner SUV 2012	0.81	0.76	0.88	40
BMW 1 Series Convertible 2012	0.81	0.78	0.83	35
Mitsubishi Lancer Sedan 2012	0.80	0.88	0.74	47
GMC Terrain SUV 2012	0.79	0.76	0.83	41
Volvo XC90 SUV 2007	0.79	0.79	0.79	43
MINI Cooper Roadster Convertible 2012	0.79	0.96	0.67	36
Rolls-Royce Phantom Drophead Coupe Convertible 2012	0.79	0.85	0.73	30
Jeep Wrangler SUV 2012	0.79	0.66	0.98	43
Acura Integra Type R 2001	0.78	0.72	0.86	44
Toyota Sequoia SUV 2012	0.78	0.69	0.89	38
Lamborghini Aventador Coupe 2012	0.78	0.79	0.77	43
Dodge Durango SUV 2012	0.77	0.73	0.81	43
Buick Enclave SUV 2012	0.77	0.71	0.83	42
Jaguar XK XKR 2012	0.76	0.97	0.63	46
Mercedes-Benz Sprinter Van 2012	0.76	0.85	0.68	41
Jeep Patriot SUV 2012	0.75	0.65	0.89	44
Ferrari 458 Italia Coupe 2012	0.74	0.67	0.83	42
Chevrolet Traverse SUV 2012	0.74	0.66	0.84	44
Buick Verano Sedan 2012	0.74	0.64	0.86	37
BMW X3 SUV 2012	0.73	0.63	0.87	38
FIAT 500 Convertible 2012	0.72	0.60	0.91	33
Lamborghini Gallardo LP 570-4 Superleggera 2012	0.72	0.79	0.66	35
BMW 3 Series Wagon 2012	0.72	0.76	0.68	41
Chevrolet Corvette ZR1 2012	0.72	0.83	0.63	46
Ford Edge SUV 2012	0.72	0.65	0.79	43
BMW Z4 Convertible 2012	0.71	0.69	0.72	40
Hyundai Tucson SUV 2012	0.70	0.84	0.60	43

Fahrzeugmodell	F1-Maß	Präzision	Trefferquote	Anzahl Instanzen
Jeep Compass SUV 2012	0.70	0.68	0.71	42
Maybach Landaulet Convertible 2012	0.69	0.57	0.90	29
Jeep Liberty SUV 2012	0.69	0.84	0.59	44
Infiniti QX56 SUV 2011	0.69	0.60	0.81	32
Bugatti Veyron 16.4 Convertible 2009	0.69	0.77	0.62	32
Audi S5 Convertible 2012	0.69	0.63	0.76	42
Chevrolet Corvette Convertible 2012	0.69	0.59	0.82	39
Jeep Grand Cherokee SUV 2012	0.68	0.72	0.64	45
Mercedes-Benz C-Class Sedan 2012	0.68	0.72	0.64	45
Toyota Camry Sedan 2012	0.68	0.60	0.79	43
Chevrolet Camaro Convertible 2012	0.68	0.96	0.52	44
Mercedes-Benz SL-Class Coupe 2009	0.68	0.69	0.67	36
Land Rover Range Rover SUV 2012	0.68	0.74	0.62	42
Chevrolet Sonic Sedan 2012	0.67	0.61	0.75	44
Infiniti G Coupe IPL 2012	0.67	0.77	0.59	34
Rolls-Royce Phantom Sedan 2012	0.67	0.60	0.75	44
GMC Acadia SUV 2012	0.67	0.57	0.80	44
Dodge Charger SRT-8 2009	0.67	0.61	0.74	42
AM General Hummer SUV 2000	0.66	0.58	0.77	44
Bentley Mulsanne Sedan 2011	0.66	0.66	0.66	35
Dodge Caravan Minivan 1997	0.65	0.57	0.74	43
Chrysler 300 SRT-8 2010	0.65	0.67	0.62	48
BMW M3 Coupe 2012	0.64	0.53	0.80	44
Chrysler Sebring Convertible 2010	0.63	0.52	0.82	40
Scion xD Hatchback 2012	0.63	0.54	0.76	41
Honda Odyssey Minivan 2012	0.63	0.58	0.69	42
Spyker C8 Coupe 2009	0.63	0.60	0.67	42
BMW 6 Series Convertible 2007	0.63	0.54	0.75	44
Suzuki Kizashi Sedan 2012	0.62	0.82	0.50	46
Bentley Continental GT Coupe 2007	0.60	0.56	0.65	46
Bentley Continental Flying Spur Sedan 2007	0.59	0.59	0.59	44
Dodge Ram Pickup 3500 Crew Cab 2010	0.59	0.67	0.52	42
Buick Regal GS 2012	0.59	0.45	0.83	35
Dodge Magnum Wagon 2008	0.59	0.57	0.60	40
BMW X5 SUV 2007	0.58	0.68	0.51	41
Audi TT RS Coupe 2012	0.58	0.70	0.49	39
Ford F-450 Super Duty Crew Cab 2012	0.57	0.59	0.56	41
Audi S4 Sedan 2007	0.57	0.57	0.58	45
Cadillac Escalade EXT Crew Cab 2007	0.57	0.55	0.59	44
Acura TL Type-S 2008	0.56	0.56	0.57	42
Ferrari 458 Italia Convertible 2012	0.56	0.72	0.46	39
Bentley Continental GT Coupe 2012	0.56	0.63	0.50	34
Mazda Tribute SUV 2011	0.55	0.45	0.72	36
Audi RS 4 Convertible 2008	0.55	0.73	0.44	36
Audi V8 Sedan 1994	0.55	0.45	0.70	43
Nissan NV Passenger Van 2012	0.55	0.71	0.45	38
Ford Expedition EL SUV 2009	0.55	0.39	0.93	44
Hyundai Sonata Sedan 2012	0.55	0.67	0.46	39
Chevrolet Silverado 1500 Classic Extended Cab 2007	0.54	0.50	0.60	42
Eagle Talon Hatchback 1998	0.54	0.41	0.78	46
Aston Martin V8 Vantage Coupe 2012	0.53	0.49	0.59	41
Nissan 240SX Coupe 1998	0.53	0.69	0.43	46
Aston Martin V8 Vantage Convertible 2012	0.53	0.51	0.56	45
Dodge Caliber Wagon 2012	0.53	0.46	0.62	40
Chevrolet Cobalt SS 2010	0.52	0.44	0.66	41
Dodge Sprinter Cargo Van 2009	0.52	0.38	0.85	39

