

Fakultät für Maschinenwesen

**Entwicklung eines kamerabasierten Kollisionsgefahrerkennungssystems  
zur Assistenz von Gabelstaplerfahrern unter Einsatz von Computer-Vision  
und Machine-Learning**

Armin Fredrik Lang

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines Doktors -Ingenieurs (Dr.-Ing.) genehmigten Dissertation.

Vorsitzende/-r: Prof. Dr. Thomas Sattelmayer

Prüfende/-r der Dissertation:

1. Prof. Dr.-Ing. Johannes Fottner
2. Prof. Dr.-Ing. Michael Freitag

Die Dissertation wurde am 15.02.2021 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 17.08.2021 angenommen.

## Danksagung

---

Während der Forschungsarbeit standen mir Kolleginnen und Kollegen jederzeit fachlich zur Seite bei denen ich mich an dieser Stelle bedanke. Besonders erwähnt seien hier: Andreas Ebner, Dirk Kauke, Matthias Jung, Markus Klevers, Johannes Lechner und Seungyong Oh. Danken möchte ich außerdem meinen Studenten, die mich sowohl in Form von Studienarbeiten als auch in Form von wissenschaftlichen Hilfskräften sehr unterstützt haben.

Ein besonderer Dank gilt meinen Eltern Alfred und Jutta Lang, die mich sowohl vor der Dissertation als auch währenddessen tatkräftig unterstützt haben.

## Kurzfassung

---

Im intralogistischen Transport überschneiden sich oft die Arbeitsbereiche von Menschen und Flurförderzeugen. Der hohe Arbeitsdruck von Gabelstaplerfahrern birgt die Gefahr der Minderung der Achtsamkeit, wodurch in Kombination mit den engen Verhältnissen in Lagern gefährliche Situationen entstehen, sowohl für den Gabelstaplerfahrer als auch für Personen, welche sich im Fahrbereich von Staplern aufhalten. Neben der Gefährdung von Personen ist auch die Gefahr der Beschädigung von Lager-einrichtung nicht unerheblich. Zur Vermeidung von Unfällen sind bereits diverse Kollisionswarnsysteme erhältlich. Allerdings haben diese den großen Nachteil, dass zur Entscheidung, ob eine Gefahr vorliegt, nur der Abstand zum nächsten Hindernis als Entscheidungsgrundlage betrachtet wird. Dies führt aufgrund geringer Distanzen des Staplers zur Umgebung zu vielen Falschwarnungen, wodurch die Akzeptanz dieser Systeme entsprechend gering ist.

In dieser Arbeit wird daher ein neuer Ansatz untersucht, bei dem nicht mehr der Abstand als Kriterium für eine Kollisionsgefahr verwendet wird. Denn bewegt sich ein Gabelstapler nur langsam auf ein Hindernis zu, ist die Kollisionsgefahr trotz der niedrigen Distanz gering. Daher wird im vorliegenden Ansatz ein System entwickelt, welches die Bewegungsvektoren zwischen dem Stapler und seiner Umgebung analysiert.

Der höchste Detailgrad in der Abbildung der Umgebung wird aktuell mit 3D-Kameras erreicht. Diese liefern sowohl die Tiefendaten der Umgebung als auch Informationen über deren Textur. Anhand dieser Daten berechnet das entwickelte System mittels Methoden des maschinellen Sehens die Bewegungsvektoren der Umgebung in Relation zum Gabelstapler. Aus diesen Vektoren kann für jedes Objekt die Zeit bis zu einer etwaigen Kollision berechnet werden.

Entsprechend ist in diesem Fall nicht der Abstand das Warnkriterium, sondern die berechnete Kollisionszeit. Neben der geschwindigkeitsabhängigen Bewertung von Gefahren ist auch der Einbezug der Richtung eine Neuerung. Fährt der Stapler nahe an einem Objekt vorbei, wird diese Situation nicht als gefährlich eingestuft. Eine weitere Neuerung in dem Ansatz findet sich in der Einstufung von etwaigen Kollisionsgefahren als solche mit und ohne Gefahr für Personen. Dies erlaubt die Verwendung unterschiedlicher Grenzwerte für die Zeit zwischen der Warnung und einer etwaigen Kollision. Vor Kollisionen mit Menschen kann entsprechend früher gewarnt werden.

## Abstract

---

In intralogistic transport, the work areas of people and industrial trucks often overlap. The tight conditions in warehouses in combination with high working pressure sometimes lead to inattentiveness of forklift drivers. This creates dangerous situations, both for the forklift driver and for people who are in the driving area of forklifts. In addition to endangering people, the risk of damage to storage facilities is also significant. Various collision warning systems are already available to avoid accidents. However, these have the major disadvantage that when deciding whether there is a danger, only the distance to the next obstacle is taken as the basis for the decision. Due to the tight situation in bearings, this leads to many false warnings, which means that the acceptance of these systems is correspondingly low.

In this work, therefore, a new approach is examined in which the distance is no longer used as a criterion for a collision risk. Because if a forklift moves slowly towards an obstacle, the risk of collision is low despite the short distance. Therefore, in this approach, a system is developed which analyzes the motion vectors between the truck and its surroundings. The highest level of detail in the mapping of the surroundings is currently achieved with 3D cameras. These provide both the depth data of the environment and information about its texture. On the basis of this data, the developed system uses machine vision methods to calculate the motion vectors of the environment in relation to the forklift. The time of a possible collision can be calculated for each object from these vectors. Accordingly, in this case the distance is not the warning criterion, but the calculated collision time. In addition to the speed-dependent assessment of hazards, the inclusion of the direction is also an innovation. If the truck passes close to an object, this situation is not classified as dangerous.

Another innovation in the approach is the classification of possible collision risks as such with and without danger to people. This allows the use of different limit values for the maximum collision time at which an impending collision is considered a danger. Accordingly, warnings can be given earlier about collisions with people.





# Inhaltsverzeichnis

---

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Ausgangssituation	2
1.2	Zielsetzung	3
1.3	Vorgehensweise	4
<b>2</b>	<b>Stand der Technik und Forschung</b>	<b>7</b>
2.1	Sensortechnologien zur Erfassung von Kollisionen	7
2.2	Kollisionswarnsysteme für Gabelstapler	12
2.3	Maschinelles Lernen	15
2.3.1	Support-Vector-Machines (SVM)	17
2.3.2	(Künstliche) Neuronale Netze	19
2.4	Methoden der Bildverarbeitung	22
2.4.1	Basisoperationen	23
2.4.2	Bildmerkmale	26
2.4.3	Maschinelles Lernen in der Bildverarbeitung	30
2.5	Bewertung des Stands der Technik	33
2.5.1	Kollisionswarnsysteme für Gabelstapler	34
2.5.2	Sensortechnologien	35
2.6	Ableitung der Forschungsfrage	38
<b>3</b>	<b>Konzeptentwicklung</b>	<b>41</b>
3.1	Anforderungen an ein Fahrerassistenzsystem für Gabelstapler	41
3.1.1	Analyse von Unfalldaten	41
3.1.2	Richtlinien und Normen	43
3.1.3	Definition des Einsatzzwecks	45
3.1.4	Ableitung von Anforderungen	45
3.2	Hardwarekonzept	49
3.3	Softwarekonzept	50
3.4	Fazit	53
<b>4</b>	<b>Implementierung</b>	<b>55</b>
4.1	Hardwarespezifikation	55
4.2	Softwareentwicklung	59

4.2.1 Eingang (engl.: „input“)	61
4.2.2 Vorverarbeitung (engl.: „preprocessing“)	62
4.2.3 Segmentierung (engl.: „segmentation“)	69
4.2.4 Personenerkennung (engl.: „person detection“)	74
4.2.5 Kollisionsgefahrerkennung (engl.: „collision detection“)	76
4.2.6 Ausgabe (engl.: „output“)	81
4.3 Fazit	83
<b>5 Vorbereitende Maßnahmen zur Evaluation des Systems</b>	<b>85</b>
5.1 Identifikation von Einflussfaktoren	85
5.2 Identifikation relevanter Verkehrssituationen	87
5.3 Definition der Testszenarien und -fälle	88
5.3.1 Personenerkennung	88
5.3.2 Kollisionsgefahrerkennung	89
5.4 Trainings- und Testdaten	93
5.5 Fazit	101
<b>6 Evaluation einzelner Algorithmen</b>	<b>103</b>
6.1 Auswertung der Personenerkennung	103
6.1.1 Auswertungsmethode	104
6.1.2 Ergebnisse – Support-Vector-Machine	107
6.1.3 Ergebnisse – Neuronales Netz	135
6.2 Auswertung der Kollisionsgefahrerkennung	142
6.2.1 Auswertungsmethode	142
6.2.2 Ergebnisse	143
6.3 Fazit	158
<b>7 Evaluation des Gesamtsystems</b>	<b>161</b>
7.1 Kollisionen mit Personenschaden	161
7.2 Durchführung im Industrieunternehmen	165
7.2.1 Randbedingungen	165
7.2.2 Systemaufbau	166
7.2.3 Auswertungsmethode	167
7.2.4 Ergebnisse	171
7.3 Fazit	177
<b>8 Zusammenfassung und Ausblick</b>	<b>181</b>

<b>Literaturverzeichnis</b>	<b>185</b>
<b>Verzeichnis der betreuten Studienarbeiten</b>	<b>196</b>
<b>Abkürzungsverzeichnis</b>	<b>198</b>
<b>Abbildungsverzeichnis</b>	<b>201</b>
<b>Tabellenverzeichnis</b>	<b>207</b>
<b>Anhang A Ergebnisse der Evaluation einzelner Algorithmen</b>	<b>A-1</b>
<b>Anhang B Ergebnisse der Evaluation des Gesamtsystems</b>	<b>B-1</b>



# 1 Einleitung

---

Im intralogistischen Transport steigt zwar der Einsatz fahrerloser Transportfahrzeuge (FTF), allerdings steigt auch weiterhin der Verkauf manuell bedienter Gabelstapler [Sch-2017; Han-2016]. Gabelstapler haben den großen Vorteil, dass Transportaufträge schneller abgewickelt werden können, da ihre Geschwindigkeit höher ist als bei FTF [STI-2020a; STI-2020b]. Die Geschwindigkeit von FTF ist niedriger, da sie in jeglichen Situationen in kürzester Zeit abbremsen können müssen um vor allem Menschen nicht zu gefährden. Außerdem sind keine Technologien zur Lokalisierung und Steuerung insbesondere zur Lastaufnahme und -abgabe notwendig. Für diese Situationen werden bei FTF mehr Sensoren benötigt als es bei Gabelstaplern der Fall ist. Dadurch sind sie in der Anschaffung um ein Vielfaches teurer.

Eine Technologie, welche manuell bediente Gabelstapler wirklich ersetzen würde, wären autonome Transportfahrzeuge. Diese könnten mit einer ähnlichen Geschwindigkeit verkehren wie manuell bediente Gabelstapler. Die fortschreitende Entwicklung autonom fahrender Kraftfahrzeuge in der Automobilindustrie zeigt, dass autonom fahrende Fahrzeuge mittlerweile möglich sind. Allerdings gibt es hierzu noch wenige am Markt erhältlichen Lösungen im Bereich der Intralogistik. Gründe hierfür könnten unter anderem die hohen Sicherheitsanforderungen der Maschinenrichtlinie und die durchaus schwierigere Fahrsituationen in Hallen sein. Optische Sensoren haben bei Kraftfahrzeugen im Straßenverkehr eine deutlich höhere Sichtweite als im intralogistischen Verkehr. In Lagerhallen verhindern oft Regale oder Lagergüter eine uneingeschränkte Sicht. Darüber hinaus treten in der Intralogistik häufig niedrige Distanzen vom Fahrzeug zur Lagereinrichtung auf. In diesen Situationen muss ein autonomes Fahrzeug entscheiden können, ob eine Gefahr vorliegt oder ob es sich um eine normale Arbeitssituation handelt.

## 1.1 Ausgangssituation

Die Intralogistik ist trotz zahlreicher Fortschritte in der Automatisierung, wie z. B. vollautomatischen Lagern, weiterhin von manuellen Tätigkeiten geprägt. Das betrifft auch, wie einleitend erwähnt, den Transport von Gütern mittels Gabelstapler. Der Verkauf von manuell bedienten Gabelstaplern stieg in den letzten Jahren weiter, wie in Abbildung 1-1 zu sehen ist:

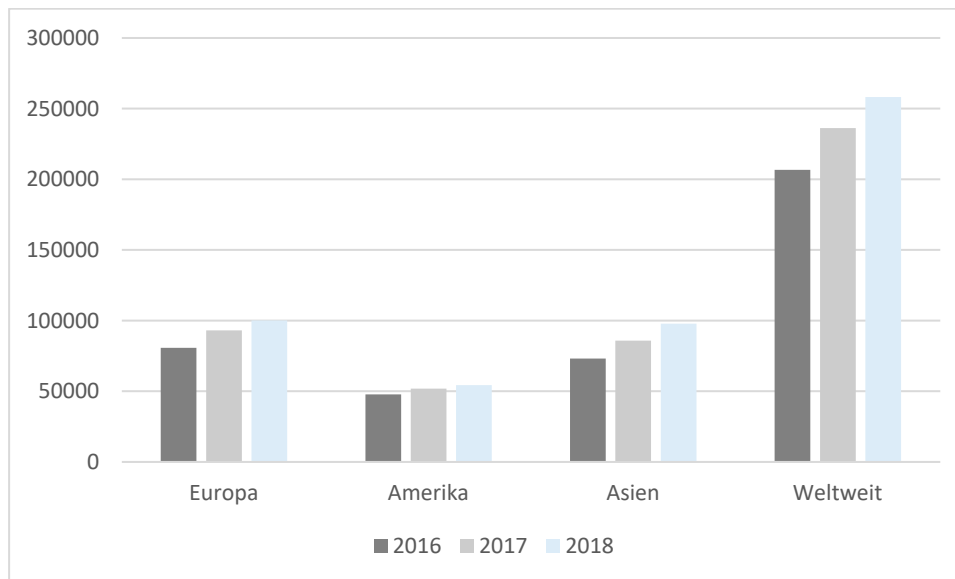


Abbildung 1-1: Entwicklung des Ordervolumens von manuell bedienten Gabelstaplern (Flurförderzeug Klasse 1) nach [EUR-2019].

Mit der steigenden Anzahl an Gabelstaplern steigt auch die Gefahr von Unfällen. In den Vereinigten Staaten ist nach dem US-amerikanischen Ministerium für Arbeitssicherheit fast jeder Gabelstapler im Laufe seines Lebenszyklus in einen Unfall mit Personenschaden verwickelt [Occ-2015]. Laut der Deutschen Gesetzlichen Unfallversicherung (DGUV) gab es von 2014 bis 2017 konstant mehr als 11.500 meldepflichtige Unfälle mit Personenschaden, bei denen Gabelstapler beteiligt waren [Deu-2015; Deu-2016; Deu-2017; Deu-2018].

Es gibt zwar verschiedene Produkte auf dem Markt, welche die Vermeidung dieser Unfälle zum Ziel haben, aber trotz alledem ist das Problem auch heute noch präsent. Die Produkte setzen auf Technologien, wie sie bereits seit langem zum Personenschutz auch in anderen Bereichen eingesetzt werden. Üblicherweise werden zu nahe Objekte über verschiedene Sensoren zur Distanzerkennung erkannt, wie z. B. Laserscanner oder Ultraschallsensoren. Der Einsatz dieser Technologien ist zwar hinsichtlich der Erkennung möglicher Gefahrenquellen sehr gut, allerdings führen sie auf-

grund der Beschränkung auf die Information der Distanz zu sehr vielen Falscherkennungen. Auch die Restriktion von Warnungen auf bestimmte Bereiche und Fahrsituationen (z. B. Warnungen nur bei einer Rückwärtsfahrt) kann Falscherkennungen nicht gänzlich vermeiden.

In der Bildverarbeitung wurden in den letzten Jahren enorme Fortschritte erreicht, vor allem im Bereich der Bildanalyse. Mithilfe von künstlicher Intelligenz wurde es möglich, nur anhand von Farbbildern Objekte eindeutig zu klassifizieren und zu lokalisieren. Parallel ermöglicht die zunehmende Leistungsfähigkeit von Prozessoren, insbesondere jene von Grafikkarten, die Berechnung vieler Bildanalysen in Echtzeit. In der Automobilindustrie findet die neue Technologie bereits Einzug in Serienprodukte. Ein Beispiel hierfür ist das Model S von Tesla, welches bereits zahlreiche Algorithmen zur Analyse der Umgebung einsetzt. Daher liegt nahe, dass diese eventuell auch für den intralogistischen Transport genutzt können.

Es sind zurzeit nur wenige Systeme bekannt, welche die Methoden der künstlichen Intelligenz aktiv im intralogistischen Transport zur Gefahrenerkennung nutzen. Aus diesem Grund soll in dieser Forschungsarbeit untersucht werden, ob und wie es möglich ist, diese sinnvoll zu nutzen. Eine Übertragung der Erkenntnisse aus der Automobilindustrie wäre zwar prinzipiell denkbar, allerdings sind über die eingesetzten Algorithmen nur wenige Informationen veröffentlicht.

## 1.2 Zielsetzung

Das übergeordnete Ziel dieser Forschungsarbeit ist die Entwicklung eines neuen Kollisionsgefahrerkennungssystems für Gabelstapler unter Berücksichtigung des aktuellen Stands der Technik, insbesondere durch die Zuhilfenahme von Methoden der künstlichen Intelligenz und des maschinellen Sehens. Zur Erreichung dieses Ziels müssen folgende Teilziele erreicht werden:

1. Identifikation der Schwachstellen aktueller Systeme
2. Identifikation relevanter Methoden zur Behebung der Schwachstellen
3. Umsetzung der Methoden in einer Software (und Hardware)
4. Vorgehensweise zur Überprüfung der Algorithmen in Testszenarien
5. Demonstratorische Umsetzung in einem Gesamtsystem
6. Verifikation der Funktionalität im universitären Umfeld und in der Praxis

Die Kenntnis über die Schwachstellen aktueller Kollisionsvermeidungssysteme ist für die Entwicklung des neuen Systems essenziell, damit es nicht die gleichen Probleme



wie z. B. Falschwarnungen hat. Für diese Schwachstellen sollen Methoden gefunden werden, welche die Qualität der Warnungen erhöht. Dabei sollen neuste Methoden des maschinellen Sehens berücksichtigt werden, damit der aktuelle Stand der Technik sich in dem System wiederfindet. Für jede Methode muss anschließend eine Aussage darüber getroffen werden können, inwieweit sie zu einer besseren Warnung beitragen kann. In jedem Fall müssen Kollisionen zuverlässig erkannt werden, bei zugleich möglichst wenig Falschwarnungen. Dazu bedarf es der Definition oder Recherche nach einer geeigneten Vorgehensweise zur Bewertung der eingesetzten Algorithmen. Ein weiteres Ziel ist die Zusammenführung der einzelnen Methoden in ein Gesamtsystem. Für dieses muss analog zu dessen Komponenten festgestellt werden, inwieweit es die angedachten Funktionen in praxisnahen, gefährlichen Situationen erfüllt.

### 1.3 Vorgehensweise

Der erste Schritt ist die Analyse des Stands der Technik und Forschung. Dabei stehen die Analyse bisheriger Systeme (Abschnitte 2.1 und 2.2) und die Recherche geeigneter Methoden zur Kollisionserfassung aus dem Bereich des maschinellen Lernens und Sehens (Abschnitte 2.3 und 2.4) im Fokus. Anschließend wird der Stand der Technik bewertet (Abschnitt 2.5) und daraus die Forschungsfrage (Abschnitt 2.6) abgeleitet.

Aufbauend auf den Erkenntnissen der Literaturrecherche wird ein Konzept für ein neues Kollisionsgefahrerkennungssystem entwickelt. Zur Ableitung von Anforderungen werden Unfalldaten (Abschnitt 3.1.1) sowie relevante Richtlinien und Normen (Abschnitt 3.1.2) analysiert und der Anwendungsfall für das System definiert (Abschnitt 3.1.3). Auf Basis dieser Anforderungen werden ein Hardware- (Abschnitt 3.2) und ein Softwarekonzept (Abschnitt 3.3) erstellt.

Darauf folgt die Implementierung, welche in Kapitel 4 beschrieben wird. Zuerst werden die notwendigen Hardwarekomponenten zusammengestellt sowie deren Position am Stapler bestimmt. Das schließt insbesondere die Montage und Ausrichtung des Sensors unter Berücksichtigung von anthropometrischen Daten und der Bremsdynamik von Gabelstaplern mit ein.

Für die Software des Systems wird zuerst in UML-Klassendiagrammen die Struktur entwickelt (Abschnitt 4.2). Auf Basis der UML-Klassen wird die Software in C++/CUDA modular umgesetzt, sodass einzelne Komponenten des Systems separat getestet werden können und darüber hinaus die Fusion zu einem System einfach möglich ist.

Auf der Basis einer Methode zur Evaluation von Fahrerassistenzsystemen werden nach der Implementierung vorbereitende Maßnahmen getroffen. Die Methode sieht die Identifikation von Einflussfaktoren (Abschnitt 5.1) und üblichen Verkehrssituationen von Staplern (Abschnitt 5.2) als erste Schritte vor. Unter Berücksichtigung der Einflussfaktoren und Verkehrssituationen werden anschließend Testfälle für die Hauptmodule – die Personen- und die Kollisionsgefahrerkennung – definiert (Abschnitte 5.3.1 und 5.3.2).

Außerdem werden Trainings- und Testdaten (Abschnitt 5.4) für die KI-Methoden und Algorithmen erstellt. Hierzu werden sowohl im universitären als auch im industriellen Umfeld Aufnahmen auf einem Gabelstapler gemacht. Die darauffolgende Markierung der Daten erleichtert die automatisierte Auswertung der Personen- und teils auch der Kollisionsgefahrerkennung.

Bei der Durchführung der Evaluation werden zuerst die Personen- und Kollisionsgefahrerkennung separat auf ihre Funktionsfähigkeit untersucht.

Bezüglich der Personenerkennung werden zwei Varianten aus dem Bereich des maschinellen Lernens verwendet. Zum einen wird eine Support Vector Machine (Abschnitt 6.1.2) verwendet, da sie deterministische Ergebnisse liefert bei zugleich moderatem Rechenaufwand. Zum anderen werden neuronale Netze (Abschnitt 6.1.3) aufgrund ihrer sehr guten Erkennungsqualität bei jedoch höherem Rechenaufwand alternativ betrachtet. Für beide Verfahren werden unterschiedliche Variationen hinsichtlich der Parameter, des Trainings und des Eingangsbilds durchgeführt und ausgewertet.

Für die Kollisionsgefahrerkennung werden verschiedene Verfahren des optischen Flusses (OF), welche die Bewegung einzelner Bereiche in sequenziellen Bildern als Ergebnis liefern, (siehe Abschnitt 2.4.2) zur Berechnung der Bewegungsvektoren untersucht (Abschnitt 6.2). Ebenso wird ein Vergleich zur distanzbasierten Kollisionsgefahrerkennung in Anlehnung an bereits erhältliche Produkte gezogen.

Mit den optimierten Ergebnissen aus den Komponententests wird abschließend das Gesamtsystem, in welchem alle Komponenten eingesetzt werden, evaluiert (Kapitel 7). Entscheidende Größen sind hierbei die richtige und rechtzeitige Erkennung von Kollisionsgefahren mit bzw. ohne Menschen bei gleichzeitiger niedriger Rate an fälschlicherweise erkannten Gefahren.



## 2 Stand der Technik und Forschung

---

Zur Identifikation von Kollisionen auf einem Gabelstapler bedarf es zusätzlicher Sensoren am Stapler, da diese im Gegensatz zu fahrerlosen Transportfahrzeugen die Umwelt nicht elektronisch erfassen. Im ersten Abschnitt dieses Kapitels werden verschiedene Sensoren vorgestellt, welche es ermöglichen, die Umwelt optisch wahrzunehmen.

Der nächste Abschnitt zeigt eine Übersicht bereits erhältlicher Kollisionswarnsysteme. Der Fokus liegt auf der verwendeten Sensorik sowie auf den Sensorwerten, welche zur Ermittlung der Kollisionsgefahr genutzt werden.

Der dritte Abschnitt ist eine Einleitung in die grundlegende Funktionsweise von Algorithmen aus dem Bereich des maschinellen Lernens. Anhand der Beispiele „Support Vector Machines“ und „künstlicher neuronaler Netze“ werden zwei wichtige Vertreter näher betrachtet.

Im vierten Abschnitt werden zuerst grundlegende Operationen im Bereich der maschinellen Bildverarbeitung erläutert. Anschließend werden die Bedeutung von Bildmerkmalen und deren wichtigste Vertreter aufgezeigt. Der letzte Teil des Abschnitts überträgt die in Abschnitt 3 aufgezeigten Methoden des maschinellen Lernens auf die Bildverarbeitung.

Abschließend werden im letzten Abschnitt die erörterten Technologien und Methoden zur Nutzung in einem Kollisionsgefahrerkennungssystem für Gabelstapler bewertet.

### 2.1 Sensortechnologien zur Erfassung von Kollisionen

Sensoren können entweder dazu genutzt werden den internen Status einer Maschine (propriozeptive Sensoren) oder die Umwelt (exterozeptive Sensoren) zu erfassen. Dabei wird zwischen aktiven und passiven Sensoren unterschieden. Aktive Sensoren (z. B. Ultraschall) emittieren Energie und messen die Reaktion der Umwelt auf die Emission. Passive Sensoren nutzen die Energie der Umgebung (z. B. Fotodiode). [Sie-2011, S. 101ff.]

Die Erfassung von Kollisionen bei Fahrzeugen erfordert primär die Messung von Distanzen. Dazu sind exterozeptive Sensoren notwendig. Als Ergänzung können propriozeptive Sensoren dienen, wie z. B. ein Drehgeber zur Messung des Lenkwinkels des Fahrzeugs. Die folgende Übersicht in Abbildung 2-1 ist auf aktive und passive exterozeptiven Sensoren zur Erfassung von Distanzen und Texturen beschränkt:

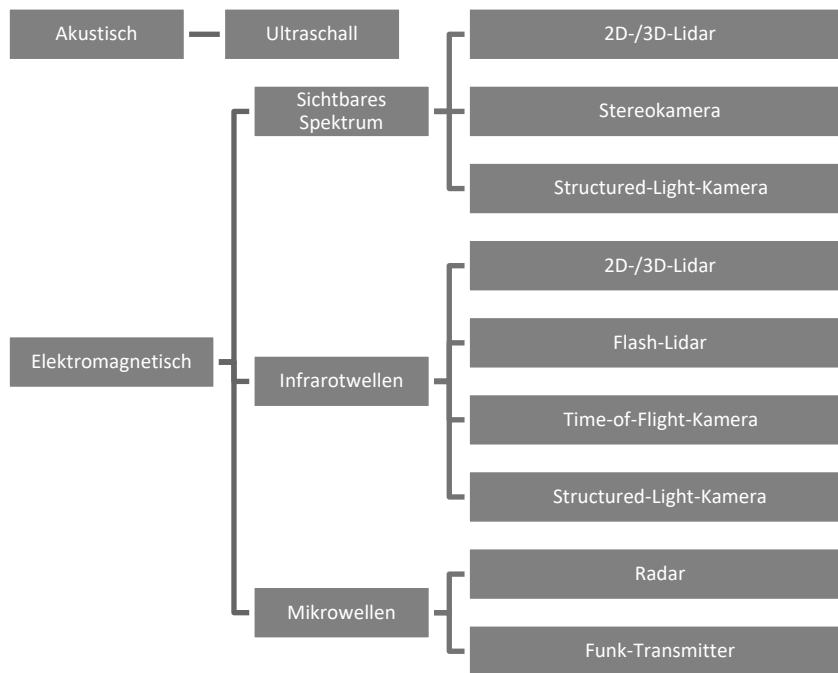


Abbildung 2-1: Technologische Einordnung von Sensoren zur Distanz- und Texturerkennung.

Der einzige akustische Sensor ist der **Ultraschallsensor**, welcher aus mindestens einem Ultraschall-Receiver und -Emitter besteht. Eines der Hauptanwendungsgebiete ist die Messung von Distanzen zur Kollisionsvermeidung (z. B. Parksensoren beim Auto). Der Sensor funktioniert nach dem „Time-of-Flight“-Prinzip: Nach der Emission einer Ultraschallwelle misst der Sensor die Zeit bis am Receiver ein bestimmter Signalpegel überschritten wird. Der Grenzwert am Receiver sinkt dabei mit zunehmender Laufzeit, da die Energie von Ultraschallwellen bei längeren Distanzen messbar abnimmt. [Sie-2011, S. 140f.] Wird ein Ultraschallsensor verwendet, ist nur die Distanz zum nächsten Objekt messbar. Durch die Verwendung mehrerer Ultraschallsensoren können über Trilateration Objekte auch lokalisiert werden. Die Lokalisierung ist je nach Objektgeometrie eher ungenau und das reflektierende Objekt wird in einen Punkt konsolidiert. Fehlmessungen können durch andere akustische Erzeuger, wie z. B. Bremsen oder bei Objekten mit weichen Stoffen (z. B. Kleidung) auftreten. [Win-2015, S. 252ff.]

Die meisten Distanzsensoren basieren auf elektromagnetischen Wellen. Zu dieser Art gehören auch **Farbkameras**, welche im sichtbaren Spektrum arbeiten. Um mit Farbkameras Distanzdaten zu erhalten, ist der Einsatz von zwei kalibrierten Kameras oder

einer **Stereokamera** nötig. Die Tiefeninformationen werden über die Disparität von zwei gleichzeitig aufgenommenen Bildern berechnet. Im Überlappungsbereich der zwei Bilder werden letztlich Tiefeninformationen für jeden einzelnen Bildpunkt berechnet. Störungen können auftreten, wenn Bildmerkmale (siehe 2.4.2), die zur Berechnung der Disparität notwendig sind, nicht mehr ausreichend extrahiert werden können. Dies tritt z. B. dann auf, wenn das Bild stark über- oder unterbelichtet ist. [Sie-2011, S. 169ff.]

**2D- und 3D-Lidarsensoren** arbeiten sowohl im sichtbaren, im ultravioletten als auch im infraroten Lichtspektrum. Zumeist werden sie als (2D-/3D-) Laserscanner bezeichnet. Die Messung der Distanz wird analog zum Ultraschallsensor über die Time-of-Flight (ToF) emittierter und empfangener Laserstrahlen berechnet. Zur Generierung von zwei- oder dreidimensionalen Daten wird der Laserstrahl durch motorbetriebene Prismen oder Spiegel in eine bzw. zwei Richtungen umgelenkt. Je nach Sensorart sind die Distanzinformationen auf einer Ebene oder ein 3D-Bild das Resultat. [Eve-2001, S. 150ff.] Prinzipiell ist auch die Ermittlung der Relativgeschwindigkeit über den Dopplereffekt wie bei Radar (s. u.) möglich. Jedoch erfordert die Messung des Dopplereffekts im Lichtspektrum einen extrem hohen technischen und finanziellen Aufwand. [Win-2015, S. 324f.]

**Flash-Lidarsensoren** unterscheiden sich von herkömmlichen Lidarsensoren durch die Verwendung eines CCD-Moduls als Receiver, wodurch keine Prismen oder Spiegel für die zweidimensionale Distanzberechnung benötigt werden. Nach jedem Laserpuls wird ein komplettes 3D-Bild anhand der reflektierten Pulse generiert. Außerdem liefern einige Varianten Kontrastdaten, welche Rückschlüsse auf die Oberflächenbeschaffenheit von Objekten geben können. [Gel-2002]

Ein komplettes 3D-Bild liefern auch **ToF-Kameras**. Die Funktionsweise unterscheidet sich nur geringfügig von Flash-Lidarsensoren. Bei dieser Technik werden Infrarotdioden zur Erzeugung eines modulierten Lichtkegels verwendet. Das modulierte Licht wird wiederum von einem CCD- oder CMOS-Sensor empfangen. [Hag-2007] Beide Sensorarten nutzen den internen photoelektrischen Effekt um Lichtwellen in elektrische Signale umzuwandeln [Car-2002].

Lidarsensoren sind vor allem anfällig gegenüber Oberflächen, welche die emittierten Wellen entweder zu stark ablenken (Metalle) oder zu wenig reflektieren (Glas). Zusätzlich können fremde Infrarotquellen (z. B. direktes Sonnenlicht) die Sensoren in ihrer Funktion beeinträchtigen. [Hag-2007]

**Structured-Light-Kameras** (SL-Kameras) ermitteln die Distanzdaten nicht über das ToF-Prinzip. SL-Kameras projizieren ein Muster, wie z. B. ein Schachbrett oder ran-

domisierte Punkte, im infraroten oder sichtbaren Spektrum in ihrem Sichtfeld. Die Kamera nimmt anschließend das Bild mit dem projizierten Muster auf und vergleicht dieses mit dem ausgesandten. Anhand der Verzerrungen können 3D-Bilder über verschiedene Techniken, auf die hier aufgrund der Komplexität nicht genauer eingegangen werden soll, berechnet werden. [Gen-2011]

**Radarsensoren** wurden im zivilen Bereich erstmals zur Detektion von Geschwindigkeitsübertretungen eingesetzt, sie fanden später aber auch Einzug in industrielle Produkte. Sie funktionieren annäherungsweise wie Ultraschallsensoren, allerdings senden sie elektromagnetische Mikrowellen (24 und 76,5 GHz) aus und der Receiver benötigt eine separate Antenne. Die Distanz wird über die ToF der emittierten und empfangenen Radarwellen berechnet. Üblicherweise können mithilfe von Radarsensoren nicht nur die Distanzdaten eines Objekts, sondern auch dessen Bewegung (Geschwindigkeit und Richtung) bestimmt werden. Die Berechnung der Geschwindigkeit wird durch den Doppler-Effekt möglich. Die Ermittlung der Richtung, in der sich das Objekt relativ zum Sensor befindet, wird durch die Verwendung einer Richtantenne möglich. Zusätzlich kann die Größe eines erkannten Objekts über die empfangene Energie geschätzt werden. Als Ergebnis liefern Radarsensoren punktförmige Objekte mit den beschriebenen Werten, sofern das jeweilige Objekt die Wellen reflektiert. Je nach Oberfläche und Reichweite von gesuchten Objekten sind nur bestimmte Frequenzbänder sinnvoll. [Win-2015, S. 260ff.]

Der Einsatz von **Funk-Transmittern** zur Kollisionsgefahrerkennung setzt voraus, dass etwaige Kollisionobjekte jeweils mit einem Modul ausgestattet sind. In diesem Fall wird demzufolge nicht die Umwelt erfasst. Hierbei dient die Signalstärke zwischen den beiden Modulen zur Ermittlung der Distanz. Etwaige Störungen können auftreten, wenn die Signalstärke durch die Umgebung beeinflusst wird. So können bspw. metallische Objekte zwischen Sender und Empfänger die Signalstärke schwächen und damit eine falsche Berechnung der Distanz bedingen.

Für die Entscheidung, welche Technologie letztlich für ein Kollisionsgefahrerkennungssystem eingesetzt werden kann (siehe Abschnitt 2.5.2), sind grundsätzlich die Merkmale der Reichweite, des Öffnungswinkels, der Störeinflüsse, der Datenfrequenz und der Datenart relevant. Eine Zusammenfassung zeigt die folgende Tabelle, welche anhand von aktuellen Produkten oder Literaturangaben einen Anhaltspunkt für die jeweiligen Parameter zeigt:

Tabelle 2-1: Ausgewählte technische Daten verschiedener Sensortechnologien zur Erfassung der Umwelt.

Sensor	Reichweite	Öffnungswinkel Tiefensensor (h x v)	Störeinflüsse	Datenfrequenz	Datenart(en)	Quellen
<b>Ultraschall</b>	<4 m	120° x 70°	Weiche Stoffe, Temperatur, Akustische Fremdstrahler	30-40 Hz	Distanz, Position bei Einsatz mehrerer Sensoren	[Sch-1993, S. 47f.; Win-2015, S. 243ff.]
<b>Stereokamera</b>	<30 m	105° x 90°	Keine	8-100 Hz	Farb- und 3D-Bild	[Ste-2018]
<b>3D-Lidar</b>	<100 m	360° x 70°	Nebel	5-25 Hz	3D-Bild	[Ocu-2015]
<b>Flash-Lidar</b>	<200 m	120° x 30°	Keine	4-100 Hz	3D-Bild	[Bli-2018]
<b>ToF-Kamera</b>	<13 m	57° x 53°	Metall, direktes Sonnenlicht	20-60 Hz	Intensitäts- und 3D-Bild	[Bas-2017; Bör-2018]
<b>SL-Kamera</b>	<10 m	85° x 58°	Keine	10-90 Hz	Intensitäts- oder Farbbild und 3D-Bild	[Int-2018]
<b>Radar</b>	<400 m	20° x 20°	Keine	16 Hz	Distanz, Größe, Bewegungsvektor und ggf. Winkel von Objekten	[Inn-2016]



## 2.2 Kollisionswarnsysteme für Gabelstapler

Zur präventiven und aktiven Kollisionsvermeidung bieten nur wenige Gabelstaplerhersteller Lösungen an. In den meisten Fällen haben die Hersteller Lösungen von Drittfirmen im Angebot, welche auf das jeweilige Modell adaptiert werden. Es sind allerdings auch Produkte erhältlich, welche unabhängig vom Staplertyp nachgerüstet werden können. Eine Übersicht zeigt Tabelle 2-2.

Das System „*Keeper*“ der Comnovo GmbH ist ein portables Modul mit einem Funktransceiver, welches im Gabelstapler mitgeführt oder befestigt wird. Es warnt akustisch, wenn sich ein anderer Gabelstaplerfahrer oder Fußgänger – ebenfalls mit einem Modul ausgestattet – innerhalb einer definierten Distanz befindet. Darüber hinaus wird die Richtung des gefährdeten Objekts visuell angezeigt. Die Distanz und die Richtung werden über Funkwellen ermittelt, die genaue Funktionsweise ist nicht näher dokumentiert. [Com-2016]

Das System „*Blaxtair*“ der Arcure GmbH nutzt eine Stereokamera als Datenquelle. Deren Daten werden zur Bestimmung des Abstands von Objekten und zur Identifikation von Personen genutzt. Das Produkt bietet durch die Personenerkennung die Möglichkeit zwei unterschiedliche Warnzonen in Abhängigkeit des Abstands einzurichten. Die Warnung erfolgt sowohl optisch als auch akustisch, sobald sich ein Objekt in einer der Warnzonen befindet. Die Funktionsweise der Personenerkennung wird nicht weiter ausgeführt. [Acu-2017]

Bei dem System *BlueSpot*<sup>TM</sup> der Linde Material Handling AG handelt es sich im Gegensatz zu den anderen Lösungen um ein passives Warnsystem, denn es besitzt keine eigene Sensorik zur Erfassung der Umwelt. Über interne Sensoren des Staplers wird erkannt, in welche Richtung der Stapler sich bewegt. Je nach Fahrtrichtung wird ein Scheinwerfer aktiviert, welcher vor bzw. hinter den Stapler einen Lichtkreis projiziert. Damit können Passanten vor herannahenden Staplern optisch gewarnt werden. [Lin-2018]

Die ELOKON GmbH bietet zwei Lösungen an, welche sich durch die verwendete Sensorik unterscheiden. Bei dem Produkt „*ELOprotect*“ werden Objekte mittels eines Laserscanners am Stapler erfasst. Der Stapler wird nach der Erkennung von Objekten im Gefahrenbereich abgebremst. Dieses System ist hauptsächlich für den Einsatz in Schmalganglagern konzipiert. „*ELOshield*“ funktioniert auf die gleiche Weise wie das

System der Comnovo GmbH. Fußgänger und Stapler sind hierbei mit Funk-Transceiver-Modulen ausgestattet, welche warnen oder den Stapler abbremsen, sobald sich ein anderes Modul im Empfangsbereich befindet. [ELO-2018]

Ein Produkt, welches die Umgebung mittels Ultraschallsensoren wahrnimmt, bietet die tbm hightech control GmbH unter dem Namen „NoColl“ an. Das System warnt akustisch, wenn sich ein Objekt innerhalb einer abstands-basierten Warnzone befindet. Die Größe der Zone ist abhängig von der Geschwindigkeit des Staplers. Hierfür ist die Voraussetzung, dass der Stapler intern die Geschwindigkeit misst und diese über den CAN-Bus bereitstellt. Dieselbe Funktionsweise findet sich bei dem System „RAM-107“ des gleichen Unternehmens wieder. Allerdings wird als Datenquelle eine ToF-Kamera eingesetzt. [tbm-2017]

Die Lösung der *U-Tech* GmbH funktioniert ähnlich der bereits erwähnten funkbasierten Systeme, allerdings auf RFID-Basis. Hierbei werden nicht Transceiver, sondern RFID-Tags und RFID-Reader eingesetzt. Ein mögliches Szenario wäre, dass auf Staplern der Reader installiert ist und Fußgänger Tags mitführen. Sobald sich ein Tag im Lesebereich des Readers befindet, gibt dieser ein logisches Signal über eine Schnittstelle aus. Das Signal muss anschließend von einer weiteren externen Einheit verarbeitet werden. [U-T-2017]

Tabelle 2-2: Übersicht erhältlicher Kollisionswarnsysteme für Gabelstapler.

#	Name	Hersteller	Herstellerspezifisch	Sensortechnik	Messparameter	Warnungsmodalität
1	Keeper Beeper Keeper Static	Comnovo GmbH	Nein	Funk	Erreichbarkeit eines Moduls, Richtung	Optisch, Akustisch, Haptisch, Geschwindigkeitsdrosselung
2	Blaxtair	Arcure	Nein	Stereokamera	Abstand, Personenerkennung	Optisch, Akustisch
3	BlueSpot™	Linde Material Handling AG	Ja	Keine	Keine	Optisch
4	ELOprotect	ELOKON GmbH	Nein	2D-Lidar	Abstand	Geschwindigkeitsdrosselung
5	ELOshield	ELOKON GmbH	Nein	Funk	Erreichbarkeit eines Moduls	Optisch, Akustisch, Geschwindigkeitsdrosselung
6	NoColl	tbm hightech control GmbH	Nein	Ultraschall	Abstand, CAN-Geschwindigkeitsdaten	Akustisch
7	RAM-107	tbm hightech control GmbH	Nein	ToF-Kamera	Abstand, CAN-Geschwindigkeitsdaten	Akustisch
8	-	U-Tech GmbH	Nein	Funk (RFID)	Erreichbarkeit eines Moduls	Keine Warnung

In diesem Abschnitt wurde ausschließlich der Stand der Technik von Kollisionswarnsystemen für Gabelstaplerfahrer behandelt. Der Grund hierfür ist, dass im Rahmen der Literaturrecherche keine passenden Veröffentlichungen aus dem Bereich der Kollisionswarnung für Gabelstapler gefunden wurden.

Veröffentlichungen, welche die Kollisionsvermeidung bzw. -erkennung zum Ziel haben, sind aber außerhalb der Logistik zu finden.

Im Automobilbereich finden sich sowohl Veröffentlichungen zur allgemeinen Kollisionsvermeidung ([Dag-2004; Str-2018; Uwe-2007]) als auch zur Vermeidung spezifischer Kollisionen, z. B. mit Fußgängern ([Llo-2011]), Tieren ([Gup-2020]) oder vorausfahrenden Fahrzeugen ([Xue-2016; Geh-2007]).

Weitere Bereiche in denen viele Forschungsergebnisse zur Kollisionsvermeidung existieren sind Drohnen ([Sed-2017; Yu-2010]) und Roboter ([Ber-2014; Moh-2017; Sha-2014]).

Da die Einsatzszenarien der veröffentlichten Arbeiten aber sehr weit von dem des Gabelstaplers abweichen, werden sie in dieser Arbeit nicht weiter vertieft.

## 2.3 Maschinelles Lernen

Das maschinelle Lernen (ML, engl.: „machine learning“) ist ein Teilgebiet der künstlichen Intelligenz. Der Begriff „machine learning“ wurde 1959 von Arthur Samuel geprägt. Er beschäftigte sich damit, wie Maschinen dazu befähigt werden können, im Brettspiel Mühle besser als Menschen zu sein. Dazu entwickelte er Algorithmen, welche trainiert werden konnten. Der Hauptaufwand sollte daher nicht mehr in der Programmierung, sondern im Training dieser Algorithmen liegen. [Sam-1959]

*Fischer* verdeutlicht die Funktionsweise von Algorithmen des maschinellen Lernens anhand der Erkennung des Typs „Baum“ durch einen Menschen: Ein Mensch lernt nicht die Form und Farbe eines jeden Baums auswendig und erkennt diesen als solchen wieder. Er lernt die Charakteristika, welche einen Baum ausmachen, er abstrahiert den Baum zu einer Sammlung von abstrakten Informationen und Prinzipien. [Fis-1999] Der Mensch trainiert sein Gehirn dementsprechend nicht auf die rohen Daten, sondern auf deren Abstraktion. Analog verhält es sich bei Algorithmen des maschinellen Lernens. Sie benötigen Trainingsdaten um lernen zu können, wie Informationen aus den zu analysierenden Daten extrahiert werden können, welche Informationen die Zielgröße beinhaltet und wie wichtig die Information für die Zielgröße ist.