Fahrzeugmodell	F1-Maß	Präzision	Trefferquote	Anzahl Instanzen
Spyker C8 Convertible 2009	0.52	0.62	0.44	45
Dodge Caliber Wagon 2007	0.51	0.53	0.50	42
GMC Yukon Hybrid SUV 2012	0.51	0.88	0.36	42
BMW 3 Series Sedan 2012	0.50	0.46	0.55	42
Bentley Continental Supersports Conv. Convertible 2012	0.50	0.57	0.44	36
Ford F-150 Regular Cab 2012	0.50	0.41	0.62	42
Acura ZDX Hatchback 2012	0.50	0.39	0.67	39
Mercedes-Benz S-Class Sedan 2012	0.49	0.43	0.57	44
Ford Focus Sedan 2007	0.49	0.40	0.62	45
Audi A5 Coupe 2012	0.48	0.36	0.76	41
Ford F-150 Regular Cab 2007	0.46	0.40	0.56	45
Toyota Corolla Sedan 2012	0.46	0.78	0.33	43
Honda Accord Sedan 2012	0.45	0.31	0.87	38
Rolls-Royce Ghost Sedan 2012	0.45	0.48	0.42	38
Hyundai Santa Fe SUV 2012	0.44	0.67	0.33	42
Audi TT Hatchback 2011	0.44	0.38	0.53	40
Hyundai Genesis Sedan 2012	0.43	0.64	0.33	43
Chrysler Town and Country Minivan 2012	0.43	0.29	0.86	37
Suzuki SX4 Hatchback 2012	0.43	0.54	0.36	42
BMW M5 Sedan 2010	0.42	0.56	0.34	41
Aston Martin Virage Convertible 2012	0.42	0.50	0.36	33
Aston Martin Virage Coupe 2012	0.42	0.38	0.47	38
Honda Accord Coupe 2012	0.42	0.37	0.49	39
BMW ActiveHybrid 5 Sedan 2012	0.41	0.48	0.35	34
Chevrolet Corvette Ron Fellows Edition Z06 2007	0.41	0.44	0.38	37
BMW M6 Convertible 2010	0.40	0.54	0.32	41
Chevrolet Avalanche Crew Cab 2012	0.40	0.27	0.76	45
HUMMER H3T Crew Cab 2010	0.39	0.55	0.31	39
Dodge Ram Pickup 3500 Quad Cab 2009	0.39	0.32	0.52	44
BMW 1 Series Coupe 2012	0.39	0.57	0.29	41
Land Rover LR2 SUV 2012	0.38	0.50	0.31	42
Audi S5 Coupe 2012	0.38	0.34	0.43	42
Ford Ranger SuperCab 2011	0.37	0.61	0.26	42
Acura TL Sedan 2012	0.37	0.65	0.26	43
Isuzu Ascender SUV 2008	0.36	0.67	0.25	40
Suzuki Aerio Sedan 2007	0.36	0.30	0.45	38
Chevrolet Tahoe Hybrid SUV 2012	0.35	0.33	0.38	37
Audi 100 Sedan 1994	0.33	0.64	0.23	40
Honda Odyssey Minivan 2007	0.33	0.44	0.27	41
Chevrolet Silverado 1500 Regular Cab 2012	0.33	0.27	0.43	44
Acura RL Sedan 2012	0.33	0.39	0.28	32
Chevrolet Express Van 2007	0.32	0.23	0.54	35
Daewoo Nubira Wagon 2002	0.31	0.69	0.20	45
Chevrolet Malibu Sedan 2007	0.31	0.28	0.34	44
Hyundai Accent Sedan 2012	0.30	0.21	0.58	24
Audi S6 Sedan 2011	0.30	0.64	0.20	46
Buick Rainier SUV 2007	0.28	0.23	0.36	42
Audi S4 Sedan 2012	0.28	0.42	0.21	39
Dodge Durango SUV 2007	0.27	0.23	0.33	45
GMC Savana Van 2012	0.27	0.85	0.16	68
Chevrolet Impala Sedan 2007	0.26	0.44	0.19	43
Hyundai Veracruz SUV 2012	0.26	0.58	0.17	42
GMC Canyon Extended Cab 2012	0.26	0.36	0.20	40
Chevrolet Silverado 2500HD Regular Cab 2012	0.25	0.31	0.21	38

Fahrzeugmodell	F1-Maß	Präzision	Trefferquote	Anzahl Instanzen
Audi 100 Wagon 1994	0.25	0.50	0.17	42
Chevrolet Malibu Hybrid Sedan 2010	0.25	0.31	0.21	38
Audi TTS Coupe 2012	0.23	0.22	0.24	42
Ford Freestar Minivan 2007	0.21	0.46	0.14	44
Chevrolet Express Cargo Van 2007	0.20	0.25	0.17	29
Chevrolet Silverado 1500 Extended Cab 2012	0.19	0.20	0.19	43
Chevrolet TrailBlazer SS 2009	0.19	0.36	0.12	40
Chevrolet Monte Carlo Coupe 2007	0.16	0.29	0.11	45
Acura TSX Sedan 2012	0.15	0.31	0.10	40
Dodge Dakota Crew Cab 2010	0.14	0.16	0.12	41
Dodge Dakota Club Cab 2007	0.13	0.38	0.08	38
Chevrolet Silverado 1500 Hybrid Crew Cab 2012	0.12	0.33	0.07	40
HUMMER H2 SUT Crew Cab 2009	0.11	0.30	0.07	43
Mercedes-Benz E-Class Sedan 2012	0.11	0.30	0.07	43
Hyundai Elantra Sedan 2007	0.11	0.23	0.07	42
Chrysler Aspen SUV 2009	0.11	0.25	0.07	43
Ford E-Series Wagon Van 2012	0.10	0.67	0.05	37
Hyundai Elantra Touring Hatchback 2012	0.10	0.08	0.12	42
Hyundai Sonata Hybrid Sedan 2012	0.07	0.07	0.06	33
Suzuki SX4 Sedan 2012	0.05	0.33	0.03	40
Plymouth Neon Coupe 1999	0.04	0.25	0.02	44
Hyundai Azera Sedan 2012	0.00	0.00	0.00	42
Chevrolet HHR SS 2010	0.00	0.00	0.00	36
Ram C-V Cargo Van Minivan 2012	0.00	0.00	0.00	41

A.4 Parametrisierung der Suchanfragen in der Nutzerstudie

Tabelle A.2: Suchbegriffe mit zugehörigen Objektklassen, die zu einer erfolgreichen Suche geführt haben. Duplikate wurden entfernt.

Aufgabe	Suchbegriff	Objektklasse
Pflastersteine	cobblestone	road
	paved road	road
	paved street	road
	paving stone	road
	paving blocks road	Gesamtbild
	paving cobblestone	road
	cobblestone road	road
	brick stones	road
Zebrastreifen	zebra crossing	road
	crossing	road
	zebra	road
	road walk	road

Aufgabe	Suchbegriff	Objektklasse
	crosswalk	road
	crosswalk	Gesamtbild
	pedestrian crossing	road
	zebra crossing	Gesamtbild
	zebra stripe	road
	zebra stripes	road
Busspur	bus lane	road
	busline	road
	buslane	road
	dedicated bus lane	Gesamtbild
	bus lane	Gesamtbild
	bus explicit lane	road
	bus track	road
	bus line	road
Bremsschwelle	judder bar	road
	speed bump	road
	bumper written on the street	road
Kreisverkehr	roundabout	road
	roundabout	Gesamtbild
	traffic circle	Gesamtbild
Palme	palm tree	vegetation
	palm tree	Gesamtbild
	palmtree	Gesamtbild
	palm	vegetation
	Palm Tree	vegetation
temporäre Fahrbahnmarkierung	construction site	road
	roadworks	Gesamtbild
	road construction	road
	temporary traffic lane	Gesamtbild
	temporary lane marker	road
	temporary road marking	Gesamtbild
	temporary road marking, construction	Gesamtbild
	line marker	road
Verkehrsleitkegel	traffic cone	Gesamtbild
	road work	Gesamtbild
	cone	traffic sign
	road marking temporary pylon cone	Gesamtbild
	cones	road
	Traffic cones	traffic sign
	traffic cone	traffic sign
	temporary road marking, construction	Gesamtbild
	traffic cone	road
Baustellenschild	construction	traffic sign
	construction-site sign	traffic sign
	road work sign	Gesamtbild
	construction side	traffic sign
	roadwork	traffic sign
	work ahead	traffic sign
	sign construction tele	Gesamtbild
	construction site	traffic sign

Aufgabe	Suchbegriff	Objektklasse
	Construction site	traffic sign
	construction sign	traffic sign
Straßenbauer	high-visibility vest	person
	road worker	person
	construction worker	person
	road builder	road
	Road builder	person
	construction site	person
Baumaschine	excavator	Gesamtbild
	construction machine	Gesamtbild
	construction machine	truck
	road construction machine	Gesamtbild
	machine road maintenance	Gesamtbild
	construction machine	car
	Construction machine	truck
	construction site machine	Gesamtbild
abgelenkte Person	distracted person	person
	distracted	person
	distracted Personen	person
	distracted person	Gesamtbild
	distracted pedestrian person	person
	distraction	person
	disrupted person	person
Fahrzeug mit Werbung	logo	car
	car with advertisement	car
	vehicle commercial	bus
	commercial	car
	advertisement	car
	advertisement car	Gesamtbild
	car with advertisement	Gesamtbild
	Advertising	car
	advertising	car
	advertisement	bus
Verkehrsstau	traffic jam	Gesamtbild
	traffic congestion	Gesamtbild
	traffic	Gesamtbild
	Traffic jam	car
Regentropfen	rain drop	Gesamtbild
	rain drops	Gesamtbild
	rain drops windshield	Gesamtbild
	raindrops	Gesamtbild
	raindrop windshield	car
	rain on windscreen	Gesamtbild
	rain	sky
	Raindrops	car
	raindrops windscreen	Gesamtbild
	rain on windshield	car
	rain trops on window	car
Schnee	snow	Gesamtbild
	snow	sidewalk

Aufgabe	**Suchbegriff**	**Objektklasse**
blendende Sonne	overexposed	Gesamtbild
	sun	Gesamtbild
	blinding sun	Gesamtbild
	sun glare	Gesamtbild
	Sun	Gesamtbild
	sun blinding	Gesamtbild
	bright daylight sun	Gesamtbild
	sun flares	sky
	sun blind	Gesamtbild
	dazzling sun	Gesamtbild

A.5 Verteilung der passenden Bilder in den Ergebnissen von Damast

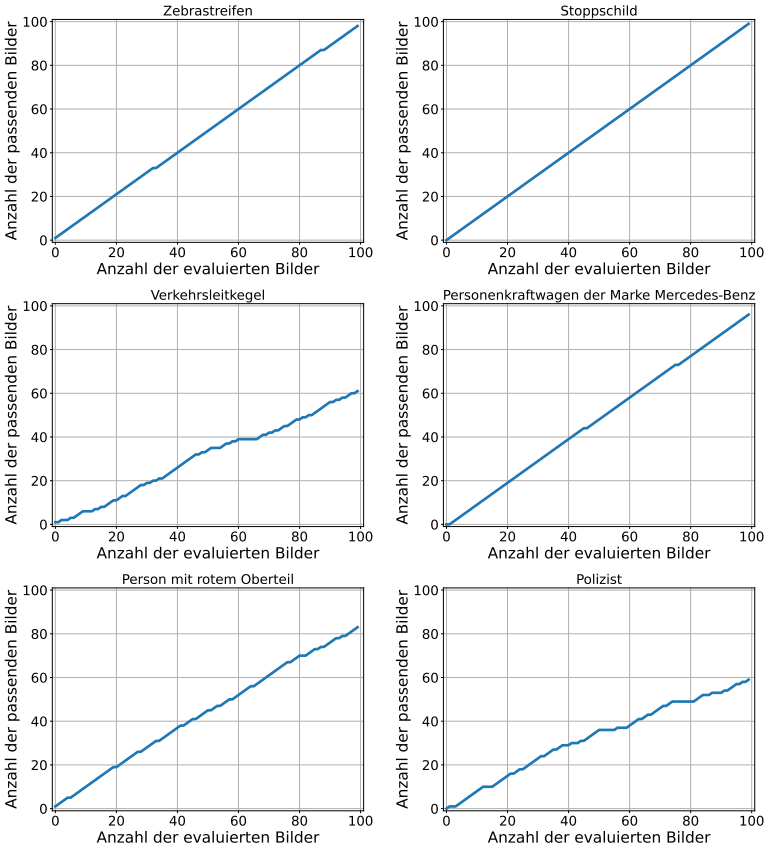


Abbildung A.1: Kumulierte Verteilung der jeweils zu den Datenanfragen passenden Bilder in den ersten 100 Ergebnissen des auf Vektorrepräsentationen basierenden Teils von Damast

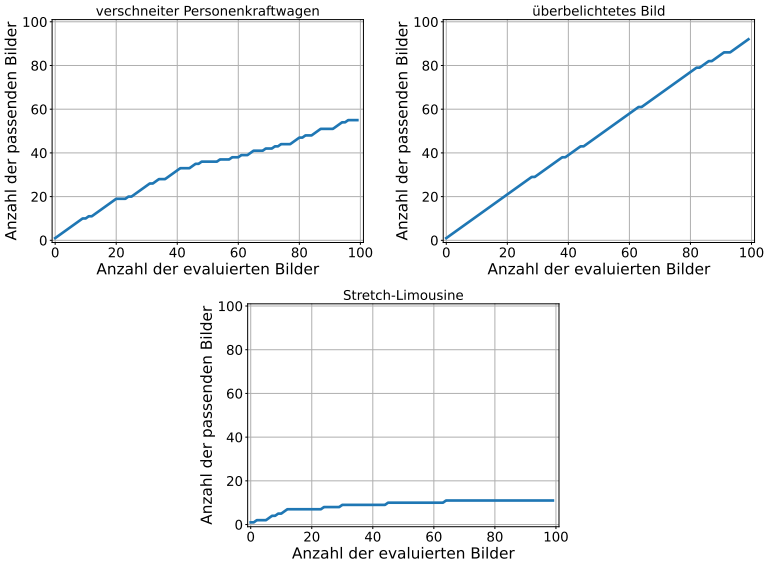


Abbildung A.1: Kumulierte Verteilung der jeweils zu den Datenanfragen passenden Bilder in den ersten 100 Ergebnissen des auf Vektorrepräsentationen basierenden Teils von Damast

Abbildungsverzeichnis

Tabellenverzeichnis

Eigene Veröffentlichungen

[PR1] P. Rigoll, P. Petersen, L. Ries, J. Langner, und E. Sax, „Augmentation von Kameradaten Mit Generative Adversarial Networks (GANs) Zur Absicherung Automatisierter Fahrfunktionen," in *Fahrerassistenzsysteme Und Automatisiertes Fahren*, VDI Wissensforum GmbH, Hrsg. VDI Verlag, 2022, S. 41–48. [Online]. Verfügbar: https://elibrary.vdi-verlag.de/index.php?doi=10.51202/9783181023945-41