Das Training der ML-Algorithmen kann je nach Datenart und -verfügbarkeit sehr unterschiedlich ausgeprägt sein. Generell wird zwischen folgenden Arten des Trainings unterschieden:

- **Überwachtes Lernen** (engl.: „supervised learning“) (SL): Das überwachte Training ist, sofern genügend Trainingsdaten verfügbar sind, für die meisten Anwendungsfälle die beste Trainingsart. Bei dieser werden zuerst die zu analysierenden Rohdaten in sog. Feature-Vektoren (extrahierte Informationen) umgewandelt. Diese Vektoren beschreiben die Eingangsdaten in numerischen Werten; beispielsweise könnte der Buchstabe „s“ durch den Vektor [0.042, 0.1] repräsentiert werden. Diese Feature-Vektoren werden in Kombination mit einem Datensatz, welcher das gewünschte Prädiktionsergebnis für jeden Vektor enthält, als Eingang für das Training des ML-Algorithmus verwendet. Der Algorithmus sagt für jeden Feature-Vektor ein Skalar oder eine reelle Zahl als Ergebnis vorher. Dieses Ergebnis wird mit dem Sollergebnis aus dem markierten Datensatz verglichen. Daraufhin wird das Prädiktionsmodell angepasst, sodass der Fehler der Vorhersage möglichst gering wird. Während des gesamten Trainings wird dieser Prozess iterativ durchgeführt, sodass sich der Fehler für möglichst viele Datensätze minimiert. Zugleich wird das Prädiktionsmodell nach einer bestimmten Anzahl an Trainingsschritten validiert. Hierbei wird mit einem ebenfalls markierten Validierungsdatensatz überprüft, ob das Prädiktionsmodell weiter generalisiert oder sich zu sehr an das Trainingsdatensatz angepasst hat. Letzteres wird als „overfitting“ bezeichnet. Im Falle von Overfitting kann das Prädiktionsmodell zwar das Ergebnis der Trainingsdaten sehr gut vorhersagen, allerdings ist die Prädiktion bei unbekanntem Daten schlecht. Meist tritt Overfitting auf, wenn die Daten des Trainingsdatensatzes zu ähnlich sind oder die Größe des Datensatzes nicht zu der Größe des eingesetzten DL-Netzes passt. [Awa-2015]
- **Unüberwachtes Lernen** (engl.: „unsupervised learning“) (UL): Bei dieser Art des Trainings werden eingangs wie beim „supervised learning“ die Rohdaten in Feature-Vektoren umgewandelt. Diese werden ohne zusätzliche Informationen vom Prädiktionsmodell ausgewertet. [Awa-2015] Das Ergebnis sind üblicherweise Cluster, welche die Daten unterteilen. Für jedes Cluster werden zugleich auch die charakteristischen Merkmale berechnet. Diese Art des Trainings ist demnach nur dann sinnvoll, wenn die Zusammenhänge von Datensätzen noch nicht bekannt sind und diese bestimmt werden sollen. [Fis-1999, S. 4]
- **Bestärkendes Lernen** (engl.: „reinforcement learning“) (RL): Bei dieser Form des Trainings sind wie beim UL keine markierten Trainingsdaten notwendig. Für RL ist die Kombination aus Exploration und Exploitation charakteristisch. Die Begriffe sollen anhand eines Beispiels zur Gewinnmaximierung am einarmigen Banditen verdeutlicht werden: Während der Exploration wird der am besten eingeschätzte einarmige Bandit evaluiert. Wechselt das Training zur Exploitation, welche deutlich seltener durchgeführt wird, wird ein anderer einarmiger Bandit eva-

luiert. Die Entscheidung, ob ein einarmiger Bandit der Beste ist, hängt von dessen Anzahl an erhaltenen „rewards“ ab. Diese erhält er, indem das Prädiktionsergebnis durch eine „reward“-Funktion bewertet wird. Die „reward“-Funktion wird beim RL meist erst während des Trainings erstellt. Im Falle des einarmigen Banditen würde die „reward“-Funktion im Laufe des Trainings einen hohen Gewinn belohnen. Die Bewertung durch die „reward“-Funktion sowie die Exploitation sind nur möglich, wenn bereits mehrere Durchläufe durchgeführt wurden. Um das Training zu starten, werden daher für die ersten Durchläufe entweder Werte außerhalb des erwarteten Wertebereichs der Zielgröße oder zufällige Werte zur Initialisierung des Prädiktionsmodells verwendet. [Bar-2012; Kub-2017; Kub-2017, S. 330ff]

Im Laufe der Zeit gab es viele unterschiedliche Ansätze zur Umsetzung von ML. Es gibt auch heute noch keinen umgesetzten Ansatz, welcher universal für alle Datenarten das beste Ergebnis liefert. Der Einsatz einer Methode ist vor allem abhängig von der Datenart und dem gewünschten Ergebnis. Folgend sollen aufgrund ihrer Relevanz zum vorliegenden Dissertationsthema zwei Ansätze des maschinellen Lernens genauer betrachtet werden: „Support Vector Machines“ (SVM) und „Neuronale Netze“ (RL). Eine detaillierte Auflistung verschiedener Ansätze findet sich in [Awa-2015].

### 2.3.1 Support-Vector-Machines (SVM)

Das mathematische Modell für die SVM wurde bereits 1963 von *Vapnik* vorgestellt. Dieses basiert auf der Funktion zur Risikominimierung. Die Neuerung war die erstmalige „Bestrafung“ fehlender Generalisierungsfähigkeit während der Minimierung. [Vap-1963]

Der Einsatz von SVM hielt sich lange in Grenzen, da die Leistung von Computern in den 1970er und 1980er Jahren noch sehr begrenzt war. In den 1990er Jahren verhalf die steigende Rechnerleistung der SVM zum Durchbruch. Der Grund für den Erfolg waren die sehr guten Ergebnisse, welche mit SVM erreicht wurden, in Kombination mit einer moderaten Rechenleistung. Der Durchbruch der SVM führte zugleich zum temporären Ende der Forschung an den deutlich rechenaufwändigeren künstlichen neuronalen Netzen.

#### Funktionsweise

Die grundsätzliche Funktionsweise einer SVM ähnelt nach *Parker* der Suche nach einer linearen Diskriminante (siehe Abbildung 2-2a), die zwei Klassen voneinander trennt. Allerdings wird bei der SVM keine lineare Diskriminante, sondern ein Grenzband gesucht, welches den größtmöglichen Abstand zu beiden Klassen besitzt (siehe Abbildung 2-2b). Neue Feature-Vektoren (Datenpunkte) außerhalb des Bands werden in diesem Fall mit höherer Wahrscheinlichkeit der richtigen Klasse zugeordnet als

durch die Trennung der Klassen über eine beliebige Diskriminante.

In der Abbildung wurde exemplarisch bei a) die rot markierte Diskriminante zum Vergleich ausgewählt. In diesem Fall wird ein neuer Feature-Vektor (rot) fälschlicherweise der linken Klasse zugeordnet werden, während er durch das Grenzband bei b) richtig klassifiziert wird.

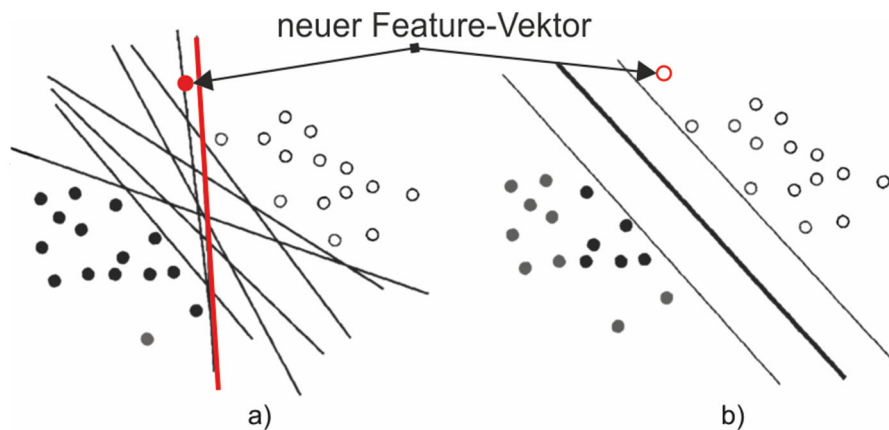


Abbildung 2-2: a) Mögliche Diskriminanten zweier Klassen b) Optimales Grenzband mit größtmöglicher Breite. [Par-2011, S. 310].

Zur Bestimmung des optimalen Grenzbandes wird mit herkömmlichen, linearen Optimierungsalgorithmen die ideale Diskriminante hergeleitet. Als Eingangsdaten dienen die Feature-Vektoren aus der umschließenden Hülle einer jeweiligen Klasse (siehe Abbildung 2-3). Die Breite des Bandes wird durch mindestens drei Feature-Vektoren der Hüllen, welche der optimalen Diskriminante am nächsten sind, bestimmt. Dabei muss jeweils mindestens ein Feature-Vektor jeder Klasse enthalten sein. Diese Feature-Vektoren werden auch Support-Vektoren genannt und bestimmen letztlich das finale Grenzband.

Das Grenzband ist im gezeigten Beispiel zweidimensional. Mehrdimensionale Feature-Vektoren sind ebenso möglich. Diese werden entsprechend durch Hyperebenen getrennt. Ebenso können die Klassen nichtlinear getrennt werden, z. B. durch einen polynomiellen Kernel. Hierbei werden allerdings keine polynomiellen Support-Vektoren berechnet, sondern die Feature-Vektoren werden in einen höherdimensionalen Raum über den polynomiellen Kernel projiziert.

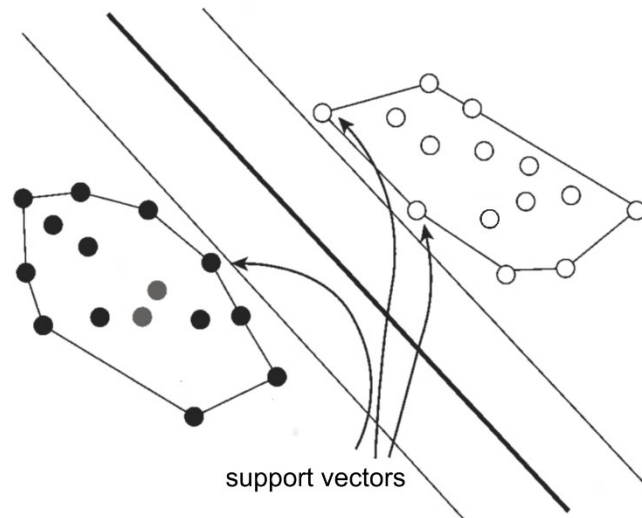


Abbildung 2-3: Hülle der Feature-Vektoren und Support-Vektoren [Par-2011, S. 310].

SVM sind grundsätzlich nur dazu in der Lage zwei Klassen voneinander zu trennen. Praktisch können auch mehrere Klassen voneinander getrennt werden. Dafür werden alle Klassen paarweise gegenübergestellt, demzufolge muss pro Klassenpaar eine SVM trainiert werden. [Par-2011, S. 306ff.]

### 2.3.2 (Künstliche) Neuronale Netze

Die Idee, künstliche neuronale Netze zur automatisierten Auswertung von Daten zu nutzen, entstand bereits in den 1940er Jahren durch *McCulloch* und *Pitts*. Über mehrere Dekaden wurden die anfangs sehr simplen, nicht lernfähigen neuronalen Netze zu SL und anschließend zu UL-Netzen weiterentwickelt. Ein bedeutender Meilenstein war 1979 die Entwicklung der „Convolutional Neural Networks“. Diese wurden nur kurze Zeit später um die effiziente Berechnungsmethode der Gewichtungen „Backpropagation“ erweitert. Allerdings gerieten, wie bereits in der Einleitung der SVM erwähnt, neuronale Netze aufgrund der Erfolge der SVM aus dem Fokus der Forschung. Im Jahr 2009 änderte sich diese Entwicklung rapide, als ein neuronales Netz in einem offiziellen ML-Wettbewerb zur Erkennung von handgeschriebenem Text allen anderen ML-Methoden deutlich überlegen war. Darüber hinaus beschleunigten neue GPU-Generationen die Berechnung tiefer Netze (siehe nächste Seite) immens. Aufgrund dieser Gegebenheiten sind neuronale Netze seitdem wieder im Fokus der ML-Forschung. [Sch-2015, S. 9ff.]

Künstliche neuronale Netze werden im Sprachgebrauch oft irrtümlicherweise als Nachbildungen des menschlichen Gehirns deklariert. Gemein haben sie allerdings nur die Struktur der uni-/bipolaren Verbindungen von Neuronen und Synapsen. Auch wenn



künstliche neuronale Netze nicht annähernd die Komplexität eines menschlichen Gehirns abbilden können, haben sie die Fähigkeit zu abstrahieren und zu lernen, weit besser als alle bisher dagewesenen ML-Algorithmen.

**Vereinfachter Aufbau und Funktionsweise neuronaler Netze**

Künstliche Neuronale Netze bestehen mindestens aus zwei Schichten: der Ein- und Ausgabeschicht. Das Netz hätte in dem Fall eine Tiefe von zwei. Üblicherweise befinden sich mehrere sogenannte versteckte (engl.: „hidden“) Schichten zwischen der Ein- und Ausgabeschicht. Diese werden versteckt genannt, da sie keinen Kontakt zur „Außenwelt“ besitzen. Werden mehr als zehn versteckte Schichten in einem Netz verwendet, wird es als tiefes neuronales Netz (engl.: „Deep Neural Network“ (DNN) oder „Deep Learning“ (DL)) bezeichnet [Sch-2015, S. 88].

Abbildung 2-4 zeigt die Darstellung eines einfachen Netzes, auch Perzeptron genannt. Die Eingabeschicht  $U_{in}$  erhält die Rohdaten  $x_1$  bis  $x_n$ . Diese werden sukzessiv bis zur Ausgabeschicht  $U_{out}$  in den Schichten bearbeitet. Jedes Neuron einer Schicht ist mit anderen Neuronen der nachfolgenden Schicht verknüpft. Das gilt allerdings nicht zwangsläufig für alle Netze. Es gibt auch Netze, bei welchen Neuronen über mehrere Schichten hinweg mit anderen Neuronen verknüpft sind. Ist jedes Neuron mit jedem Neuron der nachfolgenden Schicht verknüpft, werden diese Schichten als voll vernetzt (engl.: „fully connected layers“) bezeichnet.

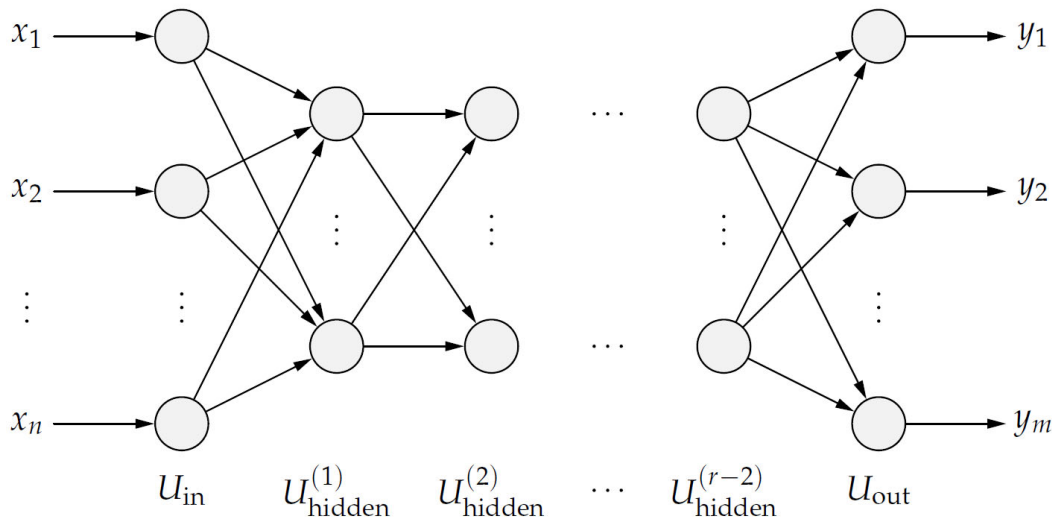


Abbildung 2-4: Schematische Darstellung eines r-schichtigen Perzeptrons [Kru-2015, S. 44].

Innerhalb eines Neurons werden die Eingabeparameter anhand einer Aktivierungsfunktion bewertet. Meist sind die Eingabeparameter der Ausgangswert des vorherigen Neurons und die Gewichtung der Verbindung der zwei Neuronen. Wenn das Ergebnis der Aktivierungsfunktion den neuronenspezifischen Schwellwert überschreitet, wird

das nächste bzw. werden die nächstverbundenen Neuronen aktiviert. Als Aktivierungsfunktion wurde längere Zeit eine sigmoide Funktion verwendet (siehe Abbildung 2-5). Diese liefert um den Nullpunkt Werte zwischen 0 und 1. Wenn die Eingangswerte sehr weit vom Nullpunkt entfernt sind, liefert die Sigmoid-Funktion nahezu konstante Werte. [Kru-2015, S. 36ff.] Dies führt zu einem stark nicht-linearen Verhalten. Das Training neuronaler Netze beruht jedoch mitunter auf Gradientenbildung, welche bei Konstanten nicht möglich ist, weswegen das Training nicht immer optimal durchgeführt werden kann. Seit einiger Zeit werden daher andere Aktivierungsfunktionen wie die „softplus“- oder „(leaky) rectifier linear unit (ReLU)“-Funktion benutzt [Ram-2017]. Diese Funktionen berechnen sich wie folgt:

Softplus:

$$f(x) = \log(1 + e^x)$$

Leaky ReLu:

$$f(x) = \begin{cases} x & \text{falls } x > 0 \\ 0,01x & \text{sonst} \end{cases}$$

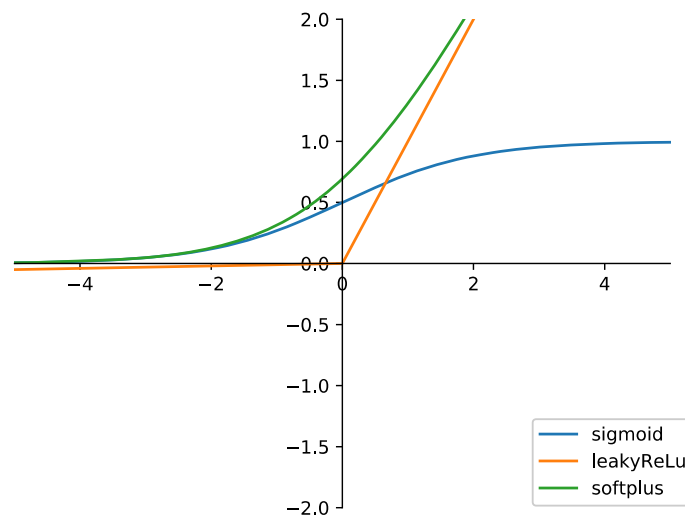


Abbildung 2-5: Vergleich verschiedener Aktivierungsfunktionen neuronaler Netze.

Die Neuronen der Ausgabeschicht  $U_{\text{out}}$  haben meist sigmoide oder lineare Aktivierungsfunktionen. Im ersten Fall sind alle Ausgabewerte  $y_1$  bis  $y_m$  im Bereich zwischen 0 und 1.

### **Training und Fehlerrückübertragung**

Während des Trainings wird der gleiche Prozess im Perzeptron durchgeführt wie bei dem, in Abbildung 2-4 gezeigten, Prozess der Detektion. Als Startwerte werden für jedes Neuron zufällige Werte für den Aktivierungsschwellwert und für jede Verbindung eine zufällige Gewichtung verwendet. Je nachdem, ob ein Netz UL, SL oder RL verwendet, wird am Ende der Prädiktion gegebenenfalls das Endergebnis mit dem Soll-Ergebnis verglichen. Der Vergleich ist mathematisch der Minimierung einer Fehlerfunktion gleichzusetzen. Der Fehler berechnet sich aus der Abweichung vom erzielten zum gewünschten Ergebnis. Ein bewährter Ansatz zur Fehlerminimierung ist die logistische Regression. Diese ist allerdings nur auf zweischichtige Perzeptren anwendbar, da die dafür benötigte Fehlerquadratsumme nur für die Ausgabeschicht berechnet werden kann. In verdeckten Schichten kann die Fehlerquadratsumme aufgrund der mangelnden Kenntnis über das zu erwartende Ergebnisse nicht errechnet werden. Aufgrund dessen wird bei tieferen neuronalen Netzen die Optimierung mittels Gradientenabstiegsverfahren verwendet. [Kru-2015, S. 58ff.]

Das Gradientenabstiegsverfahren setzt voraus, dass die Aktivierungsfunktion differenzierbar ist. Denn ist diese differenzierbar, ist auch die Fehlerfunktion differenzierbar. Für jedes Neuron muss bei diesem Verfahren der Gradient der Fehlerfunktionen berechnet werden. Da diese Vorgehensweise sehr rechenintensiv ist und die benötigte Rechenleistung exponentiell mit der Netzwerkgröße zunimmt, wurde das Fehlerrückübertragungsverfahren (engl.: „backpropagation of error“) entwickelt. In diesem Verfahren werden die Gradienten der Fehlerfunktion der jeweils vorhergehenden Schicht berechnet und für jeden Knoten gespeichert. Die gespeicherten Werte werden anschließend als Eingang für die wiederum davorliegende Schicht benutzt. Dieses Verfahren wird bis zur Eingabeschicht wiederholt. Damit reduziert sich das exponentielle auf ein lineares Problem. Anschließend werden die Gewichtungen neu festgelegt. Hierbei wird ein Trägheitsterm, das Momentum, hinzugefügt. Die neuen Gewichtungen werden demzufolge nicht nur auf Basis der neuen Bewertung festgelegt, sondern sie werden mit einer vorher festgelegten Gewichtung  $(1-\alpha)$  hinzugefügt. Ist  $\alpha$  gleich 1, hängt die neue Gewichtung nur von der letzten Änderung ab. Dieses Verfahren verhindert, dass die Gradientensuche vorzeitig beendet wird, wenn ein lokales Plateau erreicht wird. [Kru-2015, S. 58ff.]

## **2.4 Methoden der Bildverarbeitung**

Digitalisierte Bilder sind mathematischen Matrizen gleichzusetzen. Jeder Bildpunkt wird als quantitativer Wert in einer Matrixzelle repräsentiert. Demzufolge entspricht die Dimension der zugehörigen Bildmatrix der Auflösung des Aufnahmesensors. Jede

Zelle der Matrix entspricht bei der einfachen Betrachtung eines Graustufenbilds dem Helligkeitswert. Dies lässt sich analog für Farbbilder übertragen. In dem Fall beinhaltet jede Zelle drei (Intensitäts-) Werte für die Farben Rot, Grün und Blau.

Die folgenden Methoden der Bildverarbeitung sind daher reine Matrixoperationen, welche im Bereich des maschinellen Sehens das Ziel haben, digitalisierte Aufnahmen möglichst optimal für Analysezwecke vorzubereiten.

### 2.4.1 Basisoperationen

Bilder, welche maschinell analysiert werden sollen, werden zumeist von Kameras aufgenommen. Da es bislang nicht möglich ist, ein perfektes Bild mit einer Kamera aufzunehmen, sind alle aufgenommenen Bilder mit Fehlern behaftet. Rauschen, falscher Fokus, ungleichmäßige Belichtung oder Falschfarben sind nur ein paar Beispiele für ungewünschte Änderungen des Abbildes gegenüber seiner Wirklichkeit. Mithilfe von Filtern können viele dieser Bildfehler korrigiert oder entfernt werden. Allerdings können keine neuen Informationen generiert werden, welche vorher nicht vorhanden waren. Sie können nur geschätzt werden, z. B. mittels Interpolation. Ein Beispiel für die Bildverbesserung mittels Filter ist Googles „Night Sight“-Filter, welcher letztlich auf ein DL-Netz basiert. Dieser erzeugt ohne zusätzliche Hardware, wie in Abbildung 2-6 ersichtlich, ein deutlich besser ausgeleuchtetes Bild.



Abbildung 2-6: Anwendung des "Night Sight"-Filters von Google auf ein Bild [Rah-2018].

### Kanaländerung

Zu den einfachen Bildoperationen gehört die Änderung der Kanäle. Ein Beispiel hierfür ist die Konvertierung eines Farbbilds in Graustufen. Eine Kanaländerung hat je nach genutztem Farbmodell unterschiedliche Auswirkungen. Das am häufigsten verwendete Farbmodell ist das RGB- oder BGR-Modell. Bei diesem wird ein Bildpunkt als Bestandteile der Farben Rot, Grün, Blau bzw. Blau, Grün, Rot beschrieben. Wird ein Bild in diesem Farbmodell in Graustufen umgewandelt, wird jeder Bildpunktwert durch den Mittelwert aller Kanäle ersetzt. Je nach Anwendungsfall bieten sich auch andere

Farbmodelle an, wie z. B. das HSV-Farbmodell. Dieses beschreibt das Pixel durch Farbton (engl.: „hue“), Sättigung (engl.: „saturation“) und Helligkeit (engl.: „value“). Ein Wechsel der Kanalanzahl ist mit Ausnahme des RGB-Modells bei den meisten Farbmodellen nicht sinnvoll.

### **Skalierung (engl.: „scaling“)**

Die Änderung der Auflösung eines Bildes dient hauptsächlich zur Verringerung der Rechenleistung zur Analyse des Bildes. Sie kann allerdings auch dazu verwendet werden um Störungen wie z. B. Rauschen zu reduzieren. Das Ergebnis und die benötigte Rechenleistung sind stark von der verwendeten Methode zur Berechnung der neuen Pixelwerte abhängig. [Bra-2011, S. 129f.]

### **Schwelwertfilterung (engl.: „thresholding“)**

Wenn bestimmte Bildwerte interessanter Bildbereiche bekannt sind, bietet es sich an, alle Werte, die außerhalb des Bildwerts zuzüglich einer Toleranz liegen, zu entfernen. Ein Beispiel hierfür wäre die Suche eines blauen Objekts in einem Farbbild. Die Kleinladungsträger in Abbildung 2-7 haben im HSV-Farbmodell einen Farbton von etwa 217 und eine Sättigung von etwa 80. Werden alle Pixel entfernt, die stark von dem Farbton der und Sättigung der Behälter abweichen, sind überwiegend nur noch Pixel der Behälter zu sehen.

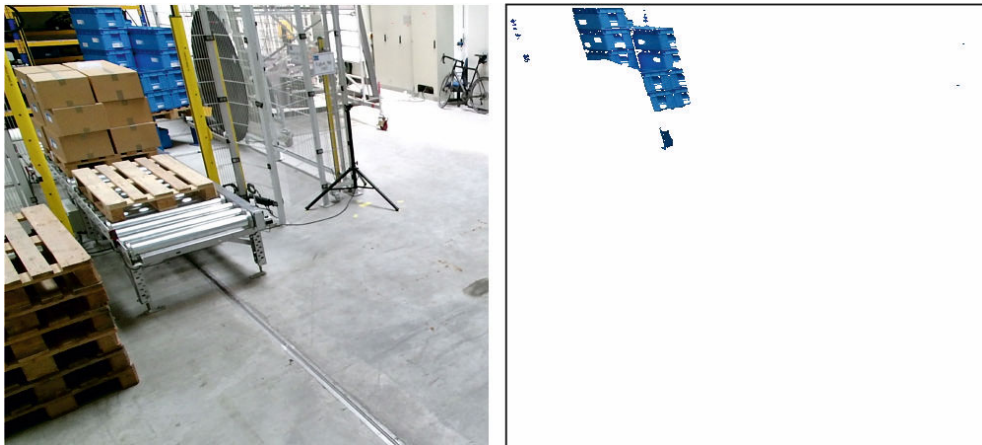


Abbildung 2-7: Beispiel für Schwellwertfilterung (H:217 ± 10, S: 80 ± 20).

### **Normalisierung (engl.: „normalization“) und Egalisierung (engl.: „equalization“) des Histogramms**

Die Belichtung eines Bilds kann in zwei nacheinander erstellten Bildern sehr unterschiedlich sein, sie kann unter anderem den Kontrast sehr beeinflussen. Dadurch verschlechtert sich z. B. die Suche nach Kanten bzw. Konturen im Bild (siehe 2.4.2). Ein Weg, um Helligkeitsunterschiede zwischen Bildern zu minimieren und Kontraste zu

erhöhen, ist das Bild bzw. dessen Histogramm zu normalisieren. Das Histogramm eines Bildes wird gebildet, indem die Häufigkeit aller möglichen Pixelwerte (normalerweise zwischen 0 bis 255) über das gesamte Bild hinweg summiert wird. Zur Normalisierung des Histogramms werden die Pixelwerte linear auf einen neuen Bereich (üblicherweise von 0 bis 255) – wie im Histogramm des mittleren Bilds in Abbildung 2-8 zu sehen – verteilt. Die Form des Histogramms wird dabei nicht verändert, der Vorgang ist reversibel.

Die Egalisierung des Histogramms hat keine Neuverteilung der Pixelwerte zum Ziel, sondern die Angleichung der Häufigkeit einzelner Pixelwerte im gesamten Pixelwertbereich. Dadurch ändert sich bei der Egalisierung auch die Form des Histogramms. Der Vorgang ist im Gegensatz zur Normalisierung irreversibel und erzeugt Bildrauschen. Aufgrund dessen werden Bilder eher normalisiert als egalisiert. [Nix-2010, S. 71ff.]

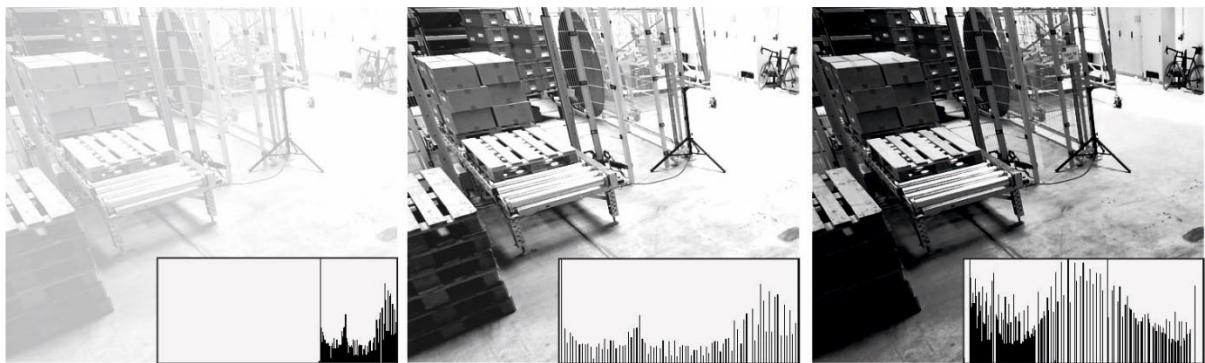


Abbildung 2-8: Originalbild, normalisiertes Bild, egalisiertes Bild, jeweils mit Histogramm.

### Weichzeichnen (engl.: „smoothing“)

Generell ist ein scharfes Bild einem unscharfen Bild vorzuziehen. Allerdings korreliert die Schärfe eines Bildes in der Realität oft mit der Stärke an Rauschen. Durch Weichzeichnen des Bildes kann das Rauschen reduziert werden. Beim Weichzeichnen werden prinzipiell ein Pixel und dessen umliegende Pixel elementweise mit einem Kernel multipliziert. (1) zeigt einen Beispiel-Kernel mit der Dimension 3 x 3 auf Basis der Gauß-Funktion (2): [Sny-2017, S. 93ff.]

$$K_g = \begin{bmatrix} 0,060 & 0,098 & 0,060 \\ 0,098 & 0,162 & 0,098 \\ 0,060 & 0,098 & 0,060 \end{bmatrix} \quad (1)$$

$$g(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)} \quad (2)$$



$$f(x, y) = g(x, y) \otimes A(x, y) \quad (3)$$

mit  $x$  = horizontaler Abstand zum zentralen Pixel,  $y$  = vertikaler Abstand zum zentralen Pixel,  $\sigma$  = Varianz

Das Weichzeichnen mittels der Gaußfunktion hat den Nachteil, dass auch wichtige Merkmale wie Kanten im Bild entfernt werden. Die Nutzung bilateraler Kernel, welche für jedes Pixel individuell berechnet werden, kann das Weichzeichnen von Kanten (vgl. Abbildung 2-9) verhindern oder zumindest mindern:



Abbildung 2-9: Beispiele für Weichzeichnen. (links nach rechts): Originalbild, Gaußsches Weichzeichnen, Weichzeichnen mit bilateralem Filter.

### 2.4.2 Bildmerkmale

Bildmerkmale (engl.: „features“) haben die Aufgabe, ein Bild auf dessen wichtigste Informationen für die nachfolgende Bildanalyse zu reduzieren. Es gibt eine Vielzahl an möglichen Bildmerkmalen, folgend werden die wichtigsten Basismerkmale, Kanten und Eckpunkte, nach *Nixon und Aguado* vorgestellt [Nix-2010, S. 115ff.].

#### Kanten (engl.: „edges“)

Für viele Bildverarbeitungsmethoden sind Kanten das wichtigste Bildmerkmal. Sie sind invariant gegenüber Helligkeitsänderungen und werden häufig dazu genutzt, Regionen eines Bildes wiederzufinden. Die Gesamtheit aller Kanten bilden die Konturen in einem Bild ab (siehe Abbildung 2-10). Sie unterscheiden sich hauptsächlich in ihrer Orientierung und Intensität. Die meisten Kanten-Detektoren nutzen das erste und/oder das zweite Differential des Bildes. Je nach Algorithmus werden dabei horizontale, vertikale oder diagonale Gradienten eines Pixelbereichs (meist quadratisch mit bis zu 5 x 5 Pixeln) berechnet.

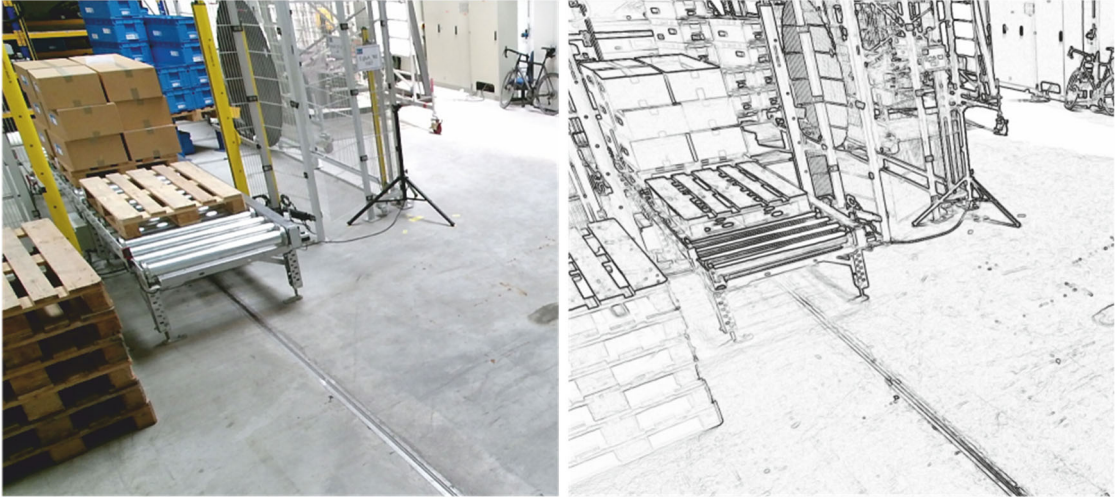


Abbildung 2-10: Beispiel zur Extraktion von Kanten.

### Eckpunkte (engl.: „corners“) und Krümmungen (engl.: „curvatures“)

Kreuzen sich zwei Kanten, handelt es sich um das Merkmal „Eckpunkt“ (siehe Abbildung 2-11). Sie können ebenso entweder durch eine starke Richtungsänderung innerhalb einer Kontur oder wenn eine Kontur endet entstehen. Bei sehr starken Änderungen innerhalb einer Kontur werden diese Regionen auch als Krümmungen bezeichnet. Diese Merkmale sind durch die Richtung und Stärke der Krümmung charakterisiert.



Abbildung 2-11: Beispiel für Eckpunkte mittels Harris-Detektor [Har-1988].

### Komplexere Merkmale (SIFT, SURF, HOG)

Bei komplexen Merkmalen werden zumeist die Basismerkmale wie Ecken und Kanten benutzt, kombiniert und z. B. mit Histogrammen erweitert. Drei wichtige komplexere Merkmale sind die „scale invariant feature transform (SIFT)“- „speeded up robust features (SURF)“- und „histogram of oriented gradients (HOG)“-Merkmale.

Beim **SIFT**-Verfahren werden Merkmale extrahiert, welche unabhängig von der Skalierung, Rotation, lokalen Deformationen und nahezu unabhängig von der Belichtung



eines Bildes sind [Low-1999]. Das SIFT-Verfahren besteht nicht nur aus der Extraktion von Features, sondern enthält zugleich einen Detektor, welcher die extrahierten Merkmale auf anderen Bildern wiederfinden kann. Abbildung 2-12 zeigt, dass die links abgebildeten Produktpackungen anhand ihrer SIFT-Merkmale eindeutig in einem anderen Bild wiedergefunden werden können, obwohl ihre Orientierungen, Skalierungen und Belichtungen nicht gleich sind. Auf Basis des SIFT-Verfahrens wurde 2006 ein schnelleres und robusteres Verfahren von *Bay* namens **SURF** entwickelt. Dieses ist bis heute Standard beim Finden und Verfolgen (engl.: „tracking“) von bereits bekannten Objekten [Bay-2006].



Abbildung 2-12: Beispiel für Textur-Matching mittels SIFT-Merkmalen [Low-1999].

Die Idee der **HOG**-Merkmale wurde in einem 1982 beantragten Patent das erste Mal veröffentlicht [Rob-1986]. Bekannt wurde das Verfahren aber hauptsächlich durch *Dalal und Triggs*. Sie nutzten die Merkmale in Verbindung mit einer SVM zur Erkennung von Fußgängern und erreichten damit eine sehr hohe Erkennungsleistung. Zur Berechnung von HOG-Merkmalen wird das Bild zunächst in der Farbe und in der mittleren Farbhelligkeit (Gamma) normalisiert. Anschließend wird das Bild in sich überlappende Blöcke und diese werden wiederum in Zellen unterteilt. Der Kontrast jedes Blocks wird normalisiert. Dadurch wird eine Invarianz gegenüber Helligkeitsänderungen erreicht. Jede Zelle enthält einen Pixelbereich von z. B. 8 x 8 Pixeln. Für diesen Bereich werden eindimensionale Histogramme der Gradientenrichtungen oder Kantenorientierungen berechnet. Durch die Kombination von Gradientenhistogrammen und die Zusammenfassung von Pixeln in Blöcke und Zellen sind die Merkmale z. B. bei Bildern von Fußgängern relativ unabhängig von deren Körperpose. [Dal-2005] Die Verwendung von HOG-Merkmalen in Verbindung mit einer SVM wird ausführlicher im Abschnitt 2.4.3 beschrieben.

### Bewegungsmerkmale – Optischer Fluss (engl.: „optical flow“)

Die bisher beschriebenen Bildmerkmale können anhand eines einzelnen Bildes erstellt werden. Zur Berechnung von Bewegungsmerkmalen werden mindestens zwei konsekutive Bilder benötigt. Die einfachste Methode, um Bewegungen in Bildern zu identifizieren, ist die Differenzbildung aus den jeweiligen Bildern (vgl. Abbildung 2-13). Das funktioniert allerdings nur, wenn die Kamera stationär ist.



Abbildung 2-13: Differenz zweier aufeinander folgender Bilder (von links nach rechts): Bild 1, Bild 2, Differenzbild.

Für Methoden des sogenannten optischen Flusses wurde eine andere Herangehensweise gewählt. Im Bereich der Bildverarbeitung wurde der optische Fluss erstmals von *Horn und Schunck* erwähnt. In ihrem beschriebenen pixelbasierten Ansatz wird angenommen, dass ein Pixel zwischen Einzelbildern

- einen konstanten Intensitätswert besitzt,
- sich minimal und/oder konstant bewegt,
- zu einer Oberfläche gehört, d. h. die umliegenden Pixel haben dieselbe Bewegung.

Die Bewegungsvektoren einzelner Pixel werden mittels partieller Derivate der Bilder berechnet. [Hor-1981] Das Ergebnis sind Bewegungsvektoren einzelner Pixel oder Bereiche (siehe Abbildung 2-14). Neben den pixelbasierten Verfahren gibt es flächenbasierte Ansätze, wie z. B. das bis heute häufig eingesetzte Lucas-Kanade-Verfahren [Luc-1981]. Deren Vorteil ist hauptsächlich der deutlich geringere Rechenaufwand, allerdings sind die Bewegungsvektoren niedriger aufgelöst.

Moderne Methoden des optischen Flusses sind deutlich komplexer und nutzen mitunter Tiefeninformationen, wie z. B. [Abr-2012] (siehe Abbildung 2-14).

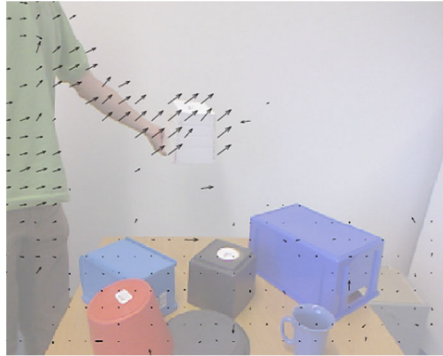


Abbildung 2-14: Darstellung des optischen Flusses, berechnet anhand von zwei konsekutiven Bildern [Abr-2012].

### 2.4.3 Maschinelles Lernen in der Bildverarbeitung

Im Abschnitt 2.3 wurde bereits die grundlegende Funktionsweise von Algorithmen aus dem Bereich ML erläutert. In diesem Abschnitt liegt der Fokus auf der Bildanalyse mittels ML-Methoden. Zuerst wird die Verwendung von SVM zur Bildanalyse beschrieben, gefolgt von DL-Netzen.

#### SVM

Die SVM ermöglicht es (wie in 2.3.1 beschrieben) Klassen voneinander zu trennen. Im Bereich der Bildverarbeitung kann sie dazu genutzt werden, um Bildobjekte zu klassifizieren. In Verbindung mit dem „Sliding-Window“-Verfahren können Objekte auf Bildern zudem lokalisiert werden. Das Sliding-Window-Verfahren teilt ein Bild in sich überlappende Blöcke auf. Beim ersten Durchlauf haben diese Blöcke die Größe des trainierten Detektors; anschließend wird jeder Block von der SVM klassifiziert. Üblicherweise wird dieser Schritt mehrmals mit zunehmend größeren Blöcken wiederholt, sodass auch größere Objekte detektiert werden können. Die Detektionsergebnisse der Blöcke werden daraufhin auf das gesamte Bild zurückgerechnet. Als Feature-Vektoren eignen sich zur Personenerkennung (wie in Abschnitt 2.4.2 beschrieben) HOG-Merkmale [Dal-2006].

Abbildung 2-15 zeigt die Verwendung von HOG-Merkmalen in Verbindung mit einer SVM zur Personenerkennung. Im Bild (a) sind die durchschnittlichen Kanten des INRIA-Trainingssets, mit dem die SVM in diesem Fall trainiert wurde, dargestellt. In den Bildern (b) und (c) sind die positiven bzw. negativen Gewichtungen der trainierten SVM dargestellt. Diese Bilder dienen nur zur Veranschaulichung, die Helligkeit der einzelnen Blöcke soll die maximale Gewichtung eines jeden Blocks repräsentieren. Auf (b) ist zu erkennen, dass bei Menschen vor allem der Kopf, die Schultern und die Füße als relevant für die SVM gelten. Auf (d) bis (g) ist der Detektionsprozess dargestellt: Aus (d) werden die HOG-Merkmale (e) berechnet. Die Anwendung der Gewichtungen

(b) und (c) auf (e) führen zu den Ergebnissen (f) und (g). Anschließend erfolgt die Klassifikation über die SVM anhand von (f) und (g).

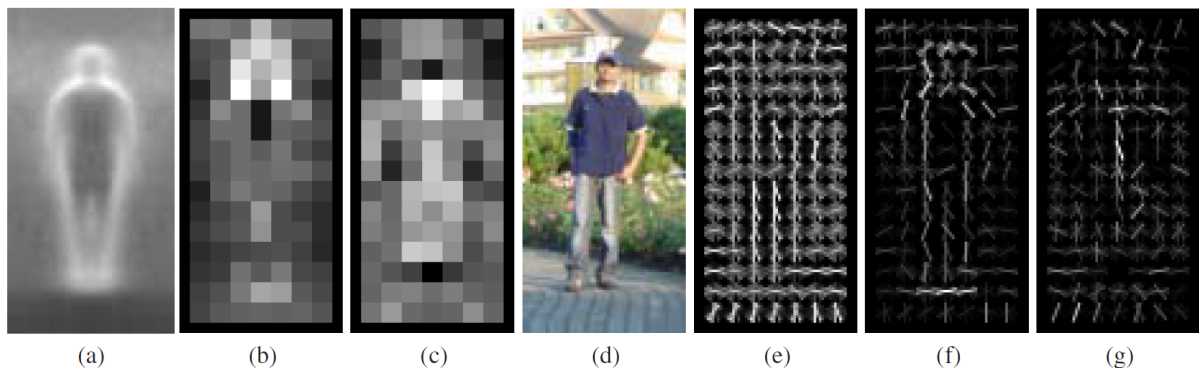


Abbildung 2-15: a) durchschnittliche Kanten des Trainingssets b) Positive Gewichtungen der SVM c) Negative Gewichtungen der SVM d) Beispielbild e) HOG-Merkmale des Beispielbilds f) durch SVM positiv gewichtete HOG-Merkmale des Beispielbilds g) durch SVM negativ gewichtete HOG-Merkmale des Beispielbilds [Dal-2005].

### Deep Learning

Der große Durchbruch von Deep-Learning gelang, wie in Abschnitt 2.3.2 beschrieben, durch den Erfolg eines Algorithmus aus der Bildverarbeitung. Auch heute ist die Bildverarbeitung einer der größten Treiber, vor allem durch die fortschreitende Entwicklung autonom fahrender Automobile, welche auf maschinelles Sehen angewiesen sind. Dadurch werden stetig neue Algorithmen veröffentlicht, welche ihre Vorgänger deutlich übertreffen. Da die Funktionsweise der Algorithmen der in Abschnitt 2.3.2 beschriebenen ähnelt, liegt der Fokus in diesem Abschnitt auf dem Aufbau von Netzen zur Objekterkennung.