[PR2] P. Rigoll, P. Petersen, J. Langner, und E. Sax, „Parameterizable Lidar-Assisted Traffic Sign Placement for the Augmentation of Driving Situations with CycleGAN," in *Advances in Systems Engineering*, L. Borzemski, H. Selvaraj, und J. Świątek, Hrsgg. Cham: Springer International Publishing, 2022, Vol. 364, S. 403–417. [Online]. Verfügbar: https://link.springer.com/10.1007/978-3-030-92604-5_36

[PR3] P. Rigoll, L. Ries, und E. Sax, „Scalable Data Set Distillation for the Development of Automated Driving Functions," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*. Macau, China: IEEE, Okt. 2022, S. 3139–3145. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/9921868/

[PR4] P. Rigoll, P. Petersen, H. Stage, L. Ries, und E. Sax, „Focus on the Challenges: Analysis of a User-friendly Data Search Approach with CLIP in the Automotive Domain," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. Bilbao, Spain: IEEE, Sep. 2023, S. 168–174. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/10422271/

[PR5] P. Rigoll, J. Langner, L. Ries, und E. Sax, „Unveiling Objects with SOLA: An Annotation-Free Image Search on the Object Level for Automotive Data Sets," in *2024 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2024, S. 1053–1059. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/10588869/

[PR6] P. Rigoll, L. Adolph, L. Ries, und E. Sax, „CLIPping the Limits: Finding the Sweet Spot for Relevant Images in Automated Driving Systems Perception Testing," in *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*. Edmonton, AB, Canada: IEEE, Sep. 2024, S. 2398–2404. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/10920034/

[PR7] P. Petersen, H. Stage, J. Langner, L. Ries, P. Rigoll, C. Philipp Hohl, und E. Sax, „Towards a Data Engineering Process in Data-Driven Systems Engineering," in *2022 IEEE International Symposium on Systems Engineering (ISSE)*. Vienna, Austria: IEEE, Okt. 2022, S. 1–8. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/10005441/

[PR8] L. Ries, P. Rigoll, T. Braun, T. Schulik, J. Daube, und E. Sax, „Trajectory-Based Clustering of Real-World Urban Driving Sequences with Multiple Traffic Objects," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. Indianapolis, IN, USA: IEEE, Sep. 2021, S. 1251–1258. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/9564636/

[PR9] P. Reis, P. Rigoll, und E. Sax, „Behavior Forests: Real-Time Discovery of Dynamic Behavior for Data Selection," in *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*. Edmonton, AB, Canada: IEEE, Sep. 2024, S. 1962–1967. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/10920178/

Betreute Abschlussarbeiten

[S1] Tillert, Wiebke H.: *Konzeptionierung einer Pipeline zur Augmentation von Bildern aus dem Straßenverkehr mittels Maschinellen Lernens*, Karlsruher Institut für Technologie, Bachelorarbeit, 2021

[S2] Leonhardt, Artur: *Merkmalsuntersuchung von mit Generative Adversarial Networks augmentierten Automotive-Bildern zur Beurteilung der Verwendung als Beispieldaten*, Karlsruher Institut für Technologie, Bachelorarbeit, 2021

[S3] Schober, Jan: *Evaluation der Einsatzmöglichkeit von Augmentation mittels Generative Adversarial Networks beim Closed-Loop Integrationstest von hochautomatisierten Fahrfunktionen*, Karlsruher Institut für Technologie, Masterarbeit, 2021

[S4] Golks, Niklas: *Objektbezogene Untersuchung eines Latent Space zur Gewinnung neuer Kontexte*, Karlsruher Institut für Technologie, Masterarbeit, 2022

[S5] Stepanov, Gleb: *Konzeption und Evaluation einer Abfrage-Methode für eine Bilddatenbank mit Automotive Realdaten*, Karlsruher Institut für Technologie, Bachelorarbeit, 2024

[S6] Wagner, Niklas: *Gestaltung eines LLM-basierten Datenassistenten mit integrierter Lernunterstützung für Unternehmen*, Karlsruher Institut für Technologie, Masterarbeit, 2025

[S7] Haverkamp, Justus: *Evaluation der Einsatzstrategien generativer KI im Requirements Engineering*, Karlsruher Institut für Technologie, Masterarbeit, 2025

Literaturverzeichnis

[1] Elsevier, „Scopus (Datenbank),” abgerufen am 06.02.2025. [Online]. Verfügbar: https://www.scopus.com/

[2] „Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles,” SAE-International, Standard J3016. [Online]. Verfügbar: https://www.sae.org/standards/content/j3016_202104/

[3] A. Leslie und D. Murray, „An Analysis of the Operational Costs of Trucking: 2023 Update,” 2023.

[4] A. Schmid, „Leere Supermarktregale: 120.000 Fahrer Fehlen,” Sep. 2024. [Online]. Verfügbar: https://www.fr.de/politik/bus-fahrer-spedition-visum-arbeitsvisum-bezahlung-deutschland-fachkraeftemangel-arbeit-jobs-lkw-zr-93298334.html

[5] NHTSA National Highway Traffic Safety Administration, „Automated Vehicles for Safety,” abgerufen am 05.11.2021. [Online]. Verfügbar: https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety#faq-60396

[6] B. Friedrich, „The Effect of Autonomous Vehicles on Traffic,” in *Autonomous Driving*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016.

[7] J. Guo, U. Kurup, und M. Shah, „Is It Safe to Drive? An Overview of Factors, Challenges, and Datasets for Driveability Assessment in Autonomous Driving,” *arXiv:1811.11277 [cs]*, Nov. 2018. [Online]. Verfügbar: http://arxiv.org/abs/1811.11277

[8] H. Yin und C. Berger, „When to Use What Data Set for Your Self-Driving Car Algorithm: An Overview of Publicly Available Driving Datasets," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. Yokohama: IEEE, Okt. 2017, S. 1–8. [Online]. Verfügbar: http://ieeexplore.ieee.org/document/8317828/

[9] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn, T. Fernandez, M. Jänicke, S. Mirashi, C. Savani, M. Sturm, O. Vorobiov, M. Oelker, S. Garreis, und P. Schuberth, „A2D2: Audi Autonomous Driving Dataset," *arXiv:2004.06320 [cs, eess]*, Apr. 2020. [Online]. Verfügbar: http://arxiv.org/abs/2004.06320

[10] M. M. Vopson, „Estimation of the Information Contained in the Visible Matter of the Universe," *AIP Advances*, Vol. 11, Nr. 10, S. 105317, Okt. 2021. [Online]. Verfügbar: https://aip.scitation.org/doi/10.1063/5.0064475

[11] A. Geiger, P. Lenz, C. Stiller, und R. Urtasun, „Vision Meets Robotics: The KITTI Dataset," *The International Journal of Robotics Research*, Vol. 32, Nr. 11, S. 1231–1237, Sep. 2013. [Online]. Verfügbar: http://journals.sagepub.com/doi/10.1177/0278364913491297

[12] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, und T. Darrell, „BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning," *arXiv:1805.04687 [cs]*, Apr. 2020. [Online]. Verfügbar: http://arxiv.org/abs/1805.04687