Die besten ML-Algorithmen werden jährlich im „ImageNet Large Scale Visual Recognition Challenge“ (ILSVRC) gekürt. Das erste DL-Netz, welches diesen Wettbewerb gewann, war das „AlexNet“. Dieses besteht aus fünf „convolutional layer“ (CL), drei „fully connected layer“ (FCL) und drei „max pooling layer“ (MPL) (siehe Abbildung 2-16):

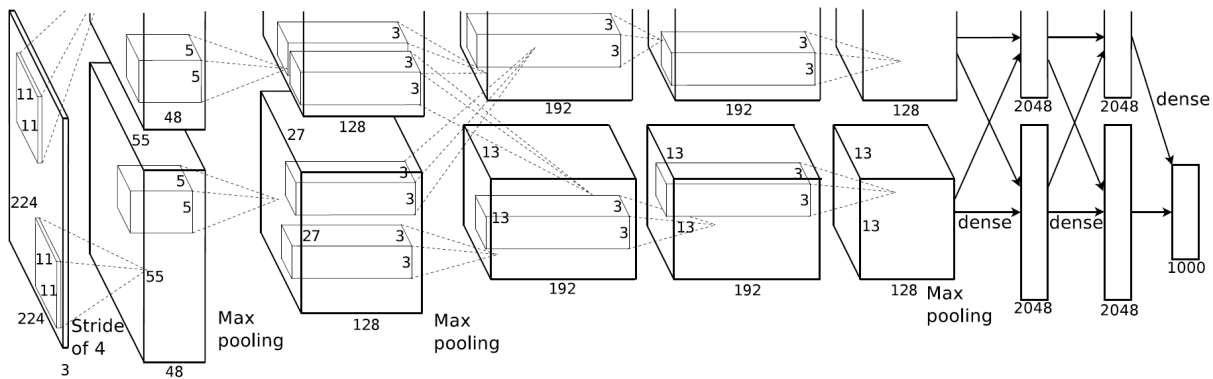


Abbildung 2-16: Architektur des AlexNet [Kri-2012].

Die CL sind dem Namen entsprechend Schichten, welche mathematische Faltungen auf die jeweiligen Matrizen ausführen. Bei diesem Vorgang wird die Auflösung kleiner, die Tiefe zugleich größer. Die Anwendung eines Gaußfilters wäre ein Beispiel für solch eine Faltungsoperation. MPL reduzieren die Größe der Matrizen, indem aus einem durchlaufenden Kernel (z. B. 3 x 3 Pixel) nur der größte Wert für die nachfolgende Schicht verbleibt. In den FCL erfolgt abschließend die Klassifizierung. [Kri-2012]

Neuere Netze wurden im Laufe der Zeit deutlich tiefer, denn je mehr Schichten enthalten sind, desto mehr Bildmerkmale können extrahiert und analysiert werden. Mit den konventionellen Schichten (CL, MPL, FCL) wurden die Ergebnisse allerdings mit zunehmender Tiefe schlechter, obwohl mehr Bildmerkmale genutzt wurden. Der Grund hierfür war, dass je weiter eine Schicht von der Eingangsschicht entfernt war, desto weniger Rohdaten bekam sie als Eingang. Die Lösung für dieses Problem wurde 2016 entdeckt: Die Ergänzung der bestehenden Architekturen um sogenannte „residual layer“ (RL). Die RL haben die Aufgabe die Identität (engl.: „identity“), sprich den Eingangswert, während der Verarbeitung im Neuron beizubehalten. Abbildung 2-17 zeigt die einfachste Variante eines RL, bei der die Summe des Eingangswertes und des Ausgangswertes an das nächste Neuron weitergegeben wird. Die Einführung des RL führte zur deutlichen Verbesserung der Ergebnisse der tiefen neuronalen Netze und zur Bezeichnung dieser Netze als „ResNet“. Das erste mit RL verbesserte Netz gewann sowohl die ILSVRC als auch andere Bilderkennungswettbewerbe im Jahr 2016. [He-2016]

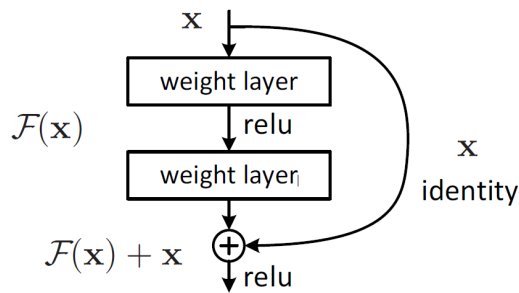


Abbildung 2-17: Beispiel eines „residual layer“ [He-2016].

Die zurzeit verwendeten Res-Net Varianten unterscheiden sich hauptsächlich in der Anzahl der Schichten (50, 101 oder 152).

Wichtig bei der Nutzung eines DL-Netzes zur Klassifikation bzw. Lokalisierung ist nicht nur die zugrundeliegende Architektur, sondern auch der verwendete Detektor: Erst der Detektor ermöglicht die Lokalisierung eines Objekts. Ohne Detektor würde ein neuronales Netz nur dazu in der Lage sein, eine Aussage darüber zu treffen, welche Kategorie einem Bild zuzuordnen ist. Der einfachste Detektor ist ein Sliding-Window, der bereits bei der SVM beschrieben wurde. Im Bereich von DL wurden im Laufe der Zeit mehrere Detektoren entwickelt, welche sich in ihrer Leistung kontinuierlich übertrafen: „Fast R-CNN“ [Gir-2015], gefolgt von „Faster R-CNN“ [Ren-2015] und kürzlich „Mask R-CNN“ [He-2017]. Letzterer liefert nicht nur ein Rechteck, in dem sich das Objekt befindet, sondern ist dazu in der Lage, jedes einzelne Pixel des gesuchten Objekts zu identifizieren. Dieser ist der zurzeit beste Detektor hinsichtlich der Erkennungsrate.

Für manche Systeme ist allerdings nicht nur die beste Erkennungsrate, sondern auch die dazu notwendige Rechenleistung von Belang. Wichtige Vertreter schnellerer Detektoren sind in chronologischer Reihenfolge: „SSD“ [Liu-2016], „YOLOv1“ [Red-2016b], „Yolo9000“ („YOLOv2“) [Red-2016a], „YOLOv3“ [Red-2018].

## 2.5 Bewertung des Stands der Technik

In diesem Kapitel wurden Sensortechnologien vorgestellt, welche es prinzipiell ermöglichen, Kollisionsgefahren über Distanzmessung zu erkennen. Diese Technologien werden mitunter in den vorgestellten Kollisionswarnsystemen für Gabelstapler verwendet. In diesem Abschnitt werden die jeweiligen Technologien bzw. Systeme kritisch auf ihre Eignung hinsichtlich der Fragestellung zur Kollisionswarnung am Gabelstapler gegenübergestellt.

### **2.5.1 Kollisionswarnsysteme für Gabelstapler**

Funkbasierte Systeme (siehe Abschnitt 1.1: 1,5,8) haben den Vorteil, dass keine Sichtverbindung zwischen Personen und Gabelstaplern notwendig ist, damit eine Annäherung erkannt wird. Jedoch können hierbei auch nahe Personen detektiert werden, wenn sie sich bspw. unsichtbar quer zur Fahrtrichtung des Staplers in einer Gasse befinden und dementsprechend nur einer geringen Gefahr ausgesetzt sind. Durch die Verwendung von Master- und Slave-Modulen besteht zudem der Nachteil, dass jede Person ein Modul tragen muss, damit ihre Nähe zu Staplern bestimmt werden kann. Außerdem können Kollisionen mit nicht ausgerüsteten Gegenständen, wie z. B. mit Waren oder Regalen, nicht verhindert werden.

Ein zusätzlicher Nachteil ergibt sich aus dem einzig auswertbaren Parameter: der Signalstärke, welche mit der Distanz indirekt proportional korreliert. Aus ihr lässt sich zwar annähernd abschätzen, ob der Abstand zwischen dem Stapler und einer Person in Luftlinie kleiner wird, allerdings nicht, ob diese Verkleinerung eine Relevanz bezüglich einer Kollision hat. Fährt beispielsweise ein Gabelstapler eine Wand entlang und befindet sich eine Person auf der anderen Seite, würde das System warnen, obwohl Stapler und Person voneinander räumlich getrennt sind. Auch die Annahme, dass die Signalstärke entsprechend der Distanz abnimmt, ist aufgrund von Abschirmeffekten nicht immer korrekt. Ein weiterer Nachteil ist der hohe Aufwand für die Erstausrüstung des Systems, da jeder Mitarbeiter, Besucher und Stapler ein Funkmodul benötigt und zum Teil eine Anbindung des Staplermoduls an den Staplerrechner über den CAN-Bus notwendig ist.

Das 2D-Lidar-basierte System ELOprotect (4) hat den Vorteil einer sehr hohen Messfrequenz und Reichweite. Die Messung über Lidar ist nahezu störungsfrei auf allen Materialien möglich. Allerdings misst das System analog zu den funkbasierten Produkten lediglich Abstände, wodurch die Gefahr von zu häufigen Warnungen (bzw. in diesem Fall auch Geschwindigkeitsdrosselungen) besteht. Vor allem bei Staplern ist dieser Aspekt einer der Schwachstellen aktueller Systeme. Stapler manövrieren oftmals mit sehr geringem Abstand, z. B. beim Be- oder Entladen von Paletten. Ein abstands-basiertes System wird in diesen Situationen sehr wahrscheinlich warnen. Das gleiche Problem besteht bei den zwei kamerabasierten Lösungen (2, 7) und bei der Ultraschalllösung (6) der tbm hightech control GmbH. Einzig das System von Blaxtair hat eine Personenerkennung integriert, welche einen separaten Abstandsgrenzwert zulässt. Die Funktionsweise der Personenerkennung ist den Informationen des Herstellers allerdings nicht zu entnehmen, daher kann auch keine Einschätzung gegeben werden, wie zuverlässig sie ist.

Als Hauptprobleme bestehender Produkte zur Kollisionsvermeidung lassen sich demnach die folgenden zwei Punkte nennen:

1. Verwendung des Abstands zur Identifikation von bevorstehenden Kollisionen
2. Fehlende Relativierung der Kollision, da keine Objekterkennung (außer Blaxtair) eingesetzt wird

Die Folge der genannten Probleme sind zu häufige Warnungen vor nicht existenten Kollisionsgefahren. Dieser Aspekt führt in der Praxis letztlich zu einem niedrigen Akzeptanzlevel der Anwender. Entweder werden die Systeme deshalb nicht gekauft oder sie werden früher oder später aufgrund der häufigen Fehlwarnungen ausgeschaltet.

### **2.5.2 Sensortechnologien**

Für die Erfassung der Umwelt ist die richtige Auswahl der Sensorik essenziell. Die Sensoren aus Abschnitt 2.1 unterscheiden sich vor allem in ihrer Reichweite, ihrer Auflösung, ihren Datenarten und in den abgebildeten Dimensionen. Im Folgenden wird die Eignung hinsichtlich einer Kollisionswarnung auf Basis der Quellen von Tabelle 2-1 beschrieben.

Ultraschallsensoren weisen für den Anwendungszweck eine zu geringe Reichweite (< 4 m) auf. Es sind zwar Produkte mit geringfügig höheren Reichweiten erhältlich, aber diese werden nur über kleine Öffnungswinkel erreicht. Des Weiteren messen Ultraschallsensoren für weiche, schallabsorbierende Stoffe andere Entfernungen als für harte Oberflächen. Eine Kalibrierung für mehrere Oberflächentypen ist nicht möglich, sodass z. B. entweder für metallische Oberflächen die richtige Entfernung gemessen wird oder aber für Menschen. Auch größere Temperaturunterschiede können die Werte negativ beeinflussen. Ferner ist die Lokalisierung eines Objekts nur durch Kombination mehrerer Sensoren möglich.

Stereokameras liefern zwei Farbbilder aus unterschiedlichen Perspektiven. Das zugehörige 3D-Bild wird bei keinem der zurzeit erhältlichen Kameras direkt übermittelt. Üblicherweise wird der Kamera eine Bibliothek zur Verfügung gestellt, welche die Berechnung auf dem Computer ermöglicht. Die Berechnung ist jedoch sehr rechenaufwändig. Sie sind relativ robust gegen äußere Umwelteinflüsse, allerdings benötigen sie bei starken Änderungen des Lichts bis zu mehrere Sekunden, um die Belichtung anzupassen. Die Kameras sind durch den Abstand der zwei Sensoren auf einen Entfernungsbereich ausgelegt. Außerhalb des Bereichs werden die Tiefenwerte zunehmend ungenauer. Ansonsten sind keine relevanten negativen Eigenschaften bezüglich der Verwendung in einem Kollisionswarnsystem vorhanden.



Ein großer Vorteil von Lidarsensoren sind die robusten Werte. In Betracht kommen aufgrund des Anwendungsfalls nur Varianten, welche 3D-Bilder liefern. Die Funktionsweise der Ablenkung eines Lasers bedingt, dass die Auflösung linear mit der Entfernung abnimmt. Entsprechend liegt die Auflösung bei einer Winkelgenauigkeit von z. B.  $2^\circ$  („Panasonic 3D-Lidar“) bei einer Meter Entfernung bei drei Zentimetern, bei fünf Metern Entfernung liegt sie bei 15 Zentimetern [Pan-2017]. Eine negative Beeinträchtigung der Distanzmessung ist nur durch Nebel möglich, innerhalb von Lagerhallen ist das aber als irrelevant zu betrachten. Kritischer sind die fehlenden Daten über die Textur. Zwar lassen die Stärke der reflektierten Strahlen Rückschlüsse auf die Oberfläche zu, eine Personenerkennung ist damit aber nur schwer realisierbar. Dieselben Schwachstellen finden sich auch bei den Flash-Lidar-Sensoren, welche überdies selbst in sehr geringen Distanzen eine niedrige Auflösung haben.

ToF-Kameras nutzen die Vorteile der Lidartechnik, haben aber zusätzlich eine Möglichkeit Texturen zu erfassen. Durch die verwendeten CCD- oder CMOS-Sensoren liefern diese neben den Tiefen- auch Intensitätsbilder (im Folgenden auch IR-Bilder genannt), welche sich mit moderatem Aufwand in graustufenähnliche Bilder konvertieren lassen. Zugleich nimmt die Auflösung nicht mit zunehmender Entfernung ab. Die aus dem Entertainmentbereich bekannte ToF-Kamera „Kinect v2“ (oder „Kinect One“) von Microsoft hat einen zusätzlichen Farbsensor, dessen Bild auf das Tiefenbild über die intern – auf der Kamera – gespeicherten Parameter kalibriert werden kann. Relevante Beeinträchtigungen treten bei direktem Sonnenlicht oder bei blanken Metallflächen auf. Da im Umfeld eines Lagers sehr viel Metall vorhanden ist, kann sich dies bei der Verwendung in einem Kollisionswarnsystem negativ auf die Funktion auswirken.

SL-Kameras nutzen das gleiche Spektrum wie ToF-Kameras. Sie liefern ebenfalls ein Intensitätsbild, ein 3D-Bild und je nach Produkt auch ein kalibriertes Farbbild. Störungen durch Metall sind bei SL-Kameras deutlich geringer als bei ToF-Kameras, da das projizierte Muster außerhalb von Metallflächen nicht beeinträchtigt wird. Störungen durch direktes Sonnenlicht werden mittels der Emission modulierter Infrarotstrahlen umgangen. Ein Nachteil des Intensitätsbilds von SL-Kameras ist, dass das projizierte Muster, welches zur Tiefenbildberechnung verwendet wird, auf diesem erkennbar ist. Vor allem im Nahbereich ist das Muster deutlich sichtbar und verfälscht die eigentliche Texturinformation. In der Reichweite ähneln die SL-Kameras den ToF-Kameras. Einzig im Hinblick auf die Auflösung und Robustheit der Tiefenwerte waren zu Beginn der Forschungsarbeit die SL- den ToF-Kameras deutlich unterlegen. Mittlerweile jedoch liefern sie vergleichbare Werte, z. B. das Modell „RealSense D435“ von Intel (vgl. [Int-2018]).

Radarsensoren haben die größte Reichweite im Feld der betrachteten Distanzmessungstechnologien. Jedoch haben sie den Nachteil eines sehr kleinen Öffnungswinkels. Die resultierenden Radardaten – Entfernung, Winkel und Größe reflektierender Flächen – lassen darüber hinaus keinen Schluss auf die genaue Umgebung zu. Es gibt bspw. keine Informationen zu Kanten von Objekten, da erkannte Objekte zu Punkten konsolidiert werden. Ähnlich wie bei Ultraschall und Lidar fehlen ausreichende Texturinformationen, um eine Objekterkennung sinnvoll umsetzen zu können.

Die Verwendung mehrerer Funkmodule zur Distanzmessung hat den Vorteil, dass die Messung unabhängig von einem freien Sichtfeld ist. Eine Personenerkennung ist in diesem Fall nicht notwendig, da Personen extra Module tragen können, die sie als solche identifizieren. Der größte Nachteil liegt darin, dass jegliches Kollisionsobjekt mit einem Sender ausgestattet sein muss. Der Aufwand für größere Unternehmen mit mehreren 100 Mitarbeitern wäre immens. Darüber hinaus muss immer gewährleistet sein, dass jeder Sender funktioniert und dass jeder Gast gefährliche Bereiche nur mit einem Sender betreten kann. Ferner ist durch die alleinige Entfernungsbestimmung nur eine eingeschränkte Aussage darüber möglich, in welcher Position sich eine Person sich relativ zum Stapler befindet.

Grundsätzlich liefern alle Sensoren mit Ausnahme der Kameras nur Tiefen- bzw. Distanzinformationen. Zur Erkennung von Personen ist die zusätzliche Information über die Textur notwendig. Neueste Algorithmen im DL-Bereich werden hauptsächlich auf die Erkennung in Farbbildern ausgelegt und trainiert. Die besten Ergebnisse sind daher auch bei der Verwendung von Farb- oder Graustufenbildern zu erwarten.

Zusammenfassend sind nur kamerabasierte Sensoren für das Forschungsvorhaben geeignet. Die Kameras wurden im Rahmen einer Nutzwertanalyse (siehe Tabelle 2-3) anhand der wichtigsten technischen Daten gegenübergestellt.

Tabelle 2-3: Ergebnisse der Nutzerwertanalyse über die Eignung für ein Kollisionsgefahrerkennungssystem. (1 – sehr schlecht, 3 – neutral, 5 – sehr gut)

	Gewichtung	Stereo-kamera	Flash-Lidar	ToF-Kamera	SL-Kamera
<b>Reichweite (Tiefendaten)</b>	15%	4	5	4	4
<b>Auflösung (Tiefendaten)</b>	15%	3	1	3	3
<b>Auflösung (Texturdaten)</b>	15%	5	1	3	2
<b>Öffnungswinkel horizontal</b>	15%	4	3	3	3
<b>Öffnungswinkel vertikal</b>	15%	4	3	3	3
<b>Rechenaufwand</b>	10%	1	5	5	3
<b>Datenqualität bei schlechten Lichtverhältnissen</b>	10%	2	5	5	5
<b>Andere Störeinflüsse</b>	5%	5	4	4	5
<b>Summe</b>	100%	3,55	3,15	3,6	3,3

Das Ergebnis der Nutzwertanalyse war, dass für die Kollisionsgefahrerkennung mit Ausnahme des Flash-Lidarsensoren keine deutlichen Unterschiede zwischen den Kameras existieren. Aufgrund der besten Eignung wurde für die weitere Entwicklung eine ToF-Kamera als Sensor vorgesehen, wobei das System aufgrund ähnlicher Datenarten auch für die zwei anderen Sensorarten funktionieren soll. Als geeignetes Produkt wurde die Microsoft „Kinect v2“ identifiziert, da sie als einzige ToF-Kamera auch ein Farbbild und die höchste Auflösung bei den Tiefendaten im Vergleich zu den seinerzeit am Markt erhältlichen Alternativen lieferte.

## 2.6 Ableitung der Forschungsfrage

In Kapitel 1 wurde dargelegt, dass Unfälle bei Gabelstaplern im intralogistischen Transport trotz zunehmenden Einsatzes von FTS ein relevantes Problem sind. Die Übersicht und abschließende Bewertung bestehender Kollisionswarnsysteme zu Beginn des Kapitels 2 zeigte, dass diese alle die Distanz als Messgröße haben, wodurch Falschwarnungen prädestiniert sind. Überdies besitzen alle außer eines dieser Systeme keine Kenntnis über verschiedene Objekttypen, somit sind keine differenzierten Warnungsstufen möglich. Zum größten Teil sind die Mängel bisheriger Systeme durch die verwendete Sensorik bedingt, teils aber auch durch die Auswertung der Daten. Letzteres betrifft vor allem diejenigen Produkte, welche Kameras einsetzen. Im Falle der restlichen Produkte sind die Möglichkeiten der Sensordaten überwiegend ausgeschöpft.

Das größte Potenzial liegt nach der Recherche in der Verwendung einer kamerabasierten Sensorik, da hierbei viele Möglichkeiten, die durch den Fortschritt im Bereich

der Bildanalyse entstanden sind, im aktuellen Stand der Technik nicht genutzt werden. Vor allem die Nutzung von Methoden des maschinellen Lernens werden in keinem bisher erhältlichen oder erforschten System ausgenutzt.

Als Forschungsfrage ergibt sich somit:

Ist es möglich, ein Kollisionswarnsystem für Gabelstapler auf Basis von 3D-Kameras zu entwickeln, welches möglichst keine Fehlwarnungen erzeugt bei gleichzeitiger robuster Erkennung gefährlicher Situationen?

Aus dieser Fragestellung ergeben sich die folgenden Teilfragen:

1. Welche Methoden der Bildverarbeitung können zur Kollisionsgefahrerkennung am Gabelstapler genutzt werden?
2. Wie können Kollisionsgefahren relativiert bzw. kategorisiert werden, sodass möglichst keine Fehlwarnungen entstehen?
3. Wie lassen sich die eingesetzten Methoden für den Anwendungsfall im intralogistischen Transport optimieren?
4. Welche Fahrsituationen können mit einem kamerabasierten Kollisionswarnsystem abgedeckt werden?

Aus der Forschungsfrage ergibt sich das Forschungsziel der Entwicklung eines kamerabasierten Kollisionswarnsystems, welches in der Lage ist:

- Kollisionen zuverlässig zu erkennen,
- zwischen Kollisionen mit hohem und niedrigem Gefährdungspotenzial für Menschen zu unterscheiden,
- die Geschwindigkeit von Stapler und etwaigen Kollisionsobjekten in die Vorhersage miteinzubeziehen.



## 3 Konzeptentwicklung

---

In Kapitel 2 wurde nach der Recherche des Stands der Technik die Forschungsfrage nach der Machbarkeit eines kamerabasierten Kollisionswarnsystems für Gabelstapler abgeleitet. In diesem Kapitel werden zuerst die Anforderungen an ein derartiges System auf Basis des Stands der Technik und anhand von Erkenntnissen aus bisherigen Unfällen sowie gabelstaplerrelevanten Normen abgeleitet. Unter Berücksichtigung der Anforderungen werden in den darauffolgenden Abschnitten sowohl ein hard- als auch softwaretechnisches Konzept zur möglichen Umsetzung beschrieben.

Teile dieses Kapitels greifen Ergebnisse aus dem Forschungsprojekt „PräVISION – Methodenentwicklung zur präventiven Steigerung der Arbeitssicherheit an Flurförderzeugen mit Umsetzung eines Assistenzsystems durch Fusion und Analyse von 2D- und 3D-Bilddaten“ auf, welche in [Jun-2016; Bör-2018; Lan-2017] dokumentiert sind.

### 3.1 Anforderungen an ein Fahrerassistenzsystem für Gabelstapler

Der erste Schritt zur Entwicklung eines Fahrerassistenzsystems (engl.: „Driver Assistance System“ (DAS)) für Gabelstapler ist die Analyse bisheriger Unfälle. Ein ausschlaggebender Punkt ist z. B. die Geschwindigkeit und Richtung des Gabelstaplers, bevor ein Unfall aufgetreten ist. Danach folgt die Behandlung relevanter Normen und Richtlinien. Wichtig hierbei sind vor allem die Vorgaben zum erlaubten Bremsverhalten von Flurförderzeugen. Da Gabelstapler sehr weitläufig eingesetzt werden und auch die Varianten zahlreich sind, wird anschließend das Anwendungsgebiet, in welchem das System eingesetzt werden soll, beschrieben. Abschließend werden in diesem Abschnitt die Systemanforderungen nach der MoSCoW-Methode priorisiert.

#### 3.1.1 Analyse von Unfalldaten

Die „Deutsche Gesetzliche Unfallversicherung“ (DGUV) veröffentlicht jährlich die Statistiken zu meldepflichtigen Arbeitsunfällen in Deutschland. Als meldepflichtig werden Unfälle betrachtet, bei welchen Menschen verletzt und aufgrund dessen mindestens vier Tage arbeitsunfähig wurden. Die Entwicklung bei meldepflichtigen Unfällen mit Gabelstaplern ab 2010 (vgl. Abbildung 3-1) zeigt, dass die Anzahl an Unfällen tendenziell leicht zunimmt. Die tödlichen Unfälle sind zwar gesunken, allerdings könnte es sich aufgrund der geringen Anzahl um zufällige Schwankungen handeln.

### 3 Konzeptentwicklung

Die Statistik aus dem Jahr 2017 gibt an, dass in 30 % der Unfälle der Gabelstaplerfahrer selbst verletzt wurde. In den übrigen 70 % der Fälle wurde demzufolge ein anderer Mensch direkt (z. B. anfahren, quetschen) oder indirekt (z. B. herunterfallende Ladung, umgestoßenes Regal) durch einen Stapler verletzt.

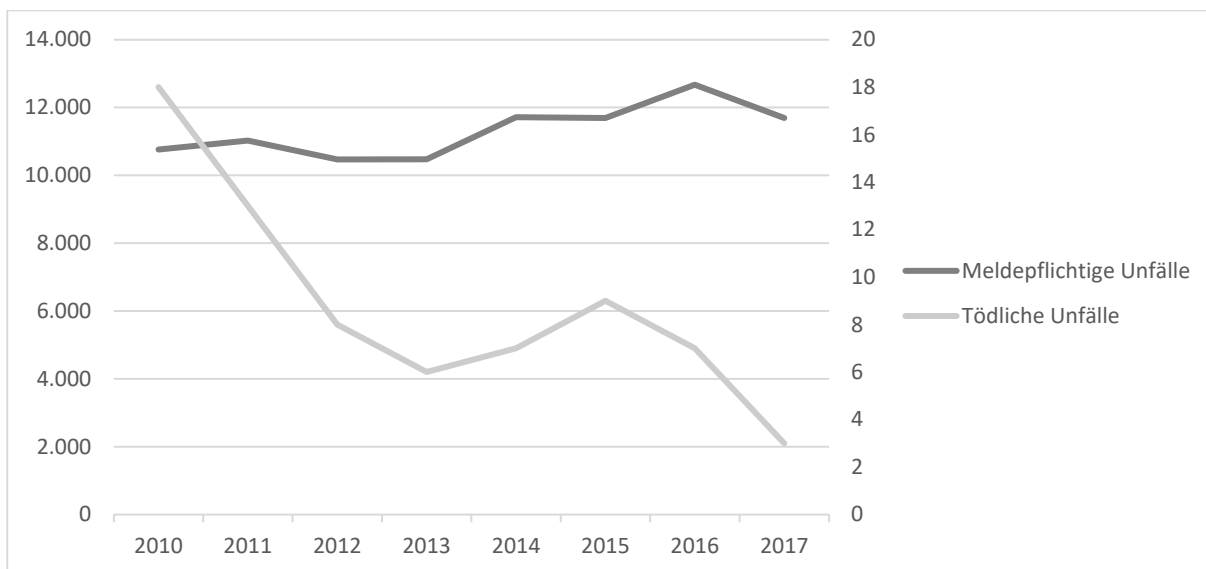


Abbildung 3-1: Entwicklung der Unfälle mit Gabelstaplern nach [Deu-2011; Deu-2012; Deu-2013; Deu-2014; Deu-2015; Deu-2016; Deu-2017; Deu-2018] [Deu-2013; Deu-2011; Deu-2014; Deu-2012; Deu-2015; Deu-2016; Deu-2017; Deu-2018].

Die von der DGUV veröffentlichten Zahlen geben allerdings keine Auskunft über den Unfallhergang. Daher wurden im Rahmen des Projektes „PräVISION“ in Zusammenarbeit mit der „Berufsgenossenschaft Handel und Warenlogistik“ (BGHW) Gabelstaplerunfälle detaillierter untersucht. Die Auswertung konnte nur für Gegengewichtsstapler durchgeführt werden, da für andere Staplerarten zu wenige Unfalldaten vorlagen. Insgesamt wurden nach *Böroid et al.* 320 Datensätze betrachtet. [Bör-2018, S. 7ff]

Die Analyse ergab, dass 55 % der Unfälle während der Rückwärtsfahrt passieren, wobei die Geschwindigkeit des Staplers zu 98 % geringer als 8 km/h war. Bei etwa jedem zweiten Unfall bewegte sich der Stapler aus der Ruhe heraus.

Unfälle während der Vorwärtsfahrt (34 %) passierten zu 57 % im normalen Geschwindigkeitsbereich (8-18 km/h). Nur 15 % der Unfälle sind Anfahrvorgängen zuzuordnen.

In Abbildung 3-2 ist dargestellt, in welchem Bereich sich die verletzten Personen in Abhängigkeit der Fahrtrichtung des Staplers befunden haben. Auffällig hierbei ist die signifikant asymmetrische Verteilung der gefährlichen Sektoren. Sowohl bei der Rückwärts- als auch bei der Vorwärtsfahrt ist der Bereich links vom Fahrer am gefährlichsten.

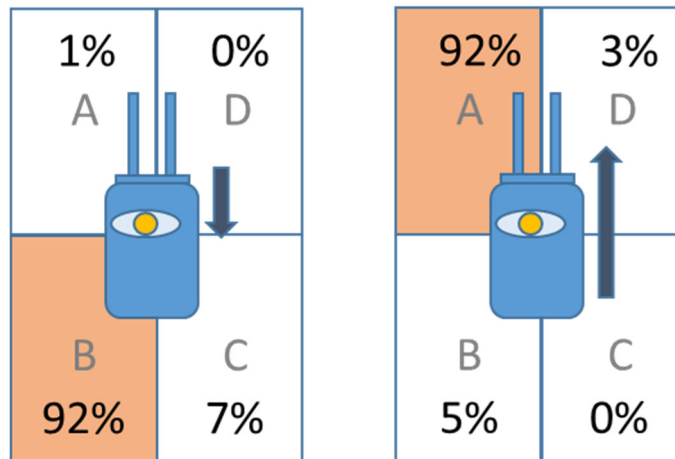


Abbildung 3-2: Verteilung der Unfälle nach Aufenthaltsbereich des Geschädigten bei Rückwärts- bzw. Vorwärtsfahrt nach [Bör-2018, S. 7ff].

#### 3.1.2 Richtlinien und Normen

Im Zusammenhang mit Gabelstaplern gibt es zwar Normen für technische Parameter der Maschinen wie bspw. den maximalen Bremsweg oder die maximale Ladung, allerdings existieren keine Vorschriften bezüglich Kollisionswarnsystemen bei Flurförderzeugen. Die BGHW empfiehlt eine Orientierung an der „DIN 15185-2“ („Flurförderzeuge – Sicherheitsanforderungen – Teil 2: Einsatz in Schmalgängen“) [DIN 15185-2; DIN 15185-2; Gau-2015; DIN 15185-2]. Diese beschränkt sich jedoch hauptsächlich auf stationäre Anlagen. Die Vorgaben zu Sensoren am Flurförderzeug sind auf Schmalgangstapler und die damit verwendeten Systeme zur Kollisionsvermeidung angepasst. [DIN 15185-2] Eine Übertragung auf andere Staplertypen gestaltet sich als schwierig.

#### Bremsweg von Gabelstaplern

Aufgrund mangelnder Normen und Richtlinien ist es zur Bestimmung der Anforderungen an ein Kollisionswarnsystem u. a. erforderlich zu wissen, wie groß der Bremsweg eines Staplers sein darf. Die Norm „DIN ISO 6292“ dient zur Auslegung von Bremsystemen von Flurförderzeugen [DIN ISO 6292]. Diese Richtlinie beinhaltet den maximal erlaubten Bremsweg in Abhängigkeit der Geschwindigkeit eines Flurförderzeugs. Die Angaben beruhen auf dem „Worst-Case“-Szenario: der Stapler ist voll beladen und die Reifen sind bis zur Grenze des zugelassenen Bereichs verschlissen. Anhand dieser Norm kann der maximale Bremsweg, den ein Gabelstapler beim Bremsen zurücklegen darf, bestimmt werden. Dementsprechend muss ein Kollisionswarnsystem dazu in der Lage sein, soweit im Voraus zu warnen, sodass zumindest der maximal zulässige Bremsweg eingehalten werden kann.



### 3 Konzeptentwicklung

Da die Richtlinie auf Hersteller von Flurförderzeugen ausgerichtet ist, fehlt in der Norm die Berücksichtigung der Reaktionszeit des Fahrers. Diese beträgt beim Gabelstapler nach Untersuchungen von unterschiedlichen Warnungsmodalitäten von *Vom Stein et al.* maximal eine halbe Sekunde [Vom-2018]. Wird der Weg des Staplers in einer halben Sekunde zum Bremsweg aus der Norm addiert, ergibt sich folgender Verlauf:

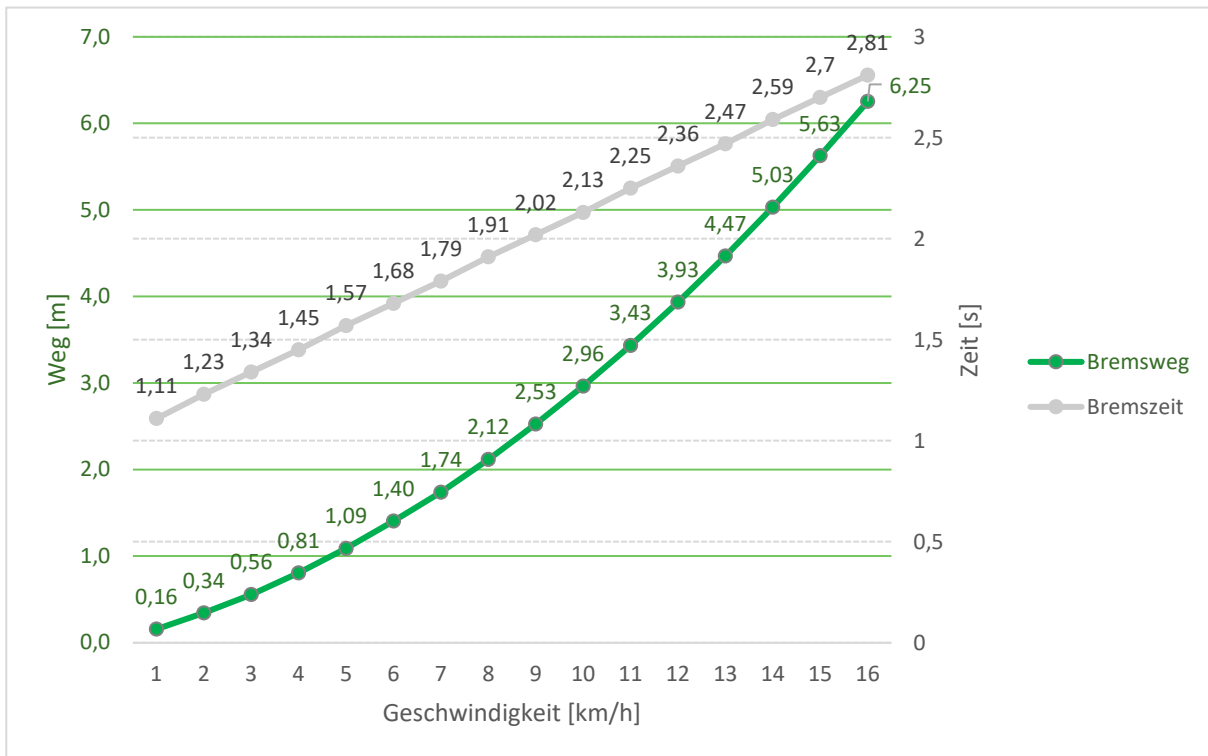


Abbildung 3-3: Maximaler Bremsweg und -zeit eines Staplers mit Berücksichtigung der Reaktionszeit.

### Statistische Verteilung der Körpermaße von Menschen

Primär soll das System auf die Vermeidung von Personenunfällen ausgelegt werden. Anhand statistischer Werte zur äußeren Gestalt von Menschen können Anforderungen für die Kollisions- und Personenerkennung abgeleitet werden. Die Norm „DIN 33402-2“ beinhaltet durchschnittliche Werte für zahlreiche Körperregionen auf Basis von Untersuchungen der 18- bis 65-jährigen deutschen Bevölkerung. Für die Kollisionsgefahrerkennung sind die minimalen bzw. maximalen Konturmaße relevant. Diese lassen sich durch die 5- und 95-Perzentilverteilungen der Körperhöhe, Schulterbreite und Körpertiefe (siehe Tabelle 3-1) ableiten.

### 3.1 Anforderungen an ein Fahrerassistenzsystem für Gabelstapler

Tabelle 3-1 Ausgewählte Körpermaßperzentile zur Beschreibung der minimalen bzw. maximalen Körperkontur [DIN 33402-2]:

Parameter	Minimal (5-Perzentil)	Maximal (95-Perzentil)
<b>Schulterbreite</b>	0,40 m (Frauen)	0,53 m (Männer)
<b>Körperhöhe</b>	1,54 m (Frauen)	1,86 m (Männer)
<b>Körpertiefe</b>	0,25 m (Frauen)	0,38 m (Männer)

Aus den Werten ergibt sich eine minimale und maximale Breite von 0,25 bzw. 0,53 Metern. Die Höhe der untersuchten Personen liegt im Bereich von 1,54 bis 1,86 Metern.

#### 3.1.3 Definition des Einsatzzwecks

Das System dient zur Erkennung von Unfallgefahren mit Gabelstaplern – insbesondere jenen, bei denen Menschen verletzt werden könnten. Der Einsatz wird auf die manuell bedienten Gabelstaplertypen Gegengewichts- und Schubmaststapler eingeschränkt bzw. ausgelegt. Eine mögliche Anwendung in anderen Szenarien soll hiermit aber nicht explizit ausgeschlossen werden.

Ein Eingriff in die Fahrzeugsteuerung ist nicht vorgesehen. Folglich wird keine Anbindung an herstellerspezifische Schnittstellen berücksichtigt. Die Montage und Verwendung müssen ebenso unabhängig vom Fahrzeug bzw. dem Hersteller möglich sein.

#### 3.1.4 Ableitung von Anforderungen

Anhand der Recherchen im Bereich des Standes der Technik, den Unfalldaten und den Normen lassen sich die folgenden Anforderungen an ein Kollisionsgefahrerkennungssystem für Gabelstaplerfahrer ableiten. Diese werden nach der MoSCoW-Methode in notwendige („**M**ust“-), gewünschte („**S**hould“-) und mögliche („**C**ould“-) Anforderungen priorisiert. Die Kategorie unschlüssiger („**W**on’t“-) Anforderungen werden nicht verwendet, da diese nur in Produktentwicklungen mit der Beteiligung mehrerer Interessensvertreter (z. B. Investoren) sinnvoll ist.

Die Anforderungen werden in mehrere Kategorien eingeteilt: Funktionalitäten, Kollisionsgefahrerkennung, Personenerkennung, Bedienung, Sensor und Sonstiges. Unter dem Punkt Funktionalität (des Gesamtsystems) ist die Befähigung zur Erkennung von Kollisionsgefahren und von Personen notwendig. Ebenso ist ein Modus obligatorisch, in welchem das System nur Kollisionen mit Menschen berücksichtigt. Dieser Modus soll bei Prozessen verwendet werden, in welchen sich die Gabelstapler in sehr engen Fahrsituationen befinden. Eine Warnung an den Fahrer ist hauptsächlich zu Testzwecken vorgesehen, da der Fokus der Arbeit nicht in der wissenschaftlichen Erarbeitung liegt, wie ein Gabelstaplerfahrer am besten vor Gefahren gewarnt werden kann.

Die Kollisionsgefahrerkennung muss rechtzeitig erfolgen. Das bedeutet, dass im Falle einer vermeidbaren Kollision das System in Berücksichtigung des Bremsweges vor der benötigten Bremszeit nach Abbildung 3-3 die Kollisionsgefahr erkennt. Für Kollisionen, welche nicht mehr vermeidbar sind, z. B. wenn ein Mensch aus einer Gasse in kurzer Distanz in den Fahrweg des Staplers läuft, wird kein spezifischer Grenzwert festgelegt. Allgemein sollte die Kollisionsgefahrerkennung mindestens in Echtzeit berechnet werden, wobei die Echtzeit auf die Frequenz des eingesetzten Sensors bezogen ist. Die ToF-Kamera Kinect v2 liefert ihre Daten mit einer Frequenz von 30 Hz, die Berechnungszeit sollte demzufolge unter 33 Millisekunden liegen. Diese Anforderung ist aber nicht obligatorisch, da eine kontinuierliche Leistungssteigerung der Rechenhardware zu erwarten ist.

Die minimale Größe der zu detektierenden Kollisionsobjekte wurde auf 20 cm in allen Dimensionen festgelegt. Damit sind etwaige heruntergefallene Objekte wie z. B. Lagersichtkästen abgedeckt.

Um Kollisionsgefahren kategorisieren zu können muss die Kollisionsgefahrerkennung die Informationen der Personenerkennung nutzen und entsprechend Kollisionen in solche mit und ohne Gefährdung von Menschen einteilen.

Ein weiterer wichtiger Aspekt der Kollisionsgefahrerkennung ist die Berechnung der relativen Bewegungsgeschwindigkeit und -richtung des Staplers gegenüber anderen Objekten im Fahrweg. Die Betrachtung der relativen Bewegung reduziert falsche Kollisionsprognosen im Bereich kurzer Distanzen mit geringer Staplergeschwindigkeit und erhöht die Treffsicherheit bei hoher Geschwindigkeit in größeren Distanzen.

Die Bestimmung der notwendigen Leistungsfähigkeit der Personenerkennung gestaltet sich schwierig, da für die Kollisionsgefahrerkennung und die damit verbundene Kategorisierung Menschen nicht in jedem Bild erkannt werden müssen. Wird eine Person in einem Bild erkannt, kann der Aufenthaltsort an dieser Position für die darauffolgenden Bilder mit größer werdendem Abstand geschätzt werden. Vor allem wenn die Person mehrmals erkannt wird, kann deren Aufenthaltsort extrapoliert werden. Auf Basis von empirischen Voruntersuchungen wurde angenommen, dass eine Erkennungsrate

### 3.1 Anforderungen an ein Fahrerassistenzsystem für Gabelstapler

von 25 % für die Kategorisierung von Kollisionen ausreicht. Dieser Wert gilt für vollständig sichtbare oder zu maximal 33 % verdeckte Personen. Verdeckungen größer als 33 % treten üblicherweise hinter Regalen auf oder wenn Personen nicht mehr vollständig im Bild sind, wodurch die Gefahr gemindert wird. Es ist daher keine obligatorische Anforderung für das System solche Fälle zu erkennen.

Für den Anwender muss die Bedienung des Systems mit minimalem Aufwand verbunden sein. Während der Fahrt muss das System vollständig automatisch agieren, etwaige Bedienungen durch den Fahrer sind zur Vermeidung von Ablenkungen auszuschließen. Die Wartung des Systems, welche mitunter die Kalibrierung des Sensors einschließt, muss ebenfalls möglichst gering und durch Laien durchführbar sein. Demzufolge ist eine automatische Kalibrierung der Hardwarekomponenten obligatorisch. Die Veränderung der Einstellungen wird über Konfigurationsdateien vorgenommen, die nur vom jeweiligen Vorarbeiter geändert werden können.

Der Sensor muss eine Mindestreichweite von 6,25 m aufweisen, sodass eine Kollision nach Abbildung 3-3 bei 16 km/h rechtzeitig erkannt wird. Der vertikale und der horizontale Öffnungswinkel sollten möglichst groß sein, damit von einem Sensor ein möglichst großer Bereich abgedeckt werden kann. Für die Personenerkennung ist die Erfassung von Texturen durch den Sensor obligatorisch. Ebenso ist die Abbildung der Umwelt durch 3D-Daten erforderlich. Eine Fusion von mehreren Sensorarten oder die Verwendung staplerspezifischer Sensorik ist nicht vorgesehen. Die Anbindung des Sensors muss eine möglichst standardisierte Schnittstelle wie RJ-45 oder USB aufweisen.

Der Installations- und Wartungsaufwand muss möglichst gering gehalten werden.

Zusammenfassend sind die definierten Anforderungen inklusive der Priorisierung in folgender Tabelle aufgelistet:

*Tabelle 3-2: Auflistung der Anforderungen an das Kollisionsgefahrerkennungssystem.*

Anforderung	Ausprägung	Kategorie
<b>Funktionalitäten</b>		
<b>Kollisionsgefahrerkennung</b>	Vorhanden	M
<b>Personenerkennung</b>	Vorhanden	M
<b>Modus zur alleinigen Warnung vor Personenkollisionen</b>	Vorhanden	M

<b>Warnung des Fahrers</b>	Vorhanden	S
----------------------------	-----------	---

## Kollisionsgefahrerkennung

<b>Erkennungszeitpunkt</b>	Rechtzeitig unter Berücksichtigung des Bremswegs aus Abschnitt 3.1.2	M
----------------------------	--	---

<b>Rechenzeit</b>	Echtzeit in Bezug auf Sensordaten und unter Berücksichtigung des Bremswegs aus Abschnitt 3.1.2	S
-------------------	--	---

<b>Minimale Größe zu erkennender Objekte</b>	20 cm je Dimension	M
--	--------------------	---

<b>Unterscheidung Kollision mit Menschen oder Objekt</b>	Vorhanden	M
--	-----------	---

<b>Berücksichtigung der Bewegungsgeschwindigkeit und -richtung des Staplers und etwaiger Kollisionsobjekte</b>	Vorhanden	M
--	-----------	---

## Personenerkennung

<b>Erkennungsrate</b>	> 25 %	S
-----------------------	--------	---

<b>Erkennung aufrechter Personen</b>	Vorhanden unter Berücksichtigung der meisten Körpermaße aus Abschnitt 3.1.2 und aus jeglichen möglichen Perspektiven <sup>1</sup>	M
--------------------------------------	---	---

<b>Erkennung teils (&lt; 33 %) verdeckter aufrechter Personen</b>	Vorhanden	M
---	-----------	---

<b>Erkennung überwiegend verdeckter (&gt; 33 %) aufrechter Personen</b>	Vorhanden	C
---	-----------	---

<b>Erkennung von Personen in anderen Posen</b>	Vorhanden	C
--	-----------	---

## Bedienung

<b>Anpassung der Einstellungen (Kollisions-, Personenerkennung, Sensorposition)</b>	Einfach, z. B. über Konfigurationsdatei	M
---	---	---

<b>Einstellung der Sensorhöhe und des -winkels</b>	Automatisch	C
--	-------------	---

<b>Komplexität der Inbetriebnahme durch den Fahrer</b>	Sehr gering, keine Schulung notwendig	M
--	---------------------------------------	---

---

<sup>1</sup> Die möglichen Perspektiven sind vor allem durch die Sensorpositionierung abhängig.