[13] H. Winner, S. Hakuli, F. Lotz, und C. Singer, Hrsgg., *Handbuch Fahrerassistenzsysteme: Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort*. Wiesbaden: Springer Fachmedien Wiesbaden, 2015. [Online]. Verfügbar: http://link.springer.com/10.1007/978-3-658-05734-3

[14] S. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. Eng, D. Rus, und M. Ang, „Perception, Planning, Control, and Coordination for Autonomous Vehicles," *Machines*, Vol. 5, Nr. 1, S. 6, Feb. 2017. [Online]. Verfügbar: http://www.mdpi.com/2075-1702/5/1/6

[15] J. Leonard und H. Durrant-Whyte, „Simultaneous Map Building and Localization for an Autonomous Mobile Robot," in *Proceedings IROS '91:IEEE/RSJ International Workshop on Intelligent Robots and Systems '91*. Osaka, Japan: IEEE, 1991, S. 1442–1447. [Online]. Verfügbar: http://ieeexplore.ieee.org/document/174711/

[16] „PEGASUS Gesamtmethode," Juni 2022. [Online]. Verfügbar: https://www.pegasusprojekt.de/files/tmpl/Pegasus-Abschlussveranstaltung/PEGASUS-Gesamtmethode.pdf

[17] PEGASUS-Projekt, „Scenario Description and Knowledge-Based Scenario Generation," 2019.

[18] Y. Kang, H. Yin, und C. Berger, „Test Your Self-Driving Algorithm: An Overview of Publicly Available Driving Datasets and Virtual Testing Environments," *IEEE Transactions on Intelligent Vehicles*, Vol. 4, Nr. 2, S. 171–185, Juni 2019. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/8667012/

[19] P. Ji, L. Ruan, Y. Xue, L. Xiao, und Q. Dong, „Perspective, Survey and Trends: Public Driving Datasets and Toolsets for Autonomous Driving Virtual Test," *arXiv:2104.00273 [cs]*, Apr. 2021. [Online]. Verfügbar: http://arxiv.org/abs/2104.00273

[20] B. Wolfe, B. D. Sawyer, und R. Rosenholtz, „Toward a Theory of Visual Information Acquisition in Driving," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, Vol. 64, Nr. 4, S. 694–713, Juni 2022. [Online]. Verfügbar: http://journals.sagepub.com/doi/10.1177/0018720820939693

[21] E. Marti, M. A. de Miguel, F. Garcia, und J. Perez, „A Review of Sensor Technologies for Perception in Automated Driving," *IEEE Intelligent Transportation Systems Magazine*, Vol. 11, Nr. 4, S. 94–108, 2019. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/8846569/

[22] F. Kluge, „Kamera," in *Kluge*. Berlin, Boston: De Gruyter, 2012.

[23] A. Erhardt, *Einführung in die digitale Bildverarbeitung: Grundlagen, Systeme und Anwendungen ; mit 35 Beispielen und 44 Aufgaben*, 1. Aufl., Serie Studium. Wiesbaden: Vieweg + Teubner, 2008.

[24] A. Tsirikoglou, G. Eilertsen, und J. Unger, „A Survey of Image Synthesis Methods for Visual Machine Learning," *Computer Graphics Forum*, Vol. 39, Nr. 6, S. 426–451, Sep. 2020. [Online]. Verfügbar: https://onlinelibrary.wiley.com/doi/10.1111/cgf.14047

[25] R. D. Fiete, *Modeling the Imaging Chain of Digital Cameras*, Serie Tutorial Texts in Optical Engineering. Bellingham, Wash: SPIE Press, 2010, Nr. v. TT92.

[26] G. C. Holst und T. S. Lomheim, *CMOS/CCD Sensors and Camera Systems*, 2. Aufl. Winter Park, FL : Bellingham, Wash: JCD Publishing ; SPIE, 2011.

[27] J. Guerrero-Ibáñez, S. Zeadally, und J. Contreras-Castillo, „Sensor Technologies for Intelligent Transportation Systems," *Sensors*, Vol. 18, Nr. 4, S. 1212, Apr. 2018. [Online]. Verfügbar: http://www.mdpi.com/1424-8220/18/4/1212

[28] I. J. Xique, W. Buller, Z. B. Fard, E. Dennis, und B. Hart, „Evaluating Complementary Strengths and Weaknesses of ADAS Sensors," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. Chicago, IL, USA: IEEE, Aug. 2018, S. 1–5. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/8690901/

[29] Merriam-Webster.com, „Metadata Definition & Meaning - Merriam-Webster," abgerufen am 03.10.2022. [Online]. Verfügbar: https://www.merriam-webster.com/dictionary/metadata

[30] A. Brand, F. Daly, und B. Meyers, *Metadata Demystified: A Guide for Publishers*. Bethesda, Md., Hanover, Pa.: NISO Press ; Sheridan Press, 2003.

[31] G. Alemu, „A Theory of Metadata Enriching and Filtering," *Libri*, Vol. 66, Nr. 4, Jan. 2016. [Online]. Verfügbar: https://www.degruyter.com/document/doi/10.1515/libri-2016-0109/html

[32] P. Petersen, H. Stage, J. Langner, L. Ries, P. Rigoll, C. Philipp Hohl, und E. Sax, „Towards a Data Engineering Process in Data-Driven Systems Engineering," in *2022 IEEE International Symposium on Systems Engineering (ISSE)*. Vienna, Austria: IEEE, Okt. 2022, S. 1–8. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/10005441/

[33] C. King, T. Braun, C. Braess, J. Langner, und E. Sax, „Capturing the Variety of Urban Logical Scenarios from Bird-view Trajectories:," in *Proceedings of the 7th International Conference on Vehicle Technology and Intelligent Transport Systems*. Online Streaming, — Select a Country —: SCITEPRESS - Science and Technology Publications, 2021, S. 471–480. [Online]. Verfügbar: https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0010441204710480

[34] J. Schiewe, *Kartographie: Visualisierung georäumlicher Daten*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2022. [Online]. Verfügbar: https://link.springer.com/10.1007/978-3-662-65441-5

[35] T. M. Mitchell, *Machine Learning*, Serie McGraw-Hill Series in Computer Science. New York: McGraw-Hill, 1997.

[36] K. Steinbuch, „Die Lernmatrix," *Kybernetik*, Vol. 1, Nr. 1, S. 36–45, Jan. 1961. [Online]. Verfügbar: https://link.springer.com/10.1007/BF00293853

[37] I. Goodfellow, *Deep Learning*, 2016.

[38] F. Rosenblatt, „The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain." *Psychological Review*, Vol. 65, Nr. 6, S. 386–408, 1958. [Online]. Verfügbar: http://doi.apa.org/getdoi.cfm?doi=10.1037/h0042519

[39] m0nhawk, „TikZ: Diagram of a Perceptron," Nov. 2022. [Online]. Verfügbar: https://tex.stackexchange.com/a/104376

[40] D. Sonnet, *Neuronale Netze kompakt: Vom Perceptron zum Deep Learning*, Serie IT kompakt. Wiesbaden: Springer Fachmedien Wiesbaden, 2022. [Online]. Verfügbar: https://link.springer.com/10.1007/978-3-658-29081-8

[41] W. Ertel, *Grundkurs Künstliche Intelligenz: Eine praxisorientierte Einführung*, Serie Computational Intelligence. Wiesbaden: Springer Fachmedien Wiesbaden, 2021. [Online]. Verfügbar: https://link.springer.com/10.1007/978-3-658-32075-1

[42] H. Alokasi und M. B. Ahmad, „Deep Learning-Based Frameworks for Semantic Segmentation of Road Scenes," *Electronics*, Vol. 11, Nr. 12, S. 1884, Juni 2022. [Online]. Verfügbar: https://www.mdpi.com/2079-9292/11/12/1884

[43] J. Long, E. Shelhamer, und T. Darrell, „Fully Convolutional Networks for Semantic Segmentation," 2015.