<b>Sensor</b>		
<b>Reichweite</b>	Größer als der maximale Bremsweg nach Abschnitt 3.1.2 (bis 16 km/h)	M
<b>Öffnungswinkel</b>	Horizontal möglichst groß. Vertikal groß genug, damit Kollisionen beim Anfahren (< 4 km/h) und normaler Fahrt (8-16 km/h) erkannt werden können	C
<b>Frequenz</b>	>20 Hz	M
<b>Texturerfassung</b>	Vorhanden	M
<b>Tiefenbilderfassung</b>	Vorhanden	M
<b>Störeinfluss gegenüber üblichen Materialien und Umgebungsbedingungen in Lagern</b>	Gering	S
<b>Schnittstelle</b>	USB, RJ-45	M
<b>Fusion mehrerer Sensoren</b>	Nein	M
<b>Verwendung interner Sensoren</b>	Nein	M
<b>Sonstiges</b>		
<b>Wartungsaufwand</b>	Keiner, selbstkalibrierendes System	M
<b>Installationsaufwand</b>	Sehr gering, staplertypunabhängige Montage	M

## 3.2 Hardwarekonzept

Gabelstapler gibt es in zahlreichen Varianten (Gegengewichts-, Schubmast-, Gabelhochhubstapler etc.). Diese unterscheiden sich in der verwendeten Technik und in ihrer Form deutlich voneinander. Nicht nur bei den Staplervarianten gibt es große Unterschiede, sondern auch zwischen Modellen verschiedener Hersteller. Daher sollte das System aus möglichst wenigen, staplerunabhängigen Komponenten bestehen, welche einfach montiert werden können. Die Unabhängigkeit vom verwendeten Stapler impliziert zudem, dass keine interne Sensorik verwendet wird.

Die Hardware besteht nur aus zwei Komponenten: einer Recheneinheit, z. B. ein Notebook oder einem Embedded-System, und einem Sensor. Da nach den Anforderungen

das System unabhängig von der Positionierung des Sensors funktionieren muss, zeigt Abbildung 3-4 nur eine beispielhafte Anordnung der Komponenten.

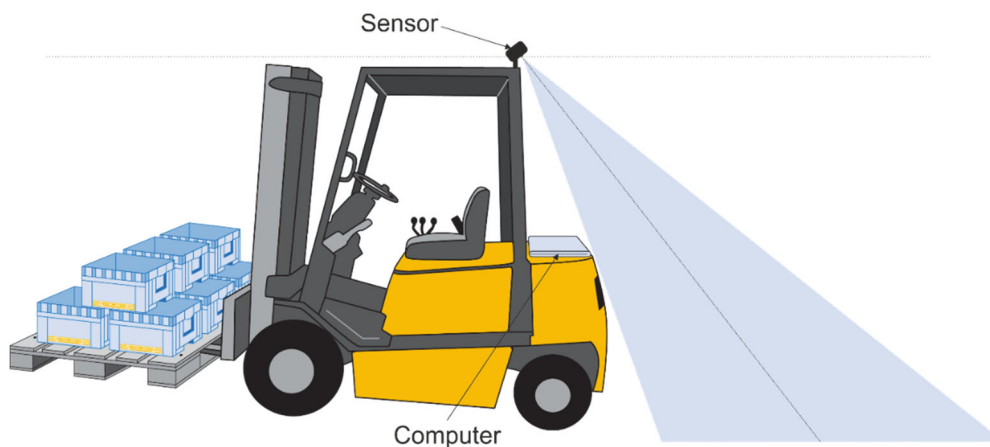


Abbildung 3-4: Hardwarekonzept für ein Kollisionsgefahrerkennungssystem.

Auf der Abbildung ist die Kamera nach hinten gerichtet, wodurch beim Gegengewichtstapler eine freie Sicht gewährleistet ist. Für eine Erkennung bei der Vorwärtsfahrt müsste die Kamera derart positioniert werden, dass die Sicht nicht durch Gabeln oder aufgeladenes Gut beeinträchtigt wird. Bei anderen Staplertypen wie z. B. dem Schubmaststapler (siehe Abschnitt 7.2.2) ist die Positionierung der Kamera in Richtung Vorwärtsfahrt einfacher, da die Gabeln senkrecht zur Fahrtrichtung sind. Je nach Staplertyp bedarf es eigener Entwicklungen um eine freie Sicht zu gewährleisten, dies soll aber im Rahmen dieser Arbeit nicht weiter erörtert werden.

### 3.3 Softwarekonzept

Zur Erkennung von Kollisionen ist grundlegend die Erkennung der Geschwindigkeit und der Richtung des Staplers und von Objekten im Fahrweg notwendig. Darüber hinaus müssen Personen im Fahrweg des Staplers erkannt werden. Zur Erreichung dieser Ziele wurde ein Konzept entwickelt, dessen Module in Abbildung 3-5 dargestellt sind:

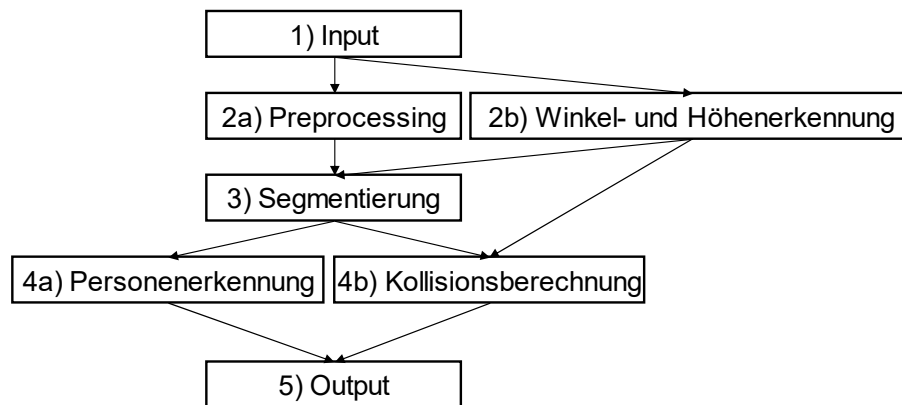


Abbildung 3-5: Module des Kollisionsgefahrerkennungssystems.

Der Empfang der Kameradaten (1) ist das grundlegende Element zur weiteren Bildverarbeitung. Die Daten müssen in Echtzeit von der Kamera empfangen und den weiteren Modulen bereitgestellt werden. Da in Abschnitt 2.5.2 keine 3D-Kamera als signifikant beste Lösung herausgearbeitet werden konnte, muss eine möglichst standardisierte Anbindung anderer Kameras möglich sein. Ungleich wichtiger ist die standardisierte, sensorunabhängige Ausgabe der Daten, sodass die folgenden Module ohne zusätzlichen Mehraufwand funktionieren.

Zu Auswertungszwecken können als Eingangsquelle auch gespeicherte Daten verwendet werden. Hierbei wird als Ausgabe ein identisches Datenformat genutzt, wie bei der Verwendung einer Kamera. Das heißt, Bilder können nicht nur sequenziell geladen, sondern auch als Stream zugeführt. Dadurch werden etwaige Auslassungen von Daten durch bspw. eine zu langsame Berechnung auch beim Laden gespeicherter Daten berücksichtigt.

Der erste Schritt nach der Datenerfassung ist die Vorverarbeitung (engl.: „Preprocessing“) (2a). In Abhängigkeit des folgenden Algorithmus werden die Daten auf diesen optimiert. Das kann unter anderem eine Konvertierung in ein anderes Farbmodell, eine Skalierung, eine Filterapplikation oder andere in Abschnitt 2.4.1 beschriebene Bildoperationen beinhalten.

Ein weitere Verarbeitung der Rohdaten findet im Modul „Winkel- und Höhenerkennung“ (2b) statt. Mithilfe der Erkennung des Bodens werden in diesem Modul der Winkel und die Höhe der Kamera berechnet. Diese zwei Parameter werden für die Segmentierung und die Kollisionsgefahrerkennung benötigt.

Im Segmentierungs-Modul (3) werden die Daten vorklassifiziert. Ein Beispiel hierfür ist die Entfernung des Bodens oder das Zusammenfassen von Bildbereichen zu Objekten (engl.: „Clustering“). Der Zweck der Segmentierung ist die Minimierung der Informati-



onen für die nachfolgenden Algorithmen, sofern in diesen nicht bereits eine Segmentierung implementiert ist. Folglich dient sie der Minimierung möglicher Falscherkennungen und des Rechenaufwands, indem unnötige Bilddaten entfernt werden.

Nach der Vorverarbeitung und -klassifizierung folgen die eigentlichen Bildanalysemodule: die Personen- und Kollisionsgefahrerkennung. Das Modul zur Personenerkennung (4a) liefert Bildbereiche, in welchen Personen mittels maschinellen Lernens erkannt werden. Dabei werden nur Personen als solche anerkannt, bei welchen der Konfidenzwert einen noch zu ermittelnden Grenzwert überschreitet. Die Höhe des Grenzwertes muss empirisch bestimmt werden, sodass alle Kollisionen mit Personen erkannt werden, aber möglichst wenig Falscherkennungen produziert werden. Parallel findet im Kollisionsgefahrerkennungsmodul (4b) die Berechnung der Bewegungsvektoren (relative Geschwindigkeit zum Stapler) jedes Pixels statt. Diese werden über die Kameraparameter, -höhe und -winkel in das Koordinatensystem des Staplers transformiert. Abschließend wird die Kollisionszeit für jedes Pixels berechnet. Einzelne Pixel, bei welchen eine Kollisionsgefahr erkannt wurde, werden zu zusammenhängenden Kollisionsbereichen zusammengefasst.

Das letzte Modul „Output“ (5) fasst die Ergebnisse aus den vorherigen Modulen zusammen. Sofern eine Kollisionsgefahr erkannt wurde, wird geprüft, ob sich in dem jeweiligen Bereich eine Person befand. Im Falle der Gefährdung einer Person wird ein höherer Grenzwert, sprich ein früherer Zeitpunkt, zur Auslösung der Warnung verwendet. Der Grenzwert zur Auslösung der Warnung ist die minimale Kollisionszeit. Liegt der Grenzwert z. B. bei fünf Sekunden, löst das System eine Warnung aus, sobald eine Kollision mit einem Objekt in unter fünf Sekunden bevorsteht. Damit nicht bei jedem Pixel eine Warnung fälschlicherweise bspw. durch Rauschen auftritt, muss der Kollisionsbereich eine konfigurierbare Mindestgröße aufweisen. Diese minimale Größe ist abhängig von der Entfernung des kollidierenden Objekts; dadurch wird vermieden, dass Objekte im Nahbereich, welche aufgrund der Nähe zum Sensor deutlich mehr Pixel beinhalten, schneller eine Warnung auslösen, als weiter entfernte Objekte, selbst wenn diese in der Realität gleich groß sind.

Sofern aufgezeichnete Daten verwendet werden, werden abschließend die Detektionsergebnisse (erkannte Personen und Kollisionsgefahren) abgespeichert. Die Speicherung dient zum Vergleich verschiedener Algorithmen und wird im operativen Betrieb nicht benötigt.

## 3.4 Fazit

In diesem Kapitel stand die Entwicklung eines Systems im Fokus, welches die Vermeidung von Unfällen bei Gabelstaplern zum Ziel hat und möglichst wenig falsche Warnungen erzeugt. Der erste Schritt war die Ableitung von Anforderungen; zu diesem Zweck wurden Unfalldaten analysiert sowie relevante Richtlinien und Normen identifiziert. Bei den Unfalldaten zeigte sich, dass bei Gegengewichtsstaplern besonders die Seite links zum Fahrer einen gefährlichen Sektor darstellt, vor allem aber bei der Rückwärtsfahrt. Bezüglich der Normen waren einerseits die Ausprägungen der minimalen und maximalen Maße von Menschen für das System von Bedeutung, andererseits gibt die Norm indirekt über den maximal zulässigen Bremsweg von Flurförderzeugen den letzten Zeitpunkt vor, bis zu welchem ein Kollisionsgefahrerkennungssystem die Kollision spätestens vorhergesagt haben muss.

Darüber hinaus wurde der Einsatzzweck des Systems auf die Assistenz für Gabelstaplerfahrer beschränkt.

Aus den Anforderungen und dem Einsatzzweck wurden schließlich ein Hardware- und Softwarekonzept entworfen. Das Hardwarekonzept sieht exemplarisch eine ToF-Kamera auf dem Dach des Staplers vor, welche entgegen der Gabeln positioniert ist. Die Hardwarekomponenten bestehen aus einer 3D-Kamera und einem Computer.

Das Softwarekonzept beinhaltet die Umsetzung eines Systems, welches anhand von 3D-Kameradaten Kollisionsgefahren vorhersagen und kategorisieren kann. Die Kategorisierung erfolgt in Kollisionsgefahren mit Hindernissen und mit Menschen. Die Unterscheidung der Kollisionsgefahr soll ein zweistufiges Warnsystem ermöglichen, bei dem unterschiedliche Schwellwerte zur Auslösung genutzt werden können. Damit dies gelingen kann, wurden folgende Module des Systems als notwendig erachtet:

1. Eingang: Mögliche Anbindung verschiedener Kameras und Standardisierung der Ausgangsdaten
2. Vorverarbeitung: Optimierung der standardisierten Kameradaten in Abhängigkeit des nachfolgenden Algorithmus (Personen-/Kollisionsgefahrerkennung)
3. Segmentierung: Entfernung unnötiger Informationen aus den Kameradaten
4. Kamerawinkel-/höhenberechnung (optional): Automatische Erkennung der Montageposition der Kamera und deren Ausrichtung
5. Personenerkennung: Zur Klassifikation erkannter Kollisionsgefahren
6. Kollisionsgefahrerkennung: Berechnung von Kollisionsgefahren
7. Ausgabe: Zusammenfassung der Ergebnisse aus Personen- und Kollisionsgefahrerkennung und gegebenenfalls Erzeugung einer Warnung



## 4 Implementierung

---

In Kapitel 3 wurde ein Konzept erarbeitet, welches es ermöglicht, Kollisionsgefahren bei Gabelstaplern zu erkennen und davor zu warnen. Die Kollisionen werden in zwei Kategorien aufgeteilt: Kollisionen mit Objekten und Kollisionen mit Personen. Je nach Kategorie wird eine andere Form der Warnung möglich. Im Folgenden wird beschrieben, wie das Konzept hardwaretechnisch und softwaretechnisch umgesetzt wurde.

### 4.1 Hardwarespezifikation

Die Komponenten, welche für das Kollisionsgefahrerkennungssystem verwendet werden, wurden bereits in dem Hardwarekonzept in Abschnitt 3.2 erläutert. Eine Spezifizierung der notwendigen Hardware in Hinsicht auf die verarbeitende Recheneinheit und den Sensor folgt in diesem Abschnitt.

Zur Auswertung der Sensordaten stehen mehrere Möglichkeiten zur Verfügung. Die Verwendung des staplereigenen Industrie-PCs kommt aufgrund zweierlei Tatsachen nicht in Betracht: Erstens würde das System auf einzelne Gabelstaplermodelle eingeschränkt werden, zweitens sind keine Industrie-PCs mit dedizierter Grafikkarte bekannt, welche für bildverarbeitende Methoden einen erheblichen Leistungsvorteil bieten. Aufgrund der angestrebten Personenerkennung wird eine Grafikkarte behilflich oder gegebenenfalls notwendig sein. Aufgrund von praktischen Gründen und der Flexibilität wurde ein handelsübliches Notebook, welches eine dedizierte Grafikkarte besitzt, verwendet<sup>2</sup>. Im Detail wurde das folgende Notebook, sofern nicht anders angegeben, im Zuge der Dissertation benutzt:

- Name: Tuxedo Book XC1707
- CPU: i7-6820HJK @ 2,70GHz\*8
- Grafikkarte: NVIDIA GTX 1070
- Arbeitsspeicher: 16 GB DDR4-2400
- Betriebssystem: Ubuntu 16.04 LTS

---

<sup>2</sup> Eingebettete Systeme mit ausreichend starker, dedizierter Grafikkarte bzw. -chip sind zum Zeitpunkt der Dissertationsverfassung von NVIDIA erhältlich. Diese Modelle gehören zur Jetson-Produktreihe. Siehe: <https://www.nvidia.com/de-de/autonomous-machines/embedded-systems/>

### Sensorposition und -ausrichtung

Die Auswahl der Montageposition des Sensors am Gabelstapler ist von äußeren Faktoren wie dem Gabelstaplertyp oder dem Einsatzgebiet abhängig. Eine generelle Aussage, wo der Sensor befestigt werden sollte, ist daher nicht möglich. Die Position des Sensors sollte prinzipiell ein möglichst freies Sichtfeld aufweisen. Darüber hinaus ist nach den analysierten Unfalldaten eine Ausrichtung entgegen den Gabeln sinnvoll. Da der ToF-Sensor bei direkter Sonneneinstrahlung gestört werden kann, ist die Montage auf dem Dach des Staplers am sinnvollsten. Dadurch wird vermieden, dass der Sensor z. B. auf ein Fenster zeigt durch das die Sonne direkt auf die Kamera strahlt. Eine mögliche Befestigung am Stapler „EFG 220“ von Jungheinrich zeigt Abbildung 4-1:



Abbildung 4-1: Beispielhafte Befestigung des Sensors an einem Gegengewichtsstapler.

Die Kinect wurde mittels einer Halterung ohne zusätzliche Bearbeitung am Stapler in einer Höhe von 3,3 m montiert. Die konstruierte Halterung dient einerseits zur einfachen Befestigung und andererseits zur Fixierung des Kamerawinkels. Letzteres ist durch die Scharniere der Kinect nicht ohne weiteres möglich.

Die optimale Ausrichtung der Kamera hängt von den jeweiligen Sensorparametern und vom Anwendungsfall ab. Die Hauptfaktoren sind hierbei der Öffnungswinkel und die Reichweite. Für die Ausrichtung wurde in Berücksichtigung der maximalen Reichweite folgende Vorgehensweise erarbeitet:

1. Bestimmung der maximal notwendigen Reichweite im Weltkoordinatensystem  
– nicht die Sensorreichweite

2. Bestimmung des Winkels unter der Annahme, dass eine 95-Perzentil große Person zu 95 % im Sichtfeld der Kamera ist. Damit soll gewährleistet sein, dass die Personenerkennung den Menschen als solchen erkennt
3. Überprüfung, ob eine 5-Perzentil große Person in geringem Abstand (0,3 m) zum Stapler noch mindestens zu 5 % sichtbar ist. Die 5 % der Körperhöhe müssen ausreichen, damit eine objektunabhängige Kollisionsgefahr erkannt wird.

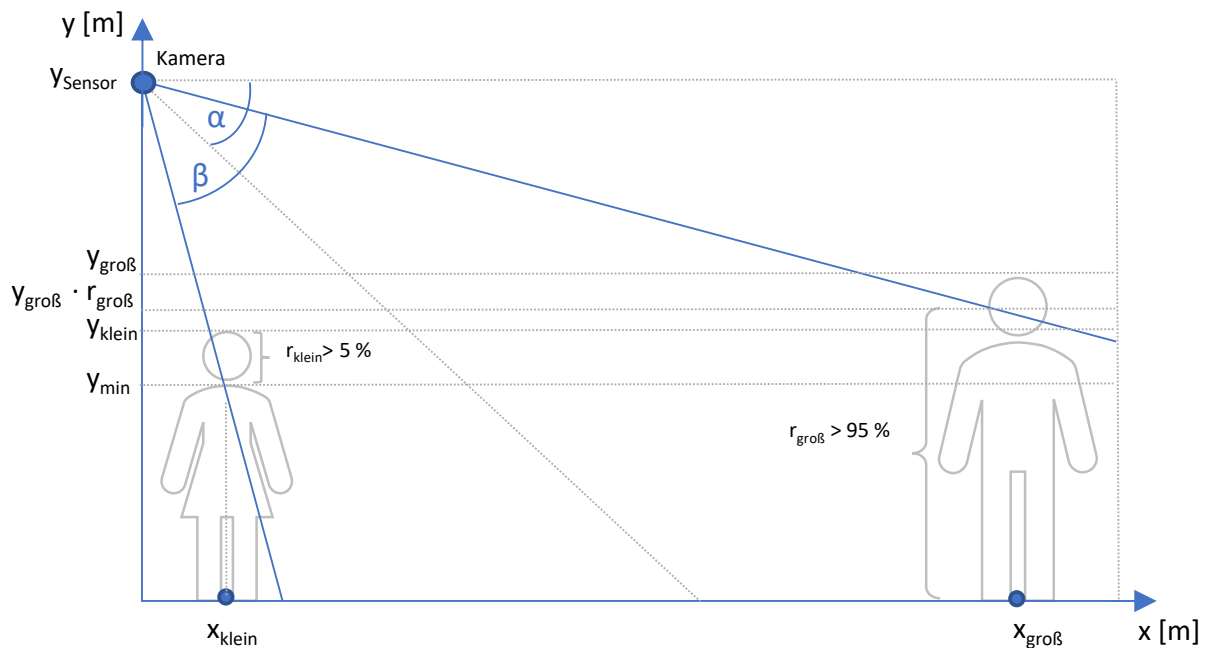


Abbildung 4-2: Schema zur Kameraausrichtung.

Der Kamerawinkel  $\alpha$  (Horizontale zur Kameraachse) ergibt sich aus trigonometrischen Beziehungen und der maximalen Reichweite ( $x_{\text{groß}}$ ) wie folgt:

$$\tan\left(\left(90^\circ - \alpha\right) + \frac{\beta}{2}\right) = \frac{x_{\text{groß}}}{y_{\text{Sensor}} - y_{\text{groß}} \cdot r_{\text{groß}}} \quad (1.1)$$

$$\alpha = 90^\circ - \tan^{-1} \frac{x_{\text{groß}}}{y_{\text{Sensor}} - y_{\text{groß}} \cdot r_{\text{groß}}} + \frac{\beta}{2} \quad (1.2)$$

mit

$\alpha$ : Kamerawinkel (Kameraachse zur Horizontalen)

$\beta$ : vertikaler Öffnungswinkel der Kamera

$x_{\text{groß}}$ : Abstand der großen Person

$y_{\text{groß}}$ : Höhe der großen Person

$r_{\text{groß}}$ : Anteil der sichtbaren Höhe einer großen Person

## 4 Implementierung

Anhand des Kamerawinkels  $\alpha$  wird anschließend die minimale Höhe  $y_{\min}$ , die im Abstand von  $x_{\text{klein}}$  noch von der Kamera erfasst werden, kann berechnet:

$$\tan\left(90^\circ - \left(a + \frac{\beta}{2}\right)\right) = \frac{x_{\text{klein}}}{y_{\text{Sensor}} - y_{\min}} \quad (2.1)$$

$$y_{\min} = y_{\text{Sensor}} - \frac{x_{\text{klein}}}{\tan\left(90^\circ - \left(a + \frac{\beta}{2}\right)\right)} \quad (2.2)$$

mit

$x_{\text{klein}}$ : Abstand der kleinen Person

$y_{\min}$ : untere Grenze des Kamerasichtfelds im Abstand  $x_{\text{klein}}$

Die Höhe, die von der Kamera eines 5-Perzentil großen Menschen vom Kopf Richtung Fuß erfasst wird, berechnet sich aus der Differenz von  $y_{\text{klein}}$  und  $y_{\min}$ . Daraus lässt sich wie folgt berechnen, wieviel Prozent des Körpers (vom Kopf gemessen)  $r_{\text{klein}}$  im Sichtfeld der Kamera ist:

$$r_{\text{klein}} = \frac{y_{\text{klein}} - y_{\min}}{y_{\text{klein}}} = \frac{y_{\text{klein}} - \left(y_{\text{Sensor}} - \frac{x_{\text{klein}}}{\tan\left(90^\circ - \left(a + \frac{\beta}{2}\right)\right)}\right)}{y_{\text{klein}}} \quad (3)$$

Die Vorgehensweise wurde anhand des Aufbaus aus Abbildung 4-2 durchgeführt (vgl. Tabelle 4-1). Als Werte für  $y_{\text{klein}}$  und  $y_{\text{groß}}$  wurden die in Tabelle 3-1 beschriebenen 5-Perzentil bzw. 95-Perzentilkörperhöhen von 1,54 m bzw. 1,86 m verwendet. Der minimale Abstand von 0,3 m wurde um den Betrag von 0,05 m erhöht, da der Sensor nicht mit dem Heck des Staplers abschließend montiert wurde. Die Montagehöhe des Sensors  $y_{\text{Sensor}}$  ist 2,4 m. Der Öffnungswinkel  $\beta$  beträgt nach den technischen Spezifikationen der „Kinect v2“  $60^\circ$ .

**Tabelle 4-1:** Beispielrechnung von  $r_{\text{klein}}$  mit  $\beta = 60^\circ$ ,  $x_{\text{klein}} = 0,35$  m,  $y_{\text{klein}} = 1,54$  m,  $y_{\text{groß}} = 1,86$  m,  $r_{\text{groß}} = 0,95$ ,  $y_{\text{Sensor}} = 2,4$  m.

Geschwindigkeit [km/h]	2	4	6	8	10
$x_{\text{groß}}$ [m]	0,52	1,21	2,11	3,17	4,45
Kamerawinkel $\alpha$ [°]	81	58	47	41	38
$y_{\min}$ [m]	0	0	0,90	1,36	1,52
$r_{\text{klein}}$ [%]	100	100	42	12	1

Je nach Geschwindigkeit des Staplers ergibt sich ein anderer maximaler Kamerawinkel  $\alpha$ . Mit zunehmender Geschwindigkeit muss der Winkel aufgrund des höheren Bremswegs ( $x_{\text{groß}}$ ) kleiner werden, damit die Sichtweite der Kamera ausreichend groß

ist. Ab 10 km/h ist wegen des geringen Kamerawinkels die 5-Perzentil große Person im Nahbereich nur noch zu 1 % ( $r_{\text{klein}}$ ) des Körpers in Bezug auf die Höhe im Sichtfeld der Kamera. Aufgrund dessen wird die Geschwindigkeit in den folgenden Tests auf 8 km/h beschränkt, da in dem Fall eine 5-Perzentil große Person bei dem maximal notwendigen Bremsweg noch zu 14 % im Kamerabild ist.

Die Grenze zur Sichtbarkeit von kleineren Personen ist stark vom Öffnungswinkel des verwendeten Sensors abhängig. Es kann davon ausgegangen werden, dass bei 3D-Kameras weitere Fortschritte bezüglich des Sensors erzielt werden. Bereits eine Erhöhung des Öffnungswinkels um  $10^\circ$  (siehe Tabelle 4-2) würde eine Abdeckung des gesamten Geschwindigkeitsbereichs von Staplern bewirken.

*Tabelle 4-2:* Beispielrechnung von  $r_{\text{klein}}$  mit einem alternativen Sensor mit  $\beta = 70^\circ$ ,  $x_{\text{klein}} = 0,35$  m,  $y_{\text{klein}} = 1,54$  m,  $y_{\text{groß}} = 1,86$  m,  $r_{\text{groß}} = 0,95$ ,  $y_{\text{sensor}} = 2,4$  m.

Geschwindigkeit [km/h]	8	12	16	20	24
$x_{\text{groß}}$ [m]	3,17	5,90	9,38	13,62	18,60
Kamerawinkel $\alpha$ [°]	46	41	39	38	37
$y_{\text{min}}$ [m]	0,09	0,98	1,19	1,28	1,32
$r_{\text{klein}}$ [%]	94	37	23	17	14

## 4.2 Softwareentwicklung

Für die Entwicklung des Systems wurde die Programmiersprache C++ verwendet. Eine der Hauptgründe ist die hohe Performanz, welche aufgrund der geforderten Echtzeitfähigkeit notwendig ist. Für die Entwicklung wurden mitunter Bibliotheken verwendet. Die wichtigste Bibliothek ist die „Open Source Computer Vision Library“ (OpenCV) in der Version 3. Diese beinhaltet eine Vielzahl an einfachen Methoden der Bildverarbeitung wie z. B. die Skalierung eines Bildes bis hin zu komplexeren Algorithmen wie dem Tracking. Im Falle der meisten Algorithmen gibt es neben der CPU-basierten Methode auch eine analoge Methode zur Berechnung auf der Grafikkarte über CUDA. Letzteres ist eine Bibliothek von NVIDIA für effiziente Berechnungen über die Grafikkarte. CUDA (Version 8) wurde im Zuge der Softwareentwicklung ebenso für eigene Methoden verwendet, welche nicht in OpenCV enthalten waren. Daher ist das entwickelte System nur in Verbindung mit Grafikkarten oder eingebetteten Systemen verwendbar, welche einen NVIDIA-Grafikchip besitzen.

### Programmablauf

Grundsätzlich folgt die Programmsequenz dem Softwarekonzept aus Abschnitt 3.3. Zur effizienten Berechnung und Minderung der Rechenzeit wurden einzelne Klassen,



die das Bild pixelweise analysieren, auf die Grafikkarte (GPU) ausgelagert. Grafikkarten haben vor allem in der parallelen Berechnung einfacher Operationen einen enormen Geschwindigkeitsvorteil gegenüber CPUs. Die verwendete Karte „GTX 1070“ hat 1920 Shader-Einheiten. Dementsprechend können 1920 Operationen in einem Takt berechnet werden, bei einem 8-Kern-Prozessor wären es 8. Im Bereich der Bildverarbeitung werden somit auf der GPU 1920 Pixel in einem Takt bearbeitet. Das funktioniert nur, wenn die Bearbeitung unabhängig von den Ergebnissen anderer Pixeloperationen ist.

Die Aufteilung der Klassen auf CPU und GPU ist in Abbildung 4-3 dargestellt:

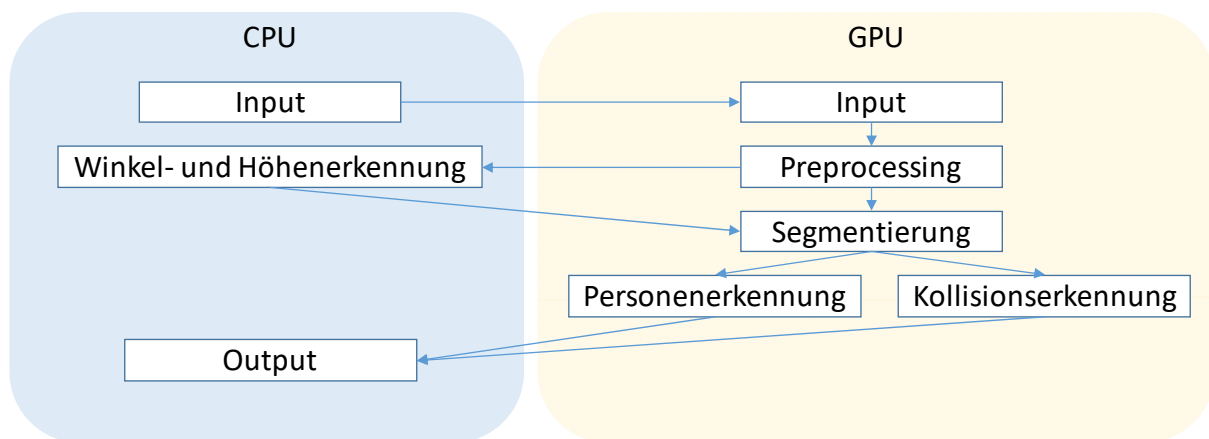


Abbildung 4-3: Aufteilung der Berechnung auf CPU & GPU.

Der Empfang der Kameradaten ist aufgrund der verwendeten Treiber nur auf der CPU möglich. Nach dem Empfang werden sie in den Arbeitsspeicher der Grafikkarte geladen und dort vorverarbeitet. Die vorverarbeiteten Daten werden wiederum für die Höhen- und Winkelerkennung in den Arbeitsspeicher des Computers kopiert. Die Berechnung ist auf der CPU schneller, da sie nicht aus Matrixoperationen, sondern sequenziellen und voneinander abhängigen Anweisungen besteht. Ansonsten wird nur noch die Ausgabe auf der CPU ausgeführt, da ähnlich wie bei der Höhen- und Winkelberechnung nur sequenzielle Operationen durchgeführt werden.

### Beschreibung der einzelnen Klassen

Die Verarbeitung der rohen Kameradaten bis hin zur fertigen Analyse des Bildes erfordert viele Zwischenschritte. Die Bearbeitung und Analyse aller zur Verfügung stehenden Daten werden im Folgenden in UML-Klassendiagrammen dargestellt und erläutert.

### 4.2.1 Eingang (engl.: „input“)

Die Input-Klasse stellt unabhängig von der gewählten Eingangsquelle je nach Verfügbarkeit Farb-, Tiefen-, und Intensitätsbilder als Stream oder zum Abruf im Arbeitsspeicher der Grafikkarte und/oder des Computers bereit (siehe Abbildung 4-4).

Als Eingangsquelle dient primär die ToF-Kamera „Kinect v2“. Als Treiber wird die Open-Source-Bibliothek „libfreenect2“ benutzt, da ein offizielles „Software Development Kit“ (SDK) für Linux nicht existiert. Die Bibliothek bietet neben dem reinen Empfang der Daten auch eine Methode zur Kalibrierung des Farbbilds auf die Daten des Infrarotsensors (Intensitäts- und Tiefenbild). Eine Anbindung an die Stereokamera „ZED“ ([Ste-2018]) wurde ebenfalls implementiert, allerdings nur zur Überprüfung der generellen Funktionsfähigkeit verwendet.

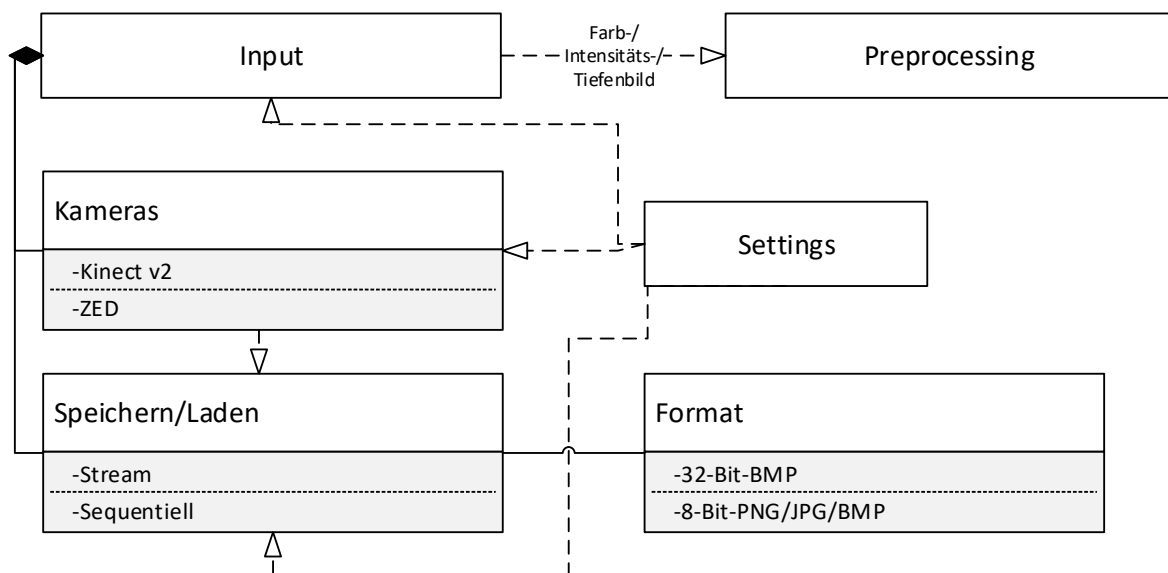


Abbildung 4-4: Klassendiagramm für Input.

Neben dem Abruf der Kameradaten lädt die Input-Klasse zuvor gespeicherte Daten ab. Grundsätzlich wurde bei Kameraaufnahmen jedes Bild als Bilddatei mittels der OpenCV-Funktion *imwrite* als Bitmap (.bmp) abgespeichert. Es wurde bewusst kein Video erstellt oder die Daten binär geschrieben, sodass die Daten mit Standard-Software möglichst einfach gesichtet und geschnitten werden können. Das Bitmap-Format wurde verwendet, da es keine Komprimierung beinhaltet und somit die Daten unverändert abspeichert. Alternativ können die Daten in komprimierter Form (JPG, PNG) abgespeichert und geladen werden. Die Klasse stellt die Bilder entweder auf Abruf (Pull) oder als Stream (Push) bereit: Erster Ansatz dient zum Leistungsvergleich unterschiedlicher Algorithmen, sodass die Analyse des Bilds durch alle nachfolgenden Algorithmen gewährleistet ist. Die Möglichkeit zum Streaming emuliert das Verhalten einer realen Kamera, sodass bei einer zu langen Berechnungszeit des

Gesamtsystems zwangsweise Bilder ausgelassen werden würden. Dementsprechend können die nachfolgenden Algorithmen z. B. nur jedes dritte Bild analysieren.

Ein Sonderfall ergibt sich bei der Speicherung des Intensitäts- und Tiefenbilds. Diese werden im Falle der „Kinect v2“ in 16-Bit geliefert. Bitmaps können allerdings nur 8-Bit-Werte beinhalten. Daher wurden die 16-Bit in zwei 8-Bit-Werte getrennt und jeweils in den Rot- bzw. Grün-Kanal eines Pixels geschrieben. Beim Laden wird der Vorgang umgekehrt durchgeführt.

Die Settings-Klasse ist in der Abbildung ebenfalls dargestellt. Sie beinhaltet alle Einstellungen des Systems. Außerhalb dieser Klasse können Einstellungen wie Kamera-parameter, Vorverarbeitungsschritte etc. nicht verändert werden. Grundsätzlich enthält jede Klasse des Systems eine Schnittstelle zur Settings-Klasse. Aus Gründen der Übersichtlichkeit wird in den nachfolgenden Klassendiagrammen diese Klasse nicht dargestellt.

### 4.2.2 Vorverarbeitung (engl.: „preprocessing“)

Das Preprocessing dient zur Vorbereitung der Daten für die nachfolgenden Algorithmen. Daher werden die Methoden zum Teil mehrfach angewandt, da z. B. die Personenerkennung andere Eingangsdaten erhält als die Kollisionsgefahrerkennung. In Abbildung 4-5 sind zwei Klassen dargestellt, welche unterschiedliche Vorverarbeitungsmethoden enthalten. Die Aufteilung hat den Grund, dass die zugrundeliegenden Methoden der „ImagePreprocessor“-Klasse in CUDA programmiert wurden. Diese wurden selber entwickelt, wohingegen die Methoden in der „Preprocessing“-Klasse auf Methoden von OpenCV zurückgreifen.

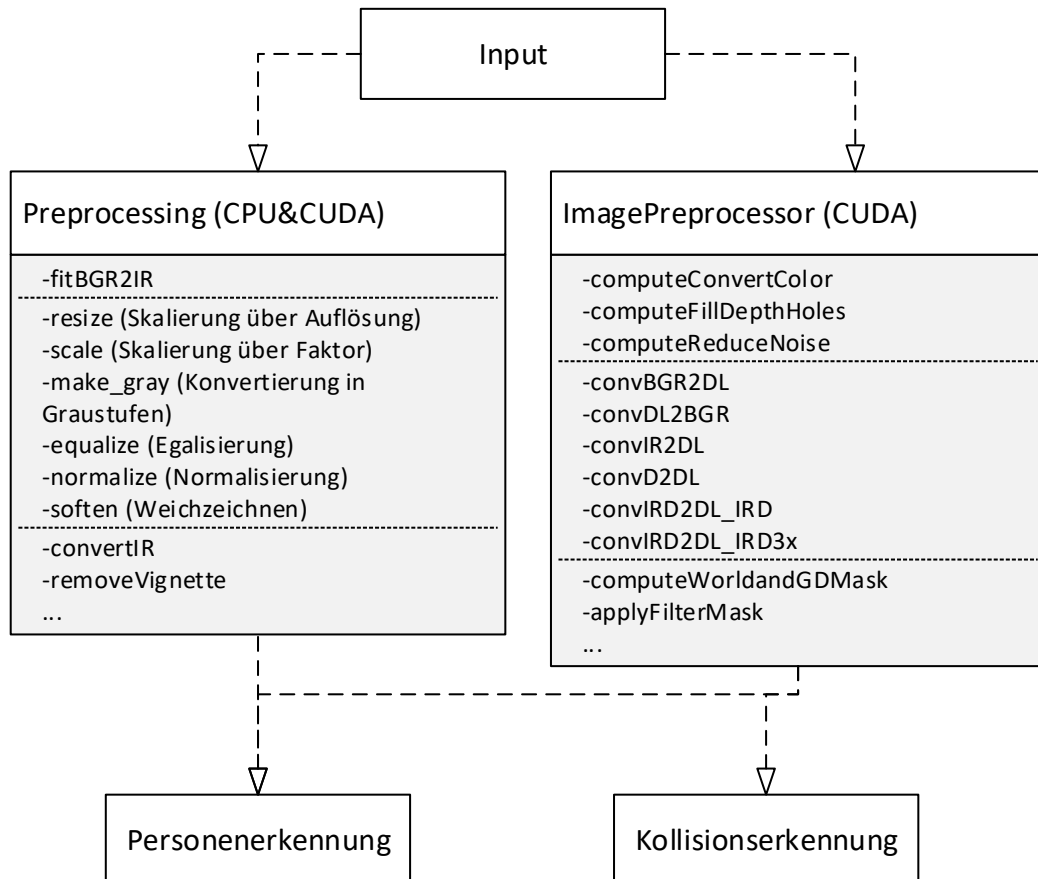


Abbildung 4-5: Klassendiagramm für Preprocessing.

Die Methoden *resize* bis *soften* sind Aliase für Methoden von OpenCV und wurden nur zur Vereinfachung von CPU- und GPU-basierten Operationen implementiert. Aufgrund dessen werden diese Operationen nicht weiter erläutert.

Die Methode *fitBGR2IR* dient zur Kalibrierung des Farbbilds auf das Intensitäts- oder Tiefenbild der Kinect. Wie eingangs erwähnt, existiert in der „libfreenect2“-Bibliothek eine Kalibrierungsmethode, allerdings weist diese den Nachteil auf, dass im sogenannten „registrierten Farbbild“ alle Pixel fehlen, bei welchen kein Tiefenwert vorhanden ist (siehe Abbildung 4-6). Aus diesem Grund wurde eine eigene Methode zur Kalibrierung implementiert. Diese liefert ein Farbbild, welches nur sehr geringfügig von dem Bild des Infrarotsensors abweicht. Die Abweichung ergibt sich aus den unterschiedlichen Öffnungswinkeln, Linsenparametern und der Position der zwei Sensoren. Die schwarzen Balken auf dem rechten Bild werden durch die unterschiedlichen Öffnungswinkel hervorgerufen.

## 4 Implementierung

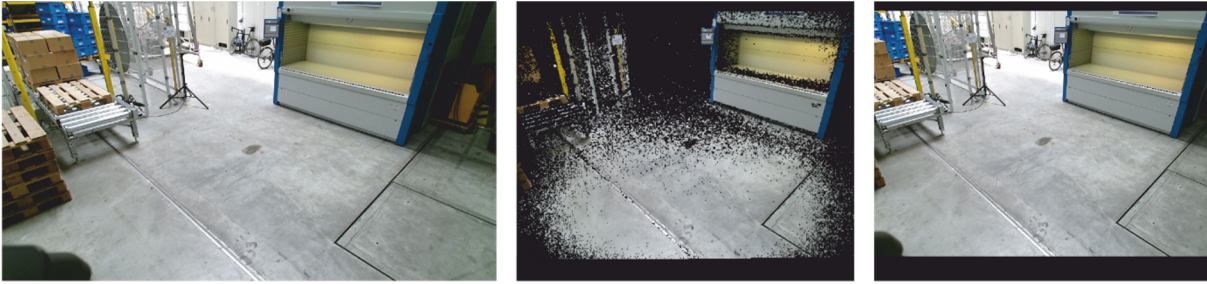


Abbildung 4-6: Vergleich der Farbbilder von links nach rechts: Originalbild, registriertes Farbbild (lib-freeenect2), kalibriertes Farbbild mit eigener Methode.

Die Methoden *convertIR* und *removeVignette* dienen zur Umwandlung des Intensitätsbildes in ein graustufenähnliches Bild. Die Umwandlung ist zur Nutzung in nachfolgenden Algorithmen notwendig, da diese auf konventionelle Farbbilder (8-Bit RGB) ausgelegt sind. Zwei hauptsächliche Unterschiede von Intensitäts- und Farbbildern sind die Beleuchtung und die Farbtiefe von 16-Bit. Letzteres führt zu Werten von 1 bis 65535, während normale Farbbilder Werte von 1 bis 255 für jedes Pixel beinhalten. Die einfachste Methode zur Anpassung der Farbtiefe ist die Division aller Pixelwerte durch 256. Das Problem der linearen Konvertierung zeigt das linke Bild in Abbildung 4-7. Wenn alle Pixelwerte durch 256 dividiert werden, verbleiben hauptsächlich sehr dunkle und helle Pixel. Wird das Bild auf den Wertebereich von 1 bis 255 normalisiert, ist das Bild zwar deutlich besser, die äußeren Bereiche des Bildes sind aber unterbeleuchtet. Der Grund hierfür sind die LED-IR-Emitter, welche hauptsächlich das Zentrum beleuchten.



Abbildung 4-7: Umwandlung von Intensitätsbildern in den RGB-8-Bit-Bereich durch lineare Skalierung. Links: Linear konvertiertes Bild, rechts: Normalisiertes, linear konvertiertes Bild.

Zur Behebung der ungleichen Belichtung wurde eine Methode zur nichtlinearen Konvertierung entwickelt. Dafür wurden sechs Wertebereiche definiert, welche unterschiedlich gewichtet werden. Die endgültige Helligkeit ergibt sich aus:

$$x_8 = \begin{cases} 255 \cdot \left( a_{factor}^{i-1}(x_{16}) + \left( \frac{x_{16} - a_{max}^{i-1}(x_{16})}{a_{max}^i(x_{16}) - a_{max}^{i-1}(x_{16})} \right) \cdot (a_{factor}^i(x_{16}) - a_{factor}^{i-1}(x_{16})) \right); & x_{16} < 9500 \\ 255; & \text{sonst} \end{cases}$$

(4)

mit

 $x_8$ : resultierender Pixelwerte (8-Bit) $x_{16}$ : Ausgangswert (16-Bit) $a_{max}^i$ : obere Grenze des i-ten Wertebereichs $a_{factor}^i$ : Prozentualer, kumulierter Anteil des i-ten Wertebereichs vom gesamten Wertebereich (1 bis 255).

Die Werte für die Parameter  $a_{max}^i$  und  $a_{factor}^i$  wurden empirisch ermittelt, sodass das Bild möglichst einem normalen Graustufenbild gleicht.

Beispielrechnung:

$$\begin{aligned} x_{16} &= 1000 \\ &\rightarrow i = 4 \text{ nach Tabelle 4-3 (links)} \\ x_{16} &= 255 \cdot \left( 0,35 + \left( \frac{1000 - 500}{1320 - 500} \right) \cdot (0,55 - 0,35) \right) = 120,34 \\ &\rightarrow \widetilde{x}_{16} = 120 \end{aligned}$$

Tabelle 4-3: Parameter zur nichtlinearen Konvertierung eines 16-Bit-Intensitätswerts in einen 8-Bit-Farbwert.

$i(x_{16})$	$i$	$a_{max}^i$	$a_{factor}^i$
$0 \leq x_{16} < 50$	0	0	0
$175 \leq x_{16} < 175$	1	50	0,11
$500 \leq x_{16} < 500$	2	175	0,26
$500 \leq x_{16} < 1320$	3	500	0,35
$1320 \leq x_{16} < 9500$	4	1320	0,55
	5	9500	1,0

Das Ergebnis der nichtlinearen Umwandlung des in Abbildung 4-7 dargestellten Originalbildes zeigt Abbildung 4-8 (links). Das Bild ist gleichmäßiger belichtet, aber die Abschattung in den Rändern ist noch deutlich zu erkennen. Die Methode *removeVignette* erzeugt eine Maske mit einer Gaußfunktion, ausgehend vom Zentrum des Bildes. Die Multiplikation der Maske mit dem konvertierten Bild führt zum rechts dargestellten Bild.

Hier ist zu erkennen, dass die Helligkeit dem restlichen Bild durch die Maske angeglichen werden kann. Das starke Rauschen an den Ecken ist bereits vorher vorhanden, es wird durch die Operation nur sichtbar.



Abbildung 4-8: Links: Ergebnis der nichtlinearen Konvertierungsmethode. Rechts: Zusätzliche Entfernung der Vignetten-Abschattung.

Die Methoden der *ImagePreprocessor*-Klasse benutzen keine Methoden der OpenCV-Bibliothek. Diese bilden Bildoperationen ab, welche in OpenCV für die Grafikkarte fehlen oder grundsätzlich nicht vorhanden sind. Hinter allen Methoden dieser Klasse stehen CUDA-programmierte „Shader“. Diese stellen pixelbasierte Operationen auf der Grafikkarte dar. Üblicherweise wird nur die Bearbeitung eines Pixels in Shadern beschrieben. Das erste Beispiel hierfür ist *computeConvertColor*. Diese hat die gleiche Funktionsweise wie *convertIR*, verarbeitet die Daten aber auf der Grafikkarte, wodurch die Berechnung schneller durchgeführt wird.

Fehlende Daten gibt es bei der ToF-Kamera grundsätzlich nur im Tiefenbild, z. B. wenn Objekte zu weit entfernt sind oder die emittierten Strahlen nicht in die Richtung der Kamera zurück reflektiert werden. Die fehlenden Daten lassen sich nur teilweise per Interpolation schätzen. Dies wird in der Methode *computeFillDepthHoles* mit einem Kernel von  $[3 \times 3]$  Pixeln durchgeführt. In dem Kernel müssen jeweils mindestens zwei Pixel im Abstand von 1 und 2 gültige Werte aufweisen. Ist diese Bedingung erfüllt, wird der fehlende Pixelwert durch den Median der umliegenden Pixel ersetzt.

Die Auswirkung der Methode in Abbildung 4-9 veranschaulicht:



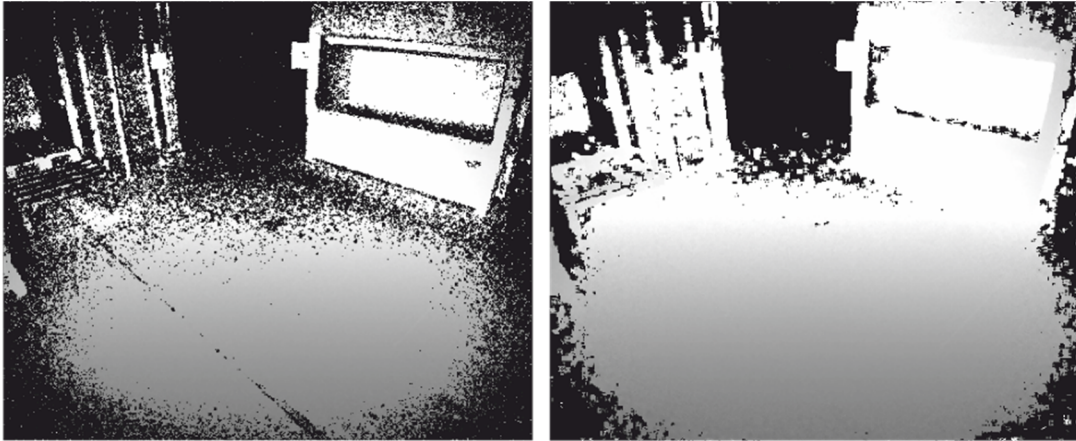


Abbildung 4-9: Interpolation fehlender Daten. Links: Originaltiefenbild, rechts: interpoliertes Tiefenbild.

Eine weitere Methode zur Verbesserung des Bildes ist das Weichzeichnen. Einfache Algorithmen zum Weichzeichnen sind bereits in OpenCV enthalten. Wie in Abschnitt 2.4.1 beschrieben führt Weichzeichnen zu Informationsverlust. Die Extraktion wichtiger Merkmale wie Ecken und Kanten ist bei einem weichgezeichneten Bild deutlich schwieriger. Daher wurde ein bilateraler Filter nach [Sny-2017, S. 93ff.] in CUDA namens *computeReduceNoise* implementiert, welcher zwar die Texturen weichzeichnet, die Kanten aber beibehält. Im Beispiel in Abbildung 4-10 wurden aus Illustrationsgründen mehrfach sehr starke und große Kernel auf ein egalisiertes Bild angewandt. Bei normaler Anwendung des Filters ist die Wirkung ungleich schwächer.



Abbildung 4-10: Anwendung des implementierten bilateralen Weichzeichnen-Filters auf egalisiertes Bild. Links: Originalbild, rechts: stark bilateral, weichgezeichnetes Bild.

Die Methoden *conv...2DL* wandeln Farb-, Intensitäts- und Tiefenbilder in ein Format um, welches für das verwendete DL-Framework „Darknet“ verwendet wird. Im einfachsten Fall wird ein Farbbild in dessen Farbkanäle unterteilt und aus den daraus entstehenden Bildern (Rot, Grün, Blau) ein Bild mit der dreifachen Höhe erzeugt. Das Beispiel aus Abbildung 4-11 veranschaulicht das Ergebnis von *convBGR2DL*, zeigt



## 4 Implementierung

---

aber nicht das tatsächlich resultierende Bild. Das wirkliche Resultat ist ein auf 1-Kanal reduziertes Bild, welches wie ein Graustufenbild nur Helligkeitswerte besitzt.

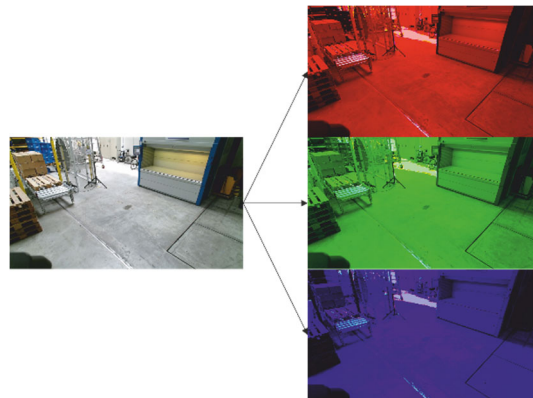


Abbildung 4-11: Veranschaulichung des DL-Formats anhand der Umwandlung eines Farbbilds.

Prinzipiell ist es bei DL-Netzen irrelevant, ob RGB- oder sonstige Werte geliefert werden. Bildverarbeitende Netze sind allerdings meist auf 8-Bit-Werte ausgerichtet und auch trainiert. Das in der Forschungsarbeit eingesetzte Darknet-Framework setzt Farbbilder, also drei Kanäle mit 8-Bit-Werten voraus. Damit ist für Farbbilder nur die bereits beschriebene Konvertierung in ein 1-kanaliges Bild erforderlich. Für die anderen Bildarten wurden Methoden zur Herstellung aller möglichen Kombinationen aus Intensitäts- und Tiefenbild (siehe Abbildung 4-12) implementiert.

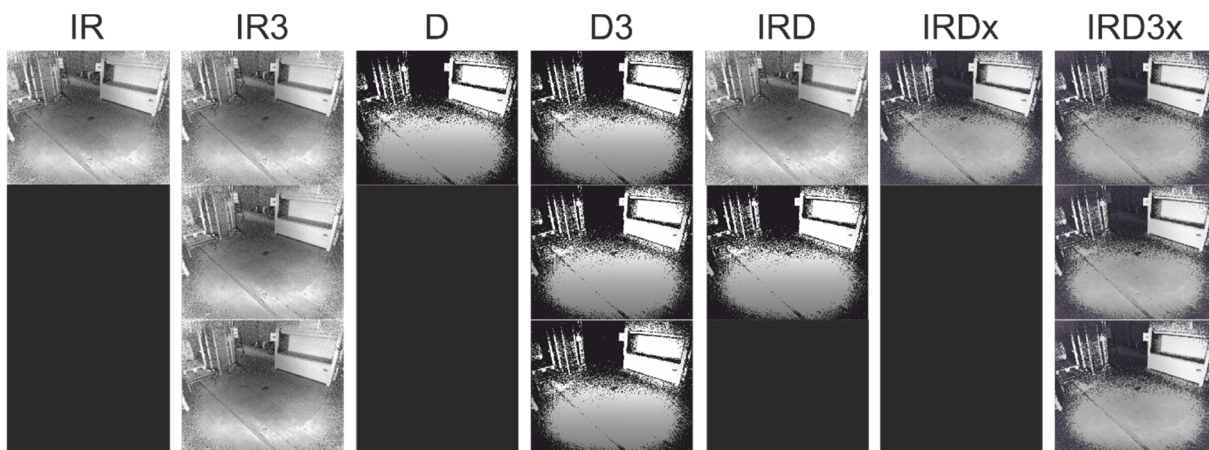


Abbildung 4-12: DL-konvertierte Bilder von links nach rechts: IR: Intensitätsbild (1 Kanal), IR3: Intensitätsbild (alle Kanäle), D: Tiefenbild (1 Kanal), D3: Tiefenbild (3 Kanäle), IRD: Infrarot (1. Kanal) & Tiefe (2. Kanal), IRDx: Infrarot & Tiefe kombiniert (1 Kanal), IRD3x: Infrarot & Tiefe kombiniert (3 Kanäle).

Die einfachste Möglichkeit der Konvertierung ist die Füllung des dreimal höheren Bildes mit den Daten des Intensitäts- oder Tiefenbilds. Die restlichen Zeilen können entweder leer bleiben bzw. mit Nullen (IR und D) aufgefüllt werden oder die gleichen Daten werden insgesamt dreimal nacheinander geschrieben (IR3 und D3). Bei der Bildart

„IRD“ wird zuerst das Intensitätsbild in das erste Drittel geschrieben und anschließend das Tiefenbild im zweiten Drittel. In der letzten implementierten Bildart „IRD(3)x“ wird aus dem Intensitäts- und Tiefenbild das arithmetische Mittel jedes Pixels gebildet. Somit sind in einem „Teilbild“ alle Informationen indirekt enthalten. Der Sinn dieser hybriden Bildart ist die Aufhellung weiter entfernter Pixel des Intensitätsbilds, da der entsprechende Tiefenwert entsprechend höher ist. Somit führt das arithmetische Mittel letztlich zur Angleichung der Belichtung.

### 4.2.3 Segmentierung (engl.: „segmentation“)

Ein Vorverarbeitungsschritt, welcher der Segmentierung zugeordnet werden kann, ist in der Methode `computeWorldAndGDMask` enthalten. In diesem wird die Segmentierung des Tiefenbildes in Hinter-, Vorder- und Untergrund vorgenommen. Gleichzeitig wird eine Maske erstellt, sodass auch in den anderen Bildarten entsprechende Segmentierungen durchgeführt werden können. Die Vorverarbeitung und Segmentierung wurden aufgrund der effizienteren Berechnung in einer Methode vereint. Folgende Prozeduren werden dabei durchgeführt (vgl. Abbildung 4-13 zur Veranschaulichung der Schritte):

1. Konvertierung der projektiven (u,v,d) Pixelkoordinaten des Tiefenbilds in das euklidische (x,y,z) Koordinatensystem (KOS) relativ zur Kamera (-achse):

Verwendung des Kameramodells der „projektiven Kamera“. Die Umrechnung von Bildpunkten in metrische Koordinaten ergibt sich nach [Sny-2017, S. 306ff.] wie folgt:

$$\vec{x}_K = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \frac{(u-c_x)}{f_x} \\ \frac{(v-c_y)}{f_y} \\ 1 \end{pmatrix} \cdot d \quad (5)$$

mit

$\vec{x}_K$ : Ergebnisvektor mit metrischen Koordinaten

u: horizontaler Bildpunktkoordinate

v: vertikale Bildpunktkoordinate

d: Tiefenwert des Bildpunkts

$c_x$ : halbe Breite des Bildes in Pixeln

$c_y$ : halbe Höhe des Bildes in Pixeln

$f_x$ : horizontale Brennweite

$f_y$ : vertikale Brennweite.

Bei der Umrechnung wird die halbe Bildbreite bzw. -höhe abgezogen, sodass  $x = 0$  der Mitte des Kamerabilds entspricht. Dementsprechend befinden sich Koordinaten mit  $x < 0$  auf der linken Hälfte des Bilds.

2. Drehung des KOS zur Horizontalen um den Kamerawinkel  $\alpha$  durch Multiplikation einer Rotationsmatrix um die x-Achse:

$$\vec{x}'_K = \vec{x}_K \cdot \mathbf{M}_{rot} = \vec{x}_K \cdot \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix} \quad (6)$$

3. Translation des KOS auf den Boden:

$$\vec{x}''_K = \vec{x}'_K + \vec{x}_{trans} = \vec{x}_K + \begin{pmatrix} 0 \\ -h_{sensor} \\ 0 \end{pmatrix} \quad (7)$$

mit  $h_{sensor}$ : Montagehöhe des Sensors

4. Erstellung einer 2D-Maske zur Entfernung des Bodens und Hintergrunds:

Die Maske enthält für alle Pixel, welche die minimale Höhe  $y_{min}$  unterschreiten oder die maximale Entfernung  $z_{max}$  überschreiten, den Wert 0, ansonsten 1. Die Maske kann anschließend auf alle Arten von Bildern mittels der Methode *applyFilterMask* angewandt werden, sodass der Boden und der Hintergrund entfernt werden. Für  $y_{min}$  sind Werte um 10 cm sinnvoll, dadurch wird überwiegend nur der Boden entfernt. Bezüglich  $z_{max}$  sind Werte über dem maximalen Bremsweg des Staplers, z. B. 6,25 m bei 16 km/h (vgl. Abbildung 3-3), zweckmäßig.

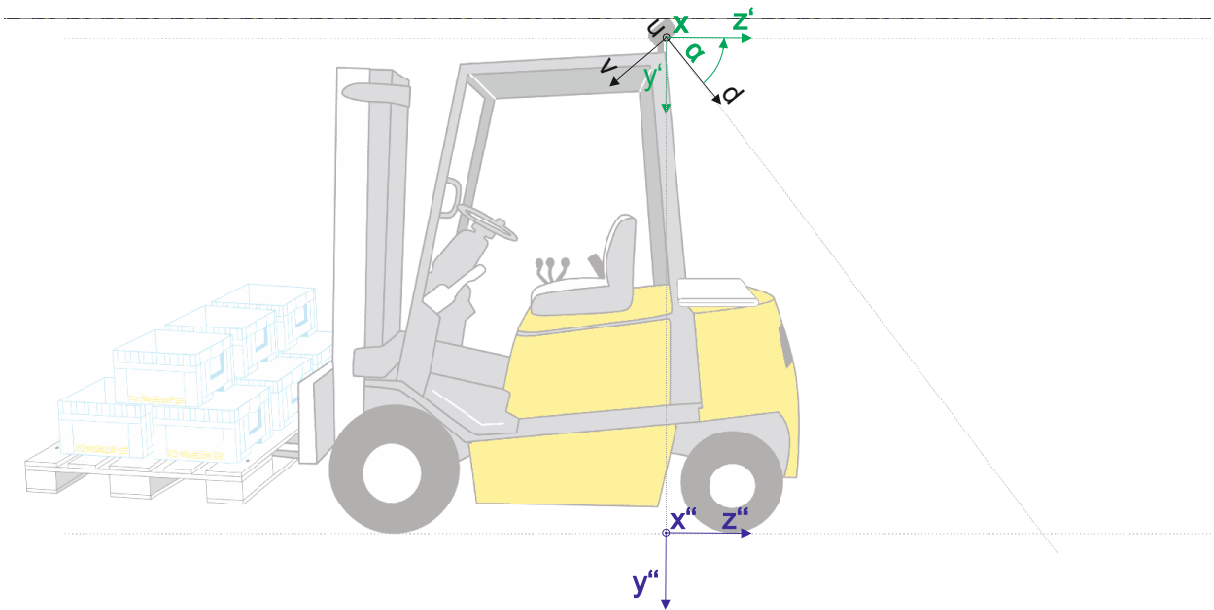


Abbildung 4-13: Schematische Darstellung unterschiedlicher Koordinatensysteme.

### Winkel- und Höhererkennung

Die zuvor beschriebenen Koordinatentransformationen benötigen als Parameter sowohl die Höhe als auch die horizontale Ausrichtung der Kamera. Zwar können diese Parameter auch gemessen werden, allerdings ist bei jeder Änderung des Kamerawinkels, z. B. durch Vibrationen, eine erneute Messung notwendig, da in der Kamera keine weitere Sensorik zur Lageerkennung vorhanden ist.

Die meisten Algorithmen zur Bodenerkennung suchen die Bodenebene in der Punktwolke über den „random sample consensus“- (RANSAC) oder einen ähnlichen Medianalgorithmus. Punktwolken-Berechnungen können weder von der CPU noch von der GPU effizient berechnet werden, weshalb diese relativ langsam sind ( $> 30$  ms) [Ema-2012; Hol-2011]. Selbst aktuelle, auf hohe Performanz ausgelegte Verfahren benötigen auf herkömmlichen Computern über 17,50 ms [van-2017]. Grundsätzlich wäre die Berechnungszeit ausreichend gering, da nach Abbildung 4-3 die Winkel- und Höhenberechnung die einzige Berechnung auf der CPU ist, welche parallel zur Kollisions- und Personenerkennung auf der GPU durchgeführt wird. Demzufolge stünden etwa 30 ms Rechenzeit zur Verfügung. Trotzdem wurde eine Methode entwickelt, welche im Anwendungsfall deutlich schneller ist und zugleich robuste Werte liefert.

Die Vorgehensweise ist ähnlich zu den Methoden zur Bestimmung des Bodens in der Punktwolke, allerdings werden alle Operationen im originalen 2 1/2-D-Tiefenbild angewandt. Zuerst werden die Tiefenwerte von fünf vertikalen Linien im zentralen unteren Bereich des Tiefenbilds extrahiert (siehe Abbildung 4-14). Der Bereich wurde gewählt, da dieser beim Stapler meist frei ist.

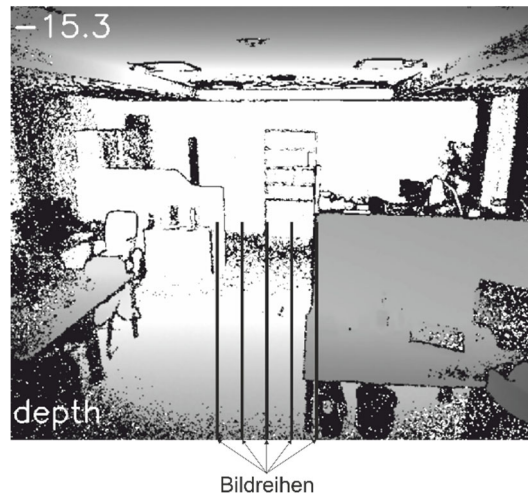


Abbildung 4-14: Schematische Darstellung der ausgewählten Linien für die Geradenschätzung [Lan-2017].

Für jede Linie wird eine Geradenfunktion mittels der „maximum likelihood“-Methode (vgl. [Hay-2001, S. 447ff.]) geschätzt, welche einer Minimierungsfunktion der quadratischen Abstände der Punkte entspricht. Der Kamerawinkel kann daraufhin über den Arkussinus der Steigung jeder Gerade ermittelt werden. Die Kamerahöhe ergibt sich durch den Nullpunktabstand der Gerade.

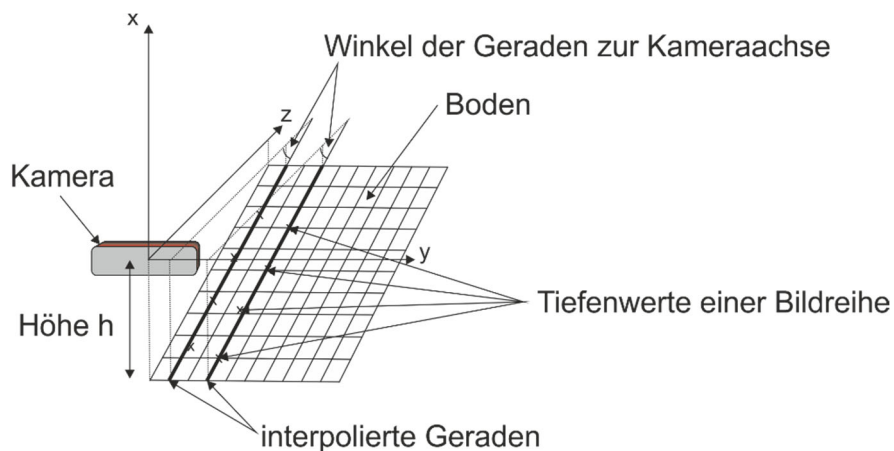


Abbildung 4-15: Ableitung des Kamerawinkels und -höhe aus den interpolierten Geraden [Lan-2017].

Die Implementierung der Vorgehensweise zur Ermittlung der Parameter entspricht den Methoden `getCameraAngle` bzw. `getCameraHeight` (siehe Abbildung 4-16) der Klasse `cameraAngle`.

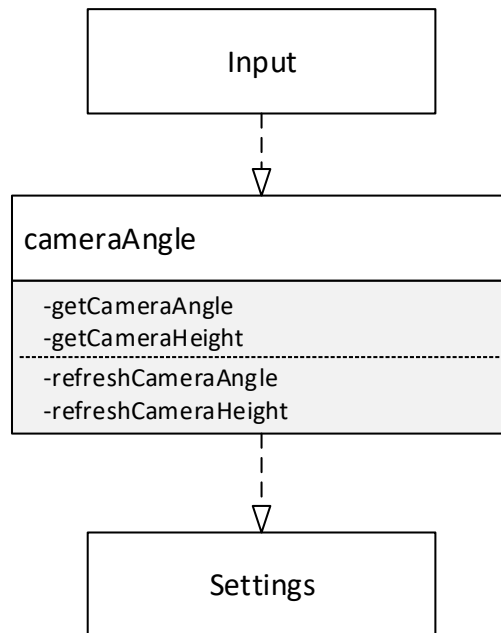


Abbildung 4-16: Klassendiagramm für Winkel- und Höhenerkennung.

Die Werte werden allerdings nicht zwangsläufig vom Hauptprogramm übernommen. Zuerst wird die Prüfung der Validität der Werte in den Methoden *refreshCameraAngle/Height* durchgeführt. Ein neuer Wert für Kamerawinkel bzw. -höhe wird nur verwendet, falls folgende Voraussetzungen gegeben sind:

- Es konnte für mindestens drei der fünf Reihen eine die Geradenfunktion berechnet werden.
- Winkel:
  - o Die Streuung der berechneten Winkel aller interpolierbaren Geraden ist kleiner als  $1^\circ$ .
  - o Der Median der neu berechneten Winkel weicht vom Durchschnitt der letzten fünf Messungen um weniger als 10 % ab.
- Höhe:
  - o Die Streuung der berechneten Höhen ist kleiner als 20 mm.
  - o Der Median der neu berechneten Höhen weicht vom Durchschnitt der letzten fünf Messungen um weniger als 10 % ab.

Aufgrund der Überprüfung der Streuungen und Abweichungen von vorhergehenden Messungen, sind Fehlmessungen, wenn sich bspw. Gegenstände im extrahierten Bildbereich befinden, nahezu ausgeschlossen. Sollte sich z. B. ein Karton auf allen fünf Linien befinden, kann keine Gerade mittels der „maximum likelihood“-Methode geschätzt werden. Der bisherige Winkel bzw. Höhe wird demzufolge beibehalten. Wenn der Karton nur zwei der fünf Linien berührt, werden die Parameter anhand der verbleibenden drei Geraden geschätzt. Die entwickelte Methode erlaubt es, die Berechnung

des Winkels und der Höhe in jedem Bild durchzuführen, da sie eine Rechenzeit von deutlich weniger als 1 ms aufweist.

#### 4.2.4 Personenerkennung (engl.: „person detection“)

Zur Erkennung von Personen gibt es unterschiedliche Herangehensweisen. In dieser Arbeit wird die Erkennung von Personen auf den neuesten Forschungsstand mittels maschinellen Lernens beschränkt. Es wurden zwei Ansätze untersucht: SVM und DL. Die SVM hat sich in Kombination mit HOG-Features (siehe 2.3.1 bzw. 2.4.2) bereits bei der Personenerkennung bewährt [Ben-2015]. Die Erkennungsleistung wird von DL-Netzen seit einigen Jahren übertroffen. Es wurden trotzdem beide Wege untersucht, da bei der Verwendung von SVM die Echtzeitfähigkeit aufgrund der deutlich schnelleren Berechnung eher gewährleistet werden kann.

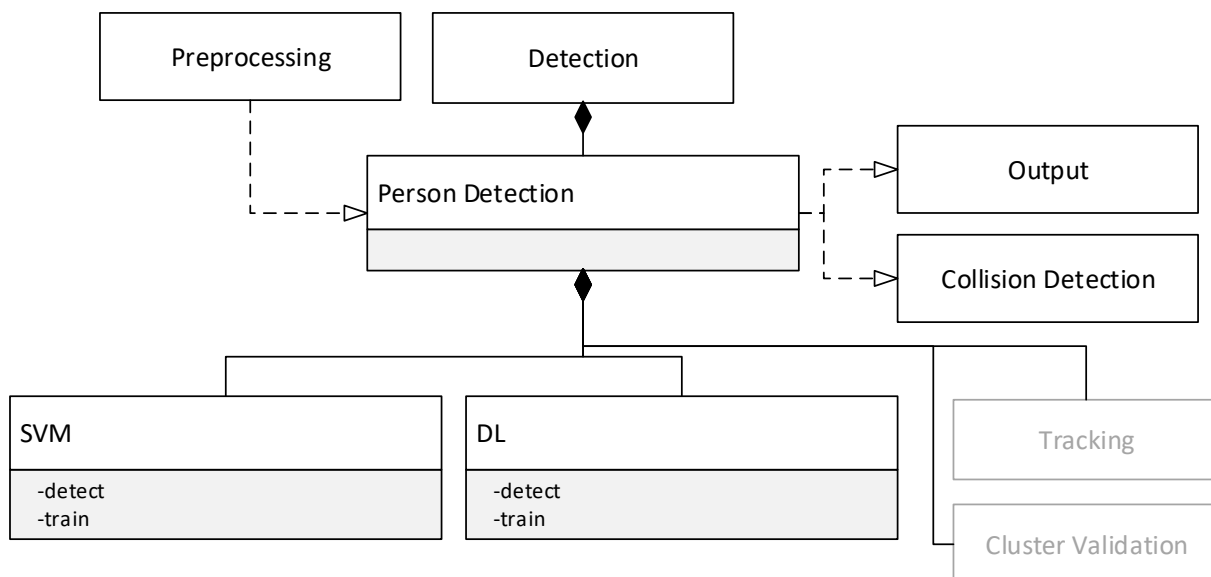


Abbildung 4-17: Klassendiagramm zur Personenerkennung.

Für die Implementierung der SVM in Verbindung mit HOG-Features wurde sowohl beim Training als auch bei der Detektion auf Methoden von OpenCV zurückgegriffen. Es wurden lediglich kleine Modifikationen vorgenommen, sodass die HOG-Features für die SVM verwendet werden konnten. Als Eingang kann das Farb-, Intensitäts- oder Tiefenbild dienen. Die Detektionsergebnisse werden als Rechtecke, welche die erkannte Person umschließen (siehe Abbildung 4-18), an die Output- und Kollisionsgefahrerkennungsklasse übergeben.



Abbildung 4-18: Personenerkennung mit einer SVM im Farbbild.

Für die Einbindung des DL-Netzes wurde das „Darknet“-Framework verwendet, da es in C++ vorliegt und das derzeit (2019) rechenärmste leistungsfähige DL-Netz „Yolo V3“ (siehe Abschnitt 2.3.2) unterstützt. Die einzigen Modifikationen, die zur Verwendung des Frameworks notwendig sind, ist die Anpassung der Eingangsdaten. Die Eingangsdaten müssen in drei Kanälen in 8-Bit vorliegen. Die entsprechende Umwandlung der Bildarten wurde bereits in 4.2.2 gezeigt. Wie bei der SVM werden als Resultate Rechtecke ausgegeben (Abbildung 4-19). Der Vergleich von Abbildung 4-18 mit Abbildung 4-19 zeigt, dass das DL-Netz die Umrisse der Schaufensterpuppe präziser erkennt.

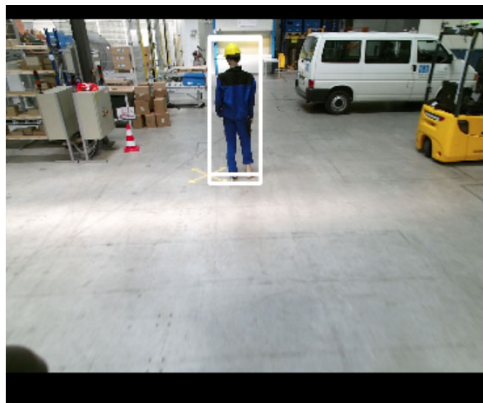


Abbildung 4-19: Personenerkennung mit einem YoloV3-Detektor im Farbbild.

Ein weiterer Unterschied fällt bezüglich der Anzahl an Rechtecken auf: Bei der Erkennung durch den Yolo-Detektor wurde zwei Mal an der gleichen Stelle ein Mensch mit minimal abweichender Größe erkannt. Die sogenannte „instant segmentation“ ist kein generelles Problem von DL-Netzen, es tritt beim Yolo-Detektor allerdings häufiger auf. Das ist für den Anwendungsfall jedoch irrelevant, da Menschen nicht gezählt werden sollen, sondern nur die Kenntnis über etwaige Aufenthaltsbereiche wichtig ist.



Im Klassendiagramm sind zwei weitere Subklassen *Tracking* und *Cluster Validation* dargestellt. Zur Verfolgung (engl.: „tracking“) bereits erkannter Personen wurden mehrere Algorithmen ([Bab-2009; Dan-2014; For-2008; Gra-2006; Kal-2010; Kal-2012]) getestet. Allerdings wurden mit den beschriebenen Methoden auch Falscherkennungen verfolgt. Darüber hinaus funktionierten die Tracking-Verfahren nur bedingt, da aufgrund der Bewegung der Kamera und der zu verfolgenden Personen sehr schnelle Änderungen im Bild erzeugt werden. Das hat zur Folge, dass die Tracking-Algorithmen innerhalb weniger Bilder die Person verlieren. Aufgrund dieser zwei Tatsachen hat der Einsatz von Tracking-Algorithmen in Voruntersuchungen zu signifikant schlechteren Ergebnissen bezüglich der Erkennung von Personen geführt. Für das System ist eine Erkennung in jedem Bild nicht notwendig, da ab der ersten Erkennung einer Person der entsprechende Bereich als kritisch für Personen gespeichert und bei mehreren Erkennungen auch extrapoliert wird. Aufgrund der fehlenden Notwendigkeit und der Verschlechterung der Personenerkennung wird im weiteren Verlauf der Arbeit diese Methode nicht weiter betrachtet.

Ein zum Tracking vergleichbar schlechtes Ergebnis wurde bei dem Versuch erzielt, erkannte Personen über deren reale Dimensionen zu validieren. Der Ansatz der *Cluster Validation* war, die Tiefendaten von erkannten Personen zu clustern und aus dem Cluster die realen Abmessungen der erkannten Person abzuleiten. Anschließend sollten die reale Breite, die Höhe, das Verhältnis der Höhe zur Breite und die Fläche des Menschen analysiert und mit den in Abschnitt 3.1.2 recherchierten anthropometrischen Daten verglichen werden. Allerdings mussten bei Voruntersuchungen die Maximalwerte der Maße derart groß gewählt werden (z. B. wenn ein Mensch mit seitwärts ausgestreckten Armen steht), dass kaum Fehlerkennungen dadurch aussortiert werden konnten. Zusätzlich wurden mehr richtige Personenerkennungen fälschlicherweise durch nicht korrektes Clustering entfernt. Diese Methode wird im Folgenden daher auch nicht weiter betrachtet, da sie den eigentlichen Zweck der Optimierung der Personenerkennung nicht erfüllen konnte und auch nicht notwendig ist.

### 4.2.5 Kollisionsgefahrerkennung (engl.: „collision detection“)

Die wichtigste Klasse des Systems ist die Erkennung von Kollisionsgefahren. Als Eingang dienen primär die Daten aus dem Preprocessing (siehe Abbildung 4-20). In der Klasse wird nach der Kollisionsgefahrberechnung auch entschieden, ob eine Kollision mit einem Menschen oder mit einem Objekt bevorsteht.

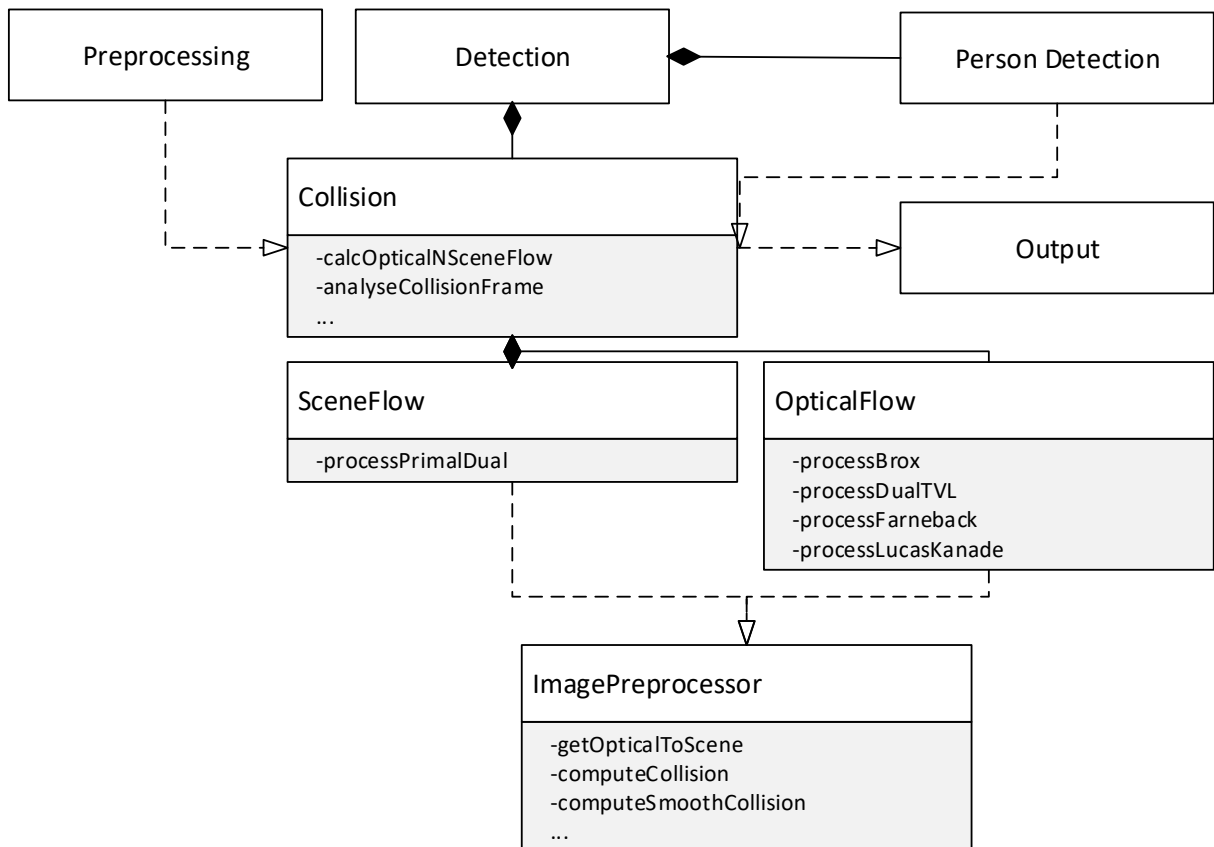


Abbildung 4-20: Klassendiagramm der Kollisionsgefahrerkennung.

Die erste Methode, welche in der Klasse aufgerufen wird, ist *calcOpticalNSceneFlow*. Die Eingangsdaten bestehen aus einem Graustufenbild (konvertiertes Farbbild oder Intensitätsbild) und aus dem Tiefenbild. Die Methode führt in den jeweiligen Unterklassen *OpticalFlow* und *SceneFlow* die Berechnung der Bewegungsvektoren aus. Das Konzept und die Implementierung dieser Methode ist an das betreute interdisziplinäre Projekt von *Kaiser* angelehnt [Kai-2017].

Die Klasse *OpticalFlow* beinhaltet die folgenden Methoden des optischen Flusses: Brox ([Bro-2004]), DualTVL ([Pér-2013]), Farneback ([Far-2003]), Lucas-Kanade ([Luc-1981]). Anhand eines Graustufenbildes berechnen diese die Bewegung von Pixeln im Bildbereich. Daher ist eine nachträgliche Berechnung des „Scene Flow“ – die Bewegung im Raum – notwendig. Diese Berechnung wird über den CUDA-Shader *getOpticalToScene* der Klasse *ImagePreprocessor* wie folgt ausgeführt:

1. Berechnung der Bewegungsvektoren  $\overrightarrow{\Delta x_k}$  aus dem Weltbild (vgl. Segmentierung auf S. 69):

$$\overrightarrow{\Delta x_k} = \overrightarrow{x_k(t+1)} - \overrightarrow{x_k(t)} = \begin{pmatrix} x(u + \Delta u, v + \Delta v) \\ y(u + \Delta u, v + \Delta v) \\ z(u + \Delta u, v + \Delta v) \end{pmatrix} - \begin{pmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{pmatrix} \quad (8)$$

2. Berechnung der relativen Geschwindigkeitsvektoren zum Stapler durch Normierung der Bewegungsvektoren auf die Zeitdifferenz:

$$\overrightarrow{v_k} = \frac{\overrightarrow{\Delta x_k}}{\Delta t} \quad (9)$$

mit

u: horizontale Pixelkoordinate

v: vertikale Pixelkoordinate

$\Delta u, \Delta v$ : Ergebnis des optischen Flusses  $\rightarrow$  horizontale bzw. vertikale Verschiebung der Pixelkoordinate

$\overrightarrow{x_k(t)}$ : Weltkoordinaten eines Pixels (u,v) zum Zeitpunkt t

$\overrightarrow{v_k}$ : resultierender Bewegungsvektor eines Pixels (u,v).

Im Gegensatz dazu wird bei der Verwendung des Algorithmus „Primal-Dual“ ([Jai-2015]) der Scene-Flow direkt berechnet, daher wird nur der letzte Rechenschritt (9) benötigt. Der Algorithmus gibt als Ergebnis die Bewegung in Weltkoordinaten durch Angabe der internen Kameraparameter aus.

Nach der Berechnung der Geschwindigkeitsvektoren erfolgt die pixelweise Überprüfung einer möglichen Kollision mit dem Stapler in der Methode *computeCollision* mit dem Ray-Box-Intersection-Verfahren [Wil-2005]. Dazu wird der Stapler durch einen virtuellen Quader abstrahiert, welcher den Stapler komplett beinhaltet (siehe Abbildung 4-21):

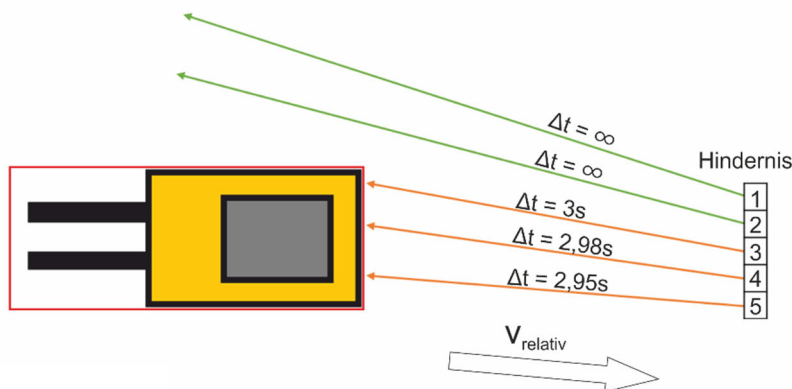


Abbildung 4-21: Veranschaulichung der Nutzung eines Quaders als Kollisionsobjekt zur Berechnung der Kollisionszeit einzelner Pixel.

Für jedes Pixel wird anhand des Tiefenwerts (aus dem Tiefenbild) und des Geschwindigkeitsvektors die Zeit bis zur Kollision mit dem virtuellen Quader berechnet. Das Ergebnis ist das „Kollisionsbild“, welches für jedes Pixel die Kollisionszeit mit dem Quader enthält. Falls aufgrund der relativen Bewegungsrichtung keine Kollision mit dem Quader bevorsteht, erhält das Pixel den Wert  $10^{10}$ . Da die Tiefenwerte teils mit starkem Rauschen behaftet sind, bietet die Methode *computeSmoothCollision* per RANSAC die Möglichkeit Ausreißer, welche zu sehr vom Median der letzten vier Bilder abweichen, zu entfernen.

Abschließend wird in der Methode *analyseCollisionFrame* das rohe oder RANSAC-gefilterte Zeitbild über folgende Vorgehensweise analysiert:

1. Clustering kollidierender Pixel über Konturensuche

Für die Entscheidung, ob eine Kollision vorliegt oder nicht, ist nicht nur die Kollisionszeit einzelner Pixel, sondern auch die Anzahl kollidierender Pixel entscheidend. Aufgrund des Rauschens im Tiefenbild gibt es zu jedem Zeitpunkt eine stark schwankende Anzahl von freien kollidierenden Pixeln im Bild. Das Zählen aller kollidierender Pixel im gesamten Bild ist daher nicht sinnvoll. Darüber hinaus drohen je nach Fahrsituation mehrere Objekte gleichzeitig zu kollidieren (siehe Abbildung 4-22). Die Einstufung, ob es sich tatsächlich um eine Kollisionsgefahr handelt, ist nur über die Zusammenfassung von Pixelbereichen zu Clustern möglich.

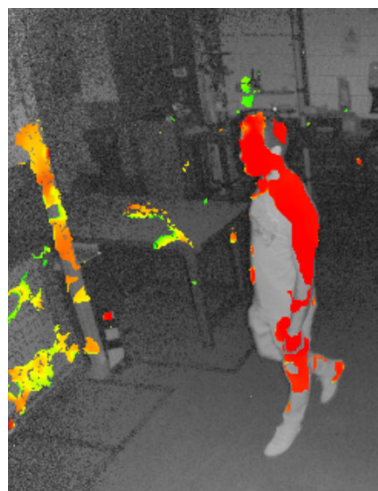


Abbildung 4-22: Beispielhafte Situation mit mehreren Kollisionsquellen. Kollidierende Pixel sind grün (Kollision < 5 s) bis rot (Kollision < 1 s) eingefärbt.

Für das Clustering wurde kein konventionelles Clustering-Verfahren (wie z. B. k-means) verwendet. Der Grund hierfür ist einerseits die hohe Rechenleistung,

welche für diese Verfahren notwendig ist; andererseits müssen bei den Verfahren die Anzahl der Cluster bekannt sein. In diesem Anwendungsfall variiert die Zahl der Cluster. Daher wurde ein anderer Lösungsweg verfolgt: Im Kollisionsbild werden die Konturen mittels des Algorithmus von *Suzuki* ([Suz-1985]) gesucht und die jeweiligen Objekte als Cluster gewertet. Abbildung 4-23 veranschaulicht das Clustering durch Konturensuchen. Die unterschiedlichen Cluster wurden mit verschiedenen Farben eingefärbt:

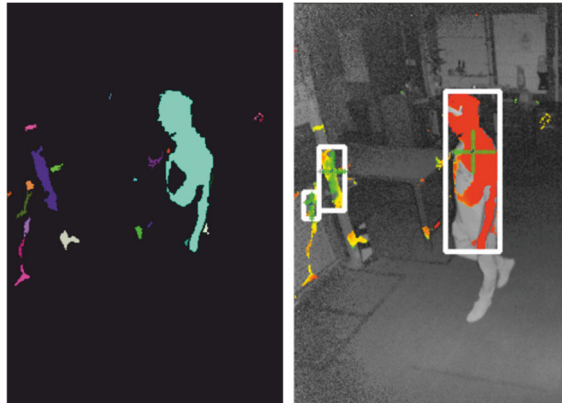


Abbildung 4-23: Links: Clustering mit Konturensuche im Kollisionsbild, Cluster farblich unterschiedlich. Rechts: Überlagerung mit Intensitätsbild und Darstellung der validen Cluster.