[44] K. He, G. Gkioxari, P. Dollar, und R. Girshick, „Mask R-CNN," in *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice: IEEE, Okt. 2017, S. 2980–2988. [Online]. Verfügbar: http://ieeexplore.ieee.org/document/8237584/

[45] A. Kirillov, K. He, R. Girshick, C. Rother, und P. Dollar, „Panoptic Segmentation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Long Beach, CA, USA: IEEE, Juni 2019, S. 9396–9405. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/8953237/

[46] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, und I. Sutskever, „Learning Transferable Visual Models From Natural Language Supervision," S. 16, 2021.

[47] K. Sohn, „Improved Deep Metric Learning with Multi-class N-pair Loss Objective," 2016.

[48] E. F. Codd, „A Relational Model of Data for Large Shared Data Banks," Vol. 13, Nr. 6, 1970.

[49] „ISO/IEC 9075 Information Technology - Database Languages - SQL," 2023.

[50] „ISO/IEC 13249 Information Technology — Database Languages — SQL Multimedia and Application Packages," 2016.

[51] N. Jatana, S. Puri, M. Ahuja, I. Kathuria, und D. Gosain, „A Survey and Comparison of Relational and Non-Relational Database," *International Journal of Engineering Research*, Vol. 1, Nr. 6, 2012.

[52] Q. S. GmbH, „Qdrant." [Online]. Verfügbar: https://qdrant.tech

[53] J. Wang, X. Yi, R. Guo, H. Jin, P. Xu, S. Li, X. Wang, X. Guo, C. Li, X. Xu, K. Yu, Y. Yuan, Y. Zou, J. Long, Y. Cai, Z. Li, Z. Zhang, Y. Mo, J. Gu, R. Jiang, Y. Wei, und C. Xie, „Milvus: A Purpose-Built Vector Data Management System," in *Proceedings of the 2021 International Conference on Management of Data*. Virtual Event China: ACM, Juni 2021, S. 2614–2627. [Online]. Verfügbar: https://dl.acm.org/doi/10.1145/3448016.3457550

[54] Rentong Guo, Li Liu, Chun Han, Ting Wang, Yuchen Gao, Jie Zeng, Chao Gao, Filip Haltmayer, und Xiaofan Luan, „Milvus Performance Evaluation 2023 - Technical Paper," Feb. 2023. [Online]. Verfügbar: https://zilliz.com/resources/whitepaper/milvus-performance-benchmark

[55] W. Dröschel, Hrsg., *Das V-Modell 97: der Standard für die Entwicklung von IT-Systemen mit Anleitung für den Praxiseinsatz*. München Wien: Oldenbourg, 2000.

[56] W. W. Royce, „Managing the Development of Large Software Systems: Concepts and Techniques," *ICSE '87: Proceedings of the 9th international conference on Software Engineering*, 1987.

[57] C. Haubelt und J. Teich, *Digitale Hardware/Software-Systeme*, Serie eXamen.press. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, Vol. 0. [Online]. Verfügbar: http://link.springer.com/10.1007/978-3-642-05356-6

[58] „ISO/IEC/IEEE 24765 International Standard - Systems and Software Engineering –Vocabulary," 2017. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/8016712/

[59] R. Zhang, A. Albrecht, J. Kausch, H. J. Putzer, T. Geipel, und P. Halady, „DDE Process: A Requirements Engineering Approach for Machine Learning in Automated Driving," in *2021 IEEE 29th International Requirements Engineering Conference (RE)*. Notre Dame, IN, USA: IEEE, Sep. 2021, S. 269–279. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/9604596/

[60] F. Reisgys, J. Plaum, A. Schwarzhaupt, und E. Sax, „Scenario-Based X-in-the-Loop Test for Development of Driving Automation," 2022.

[61] „Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," *SAE International Journal of Connected and Automated Vehicles*, Vol. 3, Nr. 1, S. 12–03–01–0003, Feb. 2020. [Online]. Verfügbar: https://www.sae.org/content/12-03-01-0003/

[62] „ISO/IEC 25012 Software Engineering - Software Product Quality Requirements and Evaluation (SQuaRE) - Data Quality Model," Dez. 2008.

[63] A. Schmidt, „Interactive Human Centered Artificial Intelligence: A Definition and Research Challenges," in *Proceedings of the International Conference on Advanced Visual Interfaces*. Salerno Italy: ACM, Sep. 2020, S. 1–4. [Online]. Verfügbar: https://dl.acm.org/doi/10.1145/3399715.3400873

[64] A. Dosovitskiy, „CARLA: An Open Urban Driving Simulator," S. 16, 2017.

[65] R. Dona und B. Ciuffo, „Virtual Testing of Automated Driving Systems. A Survey on Validation Methods," *IEEE Access*, Vol. 10, S. 24 349–24 367, 2022. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/9718588/

[66] H. Bossel, *Modellbildung und Simulation: Konzepte, Verfahren und Modelle zum Verhalten dynamischer Systeme. Ein Lehr- und Arbeitsbuch*. Wiesbaden: Vieweg+Teubner Verlag, 1994. [Online]. Verfügbar: http://link.springer.com/10.1007/978-3-322-90519-2

[67] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, und I. Polosukhin, „Attention Is All You Need," Dez. 2017. [Online]. Verfügbar: http://arxiv.org/abs/1706.03762

[68] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, und Y. Bengio, „Generative Adversarial Nets," S. 9, 2014.

[69] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, und B. Ommer, „High-Resolution Image Synthesis with Latent Diffusion Models," Apr. 2022. [Online]. Verfügbar: http://arxiv.org/abs/2112.10752

[70] A. Hu, L. Russell, H. Yeo, Z. Murez, G. Fedoseev, A. Kendall, J. Shotton, und G. Corrado, „GAIA-1: A Generative World Model for Autonomous Driving," Sep. 2023. [Online]. Verfügbar: http://arxiv.org/abs/2309.17080

[71] P. Rigoll, P. Petersen, L. Ries, J. Langner, und E. Sax, „Augmentation von Kameradaten Mit Generative Adversarial Networks (GANs) Zur Absicherung Automatisierter Fahrfunktionen," in *Fahrerassistenzsysteme Und Automatisiertes Fahren*, VDI Wissensforum GmbH, Hrsg. VDI Verlag, 2022, S. 41–48. [Online]. Verfügbar: https://elibrary.vdi-verlag.de/index.php?doi=10.51202/9783181023945-41

[72] Z. Guo, Y. Yu, und C. Gou, „Controllable Diffusion Models for Safety-Critical Driving Scenario Generation," in *2023 IEEE 35th International Conference on Tools with Artificial Intelligence (ICTAI)*. Atlanta, GA, USA: IEEE, Nov. 2023, S. 717–722. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/10356547/

[73] T. Yu, R. Feng, R. Feng, J. Liu, X. Jin, W. Zeng, und Z. Chen, „Inpaint Anything: Segment Anything Meets Image Inpainting," Apr. 2023. [Online]. Verfügbar: http://arxiv.org/abs/2304.06790

[74] X. Zhang, N. Tseng, A. Syed, R. Bhasin, und N. Jaipuria, „SIMBAR: Single Image-Based Scene Relighting For Effective Data Augmentation For Automated Driving Vision Tasks," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA: IEEE, Juni 2022, S. 3708–3718. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/9879042/

[75] P. Rigoll, P. Petersen, J. Langner, und E. Sax, „Parameterizable Lidar-Assisted Traffic Sign Placement for the Augmentation of Driving Situations with CycleGAN," in *Advances in Systems Engineering*, L. Borzemski, H. Selvaraj, und J. Świątek, Hrsgg. Cham: Springer International Publishing, 2022, Vol. 364, S. 403–417. [Online]. Verfügbar: https://link.springer.com/10.1007/978-3-030-92604-5_36

[76] F. Reway, A. Hoffmann, D. Wachtel, W. Huber, A. Knoll, und E. Ribeiro, „Test Method for Measuring the Simulation-to-Reality Gap of Camera-based Object Detection Algorithms for Autonomous Driving," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. Las Vegas, NV, USA: IEEE, Okt. 2020, S. 1249–1256. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/9304567/

[77] M. Alkhawlani und M. Elmogy, „Text-Based, Content-based, and Semantic-based Image Retrievals: A Survey," Vol. 04, Nr. 01, 2015.

[78] Y. Alemu, J.-b. Koh, M. Ikram, und D.-K. Kim, „Image Retrieval in Multimedia Databases: A Survey," in *2009 Fifth International Conference on Intelligent Information Hiding and Multimedia Signal Processing*. Kyoto, Japan: IEEE, Sep. 2009, S. 681–689. [Online]. Verfügbar: http://ieeexplore.ieee.org/document/5337432/

[79] M. Linkert, C. T. Rueden, C. Allan, J.-M. Burel, W. Moore, A. Patterson, B. Loranger, J. Moore, C. Neves, D. MacDonald, A. Tarkowska, C. Sticco, E. Hill, M. Rossner, K. W. Eliceiri, und J. R. Swedlow, „Metadata Matters: Access to Image Data in the Real World," *Journal of Cell Biology*, Vol. 189,

Nr. 5, S. 777–782, Mai 2010. [Online]. Verfügbar: https://rupress.org/jcb/article/189/5/777/35828/Metadata-matters-access-to-image-data-in-the-real

[80] J. Tesic, „Metadata Practices for Consumer Photos," *IEEE MultiMedia*, Vol. 12, Nr. 3, S. 86–92, Juli 2005. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/1490501/

[81] M. Stefanini, M. Cornia, L. Baraldi, S. Cascianelli, G. Fiameni, und R. Cucchiara, „From Show to Tell: A Survey on Deep Learning-Based Image Captioning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 45, Nr. 1, S. 539–559, Jan. 2023. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/9706348/

[82] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, und H. Laga, „A Comprehensive Survey of Deep Learning for Image Captioning," *ACM Computing Surveys*, Vol. 51, Nr. 6, S. 1–36, Nov. 2019. [Online]. Verfügbar: https://dl.acm.org/doi/10.1145/3295748

[83] S. Iyer, S. Chaturvedi, und T. Dash, „Image Captioning-Based Image Search Engine: An Alternative to Retrieval by Metadata," in *Soft Computing for Problem Solving*, J. C. Bansal, K. N. Das, A. Nagar, K. Deep, und A. K. Ojha, Hrsgg. Singapore: Springer Singapore, 2019, Vol. 817, S. 181–191. [Online]. Verfügbar: http://link.springer.com/10.1007/978-981-13-1595-4_14

[84] M. Naito, C. Miyajima, T. Nishino, N. Kitaoka, und K. Takeda, „A Browsing and Retrieval System for Driving Data," in *2010 IEEE Intelligent Vehicles Symposium*. La Jolla, CA, USA: IEEE, Juni 2010, S. 1159–1165. [Online]. Verfügbar: http://ieeexplore.ieee.org/document/5547999/

[85] L. Klitzke, C. Koch, A. Haja, und F. Köster, „Real-World Test Drive Vehicle Data Management System for Validation of Automated Driving Systems:," in *Proceedings of the 5th International Conference on Vehicle Technology and Intelligent Transport Systems*. Heraklion, Crete, Greece: SCITEPRESS - Science and Technology Publications, 2019, S. 171–180. [Online]. Verfügbar: https://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0007720501710180

[86] F. Heidecker, T. Susetzky, E. Fuchs, und B. Sick, „Context Information for Corner Case Detection in Highly Automated Driving," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*. Bilbao, Spain: IEEE, Sep. 2023, S. 1522–1529. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/10422414/

[87] P. Elspas, J. Langner, M. Aydinbas, J. Bach, und E. Sax, „Leveraging Regular Expressions for Flexible Scenario Detection in Recorded Driving Data," in *2020 IEEE International Symposium on Systems Engineering (ISSE)*. Vienna, Austria: IEEE, Okt. 2020, S. 1–8. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/9272025/

[88] T. Deselaers, D. Keysers, und H. Ney, „Features for Image Retrieval: An Experimental Comparison," *Information Retrieval*, Vol. 11, Nr. 2, S. 77–107, Apr. 2008. [Online]. Verfügbar: http://link.springer.com/10.1007/s10791-007-9039-3

[89] F. Korn, N. Sidiropoulos, und C. Faloutsos, „Fast Nearest Neighbor Search in Medical Image Databases," Okt. 1998.

[90] R. Kapoor, D. Sharma, und T. Gulati, „State of the Art Content Based Image Retrieval Techniques Using Deep Learning: A Survey," *Multimedia Tools and Applications*, Vol. 80, Nr. 19, S. 29 561–29 583, Aug. 2021. [Online]. Verfügbar: https://link.springer.com/10.1007/s11042-021-11045-1

[91] M. S. Lew, N. Sebe, C. Djeraba, und R. Jain, „Content-Based Multimedia Information Retrieval: State of the Art and Challenges," *ACM Transactions on Multimedia Computing, Communications, and Applications*, Vol. 2, Nr. 1, S. 1–19, Feb. 2006. [Online]. Verfügbar: https://dl.acm.org/doi/10.1145/1126004.1126005

[92] D. Bogdoll, M. Nitsche, und J. M. Zollner, „Anomaly Detection in Autonomous Driving: A Survey," S. 12, 2021.

[93] F. Heidecker, M. Bieshaar, und B. Sick, „Corner Cases in Machine Learning Processes," *AI Perspectives & Advances*, Vol. 6, Nr. 1, S. 1, Jan.