Auf dem rechten Bild wurden die als valide erkannte Cluster mit einem umschließenden Rechteck (engl.: „bounding box“ (BB)) gekennzeichnet.

### 2. Bestimmung der Anzahl kollidierender Pixel und deren Parameter

Nachdem die Cluster gefunden wurden, werden die Pixel im Cluster, welche die maximale Kollisionszeit (z. B. 5 s) für Menschen unterschreiten, gezählt. Darüber hinaus werden die kleinste und größte Kollisionszeit sowie die Distanz des Clusters bestimmt.

### 3. Normierung des Clusters

Die Anzahl der Pixel wird mit der mittleren Distanz des Clusters multipliziert und durch die Anzahl der Pixel des Eingangsbildes dividiert. Dadurch wird gewährleistet, dass der Schwellwert (*evaluate.collisionDetection.minCollisionArea*) zur Auslösung einer Kollision unabhängig von der Entfernung eines Objekts und der Auflösung des Sensors festgelegt werden kann.

#### 4. Kategorisierung des Kollisionsclusters

Befindet sich mindestens 70 % der BB-Fläche des gefundenen Kollisionsclusters in der BB einer gefundenen Person, wird die Kollisionsgefahr als Gefahr für einen Menschen gewertet, andernfalls Gefahr für ein Objekt. Der Wert von 70 % wurde empirisch auf Basis von Voruntersuchungen ermittelt.

Sofern der Schwellwert der normierten Anzahl an Pixel überschritten wurde, wird die kleinste berechnete Kollisionszeit aus Punkt 2 überprüft. Handelt es sich um ein Kollisionscluster, indem sich ein Mensch befindet, wird der höhere Schwellwert für Menschen (z. B. fünf Sekunden) benutzt, d. h. es wird früher gewarnt. Andernfalls wird der niedrigere Schwellwert (z. B. zwei Sekunden) für Objekte verwendet. Unterschreitet die kleinste Kollisionszeit den Schwellwert, übergibt die Methode die Kollisionskategorie an die Output-Klasse.

#### 5. Bestimmung der Historie

Zur Evaluations- und Optimierungszwecken werden Historien von Kollisionsgefahren erstellt, damit bei der Auswertung bspw. festgestellt werden kann, wann das erste Mal gewarnt wurde oder wie oft die Kollisionsgefahr der Kategorie „Kollisionsgefahr mit Mensch“ zugeordnet wurde. Kandidaten für ältere Kollisionsgefahren sind jene, deren BB sich mit den BB der neu erfassten Kollisionsgefahren überschneiden. Sollten dies mehrere sein, wird die neue Kollisionsgefahr derjenigen älteren zugeordnet, deren Zentrum am nächsten ist. Wurde keine passende ältere Kollision gefunden, wird ein neues Objekt in der Historie angelegt.

Die Sammlung der Kollisionsgefahren in einer Historie ermöglicht zudem die Möglichkeit nur in dem Fall zu warnen, wenn die gleiche Kollisionsgefahr mehrmals erkannt wurde.

#### 4.2.6 Ausgabe (engl.: „output“)

Im Gegensatz zu den vorher beschriebenen Klassen existiert für den letzten Prozessschritt – der Ausgabe – keine eigene Klasse in der Implementierung, vielmehr werden in verschiedenen Klassen Methoden zur Ausgabe ausgeführt.

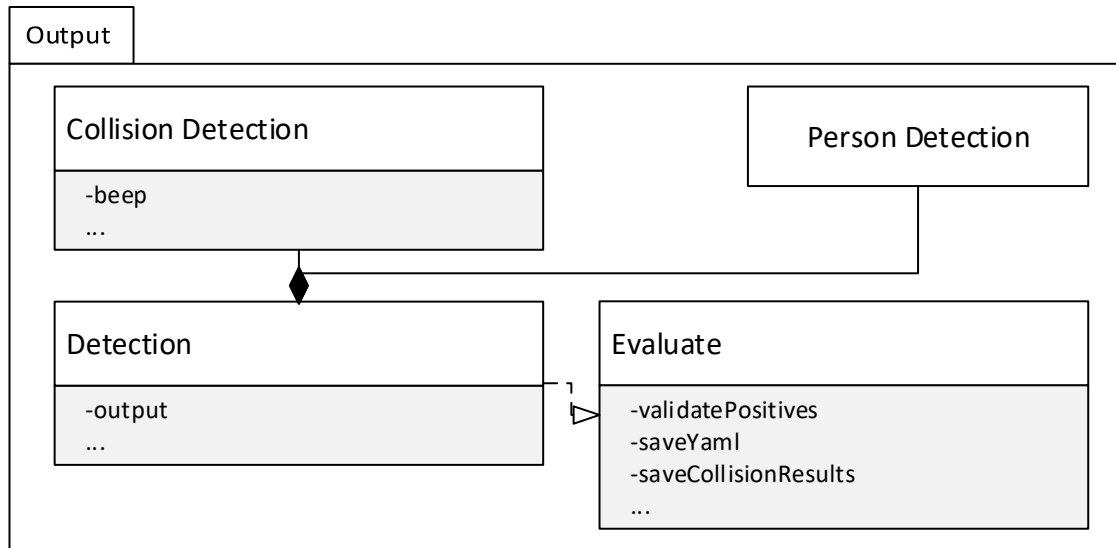


Abbildung 4-24: Klassendiagramm der an der Ausgabe beteiligten Klassen.

Wenn eine Kollisionsgefahr erkannt und klassifiziert wurde, wird durch die Methode *beep* ein Ton ausgegeben. Je nachdem wo sich die Kollision befindet, wird links, rechts oder auf beiden Lautsprecherkanälen ein Ton erzeugt. Wenn es sich um eine Kollision mit einem Menschen handelt, wird eine hohe Frequenz (1,5 kHz) verwendet, andernfalls eine tiefere Frequenz (0,5 kHz).

Diese Methode wurde zu reinen Testzwecken implementiert. Das heißt, die Frequenzen und die Modalität wurden nicht wissenschaftlich fundiert gewählt, da wie eingangs dieser Arbeit erwähnt, der Fokus der Arbeit nicht auf der korrekten Art und Weise der Warnung an den Fahrer liegt.

Die Methode *output* der *Detection*-Klasse dient zur Veranschaulichung der Ergebnisse der einzelnen Bildverarbeitungsschritte. Mit ihr können z. B. die Ergebnisse der Personen- oder Kollisionsgefahrerkennung dargestellt werden (vgl. Abbildung 4-19 und Abbildung 4-22). Sie übergibt außerdem der *Evaluate*-Klasse die Ergebnisse der Personen- und Kollisionsgefahrerkennung sowie die Bilddaten. In dieser Klasse werden in der Methode *validatePositives* die Ergebnisse der Personenerkennung in falsche, fehlende und richtige Erkennungen unterteilt und mit den Methoden *saveYaml* und *saveCollisionResults* gespeichert. Näheres zur Auswertungsmethode wird in Abschnitt 6.1.1 beschrieben.

## 4.3 Fazit

In diesem Kapitel wurde das Hard- und Softwarekonzept umgesetzt. Bezüglich der Hardware wurde geklärt, wo und wie die Kamera am Stapler positioniert bzw. ausgerichtet wird. Bei der Positionierung wurde das Dach des Staplers aus sensortechnischen Gründen präferiert. Die Ausrichtung der Kamera wurde anhand der Sensorparameter, des maximal erlaubten Bremswegs des Staplers und anthropometrischen Daten festgelegt.

Die Module des Softwarekonzepts wurden als Klassen in C++ bzw. in CUDA entwickelt. Die wichtigsten zwei Module sind die Personen- und die Kollisionsgefahrerkennung. Für die Personenerkennung wurden zwei ML-Methoden implementiert: Die SVM und der DL-Detektor „Yolov3“. Die SVM wurde implementiert, da sie in Verbindung mit HOG-Merkmalen eine bewährte, performante Lösung zur Klassifikation und Lokalisierung von Personen ist. Der Yolov3-Detektor wurde aus den zahlreich verfügbaren DL-Algorithmen gewählt, da dieser eine nahezu echtzeitfähige Berechnung bei gleichzeitig hoher Detektionsquote erlaubt. Auch wenn der DL-Detektor deutlich bessere Ergebnisse erwarten lässt, wurde die SVM trotzdem aufgrund des sehr geringen Rechenaufwands im Vergleich zu dem DL-Detektor implementiert.

Für die Kollisionsgefahrerkennung wurden die in OpenCV verfügbaren OF-Algorithmen und ein SF-Algorithmus, welcher für Farbbilder mit Tiefendaten (RGBD-Daten) entwickelt wurde, ausgewählt. Die Berechnung der Kollisionszeit aus den Daten der OF-Algorithmen wurde über das Ray-Box-Intersection-Verfahren realisiert. Bei diesem wird anhand der Bewegungsvektoren für jeden Pixel die Kollisionszeit mit einem virtuellen Quader, welcher den äußeren Staplerabmessungen entspricht, berechnet.





## 5 Vorbereitende Maßnahmen zur Evaluation des Systems

---

Im letzten Kapitel wurden die zur Funktion des Systems notwendigen Algorithmen vorgestellt, in diesem Kapitel die Vorgehensweise zur Evaluation dieser Algorithmen. Das zu entwickelnde System gehört zur Gruppe der Fahrerassistenzsysteme (engl.: „driver assistance systems“ (DAS)). Bei kamerabasierten Systemen gibt es grundsätzlich zwei Möglichkeiten ein DAS zu testen: während des Einsatzes in der realen Anwendung oder durch aufgenommene Daten. Da es unmöglich ist, einzelne Testfälle in der Realität unter den exakt gleichen Bedingungen (Wetter, Lichtverhältnisse) durchzuführen, ist die Variante mit den aufgenommenen Daten vorzuziehen. [Tan-2013] Nach *Cioran* wird folgende Vorgehensweise für die Evaluation eines DAS vorgeschlagen [Cio-2015, S. 43]:

1. Definition des Einsatzzwecks: Welche Funktionen müssen getestet werden und unter welchen Bedingungen?
2. Identifikation von Einflussfaktoren: Welche Umstände können die Funktionsfähigkeit der Systems beeinflussen?
3. Definition relevanter Verkehrssituationen: Welche Situationen sind besonders kritisch oder treten häufig auf?
4. Definition von Testfällen: Wie kann das System unter der Berücksichtigung von Schritt 1-3 getestet werden?
5. Evaluation der Testfälle: Wie gut funktioniert das System in den Testfällen?

Der Einsatzzweck wurde bereits in der Konzeptentwicklung in Abschnitt 3.1.3 definiert, dieser gilt für die Evaluation gleichermaßen. Die Erkenntnisse über die weiteren Schritte zwei bis vier werden in den folgenden Abschnitten 5.1 bis 5.3 beschrieben. Darauf folgt die Beschreibung der verwendeten öffentlich zugänglichen und den erzeugten Trainings- und Testdaten. Die Ergebnisse des letzten Schritts – der Evaluation – werden in Kapitel 6 und 7 beschrieben.

### 5.1 Identifikation von Einflussfaktoren

Die Faktoren, welche die Funktionsfähigkeit des Systems beeinflussen, sind aufgrund der Wahl eines exterozeptiven Sensors durch die Umwelt bedingt. Diese können entweder zur Störung der Kameradaten oder der nachfolgenden Bildanalysealgorithmen führen. Folgende Einflussfaktoren konnten identifiziert werden:

1. Einflüsse auf die Bilderzeugung bzw. Qualität der Kameradaten:
  - a. Änderung der Lichtverhältnisse
  - b. Erschütterungen bspw. durch Unebenheiten im Boden oder Befahren einer Rampe
  - c. Verschmutzung der Sensoren
  - d. Spiegelnde, metallische Objekte
  - e. Direkte Sonneneinstrahlung
2. Einflüsse auf die Bildanalysealgorithmen:
  - a. Personenerkennung
    - i. Körperdimensionen
    - ii. Posen von Menschen
    - iii. Getragene Kleidung oder Accessoires
    - iv. Grad der Verdeckung eines Menschen
  - b. Kollisionsgefahrerkennung
    - i. Geschwindigkeit und Fahrtrichtung des Staplers
    - ii. Bewegung von Kollisionsobjekten
    - iii. Dimension von Kollisionsobjekten
    - iv. Abstand zu Kollisionsobjekten

Die Änderung der Lichtverhältnisse führt zur Unter- oder Überbelichtung bei Kameras. Allerdings betrifft das im Falle der verwendeten ToF-Kamera nur das Farbbild, das Intensitäts- und Tiefenbild sind durch die Nutzung eines anderen Lichtspektrums (IR-Emitter/-Receiver) davon nicht betroffen.

Erschütterungen haben den größten Einfluss auf den Farbsensor, da bei diesem die Belichtungszeit im Vergleich zum IR-Sensor ungleich höher ist. Dadurch treten im Farbbild Mehrfachbelichtungen auf, welche u. a. zur Minderung der Schärfe führen. Gegenüber Verschmutzungen sind alle Sensoren aufgrund der Nutzung optischer Strahlen gleichermaßen anfällig.

Einige Störungseinflüsse treten aufgrund der Sensortechnik nur beim IR-Sensor und damit im Intensitäts- und Tiefenbild auf. Spiegelnde Oberflächen (im Anwendungsfall sind das hauptsächlich blanke Metalle oder Warnwesten) führen dazu, dass das emittierte Infrarotlicht teilweise oder überhaupt nicht zum IR-Receiver zurückgestreut wird. Dadurch fehlen im Bereich des metallischen Objekts und über diesen Bereich hinaus Daten. Auch direktes Sonnenlicht stört den IR-Sensor; durch die starke Infrarotstrahlung der Sonne kann der Sensor nicht mehr zwischen emittierten und fremden IR-Strahlen unterscheiden. Das gilt zwar für die verwendete Kinect v2, aber nicht für alle ToF-Kameras. Einige ToF-Kameras nutzen moduliertes IR-Licht, wodurch externe IR-Strahlungsquellen ignoriert werden können.

Die Einflussfaktoren auf die Personenerkennung sind auf das äußere Erscheinungsbild von Menschen beschränkt. Die Qualität der Erkennung kann z. B. durch die Größe eines Menschen oder durch dessen Pose beeinflusst werden. Auch die Textur eines Menschen, welche durch Hautfarbe, Kleidung und sonstige Accessoires sehr variabel ist, kann die Erkennung begünstigen oder verschlechtern. Darüber hinaus kann nicht davon ausgegangen werden, dass eine Person immer vollständig sichtbar ist. Ist die Person teilweise verdeckt, ist die Erkennung folglich schwieriger.

Die Erkennung von Kollisionen wird maßgeblich durch die Funktionsfähigkeit der Berechnung des optischen Flusses bestimmt. Dieser kann nur berechnet werden, wenn möglichst viele Bildmerkmale in konsekutiven Bildern wiedergefunden werden. Eine hohe Geschwindigkeit oder starke Drehung des Staplers kann zu verwaschenen Bildern führen, wodurch die Suche nach Merkmalen erschwert wird. Dasselbe kann auch auftreten, wenn sich ein Objekt im Sichtfeld der Kamera zu schnell bewegt.

Die Größe eines Objekts wirkt sich dahingehend auf die Kollisionsgefahrerkennung aus, dass bei größeren Objekten mehr Pixel analysiert werden können, wodurch sich etwaige Abweichungen bei der Distanz normalisieren. Analog kann dieser Zusammenhang auf den Abstand eines Objekts von der Kamera übertragen werden.

## **5.2 Identifikation relevanter Verkehrssituationen**

Im betrachteten Anwendungsfall, der Bewegung im intralogistischen Transport, treten ähnliche Verkehrssituationen wie im Straßenverkehr auf. Gabelstapler bewegen sich vorwärts, rückwärts und biegen ab. Rückwärtsfahrten treten aufgrund der eingeschränkten Sicht bei einigen Staplertypen (z. B. beim Gegengewichtsstapler) häufiger als bei Fahrzeugen im Straßenverkehr auf. Die Spitzengeschwindigkeit von handelsüblichen Modellen liegt in etwa bei 14-16 km/h. [Jun-2018; STI-2019a; STI-2019b]

Ein Sonderfall ergibt sich bei Gabelstaplern durch die oftmals benutzte 3-Rad-Kinematik. Dadurch ist die Wendigkeit im Vergleich zu Kraftfahrzeugen ungleich höher, wodurch eine Drehung auf der Stelle möglich wird. Kollisionen können aufgrund dessen auch seitwärts auftreten.

Potenzielle Kollisionsobjekte sind die Lagerinfrastruktur, andere Fahrzeuge und Menschen, wobei sich Letztgenannte ebenso in Bewegung befinden können. Abbildung 5-1 zeigt beispielhaft die möglichen Bewegungsrichtungen eines Staplers und von Menschen. Im Falle des Menschen ist eine Drehung um die eigene Achse zwar ebenso möglich, aber für die Kollisionsgefahrerkennung unbedeutend.

Die relevanten Verkehrssituationen resultieren aus der Kombination von Relativbewegungen, welche in einer Kollision enden.

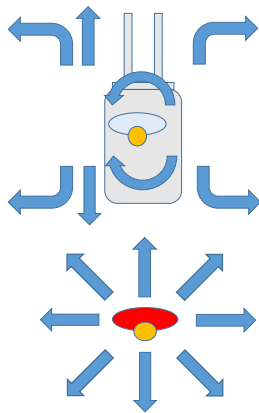


Abbildung 5-1: Darstellung möglicher Bewegungskonstellationen im intralogistischen Transport.

In Verkehrssituationen mit einer Kollisionsgefahr zeigt der relative Bewegungsvektor zwischen Gabelstapler und einem Hindernis aus der Sicht des Staplers immer zum Hindernis. Folglich gibt es nur eine Verkehrssituation, bei welcher die Relativgeschwindigkeit und der Abstand die einzigen Variablen sind, welche für die Bewertung der Gefahr entscheidend sind. Die sich daraus ergebenden Konstellationen werden in der folgenden Definition der Testfälle ausgearbeitet.

### 5.3 Definition der Testszenarien und -fälle

Zur Auswahl von Algorithmen, welche für das beschriebene Konzept eingesetzt werden können, werden für den objektiven Vergleich der Qualität der Personen- und Kollisionsgefahrerkennung Testszenarien bzw. -fälle benötigt. Anhand dieser soll entschieden werden, ob die gewünschte Funktion erbracht wird. Entscheidend für die Wirksamkeit des Systems ist einerseits die Personen- und andererseits die Kollisionsgefahrerkennung. Folgend wird beschrieben, welche Szenarien für die Personenerkennung und welche für die Kollisionsgefahrerkennung zweckgemäß sind.

#### 5.3.1 Personenerkennung

Das entwickelte System wurde zum Einsatz in Lagern ausgelegt. Die Haupttätigkeiten von Menschen, welche kein Fahrzeug bedienen, sind Manipulationsaufgaben an verschiedenen Orten, zwischen denen sie sich bewegen müssen. Dabei nehmen sie unterschiedliche Posen (aufrecht, sitzend, laufend, bückend) ein. Zwischen einem Stapler und einem Menschen können Hindernisse stehen, demzufolge können Menschen aus der Sicht des Staplers auch nur zum Teil sichtbar sein. Da im Lager auch Besucher verkehren, kann nicht davon ausgegangen werden, dass alle Menschen im Lager eine

Arbeitskleidung (ggf. inkl. Helm und Warnweste) tragen.

Zusammengefasst müssen folgende Testszenarios bezüglich der Personenerkennung überprüft werden:

Menschen, welche

- aufrecht sind,
- sitzen,
- laufen,
- sich bücken,
- nur teilweise sichtbar sind,
- ein Flurförderzeug (z. B. Hubwagen, Ameise) bedienen,
- oder Schutzkleidung tragen.

### 5.3.2 Kollisionsgefahrerkennung

Potenzielle Kollisionsobjekte für Gabelstapler sind alle Objekte, die während des Betriebs in die unmittelbare Nähe des Staplers gelangen. Insbesondere die Lagerinfrastruktur ist stark gefährdet, da die Gabelstapler in engen Gassen verkehren müssen. Aber auch Personen sind, wie den Statistiken aus Abschnitt 3.1.1 zu entnehmen ist, besonders gefährdet. Aufgrund dessen werden sowohl Testszenarios mit als auch ohne Personen für die Kollisionsgefahrerkennung definiert.

Die ersten Szenarios (vgl. 1a bis 1d in Abbildung 5-2) sind Kollisionen mit einem Hindernis. Der Stapler fährt entweder frontal auf das Hindernis zu, dreht in Hindernisnähe oder fährt um das Hindernis, woraufhin ein anderes Hindernis folgt. In der zweiten Reihe von Szenarios (2a bis 2d) soll geprüft werden, ob das System erkennt, dass keine Kollisionsgefahr besteht, obwohl der Gabelstapler sehr nah an Hindernissen vorbei- bzw. durchfährt oder in ausreichendem Abstand auf der Stelle dreht. Die verbleibenden Szenarios (3a bis 3d) bilden Kollisionen mit Menschen ab. Diese sind ähnlich zu den ersten Szenarios, aber das System muss hierbei aufgrund der Gefährdung eines Menschen früher warnen.

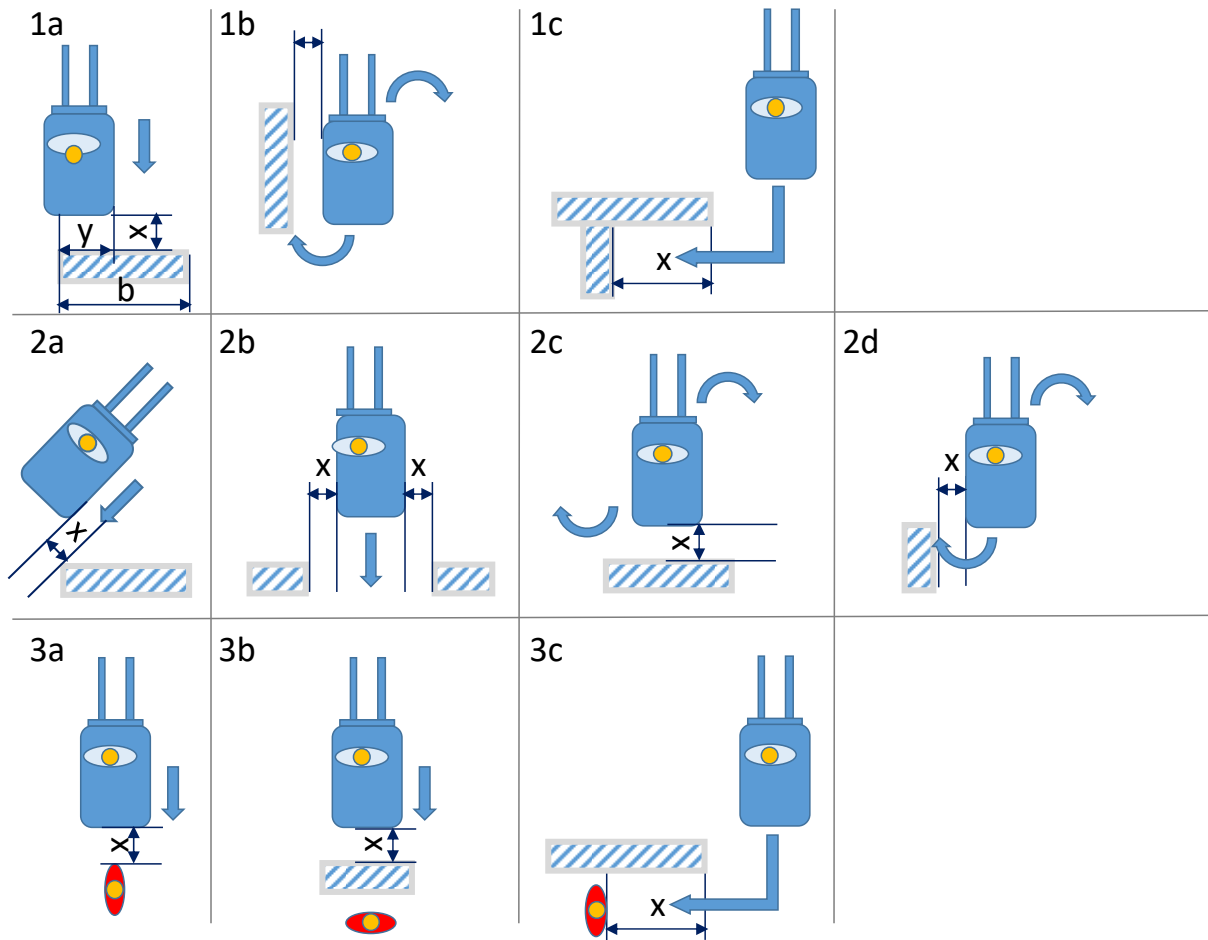


Abbildung 5-2: Übersicht zu den Testszenarien der Kollisionsgefahrerkennung.

Zusätzlich werden für vereinzelte Szenarien in Testfällen Parametervariationen durchgeführt. Je nach Testfall werden entweder die Geschwindigkeit des Gabelstaplers, die Höhe des Hindernisses oder die Abstände  $x$  bzw.  $y$  variiert.

Die Geschwindigkeit des Staplers wird zwischen 2, 4 und 6 km/h variiert, da diese Geschwindigkeiten über Modi beim verwendeten Stapler fixiert werden können und damit vergleichbare Durchläufe produziert werden können. Als Kollisionsobjekte werden Kleinladungsträger (KLT) verwendet, welche eine Höhe von 0,28 m aufweisen. Im Versuchsplan in Tabelle 5-1 sind die unterschiedlichen Hindernishöhen Vielfache der KLT-Höhe. Die Abstände  $x$  und  $y$  werden entweder inkrementell erhöht oder ergeben sich aus dem maximalen Bremsweg nach Abbildung 3-3. Im Falle der Kollisionen mit Menschen wird zusätzlich ein Sicherheitszuschlag von 50 % beim notwendigen Bremsweg berücksichtigt. Der Sicherheitszuschlag wurde entsprechend hoch gewählt, da die Vermeidung von Kollisionen mit Menschen im Vordergrund steht. Etwaige zu spät erkannte Kollisionsgefahren mit Menschen, z. B. durch nicht analysierte Bilder (Kollisionsberechnung dauert länger, als die Kamera für das Senden eines neues Bilds benötigt), werden somit gemindert.

Tabelle 5-1: Versuchsplan für die Kollisionstests (n. b.: nicht bestimmt).

Testfall	Szenario	Geschwindigkeit des Staplers [km/h]	Höhe des Hindernisses [m]	x [m]	y [m]
1	1a	2	1,12	> 0,34	n. b.
2	1a	4	1,12	> 0,81	n. b.
3	1a	6	1,12	> 1,40	n. b.
4	1a	6	0,28	> 2,12	n. b.
5	1a	6	0,56	> 2,12	n. b.
6	1a	6	1,68	> 2,12	n. b.
7	1a	6	2,24	> 2,12	n. b.
8	1a	6	1,12	> 2,12	0
9	1a	6	1,12	> 2,12	b/4
10	1a	6	1,12	> 2,12	b/2
11	1b	n. b.	1,12	0,1	n. b.
12	1b	n. b.	1,12	0,2	n. b.
13	1b	n. b.	1,12	0,3	n. b.
14	1b	n. b.	1,12	0,4	n. b.
15	1c	2	1,12	0,34	n. b.
16	1c	2	1,12	0,4	n. b.
17	1c	2	1,12	0,5	n. b.
18	1c	2	1,12	0,6	n. b.
19	2a	6	1,12	0,1	n. b.
20	2a	6	1,12	0,2	n. b.
21	2a	6	1,12	0,3	n. b.
22	2a	6	1,12	0,4	n. b.
23	2b	6	1,12	0,1	n. b.
24	2b	6	1,12	0,2	n. b.
25	2b	6	1,12	0,3	n. b.
26	2b	6	1,12	0,4	n. b.
27	2c	n. b.	1,12	0,1	n. b.
28	2c	n. b.	1,12	0,2	n. b.
29	2c	n. b.	1,12	0,3	n. b.
30	2c	n. b.	1,12	0,4	n. b.
31	2d	n. b.	1,12	0,1	n. b.
32	2d	n. b.	1,12	0,2	n. b.
33	2d	n. b.	1,12	0,3	n. b.
34	2d	n. b.	1,12	0,4	n. b.
35	3a	2	n. b.	> 2,12·1,5	n. b.
36	3a	4	n. b.	> 2,12·1,5	n. b.
37	3a	6	n. b.	> 2,12·1,5	n. b.
38	3b	6	0,28	> 2,12·1,5	n. b.
39	3b	6	0,56	> 2,12·1,5	n. b.
40	3b	6	1,68	> 2,12·1,5	n. b.
41	3b	6	2,24	> 2,12·1,5	n. b.
42	3c	n. b.	n. b.	0,34·1,5	n. b.
43	3c	n. b.	n. b.	0,4·1,5	n. b.



## 5 Vorbereitende Maßnahmen zur Evaluation des Systems

---

<b>44</b>	3c	n. b.	n. b.	0,5-1,5	n. b.
<b>45</b>	3c	n. b.	n. b.	0,6-1,5	n. b.

## 5.4 Trainings- und Testdaten

Eingangs wurde im Kapitel festgelegt, dass zur Auswertung der Algorithmen Aufnahmen verwendet werden. Für die ML-Algorithmen der Personenerkennung sind neben den Aufnahmen zur Auswertung der Testfälle auch Daten zum Training notwendig. Folgend werden zuerst öffentlich zugängliche Datenquellen für Trainingsdaten beschrieben und anschließend wie eigene Trainings- als auch Testdaten für die Personen- und Kollisionsgefahrerkennung erzeugt wurden.

### Personenerkennung

Das Training von ML-Algorithmen ist entscheidend für deren Leistungsfähigkeit bei der Detektion. Die Trainingsdaten sollten den später zu analysierenden Bildern ähneln, aber diesen nicht exakt entsprechen.

Es sind jeweils zwei Kategorien von Bildern notwendig: Negativ- und Positivbilder. Auf den Negativbildern ist keines der gesuchten Objekte (hier: keine Person) enthalten (siehe Abbildung 5-3). Positivbilder dürfen nur das gesuchte Objekt beinhalten.

Im Falle von DL werden meist keine separaten Negativ- und Positivbilder mehr verwendet, sondern es werden die zu suchenden Objekte in beliebigen Bildern markiert bzw. deren Positionen in einer separaten Datei gespeichert.



Abbildung 5-3: Beispiel für Bilderkategorien, von links nach rechts: Negativbild, Positivbild, markiertes Bild.

In den letzten Jahren gab es intensive Forschungsaktivitäten im Bereich Personenerkennung. Daraus sind einige Bilderdatenbanken entstanden, welche einerseits zum Training verwendet werden können, andererseits dienen die Datenbanken als Bewertungsgrundlage für den Vergleich neuer Algorithmen. Populäre Beispiele solcher (Personen-) Datenbanken sind: „Daimler Mono Pedestrian Detection Benchmark Dataset“ ([Enz-2009]), „INRIA Person Dataset“ ([Dal-2005]), „Caltech Pedestrian Dataset“ ([Dol-2009]) oder das mit 80 Objektkategorien deutlich größere „Common Objects in Context“ ([Lin-2014]).

Neben der Verwendung realer Daten wird in der Forschung auch der Ansatz untersucht, künstlich erzeugte Daten (z. B. aus einer Simulation) zum Training von Algorithmen des maschinellen Lernens zu verwenden [Bör-2019; Tha-2018].






Die Bilder aus den Datenbanken bestehen aus Farb- oder Graustufenbildern. Meistens wurden die darauf dargestellten Personen frontal von einer stationären Kamera aufgenommen. Der Blickwinkel der Kamera auf dem Gabelstapler ist jedoch nicht frontal, sondern schräg von oben herab. Die Konturen von Menschen unterscheiden sich daher im Anwendungsfall gegenüber den Frontalaufnahmen aus den Datenbanken. Dennoch wurden zwei Bilderdatenbanken (siehe Abbildung 5-3) exemplarisch zum Training verwendet. Der Hintergrund ist die Möglichkeit prüfen zu können, ob ein Training mit diesen Datenbanken ausreichend ist, da die Erstellung eigener Trainingsdaten sehr aufwändig ist. Dies gilt insbesondere für die Positivbilder. Negativbilder können sehr aufwandsarm produziert werden, indem z. B. ein Video vom Anwendungsgebiet erstellt wird und Sequenzen mit Personen vermieden oder entfernt werden.

Im Falle der verwendeten Daimler-Datenbank liegen die Daten getrennt als Negativ- und Positivbilder (N/P) in Graustufe vor.

Die Bilder der zusätzlich verwendeten INRIA-Datenbank liegen sowohl als markierte Bilder (M) als auch in der Form getrennter Negativ- und Positivbilder vor. Im Gegensatz zur Daimler-Datenbank sind die Bilder farbig und des Öfteren sind mehrere Personen im Bild zu sehen.

Eine Quelle für Negativbilder sind Werbevideos aus Lagern. Diese sind unter anderem auf der Video-Plattform „YouTube“ zu finden. Bei der Verwendung von Werbevideos sind zwei Dinge zu beachten: In den meisten Werbevideos sind Logos oder Beschriftungen eingefügt, diese müssen vor dem Training entfernt werden. Darüber hinaus sind in vielen Videos Menschen zu sehen, diese Sequenzen müssen zur Sammlung von Negativbildern auch entfernt werden.

Tabelle 5-2: Verwendete Bilddatenquellen für das Training und teilweise für die Evaluation der ML-Algorithmen.

Name	Quelle	Kategorie	Bildart	Beispiele
Daimler	[Enz-2009]	N/P	Graustufenbild	
INRIA	[Dal-2005]	M, N/P	Farbbild	
Versuchshalle	Eigene Aufnahmen	M, N/P	Farb-, Intensitäts-, Tiefenbild	
Industrielager	Eigene Aufnahmen	M, N/P	Farb-, Intensitäts-, Tiefenbild	
Lagerwerbevideos	YouTube	N	Farb- oder Graustufenbild	

<sup>3</sup> Aus urheberrechtlichen Gründen nicht dargestellt

Neben dem frei erhältlichen Bildmaterial wurden eigene Aufnahmen zum Training und zum Testen gemacht. Für die Aufnahmen wurde der Gegengewichtsstapler „EFG 220“ der Jungheinrich AG (siehe Abschnitt 3.2) in der Versuchshalle des Lehrstuhls fml verwendet. Die Kamera wurde rückwärtig auf dem Dach des Staplers montiert. Die Aufnahmen wurden unter Beteiligung von fünf männlichen Personen gemacht, wobei diese die Kleidung während den Aufnahmen wechselten. Dabei wurden alle Bildarten der Kinect aufgezeichnet, sodass die Aufnahmen in Farb-, Intensitäts- als auch Tiefenbildern vorliegen. Insgesamt wurden über 16.000 Bilder je Bildart aufgenommen.

Damit die Ergebnisse auch bezüglich der Anwendung in einer realen Lagerhalle validiert werden können, wurden darüber hinaus über zwei Tagesschichten hinweg Aufnahmen im realen Betrieb in der Lagerhalle eines Weiße-Ware-Herstellers gemacht (siehe Abschnitt 7.2).

Aufgrund der enormen Datenmenge von 1,8 Millionen Bildern pro Bildart wurden für die Evaluation der Personenerkennung nur Sequenzen mit Personen im Sichtfeld der Kamera ausgewählt. Diese Sequenzen umfassen insgesamt über 17.000 Bilder pro Bildart.

### **Markierung der Daten**

Die Auswertung der Personenerkennung kann entweder automatisiert oder manuell durchgeführt werden. Im Falle einer Automatisierung sind markierte Daten notwendig. Andernfalls müssen die Ergebnisse nach der Erkennung manuell in falsche, richtige oder in fehlende Erkennungen eingeteilt werden. Da jedes Bild mehrmals mit verschiedenen Einstellungen der ML-Algorithmen analysiert wird, ist die Markierung der Daten zeitlich sinnvoller.

Die Markierungen entsprechen Rechtecken, welche die Menschen auf den Bildern umschließen. Es wurden die Positions- als auch die Größendaten der jeweiligen Rechtecke gespeichert. Für die Markierung wurde ein eigenes Tool entwickelt, da die frei erhältlichen<sup>4</sup> nicht alle folgenden gewünschten Funktionen beinhalteten:

- Zuweisung der Markierungen zu (folgenden) Kategorien:
  - Mensch
  - Mensch (Aufnahme bei stillstehendem Gabelstapler)
  - Mensch, der einen Gabelstapler bedient/fährt

---

<sup>4</sup> Betrachtete Label-Tools:

LableMe: <https://github.com/CSAILVision/LabelMeAnnotationTool>;

SuperVisely: <https://github.com/supervisely/supervisely>;

LabelBox: <https://github.com/Labelbox/Labelbox#local-development-of-labeling-interfaces>

- Mensch, der sich bückt
- Mensch, der sitzt
- Mensch, der einen Hubwagen bedient
- Mensch, der einen Hubwagen zieht
- Mensch, der einen Rollbehälter zieht bzw. schiebt
- Speicherung der Verdeckung des Menschen im Bild:
  - Keine Verdeckung
  - Ungefähr zu 33 % (z. B. nur der Kopf) verdeckt
  - Ungefähr zu 50 % (nur Ober-, Unterkörper oder linke bzw. rechte Hälfte) verdeckt
  - Ungefähr zu 66 % (z. B. alles außer dem Kopf oder der Füße) verdeckt
- Interpolation der Rechtecke und der Verdeckung zwischen mehreren Bildern
- Automatische Entfernung von Markierungen in Bildarten, in welchen der Mensch nicht sichtbar ist
- Automatische Korrektur der Position und der Größe von Markierungen bei der Konvertierung in andere Bildarten

Die Zuweisung von den Kategorien (siehe Abbildung 5-4) dient der späteren Auswertung. Wenn die Markierungen kategorisiert sind, kann beurteilt werden, inwieweit z. B. sitzende Menschen erkannt werden, stehende aber nicht. Analog dient der Grad der Verdeckung der späteren Auswertung.

Die Interpolation der Rechtecke hat den Zweck die manuelle Markierung bei Serienaufnahmen effizienter zu gestalten. Dazu werden in einem Bild alle Menschen markiert. Die nächsten Bilder werden ausgelassen und anschließend z. B. das zehnte darauffolgende Bild markiert. Die Markierungen dazwischen werden sowohl in der Größe des Rechtecks als auch bezüglich der Position interpoliert. Darüber hinaus wird der Verdeckungsgrad interpoliert. Wenn z. B. eine Person in Bild 1 vollständig sichtbar ist und diese ist in Bild 10 zu 66 % verdeckt, wird in Bild 5 eine Verdeckung von 33 % angenommen.



Abbildung 5-4: Screenshots des Markierungstools. Links: Verwendung von Kategorien – Mensch (grün) und Gabelstaplerfahrer (pink), rechts: Zuweisung der Verdeckung – 33 % Verdeckung → grau.

Die automatische Entfernung von Markierungen in verschiedenen Bildarten dient ebenso einer effizienteren Markierung. Grundsätzlich müssten die Markierungen in allen Bildarten separat gemacht werden, da aufgrund der unterschiedlichen Sensoren Menschen auf manchen Bildarten zu sehen sind und auf anderen nicht oder nur teilweise. In letzterem Fall wird die Größe der Markierung bis zum Bildrand angepasst. Darüber hinaus ist eine Größen- und Positionskorrektur der Markierungen bei der Konvertierung in andere Bildarten notwendig, da die Rechtecke der Menschen im Bild aufgrund der unterschiedlichen Sensorparameter in den Pixelkoordinaten nicht übereinstimmen.

Das größte Sichtfeld hat bei der verwendeten Kamera wegen des Öffnungswinkels das Farbbild. Aufgrund dessen wurden Menschen nur auf diesem markiert. Für die restlichen Bildarten (Intensitäts- und Tiefenbild) wurden die Markierungen durch das Tool automatisch angepasst.

Abbildung 5-5 zeigt die automatische Anpassung der Markierungen. Die linke Person auf dem Farbbild ist auf dem Tiefenbild nicht mehr zu sehen, daher wurde diese entfernt. Im Intensitätsbild ist sie nur teilweise zu sehen, daher wurde die Größe des Rechtecks verkleinert. Zusätzlich wurde die Position des Rechtecks angepasst.

Zur Überprüfung, ob eine Markierung eine Person im Tiefenbild sichtbar ist, wird im Tiefenbild der Bereich innerhalb der Markierung analysiert. In diesem müssen mindestens 20 % der Pixel valide Distanzdaten enthalten. Sind es weniger, wird die Markierung entfernt. Der Wert von 20 % wurde empirisch durch die Auswertung von etwa 2.000 Bildern bestimmt. Als Kriterium diente eine manuelle Sichtung, ob der Mensch bzw. Teile von ihm auf dem Bild als menschlich identifiziert werden konnten.



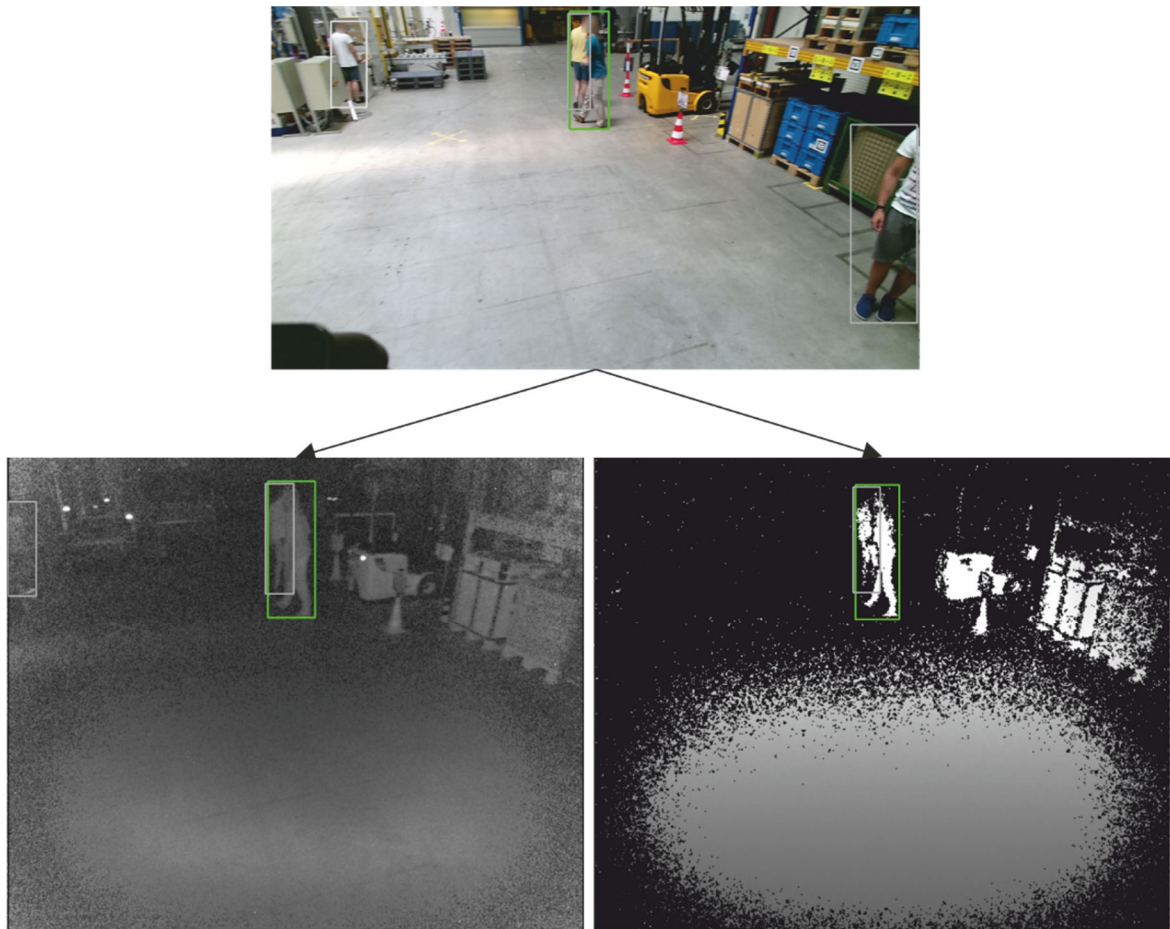


Abbildung 5-5: Ergebnis der automatischen Markierungsanpassung für das Intensitäts- bzw. Tiefenbild. Dargestellt sind die Bilder eines Aufnahmezeitpunktes. Oben: Farbbild, unten links: Intensitätsbild, unten rechts: Tiefenbild.

Bei der Verwendung von ML-Algorithmen werden üblicherweise 80 % der Daten zum Training (und zur Validierung) verwendet und die restlichen 20 % zum Testen [Lea-2016]. Diese Aufteilung wurde auch für die eigenen Aufnahmen verwendet. Es wurde dabei darauf geachtet, dass im Trainings- und Testdatensatz keine Bilder aus den gleichen Videos sind. Aus diesem Grund wurden nicht einzelne Bilder, sondern einzelne Videos in Trainings- und Testvideos unterteilt. Das Training des DL-Algorithmus „YoloV3-Detektor“ erfolgte in Zusammenarbeit mit *Daßler* im Rahmen einer Masterarbeit [Daß-2018].

### Kollisionsgefahrerkennung

Die Testdaten für die Kollisionsgefahrerkennung wurden in der Versuchshalle des Lehrstuhls fml aufgenommen. Für diesen Fall sind keine Daten aus dem industriellen Umfeld notwendig, da die Kollisionsgefahrerkennung unabhängig von der Umgebung



ist. Alle Testfälle aus dem Versuchsplan aus Tabelle 5-1 wurden zehnmal durchgeführt<sup>5</sup>.

Als Hindernisse dienten je nach geforderter Hindernishöhe gestapelte KLT (Größe 594 mm × 396 mm × 280 mm). Für den Test von Kollisionen mit einem Menschen wurde eine Schaufensterpuppe (siehe Abbildung 5-3) (Größe: 1,90 m) verwendet.

In den Videos mit Kollisionen wurde zusätzlich das Bild, in welchem die Kollision stattgefunden hat, markiert. Anhand dieser Markierung ist es über den Zeitstempel des jeweiligen Bildes möglich herauszufinden, wie viele Sekunden vor der Kollision das System die Gefahr erkannt hat.