2024. [Online]. Verfügbar: https://aiperspectives.springeropen.com/articles/10.1186/s42467-023-00015-y

[94] Y. Shoeb, R. Chan, G. Schwalbe, A. Nowzard, F. Güney, und H. Gottschalk, „Have We Ever Encountered This Before? Retrieving Out-of-Distribution Road Obstacles from Driving Scenes," Sep. 2023. [Online]. Verfügbar: http://arxiv.org/abs/2309.04302

[95] X. Zhang, J. Tao, K. Tan, M. Törngren, J. M. G. Sánchez, M. R. Ramli, X. Tao, M. Gyllenhammar, F. Wotawa, N. Mohan, M. Nica, und H. Felbinger, „Finding Critical Scenarios for Automated Driving Systems: A Systematic Literature Review," Okt. 2021. [Online]. Verfügbar: http://arxiv.org/abs/2110.08664

[96] H. M. Nguyen, V. Thanh The, und T. V. Lang, „A Method of Semantic-Based Image Retrieval Using Graph Cut," *Journal of Computer Science and Cybernetics*, Vol. 38, Nr. 2, S. 193–212, Juni 2022. [Online]. Verfügbar: https://vjs.ac.vn/index.php/jcc/article/view/16786

[97] Merantix, „Natural Language Search," abgerufen am 06.03.2024. [Online]. Verfügbar: https://nucleus.scale.com/docs/natural-language-search

[98] G. Hess, A. Tonderski, C. Petersson, K. Åström, und L. Svensson, „LidarCLIP or: How I Learned to Talk to Point Clouds," Mai 2023. [Online]. Verfügbar: http://arxiv.org/abs/2212.06858

[99] S. Sai Gannamaneni, A. Sadaghiani, R. Prakash Rao, M. Mock, und M. Akila, „Investigating CLIP Performance for Meta-data Generation in AD Datasets," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Vancouver, BC, Canada: IEEE, Juni 2023, S. 3840–3850. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/10208693/

[100] H. Stage, L. Ewecker, J. Langner, T. S. Sohn, T. Villmann, und E. Sax, „Reducing Computer Vision Dataset Size via Selective Sampling," in *2023 IEEE 26th International Conference on Intelligent Transportation*

Systems (ITSC). Bilbao, Spain: IEEE, Sep. 2023, S. 1422–1428. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/10422460/

[101] K. Zhou, J. Yang, C. C. Loy, und Z. Liu, „Learning to Prompt for Vision-Language Models," *International Journal of Computer Vision*, Vol. 130, Nr. 9, S. 2337–2348, Sep. 2022. [Online]. Verfügbar: https://link.springer.com/10.1007/s11263-022-01653-1

[102] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, und R. Ranftl, „Language-Driven Semantic Segmentation," Apr. 2022. [Online]. Verfügbar: http://arxiv.org/abs/2201.03546

[103] T. Lüddecke und A. S. Ecker, „Image Segmentation Using Text and Image Prompts," Mar. 2022. [Online]. Verfügbar: http://arxiv.org/abs/2112.10003

[104] J. Li, D. Li, S. Savarese, und S. Hoi, „BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models," Juni 2023. [Online]. Verfügbar: http://arxiv.org/abs/2301.12597

[105] V. Ostankovich, R. Yagfarov, M. Rassabin, und S. Gafurov, „Application of CycleGAN-based Augmentation for Autonomous Driving at Night," in *2020 International Conference Nonlinearity, Information and Robotics (NIR)*. Innopolis, Russia: IEEE, Dez. 2020, S. 1–5. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/9290218/

[106] I. Reda und A. Andreas, „Solar Position Algorithm for Solar Radiation Applications (Revised)," Tech. Rep. NREL/TP-560-34302, 15003974, Jan. 2008. [Online]. Verfügbar: http://www.osti.gov/servlets/purl/15003974/

[107] OpenStreetMap contributors, „Planet Dump Retrieved from https://planet.osm.org." [Online]. Verfügbar: https://www.openstreetmap.org

[108] W. Chen, Y. Liu, W. Wang, E. M. Bakker, T. Georgiou, P. Fieguth, L. Liu, und M. S. Lew, „Deep Learning for Instance Retrieval: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 45, Nr. 6, S. 7270–7292, Juni 2023. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/9933854/

[109] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, und R. Girdhar, „Masked-Attention Mask Transformer for Universal Image Segmentation," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA: IEEE, Juni 2022, S. 1280–1289. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/9878483/

[110] National Institute of Standards and Technology (NIST), „COSINE DISTANCE, COSINE SIMILARITY, ANGULAR COSINE DISTANCE, ANGULAR COSINE SIMILARITY," abgerufen am 08.03.2023. [Online]. Verfügbar: https://www.itl.nist.gov/div898/software/dataplot/refman2/auxillar/cosdist.htm

[111] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, und I. Sutskever, „Learning Transferable Visual Models From Natural Language Supervision," Feb. 2021. [Online]. Verfügbar: http://arxiv.org/abs/2103.00020

[112] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, und B. Schiele, „The Cityscapes Dataset for Semantic Urban Scene Understanding," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, Juni 2016, S. 3213–3223. [Online]. Verfügbar: http://ieeexplore.ieee.org/document/7780719/

[113] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, und B. Guo, „Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, Okt. 2021, S. 9992–10 002. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/9710580/

[114] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, und Li Fei-Fei, „ImageNet: A Large-Scale Hierarchical Image Database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Miami, FL: IEEE, Juni 2009, S. 248–255. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/5206848/

[115] D. Bogdoll, F. Schreyer, und J. M. Zöllner, „Ad-Datasets: A Meta-Collection of Data Sets for Autonomous Driving," in *Proceedings of the 8th International Conference on Vehicle Technology and Intelligent Transport Systems*, 2022, S. 46–56. [Online]. Verfügbar: http://arxiv.org/abs/2202.01909

[116] C. Sakaridis, D. Dai, und L. Van Gool, „ACDC: The Adverse Conditions Dataset with Correspondences for Semantic Driving Scene Understanding," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, Okt. 2021, S. 10 745–10 755. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/9711067/

[117] J. Krause, M. Stark, J. Deng, und L. Fei-Fei, „3D Object Representations for Fine-Grained Categorization," in *2013 IEEE International Conference on Computer Vision Workshops*. Sydney, Australia: IEEE, Dez. 2013, S. 554–561. [Online]. Verfügbar: http://ieeexplore.ieee.org/document/6755945/

[118] M. Palatucci, D. Pomerleau, G. Hinton, und T. M. Mitchell, „Zero-Shot Learning with Semantic Output Codes," 2009.

[119] D. Liu, „Progressive Multi-Task Anti-Noise Learning and Distilling Frameworks for Fine-Grained Vehicle Recognition," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 25, Nr. 9, S. 10 667–10 678, Sep. 2024. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/10623841/

[120] S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang, und W. Shi, „Edge Computing for Autonomous Driving: Opportunities and Challenges," *Proceedings of the IEEE*, Vol. 107, Nr. 8, S. 1697–1716, Aug. 2019. [Online]. Verfügbar: https://ieeexplore.ieee.org/document/8744265/

[121] A. Bruno, D. Moroni, und M. Martinelli, „Efficient Adaptive Ensembling for Image Classification," *Expert Systems*, Vol. 42, Nr. 1, S. e13424, Jan. 2025. [Online]. Verfügbar: http://arxiv.org/abs/2206.07394