---

<sup>5</sup> Beim ersten Testfall (Szenario 1a) wurde im Nachgang ein Aufzeichnungsfehler in einem Video entdeckt. Daher wurden in der späteren Auswertung nur neun Videos ausgewertet.

## 5.5 Fazit

In diesem Kapitel wurden die Maßnahmen beschrieben, die zur folgenden Evaluation des Systems notwendig sind. Zuerst wurden Faktoren identifiziert, welche die Funktionalität des Systems beeinflussen können. Diese werden in der darauffolgenden Definition der Testfälle berücksichtigt. Dabei handelt es sich einerseits um externe Faktoren bspw. das Umgebungslicht, welche die Rohdaten der Kamera verschlechtern können. Andererseits gibt es einige Faktoren, deren Einfluss abhängig vom verwendeten Algorithmus ist. So können Kleidung, Pose oder Größe eines Menschen die Wirksamkeit der Personenerkennung beeinflussen. Bei der Kollisionsgefahrerkennung sind vor allem die Bewegungsvektoren und Dimensionen des Staplers sowie etwaiger Hindernisse entscheidend für die korrekte Berechnung.

Aus letzterem Punkt folgt die Aufstellung zu prüfender Verkehrssituationen zwischen Staplern und Hindernissen. Bei stationären Hindernissen sind nur die Bewegungsfreiheitsgrade des Staplers von Interesse. Dieser hat die Möglichkeit sich in eine Richtung zu bewegen, eine Kurve zu fahren oder sich um die eigene Achse zu drehen. Gleiches gilt für einen Menschen, allerdings ist bei diesem die Drehung um die eigene Achse irrelevant.

Auf Basis der Einflussfaktoren und Verkehrssituationen wurden Testszenarien für die Personen- und Kollisionsgefahrerkennung definiert. In den Szenarien der Personenerkennung steht insbesondere die Änderung des äußeren Erscheinungsbilds, z. B. durch die Änderung der Pose, eines Menschen im Fokus. Die Szenarien der Kollisionsgefahrerkennung weisen u. a. unterschiedliche geometrische Ausprägungen des Hindernisses, dessen Standort und Abstand zum Stapler auf. Überdies wurden die unterschiedlichen Bewegungsmöglichkeiten (Fahren, Drehen) in den Szenarien berücksichtigt.

Damit das System getestet werden konnte, war die Aufnahme von Testdaten notwendig. Die Aufnahme der Testszenarien hat den Vorteil, dass verschiedene Algorithmen mit exakt den gleichen Daten verglichen werden können. Die Alternative, das System im laufenden Betrieb zu validieren, hätte mehrere Nachteile; der entscheidende ist die praktisch unmögliche Reproduzierbarkeit derselben Versuchsbedingungen während der Durchführung aller Testszenarien. Daher wurden sowohl im universitären als auch im industriellen Umfeld Aufnahmen der Testszenarien gemacht.

Die Aufnahmen wurden markiert, sodass diese automatisch ausgewertet werden können. Für die Markierung der aufgenommenen Daten wurde ein Tool entwickelt, welches die Tätigkeit effizienter gestaltet. Im Falle der Personenerkennung wurden alle

Menschen auf den Aufnahmen markiert und kategorisiert. Bei der Kollisionsgefahrerkennung wurde der Kollisionszeitpunkt markiert.

Da die ML-Algorithmen der Personenerkennung nicht ohne vorheriges Training funktionieren, wurden zusätzlich separate Trainingsdaten an unterschiedlichen Orten erzeugt.

## 6 Evaluation einzelner Algorithmen

---

Im vorigen Kapitel wurden die Testszenarien definiert und die Daten für das Training bzw. für die Auswertung beschrieben. In diesem Kapitel folgt die Anwendung der Algorithmen auf die Testdaten.

Im ersten Abschnitt wird die Evaluation der Personenerkennung beschrieben. Dazu wird die Auswertungsmethode vorgestellt und anschließend werden die Ergebnisse zur Personenerkennung mittels SVM und neuronalem Netz dargelegt. Die Auswertung der SVM enthält die Ergebnisse zahlreicher Variationen im Training, in der Bildvorverarbeitung und in den SVM-Parametern. Beim neuronalen Netz sind in der Hinsicht nur wenig Variationen möglich.

Im zweiten Abschnitt wird eine Methode zur Auswertung der Kollisionsgefahrerkennung vorgestellt und mithilfe dieser entschieden, welche Methoden des optischen Flusses sich zur Kollisionsgefahrerkennung eignen. Außerdem werden verschiedene Ansätze gezeigt, um das Ergebnis zu verbessern. Abschließend werden die Ergebnisse der Evaluation resümiert und die besten Algorithmen sowie Konfigurationen für die Evaluation des Gesamtsystems ausgewählt.

### 6.1 Auswertung der Personenerkennung

Ein wichtiges Unterscheidungsmerkmal des entwickelten Systems zu erhältlichen Kollisionswarnsystemen ist die automatische Erkennung von Personen in der Fahrbahn des Staplers. Die zuverlässige Funktionsweise ist daher unabdinglich. Kein öffentlich zugänglicher Algorithmus funktioniert dabei perfekt im Erkennen von Personen: Falscherkennungen und nicht erkannte Personen sind daher trotz der immensen Entwicklung in diesem Bereich weiter üblich. Generell korreliert die Anzahl an falschen Erkennungen mit denen richtiger.

Der erste Abschnitt handelt von der Methode, wie trotz der Korrelation zwischen falschen und richtigen Erkennungen ohne Kenntnis der maximal zulässigen Anzahl an Falscherkennungen eine Bewertung bzw. ein Vergleich mehrerer Algorithmen möglich ist. Daraufhin werden die Ansätze beschrieben, welche zur Erreichung eines optimalen Ergebnisses mit dem Einsatz von SVM führen sollen, ebenso wie die Ansätze zum Einsatz eines neuronalen Netzes.

### 6.1.1 Auswertungsmethode

Die Auswertung von ML-Algorithmen ist schwierig, weil durch die Nutzung verschiedener Grenzwerte für die minimal notwendige Konfidenz unterschiedliche Ergebnisse erzeugt werden. Wird ein niedriger Schwellwert verwendet, werden zwar viele Personen erkannt, allerdings werden bei den Ergebnissen auch vermehrt falsche Erkennungen enthalten sein. Ist der Schwellwert zu hoch, werden nicht nur Falscherkennungen, sondern auch richtige Erkennungen vermindert.

Neben der Problematik des richtigen Schwellwerts ist die Entscheidung, wann eine Erkennung als richtig oder falsch einzustufen ist, schwierig. Wenn bspw. nur die Hälfte einer Person erkannt wurde, stellt sich die Frage, ob die Erkennung als richtig oder falsch einzuordnen ist.

2012 stellten *Dollar et al* in „Pedestrian Detection: An Evaluation of the State of the Art“ eine Methode zur Bewertung vor, welche die zwei Problematiken beseitigen [Dol-2012]. Diese hat sich bezüglich der Bewertung von Objekterkennungsalgorithmen mittlerweile bewährt und etabliert.

Anhand der Abbildung 6-1 soll die Vorgehensweise aufgezeigt werden. Beide Koordinatenachsen werden logarithmisch dargestellt. Auf der Abszisse wird die Anzahl der Falscherkennungen (falsche Positive) pro Bild (engl.: „false positives per image“ (fppi)) aufgetragen. Auf der Ordinate kann der prozentuale Anteil fehlender Erkennungen (engl.: „miss rate“) abgelesen werden. Eine Linie beschreibt die Leistungsfähigkeit der Personenerkennung in einer bestimmten Konfiguration. Diese Linie wird durch mehrere Stützpunkte erzeugt, welche wiederum anhand verschiedener Schwellwerte (siehe a)) zur Konfidenz (oder Erkennungsakzeptanz) entstehen. Ein Beispiel soll dies verdeutlichen: Beim ersten Durchlauf (1) wird ein Schwellwert von 0,4 verwendet; der Detektor erkennt eine Person mit einer Konfidenz von 0,45. Die Erkennung wird daher als richtige positive Erkennung gewertet. Wenn die Konfidenz unter 0,4 wäre oder keine Person in diesem Bereich ist, würde es als falsche positive Erkennung gewertet. Dieser Vorgang wiederholt sich über alle Bilder hinweg. Die Falscherkennungen pro Bild (engl.: „false positives per image“ (fppi)) errechnen sich anschließend durch den Quotienten aller falschen Erkennungen  $n_{falsche\ Positive}$  und der Anzahl analysierter Bilder  $n_{Bilder}$  (10). Die miss rate wird aus dem Komplementärwert des Verhältnisses von richtigen Positiven  $n_{richtige\ Positive}$  und der Anzahl aller Markierungen bzw. zu erkennenden Personen  $n_{Markierungen}$  berechnet (11).

$$fppi: \frac{n_{falsche\ Positive}}{n_{Bilder}} \quad (10)$$

$$\text{miss rate: } 1 - \frac{n_{\text{richtige Positive}}}{n_{\text{Markierungen}}} \quad (11)$$

Im zweiten Durchlauf (2) wird ein Schwellwert von 0,5 verwendet. Die erste Person wird wieder mit einer Konfidenz von 0,45 erkannt, aufgrund des höheren Schwellwertes wird sie in diesem Fall aber als richtige positive Erkennung gewertet. Aus der Variation des Schwellwertes ergeben sich üblicherweise monoton fallende Kurven.

Zum Vergleich mit anderen Konfigurationen (siehe b)), z. B. verschiedene Parameter-einstellungen für die SVM, eignet sich die Darstellung in einem Diagramm. Je näher eine Linie am Koordinatenursprung ist, desto besser ist die Erkennungsleistung mit der jeweiligen Konfiguration.

In den folgenden Auswertungen werden in der Legende verwendete SVM-Charakteristika angegeben:

- Bilderart der Eingangsdaten (siehe c)): „**color**“ (Farbbild), „**ir**“ (Intensitätsbild) oder „**depth**“ (Tiefenbild).
- **amr** (siehe d)): Gemittelte miss rate. Diese repräsentiert das normierte Integral (12) von  $fppi = 0,01$  bis  $0,1$  (im Diagramm gelb markierter Bereich). Wenn kein Wert innerhalb des Integrationsbereiches vorliegt, wird dies mit „n/a“ gekennzeichnet.
- **afps** (siehe e)): Mittlere Bildrate in Bildern pro Sekunde (engl.: „frames per second“ oder fps)

$$amr = \frac{1}{0,09} \cdot \int_{0,01}^{0,1} mr(fppi) \quad (12)$$

## 6 Evaluation einzelner Algorithmen

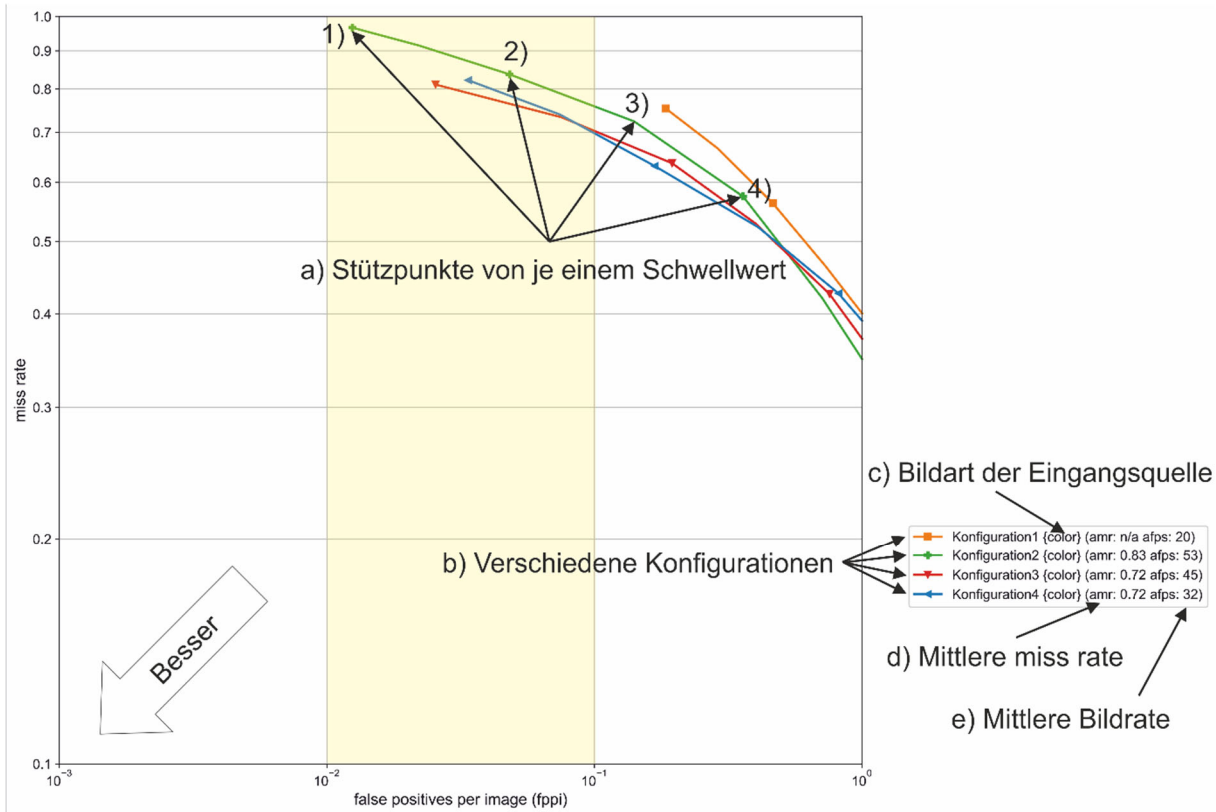


Abbildung 6-1: Grafische Auswertung nach Dollar et al.

Zur Entscheidung, ob es sich bei einer Erkennung um eine falsch oder richtig positive handelt, werden in der Vorgehensweise die Rechtecke der Markierung und der Erkennung verglichen. Im Detail wird die Fläche des Rechtecks (in Pixeln) der Schnittmenge in das Verhältnis zur Fläche des Rechtecks der Vereinigungsmenge gesetzt (siehe Abbildung 6-2). Ist dieser Wert  $r_a$  größer als 0,5, gilt die Erkennung als „richtig positiv“, ansonsten als „falsch positiv“:

$$r_a = \frac{A_n}{A_U} > 0,5 \quad (13)$$

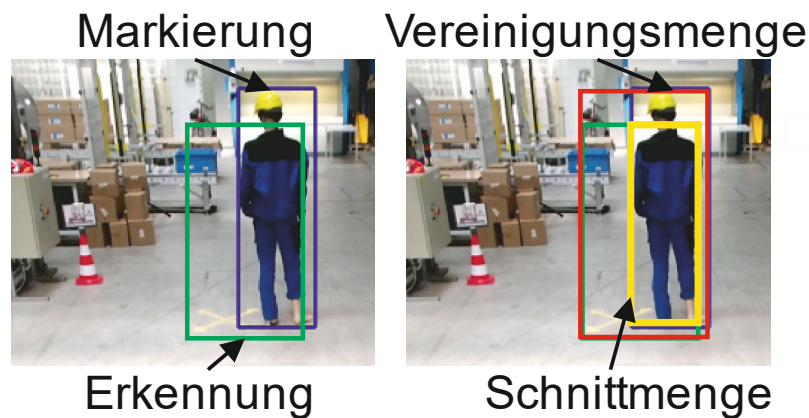


Abbildung 6-2: Darstellung der Bildung der Rechtecke aus Schnitt- und Vereinigungsmenge erkannter und markierter Rechtecke. Links: Markierung (blau) und Erkennung (grün), rechts: Rechtecke der Schnitt- (rot) und Vereinigungsmenge (gelb).

Im Folgenden werden nur die wichtigsten Ergebnisse grafisch dargelegt. Die vollständigen Auswertungen inklusive der Rohdaten und anderen Messwerten („recall“, „precision“) sind im Anhang A.1 zu finden.

### 6.1.2 Ergebnisse – Support-Vector-Machine

Für das Training von ML-Algorithmen gibt es keine Vorgehensweise, welche das beste Ergebnis garantiert. Daher wurden die möglichen Variablen identifiziert und getrennt voneinander variiert.

Die ersten Variationsmöglichkeiten sind beim Training möglich. Hierbei sind die Größe des Detektors, die Trainingsdaten und diverse SVM-Parameter veränderbar. Weitere Variationen sind während der Detektion möglich, insbesondere trifft dies auf die Vorverarbeitung des zu analysierenden Bildes zu.

Die folgenden Auswertungen konnten nicht sukzessiv durchgeführt werden, sondern beruhen auf der iterativen Durchführung aller Testläufe. Zum Teil werden in der Konfiguration daher auch Ergebnisse von Voruntersuchungen, wie z. B. die notwendige Anzahl an Positivbildern für das Training, verwendet.



Folgende (Hyper-)Parameter wurden, sofern nicht anders angegeben, bei allen Trainings standardmäßig verwendet:

Tabelle 6-1: Standard-Parameter für SVM-Training.

Parameter	Wert
<b>SVM-Typ</b>	EPS-SVR
<b>SVM-Kernel</b>	Linear
<b>C</b>	Auto (min: 0,1;max: 500;step: 5)
<b>P</b>	Auto (min: 0,001;max: 1,0;step: 3)
<b>Bildart</b>	Farb-, Intensitäts-, Tiefenbild
<b>Anzahl der Positivbilder</b>	7.000
<b>Faktor für Negativbilder</b>	2,4

Es wurde die Kombination aus einer Regressions-SVM mit einem linearen Kernel verwendet, da Vorabuntersuchungen mit anderen SVM-Typen (C-SVC, NU-SVC, NU-SVR) sowie anderen Kernen (polynomiell, sigmoid, „radial basis function“) zeigten, dass diese für den Anwendungsfall nicht nutzbar sind, da sie einerseits mehr Rechenaufwand als z. B. die eingesetzten DL-Netze hatten. Andererseits war die Klassifikation signifikant schlechter. Für die Hyperparameter C und P wurden während des Trainings Vergleichsprüfungen (engl.: „cross validations“) mit den angegebenen Grenzen durchgeführt und das jeweils beste Ergebnis gespeichert. Üblicherweise wurden die Trainings jeweils mit 7.000 Positivbildern und der entsprechend 2,4-fachen Anzahl an Negativbildern durchgeführt.

Folgende Einstellungen wurden, sofern nicht anders angegeben, als Standard bei der Detektion verwendet:

Tabelle 6-2: Standardeinstellungen des Systems für die Parametervariation.

Parameter	Wert
<b>Vorverarbeitung</b>	
<b>Farbbildkalibrierung</b>	An: Eigene Methode (siehe S. 63)
<b>Intensitätsbildkonvertierung</b>	An: Eigene Methode (siehe S. 64 f.)
<b>Skalierungsfaktor</b>	Farbbild: 0,4 Intensitäts-/Tiefenbild: 0,8
<b>Normalisierung</b>	Aus
<b>Egalisierung</b>	Aus
<b>Segmentierung</b>	Aus
<b>Weichzeichnen</b>	Aus
<b>Interpolation fehlender Daten im Tiefenbild</b>	Aus
<b>SVM-Parameter</b>	
<b>gr_threshold</b>	5
<b>h_scale</b>	1,05
<b>n_levels</b>	15
<b>Automatische Berechnung von n_levels</b>	Aus
<b>nbins</b>	9
<b>win_stride_width / block_stride_width</b>	8
<b>win_stride_height/ block_stride_height</b>	8
<b>gamma correction</b>	Aus

### Variation des Trainings – Detektorgröße

Die Detektorgröße beschreibt die Größe des Bildes, welches von der SVM analysiert, sprich klassifiziert wird. Sie legt damit auch die Mindestgröße zu erkennender Objekte fest. Wurde ein Detektor auf die Größe von 64 x 128 Pixel trainiert, können in einem Eingangsbild nur Objekte erkannt werden, welche mindestens diese Größe aufweisen. Größere Objekte sind durch das „Sliding Window“-Verfahren (siehe S. 30) ebenfalls möglich. Aufgrund von Limitierungen, welche durch die Verwendung von HOG-Features entstehen, können nur Vielfache von 16 als Detektorbreite genutzt werden (siehe Tabelle 6-3). Aufgrund dessen, dass ein Mensch auf einem Bild mit einer Breite von 16 bzw. 32 Pixeln kaum erkennbar ist, wurden nur Detektorbreiten von mindestens 48 Pixeln untersucht. Die Detektorhöhe entspricht – analog zum Größenverhältnis von Menschen (Körperhöhe zu -breite) – dem Doppelten der Detektorbreite. Als Trainingsdaten wurden alle Bildarten des selbst erstellten Trainingsdatensatzes verwendet. Der Skalierungsfaktor für das Eingangsbild wurde bei der Analyse entsprechend der Detektorgröße angepasst, sodass für jeden Detektor die Möglichkeit besteht, die gleichen Personen zu erfassen.

Grundsätzlich können bei einem größeren Detektor mehr Merkmale extrahiert werden, dementsprechend sind mehr Informationen für die darauffolgende Analyse vorhanden. Daher wäre prinzipiell ein großer Detektor einem kleineren vorzuziehen. Allerdings steigt mit zunehmender Detektorgröße der Rechenaufwand und die Erkennungsleistung muss nicht ebenso steigen. Letztlich ist entscheidend, welche Auflösung und Größe die Menschen auf dem zu analysierenden Bild aufweisen.

Tabelle 6-3: Trainingskonfigurationen für Variation der Detektorgröße.

Parameter	Werte
Detektorgröße	48 x 96, 64 x 128, 96 x 192, 128 x 256
Skalierungsfaktor	0,40, 0,53, 0,80, 1,06
Trainingsdaten - Positivbilder	Eigener Trainingsdatensatz („rec“)
Trainingsdaten - Negativbilder	Eigener Trainingsdatensatz („rec“)

Die Ergebnisse der Variation der Detektorgröße zeigt folgende Abbildung:

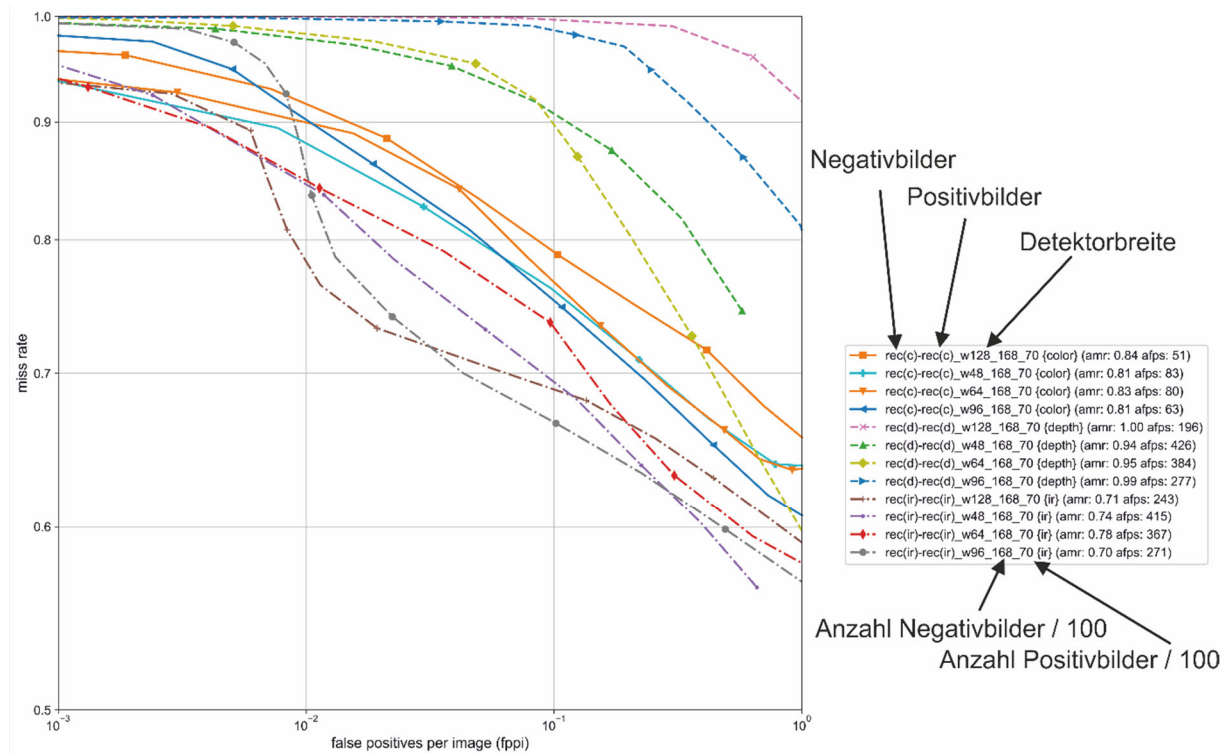


Abbildung 6-3: Ergebnis der Variation der Detektorgröße.

Die Nomenklatur der Trainingskonfigurationen in der Legende ist folgende:

Negativbildquelle(trainierte Bildart<sup>6</sup>)-Positivbildquelle(trainierte Bildart)\_wDetektorbreite\_Anzahl der Negativbilder\_Anzahl der Positivbilder {analysierte Bildart} (amr: durchschnittliche miss rate afps: durchschnittliche Bildrate)

Das wichtigste Ergebnis ist der große Unterschied zwischen den verschiedenen Bildarten. Die Detektoren, welche auf Intensitätsbilder trainiert wurden (Strich-Punkt-Linien), waren deutlich besser als die auf Farbbilder (durchgezogene Linien) und Tiefenbilder (gestrichelte Linien) trainierten Detektoren.

Für die Variation der Detektorgröße ergab sich kein eindeutiges Bild. Beim Intensitätsbild war sowohl eine Detektorbreite von 96 (rot) und 128 (blau) Pixeln am besten. Bei der Verwendung von Farbbildern waren Detektorbreiten von 48 (orange) und 96 (grün) Pixeln besser, jedoch ist der Unterschied zu den restlichen Detektorbreiten relativ gering. Die auf Tiefenbildern trainierten Detektoren haben mit einer Breite von 48

<sup>6</sup> color=Farbbild, depth=Tiefenbild, ir=Intensitätsbild

(orange) und 64 (dunkelblau) Pixeln die besten Ergebnisse geliefert. Die beste mittlere miss rate lag bei diesen mit 94 % jedoch weitab von einem praxistauglichen Bereich.

Ein Grund, dass nicht eine Detektorbreite bei allen Bildarten als beste hervorging, könnten die Qualitätsunterschiede der jeweiligen Bildarten sein. Das Farbbild ist deutlich höher aufgelöst und weist ein geringeres Rauschen als das Tiefen- oder Intensitätsbild auf. Aufgrund der höheren Auflösung wird es bei der Bildanalyse umso stärker verkleinert. Das hat zur Folge, dass die resultierenden Detektorbilder von z. B. 64 x 128 Pixel oder 96 x 192 Pixel keine großen qualitativen Unterschiede aufweisen. Der qualitative Unterschied der Bilder bei einer Skalierung auf 64 x 128 Pixel oder 96 x 192 Pixel ist im Falle des Tiefen- oder Intensitätsbilds deutlich größer, da die Ausgangsauflösung nur etwa ein Viertel des Farbbilds beträgt.

### **Variation des Trainings – Bildquellen für Positivbilder**

Die Wahl der Quelle für die Trainingsdaten ist abhängig von den Daten, die später analysiert werden sollen. Anhand dieser Daten lernt der ML-Algorithmus die Klassifikation gesuchter Objekte und des Hintergrunds. In Abschnitt 5.4 wurden bereits die in Frage kommenden Datenquellen für die positiven Beispiele (Menschen) beschrieben. Der klare Vorteil der Nutzung von Bildern aus frei verfügbaren Datenbanken ist die Einsparung der sehr aufwändigen Markierung von Personen auf den Bildern.

Mit der Verwendung von Bildern dieser Datenbanken wurde überprüft, ob diese bereits verfügbaren Daten für den Anwendungsfall ausreichen. Für jede Datenquelle wurden mehrere SVM (siehe Tabelle 6-4) trainiert. Da die Bilder der Datenbanken nur als Farb- bzw. Graustufenbilder vorliegen, konnten folglich auch nur diese Bildarten trainiert werden. Die Auswertung der Detektion wurde für alle Bildarten durchgeführt. Als Negativbilder wurden jeweils die eigenen Aufnahmen verwendet. Die Detektorgrößen wurden je nach Bildart auf Basis der Erkenntnisse des vorherigen Versuchs ausgewählt.

Tabelle 6-4: Trainingskonfigurationen für Variation der Positivbilder.

Parameter	Werte
<b>Detektorgröße</b>	Farbbild: 48 x 96
	Tiefenbild: 64 x 128
	Intensitätsbild: 96 x 192
<b>Trainingsdaten - Positivbilder</b>	Daimler, INRIA, Daimler&INRIA gemischt ( <i>daimler_inria</i> ), Eigener Trainingsdatensatz ( <i>rec</i> )
<b>Trainingsdaten - Negativbilder</b>	Eigener Trainingsdatensatz ( <i>rec</i> )

Die Ergebnisse in Abbildung 6-4 zeigen, dass mit Ausnahme des Tiefenbilds die eigenen Trainingsdaten für die jeweils trainierte Bildart das beste Ergebnis liefern.

Beim Tiefenbild ist zwar der mit Daimler-Bildern trainierte Detektor besser als der mit eigenen Daten trainierte, aber mit einer durchschnittlichen miss rate von 85 % nicht im angestrebten Bereich von unter 75 %. Deshalb wird in den folgenden Versuchen das Tiefenbild nicht weiterverfolgt.

Im Farbbild konnte mit einem Detektor (*rec(c)-inria\_w48\_168\_70*), welcher mit Datenbankbildern trainiert wurde, ein nahezu gleichwertiges Ergebnis im Vergleich zu dem mit eigenen Daten trainierten Detektor erreicht werden. Dementsprechend wäre für die Personenerkennung in Farbbildern trotz der unterschiedlichen Perspektiven in den Trainingsbildern die Erstellung eines eigenen Datensatzes nicht notwendig.

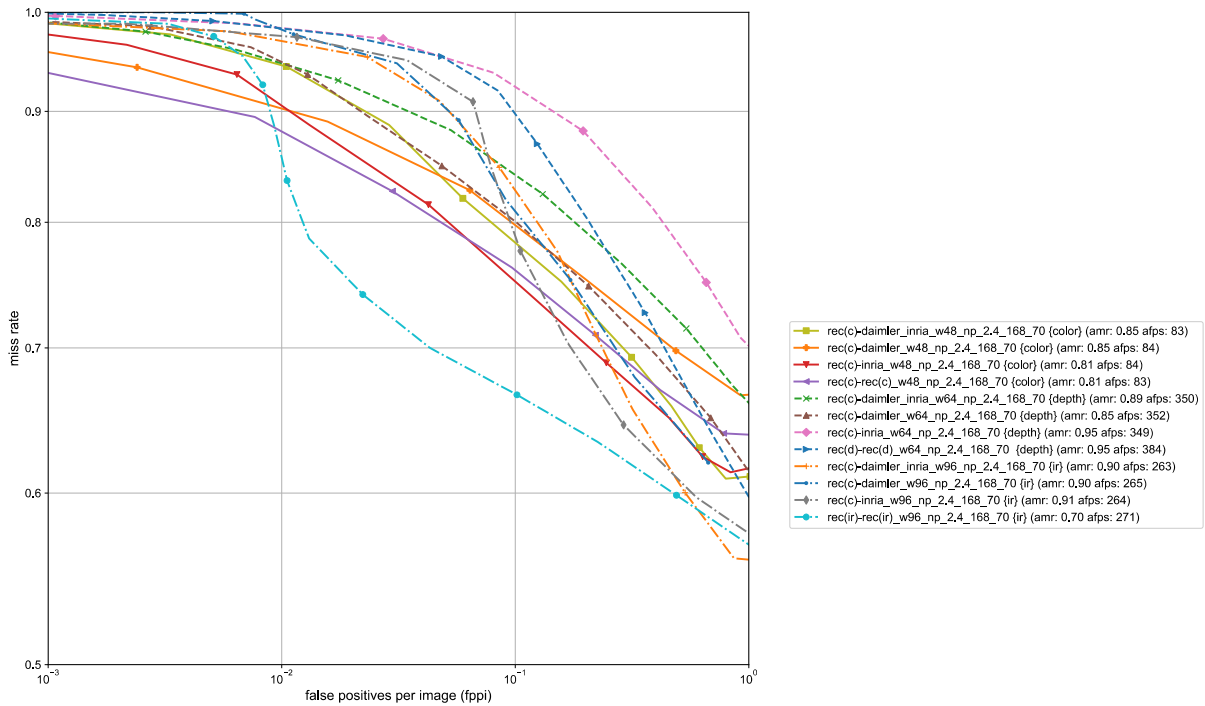


Abbildung 6-4: Ergebnis der Variation der Bildquellen für die Positivtrainingsbilder.

### Variation des Trainings – Bildquellen für Negativbilder

Wie die Variation der Positivbilder hat auch die Wahl der Negativbilder einen Einfluss auf die Detektion. Die Erzeugung von Negativbildern ist bei weitem aufwandsärmer als die der Positivbilder, da nur darauf geachtet werden muss, dass keine Personen auf den Bildern zu sehen sind. Trotzdem wurde getestet, ob mit der Beimischung oder alleinigen Verwendung von YouTube-Werbevideos aus Lagern das Training durchgeführt werden kann. Etwaige Logos oder sonstige Schriftzüge wurden aus dem Material entfernt.

Tabelle 6-5: Trainingskonfigurationen für Variation der Negativbilder.

Parameter	Werte
<b>Detektorgröße</b>	Farbbild: 48 x 96 Intensitätsbild: 128 x 254
<b>Trainingsdaten - Positivbilder</b>	Eigener Trainingsdatensatz
<b>Trainingsdaten - Negativbilder</b>	Eigener Trainingsdatensatz, Eigener Trainingsdatensatz & YouTube-Video, YouTube-Videos

Die Auswertung in Abbildung 6-5 zeigt, dass sowohl die Beimischung von YouTube-Lagervideos (orange) als auch deren alleinige Verwendung (grün) das Ergebnis verschlechtern. Der negative Effekt ist mit einem Anstieg von 0,3 Prozentpunkten bei der amr aber relativ gering. Daher könnten auch fremde Quellen als Negativbilder verwendet werden.

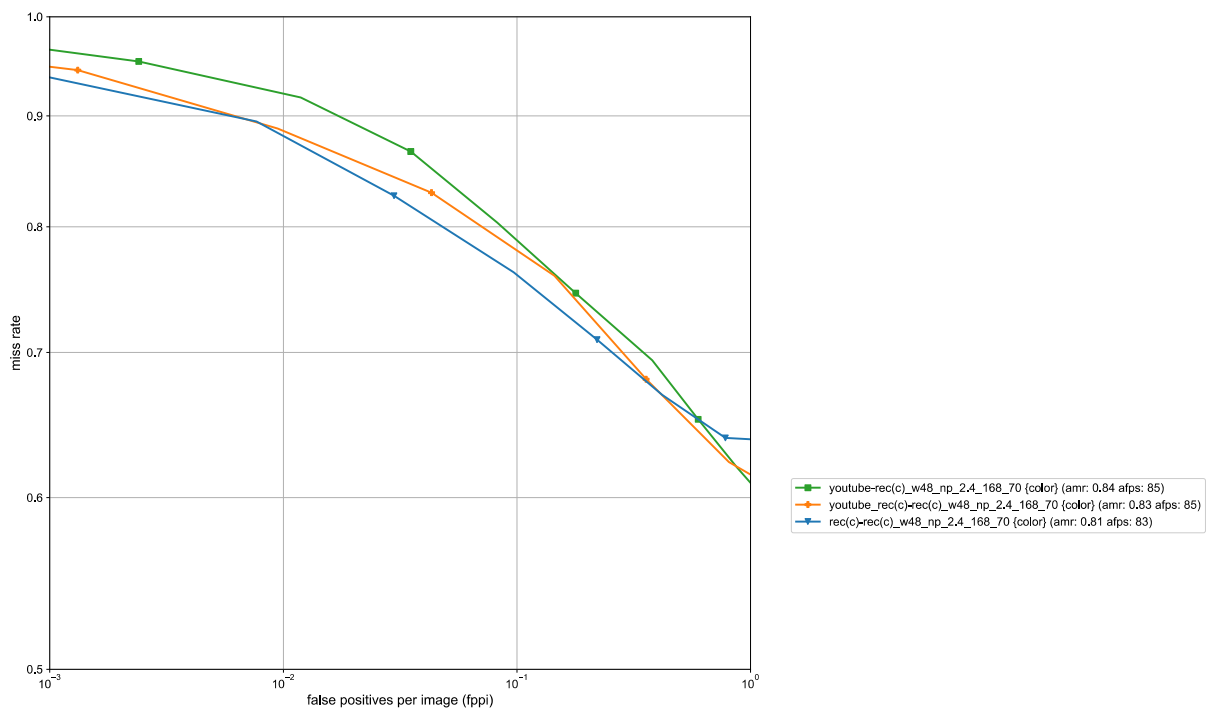


Abbildung 6-5: Ergebnis der Variation der Bildquellen für die Negativtrainingsbilder.

### Variation des Trainings – Anzahl der Trainingsbilder

Die Anzahl der Trainingsbilder beeinflusst die Detektionsqualität dahingehend, dass bei wenigen Trainingsbildern der ML-Algorithmus eventuell nicht ausreichend Daten zur Verfügung hat, um die Merkmale eines Objekts ausreichend zu lernen.



Bei einer großen Anzahl an Trainingsbildern kann das sogenannte „Overfitting“ auftreten. In diesem Fall verliert der Algorithmus die Abstraktionsfähigkeit und lernt nur die Objekte im Trainingsdatensatz zu erkennen, wobei die Detektion in anderen Bildquellen deutlich schlechter wird. Daher wurde das Verhalten der SVM bei der Verwendung unterschiedlicher Anzahl an Bildern untersucht. Das Verhältnis von Positiv- zu Negativbildern wurde beibehalten.

Tabelle 6-6: Trainingskonfigurationen für Variation der Anzahl der Trainingsbilder.

Parameter	Werte
<b>Detektorgröße</b>	Farbbild: 48 x 96 Intensitätsbild: 128 x 254
<b>Anzahl positiver Bilder</b>	1.500, 3.500, 7.000, 10.000, 15.000
<b>Trainingsdaten - Positivbilder</b>	Eigener Trainingsdatensatz
<b>Trainingsdaten - Negativbilder</b>	Eigener Trainingsdatensatz

Die Auswertung in Abbildung 6-6 zeigt, dass sowohl für das Farb- (durchgezogene Linien) als auch für das Intensitätsbild (gestrichelte Linien) die beste Detektion bei 3.500 Trainingsbildern (lila bzw. rosa Linie) erreicht wurde. Die Verdoppelung auf 7.000 Trainingsbilder führte zu einer minimalen Verschlechterung der Erkennung, eine weitere Erhöhung der Trainingsbilderanzahl verschlechterte in beiden Bildarten das Ergebnis erheblich.

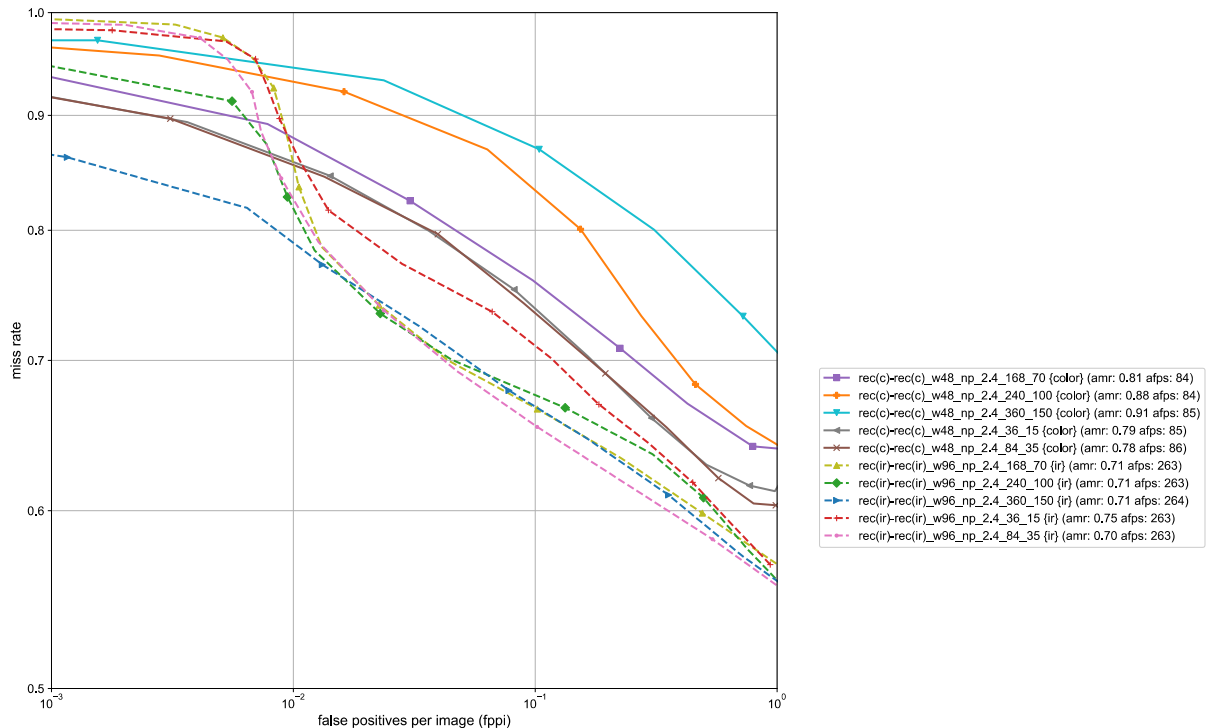


Abbildung 6-6: Ergebnis der Variation der Anzahl an Positivtrainingsbildern.

### Variation des Trainings – Verhältnis negativer zu positiven Bildern

Eine weitere Stellgröße beim Training ist das Verhältnis von negativen zu positiven Bildern bzw. die Anzahl an Negativbildern. Analog zur Anzahl an Positivbildern ist im Vorhinein schwer abzuschätzen wie viele Bilder notwendig sind.

Tabelle 6-7: Trainingskonfigurationen für Variation des Verhältnisses zwischen Negativ- zu Positivbildern.

Parameter	Werte
<b>Detektorgröße</b>	Farbbild: 48 x 96 Intensitätsbild: 128 x 254
<b>Anzahl positiver Bilder</b>	3.500
<b>Trainingsdaten - Positivbilder</b>	Eigener Trainingsdatensatz
<b>Trainingsdaten - Negativbilder</b>	Eigener Trainingsdatensatz
<b>Faktor für Negativbilder</b>	0,8, 1,2, 1,6, 2,0, 2,4, 2,8

Eine signifikante Tendenz zum besten Verhältnis trainierter Negativ- zu Positivbildern ist der Auswertung in Abbildung 6-7 nicht zu entnehmen, allerdings unterscheidet sich

die Erkennungsleistung auch hierbei um bis zu 10 %. Die doppelte Anzahl an Negativbildern (np\_2.0) ist nach der Auswertung ein guter Ausgangswert um das optimale Verhältnis herauszufinden. Im Falle des Farbbilds liegt es bei 2,0 (rec(c)-rec(c)\_w48\_np\_2.0\_70\_35), beim Intensitätsbild bei 2,4 (rec(ir)-rec(ir)\_w96\_np\_2.4\_84\_35).

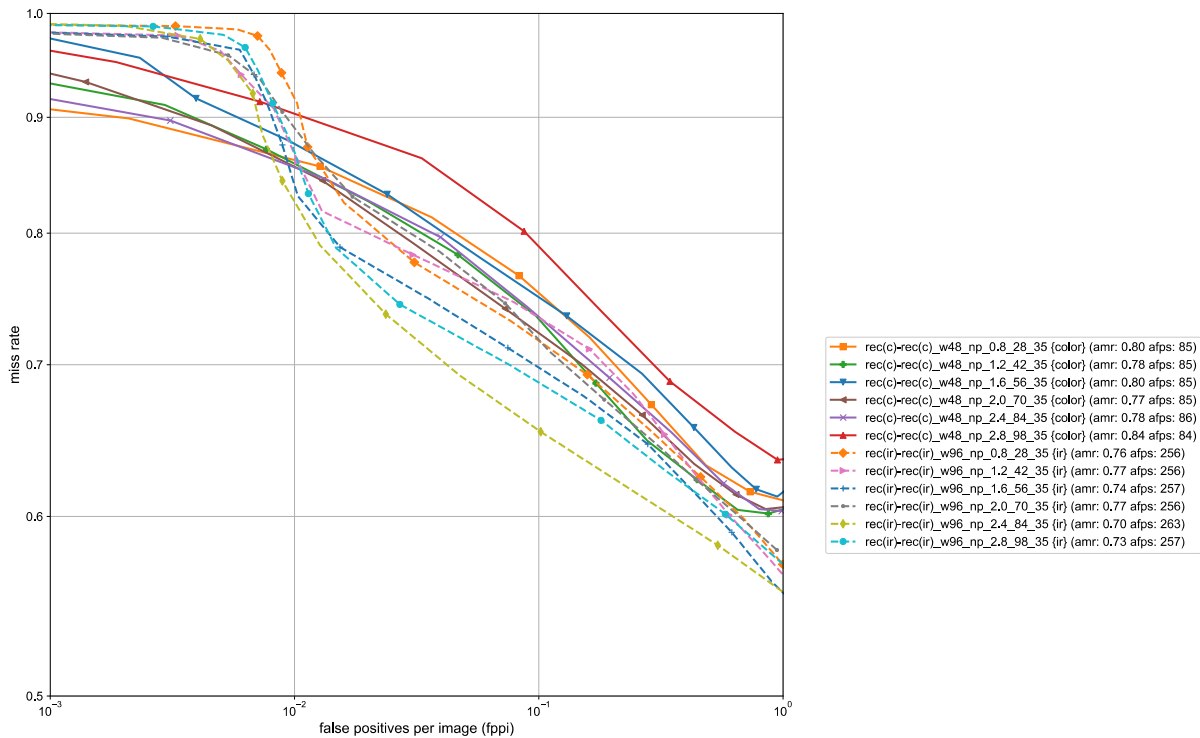


Abbildung 6-7: Ergebnis der Variation des Verhältnisses trainierter Negativ- zu Positivbildern (np\_\*).

### Variation des Eingangsbildes

In den vorhergehenden Untersuchungen wurden verschiedene Konfigurationen für das Training der SVM untersucht. Im Folgenden werden die Ergebnisse zur Veränderung des Eingangsbildes, also dem Preprocessing dargestellt. Hierfür wurden die besten SVM der bisherigen Untersuchungen verwendet. Bei ähnlichen Ergebnissen wurde die SVM gewählt, welche schneller war oder mit welcher weniger Trainingsdaten notwendig waren. Deshalb wurde im Falle des Farbbilds der Detektor mit nur 4.200 Negativbildern gewählt, da er eine ähnliche Erkennungsleistung wie der mit 7.000 Bildern trainierte vorwies. Im Gegensatz zu den Versuchsreihen zuvor, wurden die besten Parameter nicht direkt in der nächsten Testreihe verwendet, sondern die Tests wurden unabhängig voneinander durchgeführt.

SVM für das Farbbild:

Detektorgröße: 48 x 96 Pixel

Negativbilder: 4.200 aus den eigenen Farbbild-Trainingsdaten

Positivbilder: 3.500 aus den eigenen Farbbild-Trainingsdaten

SVM für das Intensitätsbild:

Detektorgröße: 96 x 192 Pixel

Positivbilder: 8.400 aus den eigenen Intensitätsbild-Trainingsdaten

Negativbilder: 3.500 aus den eigenen Intensitätsbild-Trainingsdaten

### Variation des Eingangsbildes – Skalierungsfaktor

Der Skalierungsfaktor beschreibt die Reduktion bzw. Vergrößerung des Eingangsbildes für die Detektion. Die verwendete Kamera Kinect v2 hat eine Auflösung von 512 x 424 Pixel beim Infrarotsensor. Das Farbbild wird ebenso auf diese Größe kalibriert. Der Skalierungsfaktor 0,5 steht bspw. für die halbe Auflösung des Ausgangsmaterials, also 256 x 212 Pixel. Eigentlich sind die HOG-Features invariant gegenüber Skalierungen. In Vorabuntersuchungen hat sich allerdings gezeigt, dass die Größe des Eingangsbildes einen Einfluss auf die Detektionsqualität hat. Die minimal detektierbare Größe ist mitunter ein Faktor, welcher bei unterschiedlichen Skalierungen bei gleicher Detektorgröße eine Rolle spielt. Ebenso unterscheidet sich die maximal detektierbare Größe, da das Sliding-Window-Verfahren standardmäßig nur für  $n_{levels}$ -Durchläufe das Suchfenster um den Faktor  $h_{scale}$  vergrößert. Damit die Höhe des Suchfensters im letzten Durchgang immer der Bildhöhe entspricht, wird  $n_{levels,max}$  wie folgt berechnet:

$$n_{levels,max} = \frac{\log_{10}\left(\frac{h_{Bild}}{h_{Detektor}}\right)}{\log_{10}(h_{scale})} \quad (14)$$

Mit  $h_{Bild}$ : Bildhöhe in Pixeln;  $h_{Detektor}$ : Höhe des Detektors in Pixeln;  $h_{scale}$ : Skalierungsfaktor des Sliding-Windows (Standard: 1,05)

Grundsätzlich ist ein möglichst großes Bild einem kleineren vorzuziehen, da es mehr Informationen enthält. Allerdings ist der Detektor auf bestimmte Auflösungen trainiert worden. Für den auf Farbbildern trainierten Detektor wurden Bilder mit 48 x 92 Pixeln und für den auf Intensitätsbildern trainierte wurden Bilder mit 96 x 192 Pixeln verwendet. Aufgrund des größeren Detektors in letzterem Fall ist davon auszugehen, dass bei diesem für ein besseres Ergebnis das Eingangsbild und demzufolge der Skalierungsfaktor größer sein sollte.

## 6 Evaluation einzelner Algorithmen

Die Auswertung in Abbildung 6-8 zeigt, dass diese Annahme berechtigt war. Für das Intensitätsbild ist ein großer Einfluss der Skalierung auf die Erkennungsleistung ersichtlich. Bei der kleinsten Skalierung (0,6) ist die Erkennungsleistung am geringsten (amr: 0,76), bei Skalierungen auf die 1,1-/1,2-fache Ausgangsauflösung am höchsten (amr: 0,63).

Dagegen wurden im Farbbild nur kleine Unterschiede bei der Variation des Skalierungsfaktors festgestellt (amr 0,69 bis 0,72). Die Ursache für die geringe Sensitivität liegt an der kleineren Detektorgröße. Bei einem kleinen Skalierungsfaktor sind entsprechend auch Menschen in einer niedrigen Auflösung auf den Bildern. Aufgrund der geringen Detektorgröße können diese erkannt werden. Mit steigendem Skalierungsfaktor werden die Menschen zwar höher aufgelöst bzw. sind größer, aber sie können durch das sich vergrößernde Suchfenster des Sliding-Window-Verfahrens erkannt werden. Im Gegensatz dazu werden mit dem größeren Detektor im Intensitätsbild deutlich weniger klein abgebildete (niedrig aufgelöste) Menschen erkannt. Sofern sie doch erkannt werden, ist die BB der Erkennung um einiges größer als die der Markierung. Das hat zur Folge, dass die Erkennung als falsch positiv gewertet wird, da das Verhältnis der Rechteckflächen der BB außerhalb des Grenzwerts liegt.

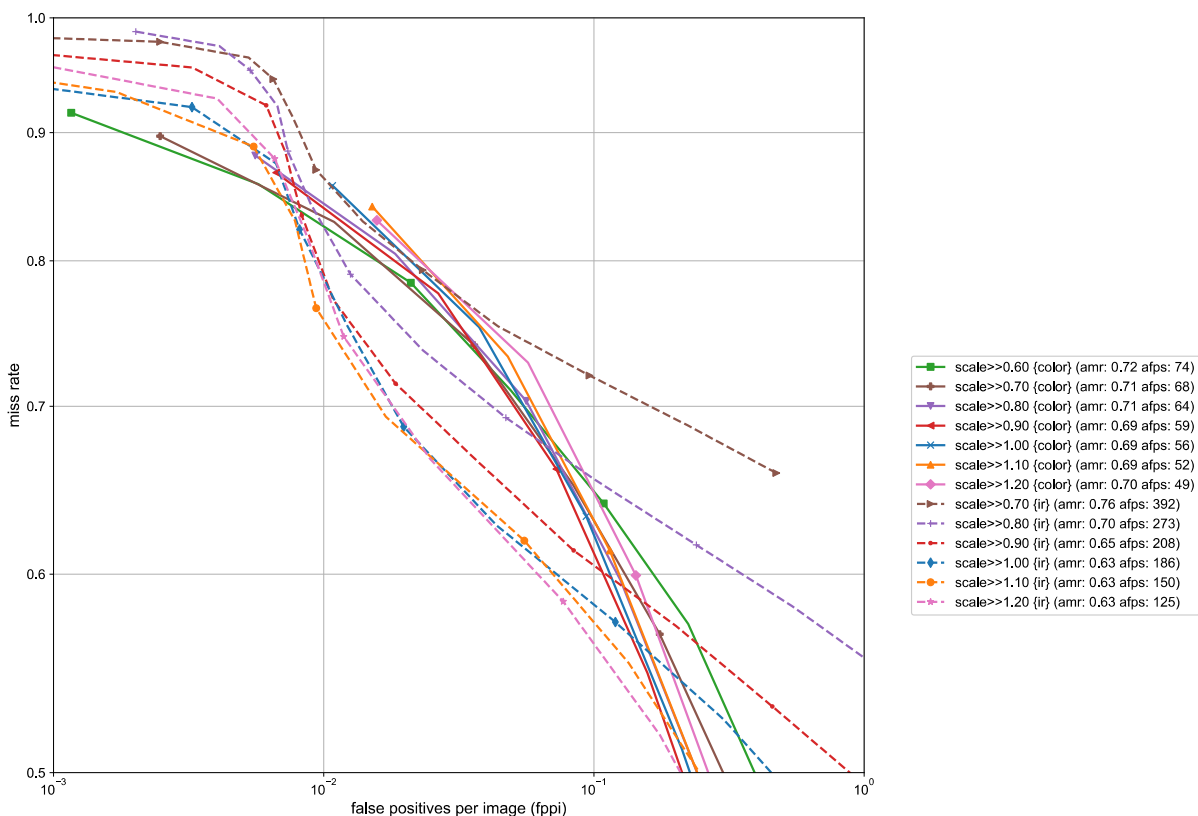


Abbildung 6-8: Ergebnis der Variation der Skalierung des Eingangsbildes.

### Variation des Eingangsbildes – Weichzeichnen (Gauß-Filter & Bilateraler Filter)

Das Weichzeichnen des Eingangsbildes hat zur Folge, dass Bildfehler durch Rauschen reduziert werden können. Allerdings gehen durch das Weichzeichnen auch wichtige Informationen wie z. B. Kanten verloren. Deshalb wurde neben einem schnell zu berechnenden Gauß-Filter auch ein bilateraler Filter, welcher Kanten beibehält, untersucht. Vor allem beim Intensitätsbild könnte eine Verbesserung erreicht werden, da dieses aufgrund des Sensors und der Umwandlung in ein Graustufenbild von starkem Rauschen betroffen ist.

Für das Weichzeichnen wurden bewusst kleine Kernel verwendet, damit etwaige Kanten noch identifiziert werden können. Sowohl für den Gauß- als auch für den bilateralen Filter wurden symmetrische Kernel von drei und fünf Pixeln untersucht. Der Gaußfilter („soften\_“) wurde mit  $\sigma=0,5$  und  $\sigma=1,0$  appliziert. Für den bilateralen Filter („soften\_bilateral\_“) wurde  $\sigma=50$  und  $\sigma=100$  verwendet.<sup>7</sup> Die Werte stammen aus empirischen Voruntersuchungen, welche zum Ziel die größtmögliche Entfernung von Rauschen bei gleichzeitiger Beibehaltung von Kanten hatten.

Das Ergebnis in Abbildung 6-9 zeigt einen großen Unterschied zwischen dem Einfluss von Weichzeichnen auf das Farbbild im Vergleich zu dem Einfluss auf das Intensitätsbild. Im Falle des Farbbilds waren die Einbußen in der Erkennungsleistung derart groß, dass im Bereich von  $fppi < 0,1$  keine Erkennung mehr stattfand. Deshalb wurde für diese Abbildung ein anderes Intervall für die Darstellung verwendet.

Im Intensitätsbild war der Einfluss des Weichzeichnens allgemein sehr gering. Unter der Verwendung des bilateralen Filters mit einem Kernel von  $3 \times 3$  Pixeln und  $\sigma = 50$  wurde gegenüber dem originalen Eingangsbild eine leichte Verbesserung um zwei Prozentpunkte in der amr erreicht.

Anmerkung zur fehlenden Monotonie: Bei einem großen Wert der fppi steigen die Fehlererkennungen wieder. Dies liegt dadurch begründet, dass der Schwellwert zur Akzeptanz von Personenerkennungen gegen Null geht. In diesem Fall häufen sich falsche Erkennungen im Bild an unterschiedlichsten Positionen. Der in der SVM verwendete Sliding-Window-Algorithmus fügt naheliegende Erkennungen zu einer zusammen. Die BB der zusammengefassten Erkennungen ist anschließend nahezu so groß wie die Bildhöhe. Auch wenn sich nun in dieser BB Menschen befinden, wird die Erkennung aufgrund der Auswertungsmethode als falsch bewertet. Deshalb nimmt die miss rate ab einem bestimmten Wert wieder zu.

---

<sup>7</sup> Die Standardabweichung (Sigma) hat im Falle des Gauß-Filters eine andere Wirkung als beim Bilateral-Filter, daher weichen die Werte stark voneinander ab.

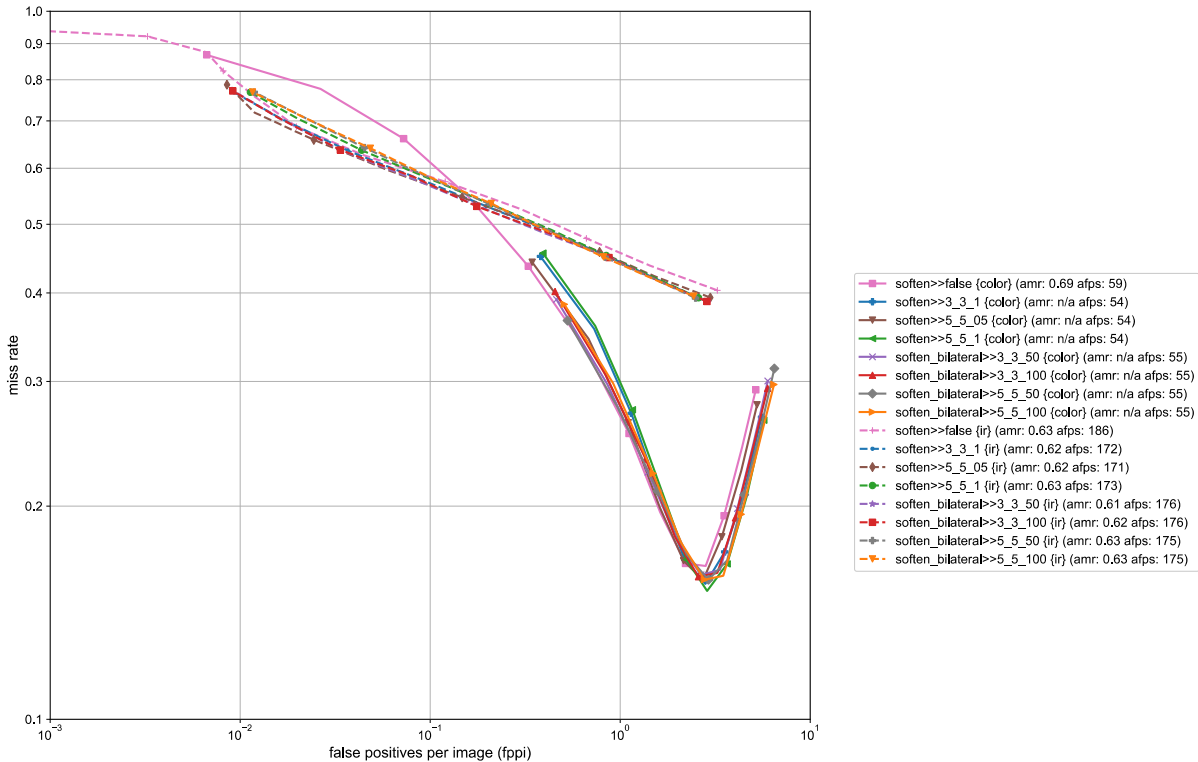


Abbildung 6-9: Ergebnis der Variation der Weichzeichnung des Eingangsbildes.

### Variation des Eingangsbildes (Intensitätsbild) – IR-Konvertierung

Die Konvertierung des Intensitätsbildes in ein graustufenähnliches Bild wird in der einfachsten Variante durch die Normierung oder Egalisierung des Bildes in den 8-Bit-Bereich vorgenommen. Aufgrund der deutlichen Abweichung zu einem normalen Graustufenbild wurde die in Abschnitt 4.2.2 beschriebene Methode zur Umwandlung entwickelt. Ob diese Konvertierung besser ist als eine einfache Normalisierung oder Egalisierung, wurde in dieser Versuchsreihe überprüft.

In der Abbildung 6-10 ist klar erkennbar, dass die eigene, nichtlineare Konvertierungsmethode („own“) eine deutlich bessere Erkennung (mehr als 15 %) als die anderen untersuchten Methoden ermöglicht. Wird die lineare Streckung („normalize“) auf den 8-Bitwertebereich angewandt, ist die Erkennung nur geringfügig höher als bei der Verwendung der Egalisierung („equalize“).

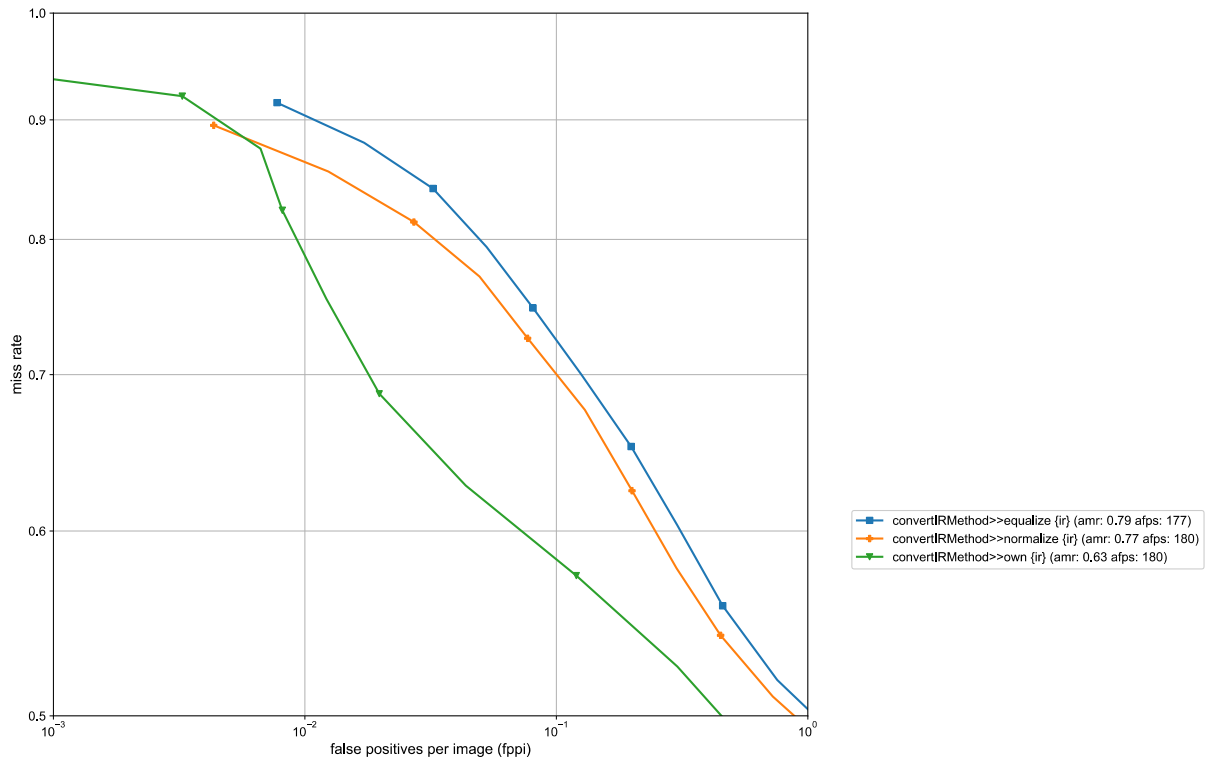


Abbildung 6-10: Ergebnis der Variation der IR-Konvertierungsmethode.

### Variation des Eingangsbildes (Intensitätsbild) – Normalisierung und Egalisierung

Die Normalisierung trägt zur Erhöhung des Kontrastes eines Bildes bei (siehe Abschnitt 2.4.1). Bei der Erzeugung von HOG-Features wird zwar eine Normalisierung durchgeführt, allerdings ist diese nicht auf das gesamte Bild bezogen, sondern auf die einzelnen extrahierten Blöcke und Fenster. Daher wurde untersucht, ob sich durch die Normalisierung des gesamten Bildes die Detektionsqualität ändert. Die Normalisierung wurde nicht auf das rohe Eingangsbild wie in der letzten Versuchsreihe angewandt, sondern auf das bereits konvertierte Graustufenbild. Zugleich wurde die Wirkung der Egalisierung des bereits konvertierten IR-Bildes untersucht. Die Auswirkungen wurden nur für das Intensitätsbild betrachtet, da eine Normalisierung und Egalisierung nur geringfügige Änderungen in einem Farbbild bewirken.

Die Normalisierung auf das gesamte Bild führte nach der Auswertung in Abbildung 6-11 zu keiner Änderung in der Detektion. Die Egalisierung des konvertierten Intensitätsbildes bewirkte jedoch eine Steigerung der Erkennungsrate um 6 %.



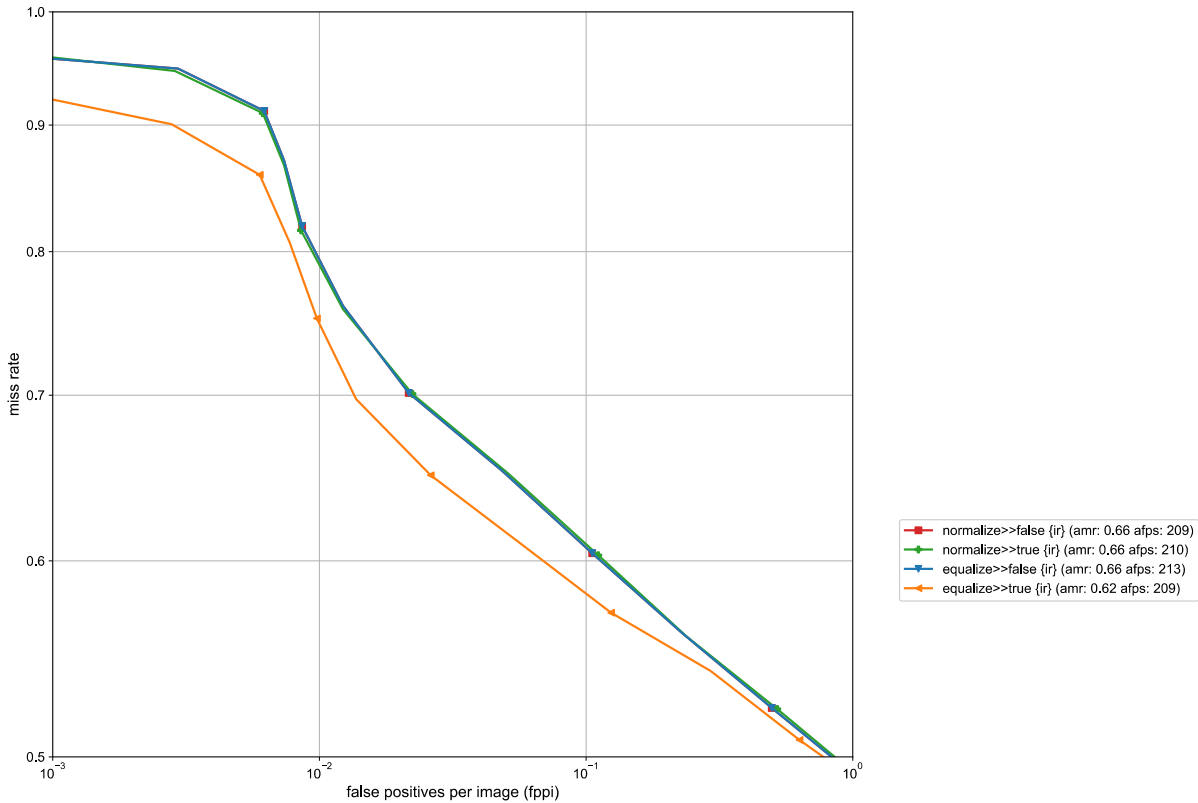


Abbildung 6-11: Ergebnis der Erkennungsleistung bei Vorverarbeitung des Intensitätsbildes durch Normalisierung und Egalisierung.

### Variation der SVM-Parameter – $h_{scale}$

Der Parameter  $h_{scale}$  gehört zu den Parametern des Sliding-Windows-Verfahrens. Er gibt an, um wieviel Prozent das Suchfenster bei jedem Durchlauf vergrößert wird. Umso größer dieser Wert ist, desto performanter ist die Analyse eines Bildes, da weniger Durchläufe notwendig sind. Allerdings kann dies eine Minderung der Detektion zur Folge haben, weil weniger Größen des Suchfensters analysiert werden. Daher wurde im Bereich des Standardwertes von 1,05 (Steigerung der Suchfenstergröße um 5 % in jedem Durchlauf) die Variation des Parameters untersucht. Die Anzahl der Vergrößerungen  $n_{levels}$  wurde in Abhängigkeit der Bildgröße auf den maximalen Wert (siehe Formel (14) auf S. 119) gesetzt.

Die Auswertung in Abbildung 6-12 bestätigt die Vermutung, dass mit einem größeren Wert von  $h_{scale}$  die Detektion schlechter wird. Vor allem beim Farbdetektor ist der Einfluss deutlich zu sehen. Ausgehend von einem Ausgangswert von  $h_{scale} = 1,04$  bis hin zu  $h_{scale} = 1,13$  erhöht sich die amr konstant von 0,71 auf 0,77. Die Bildrate steigt analog im Schnitt um 1-2 FPS pro erhöhtem Prozentpunkt von  $h_{scale}$ .

Der auf Intensitätsbilder trainierte Detektor reagiert weniger sensitiv auf die Erhöhung von  $h_{scale}$ . Die Hauptursache hierfür ist, dass das Eingangsbild nur geringfügig größer ist, als das des Farbbilds (Skalierungsfaktor Farbe: 0,9  $\leftrightarrow$  Intensitätsbild: 1,0), der

Detektor ist aber doppelt so groß. Daher finden während der Bildanalyse ohnehin wenig Vergrößerungen statt. Dieses Verhalten sollte auch in der nächsten Testreihe auftreten, bei welcher die Auswirkung der Anzahl an Vergrößerungen des Sliding-Windows untersucht wurde.

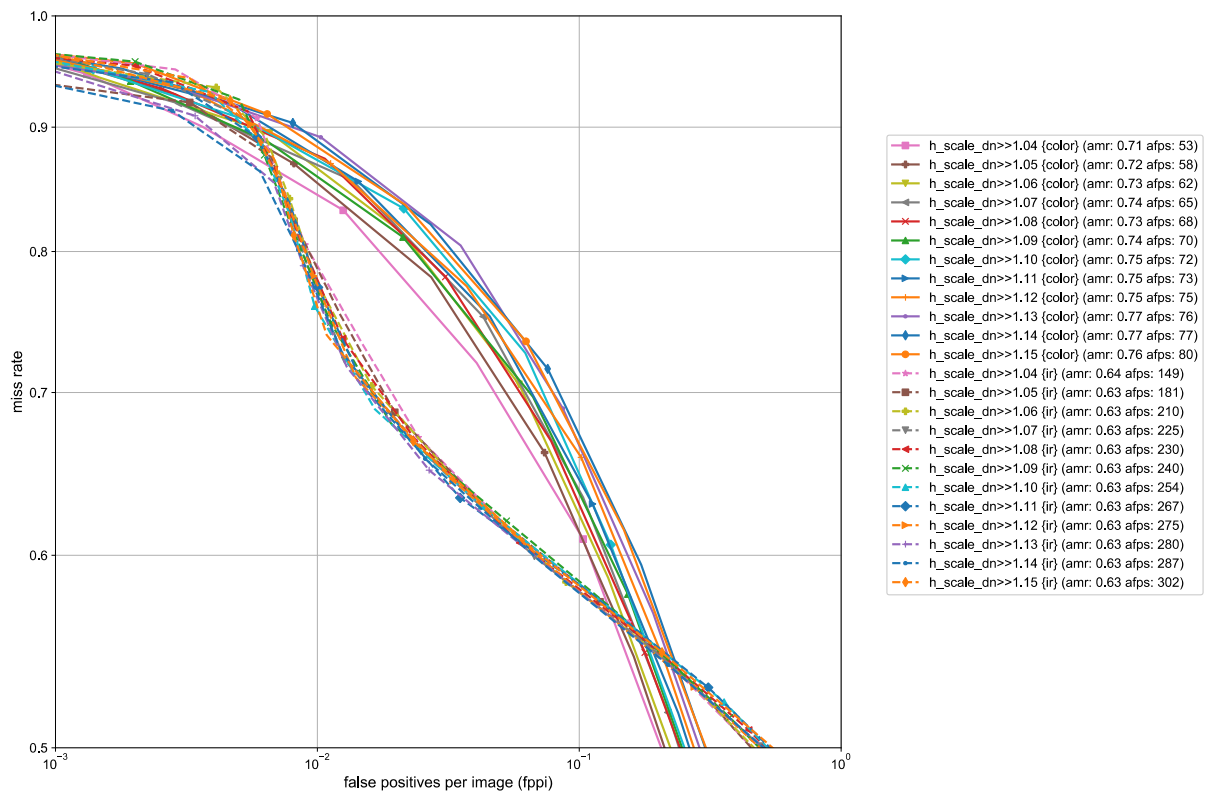


Abbildung 6-12: Ergebnis der Variation des SVM-Parameters  $h_{scale}$ .

### Variation der SVM-Parameter – $n_{levels}$

Analog zum Parameter  $h_{scale}$  ist auch  $n_{levels}$  ein Parameter des Sliding-Window-Verfahrens. Er beschreibt die maximale Anzahl an Vergrößerungen des Suchfensters. Die Erhöhung dieses Werts wird ebenso zur Minderung der miss rate als auch der Bildrate führen. Der maximale Wert kann zwar, wie in Formel (14) auf S. 119 beschrieben, über die Bildhöhe berechnet werden, allerdings ist anzunehmen, dass das Suchfenster nicht zwangsläufig bis zur Bildhöhe vergrößert werden muss. Sofern dies zutrifft, ist durch die Begrenzung der Vergrößerungen eine Senkung der notwendigen Rechenzeit möglich.

Sowohl im Intensitäts- als auch im Farbbild sanken erwartungsgemäß die miss rates mit zunehmendem  $n_{levels}$ . Im Farbbild konnte bis zur maximalen Anzahl von 26 Vergrößerungen keine Stagnation beobachtet werden, wohingegen im Intensitätsbild die miss rate ab 13 von 16 möglichen Vergrößerungen stagniert. Durch die Verwendung von

maximal 13 Vergrößerungen kann der Rechenaufwand gegenüber der maximalen Anzahl an Vergrößerungen (16) um über 10 % gesenkt werden.

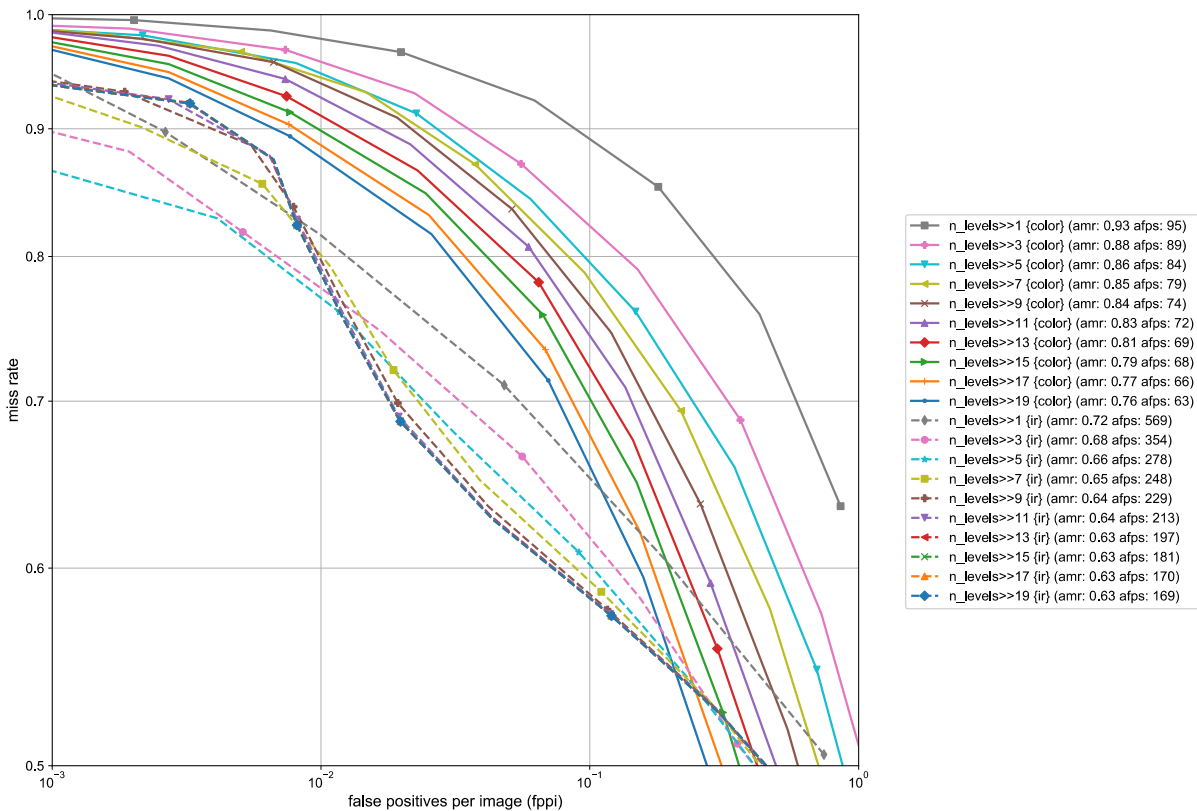


Abbildung 6-13: Ergebnis der Variation des SVM-Parameters  $n_{levels}$ .

### Variation der SVM-Parameter – Group Threshold

Der SVM-Parameter “group threshold” ist ein Schwellwert, welcher die Mindestanzahl gefundener Detektionen in einer Region definiert. Ist der Wert bei Null, werden alle Detektionen der SVM ausgegeben. Wird z. B. ein Wert von Vier verwendet, werden nur Detektionen, welche mindestens durch drei naheliegende Detektionen mit ähnlicher Größe bestätigt wurden, ausgegeben. Das ausgegebene Rechteck ist der Durchschnitt der Rechtecke des jeweiligen Clusters. Die Auswirkung der Erhöhung dieses Schwellwerts ist vorab nicht prognostizierbar, denn wenn ein Detektor sehr wenige Falscherkennungen hat und Personen mit sehr hoher Konfidenz erkennt, kann die Erhöhung des Schwellwerts dazu führen, dass weniger Personen erkannt werden. Liefert ein Detektor jedoch eher Detektionen mit niedrigerer Konfidenz und mehr Falscherkennungen, kann die Erhöhung des Schwellwerts die Falscherkennungen – bei gleichzeitiger Beibehaltung der richtigen Erkennungen – reduzieren.

Die Auswertung in Abbildung 6-14 zeigt die beiden beschriebenen Effekte. Der auf Farbbilder trainierte Detektor verliert bei der Erhöhung des Schwellwerts richtige Erkennungen, wenn auch der Einfluss nicht sehr stark ist.

Der Einfluss beim Intensitätsbild ist deutlich stärker. Mit zunehmender Größe des Schwellwerts sinkt die miss rate kontinuierlich. Dies spricht für eine Abnahme der Falscherkennungen durch die Fusion der Rechtecke.

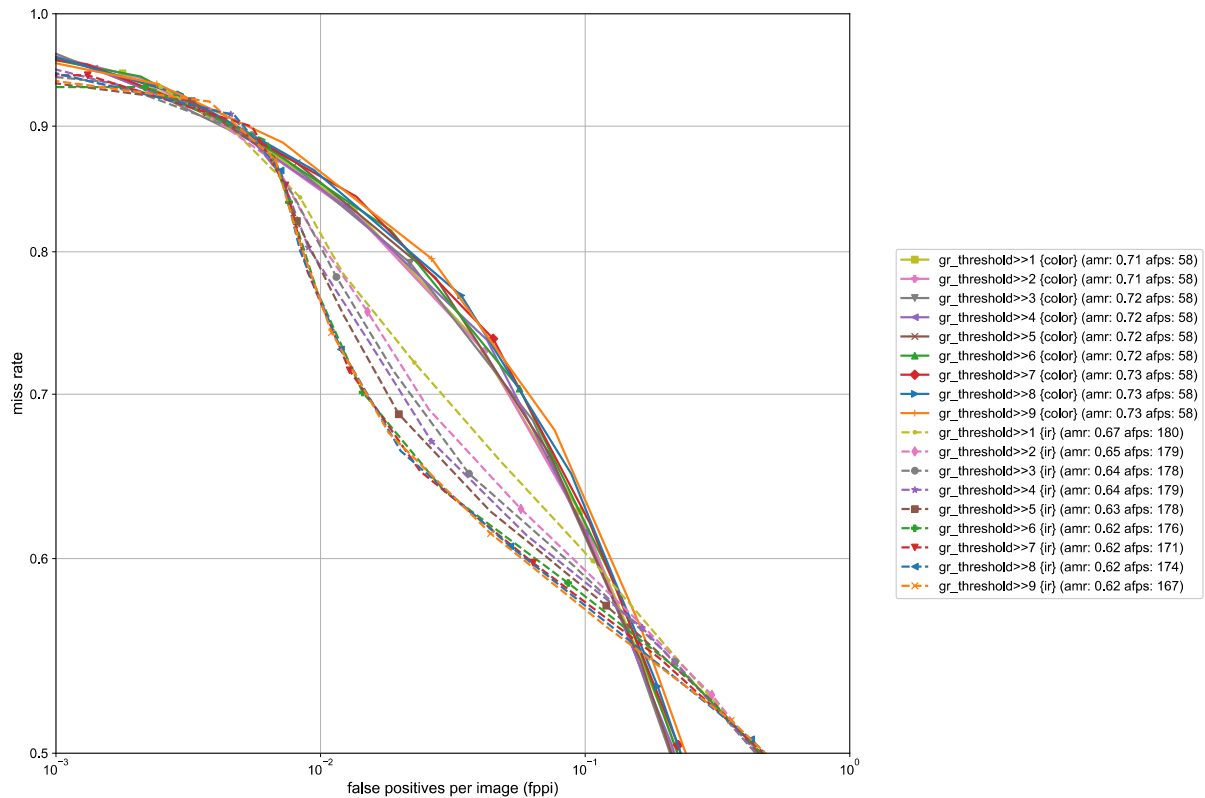


Abbildung 6-14: Ergebnis der Variation des SVM-Parameters „group threshold“.

### Variation der SVM-Parameter – Anwendung der besten Parameter

Die vorhergehende Evaluation der verschiedenen SVM-Trainingsansätze und -Einstellungen führten je nach Versuchsreihe zu Optimierungsmöglichkeiten hinsichtlich der Detektion im Anwendungsfall. Das Tiefenbild wurde zur Bildanalyse aufgrund der schlechten Ergebnisse verworfen. Die in Tabelle 6-8 gezeigten Konfigurationen wurden als beste Werte für die Testdaten identifiziert. Bei ähnlichen amr wurde entweder der Standardwert oder die rechenärmere Konfiguration verwendet. Zu den Konfigurationen sei angemerkt, dass diese keine allgemeingültigen Lösungen für jegliche Anwendungsgebiete darstellen, da die Optimierungen auf die Testdaten durchgeführt wurden; alleine die Verwendung eines anderen Sensors oder Kamerawinkels kann deutliche Abweichungen zu den gewonnenen Ergebnissen nach sich ziehen. Ungleich wichtiger ist die Vorgehensweise die SVM mittels der gezeigten Testreihen zu optimieren.

## 6 Evaluation einzelner Algorithmen

Tabelle 6-8: Zusammenfassung der Ergebnisse für das Farb- und Intensitätsbild zur Personenerkennung mit einer SVM.

Parameter	Farbbild	Intensitätsbild
Detektorgröße [Px]	48 x 96	128 x 254
Anzahl positiver Trainingsbilder	3.500	3.500
Anzahl negativer Trainingsbilder	4.200	8.400
Trainingsdaten - Positivbilder	Eigener Trainingsdatensatz	Eigener Trainingsdatensatz
Trainingsdaten - Negativbilder	Eigener Trainingsdatensatz	Eigener Trainingsdatensatz
Skalierungsfaktor	0,9	1,0
Weichzeichnen	Aus	Bilateral (Kernel 3 x 3, $\sigma=50$ )
IR-Konvertierung	-	Eigene Methode
Egalisierung	-	Aus
Normalisierung	-	Aus
$h_{scale}$	1,04	1,05
$n_{levels}$	Automatisch	Automatisch
$gr_{threshold}$	1	5

Durch die Variation der Vorverarbeitung und der SVM-Parameter konnten die Detektionsleistung der SVM im Vergleich zu den Ergebnissen aus Abbildung 6-7 gesteigert werden. Im Falle des Farbbilds wurde die miss rate von 78 % auf 71 % vermindert (siehe Abbildung 6-15). Im Intensitätsbild konnte die miss rate von 70 % auf 59 % reduziert werden. Somit wurden im Schnitt etwa 40 % der Personen erkannt, bei einem fppi-Wert von 0,01 bis 0,1. Ob diese Erkennungsleistung oder die fast 30 % des Farb-Detektors für die zweistufige Kollisionsgefahrerkennung bei anderen Testdaten ausreichen, wird in Kapitel 7 erörtert.

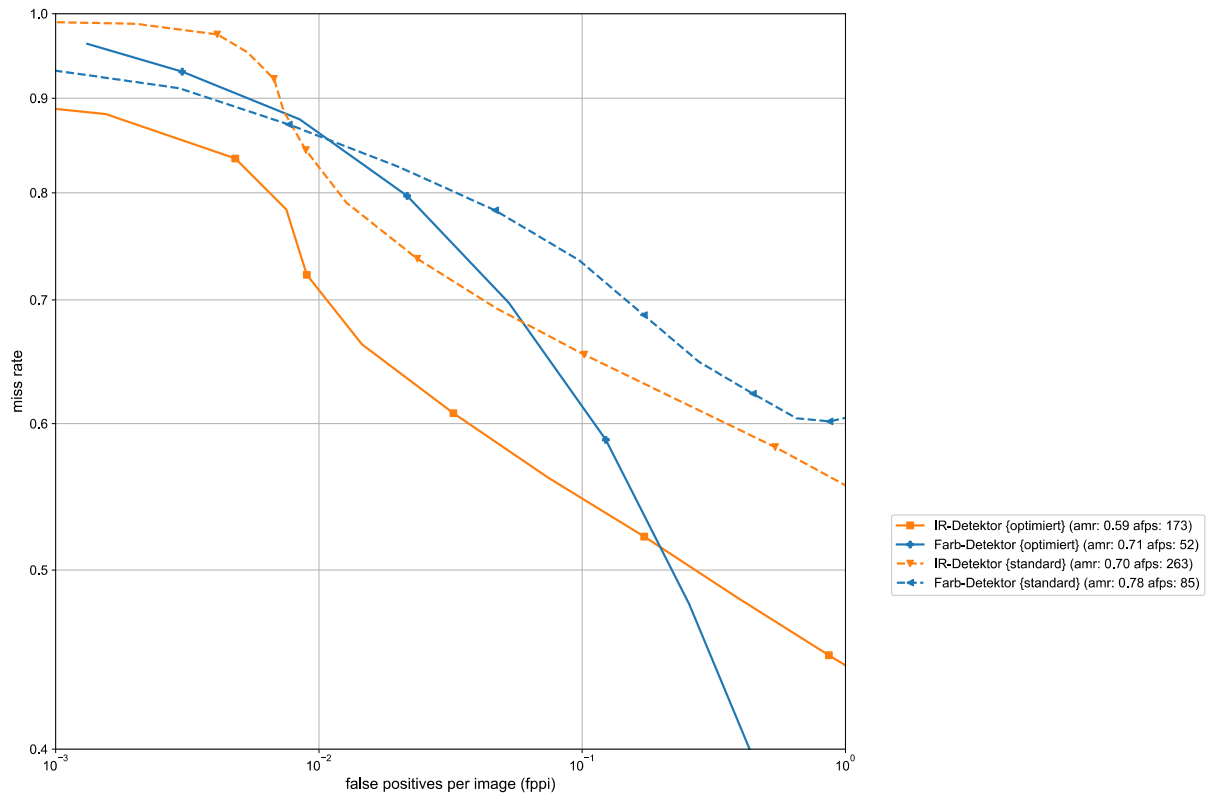


Abbildung 6-15: Unterschied der Erkennungsleistungen der SVM nach Optimierung des Eingangsbildes und SVM-Parameter.

### Variation der detektierten Personen – Verdeckungsgrad (Intensitätsbild)

In den bisherigen Untersuchungen wurden nur Personen ausgewertet, welche maximal zu 33 % verdeckt waren. Der Grad der Verdeckung wurde qualitativ bei der Markierung der Daten in die folgenden Kategorien (siehe Abbildung 6-16) eingeteilt:

- 0 %: Person ist vollständig sichtbar
- < 33 %: Maximal eine Körperregion (Kopf, Brust, Füße) ist verdeckt
- 33 %-66 %: Minimal eine Körperregion und maximal zwei sind verdeckt
- > 66 %: Mehr als zwei Körperregionen sind verdeckt

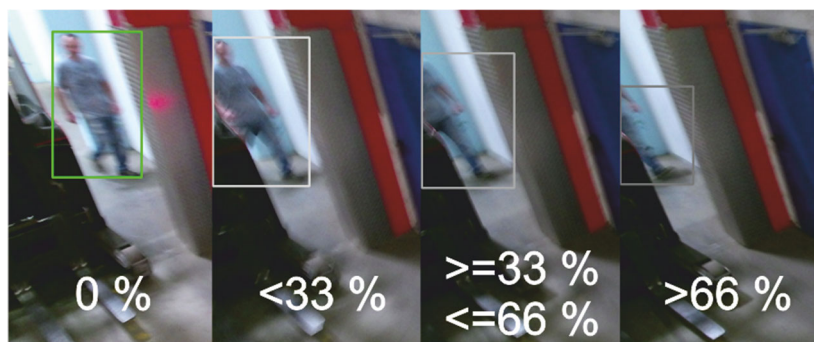


Abbildung 6-16: Beispiele für unterschiedliche Verdeckungsgrade von Personen.

Die größte Gefahr besteht für Menschen, wenn sie nahezu vollständig sichtbar im Bild sind, da somit der Weg zwischen Stapler und Mensch frei ist. Aber auch wenn Menschen nur teilweise sichtbar sind oder durch Hindernisse verdeckt werden, kann ihnen eine Gefahr drohen. Daher wurden für die SVM-Konfigurationen aus Tabelle 6-8 die Testdaten nochmals in Bezug auf den Verdeckungsgrad am Beispiel des Intensitätsbilds ausgewertet. In den folgenden Auswertungen ist zusätzlich die Markierungsanzahl (N:... ) zur Beurteilung der Stichprobe in der Legende angegeben.

Eine große Schwäche der SVM zeigt sich in Abbildung 6-17. Während vollständig sichtbare Personen zu fast 50 % im Integrationsbereich erkannt wurden, steigt die amr mit dem Grad der Verdeckung rapide. Bei der Verdeckung eines Körperteils liegt die amr bei 80 %, bei mehreren verdeckten Körperteilen werden nahezu keine Personen mehr erkannt.

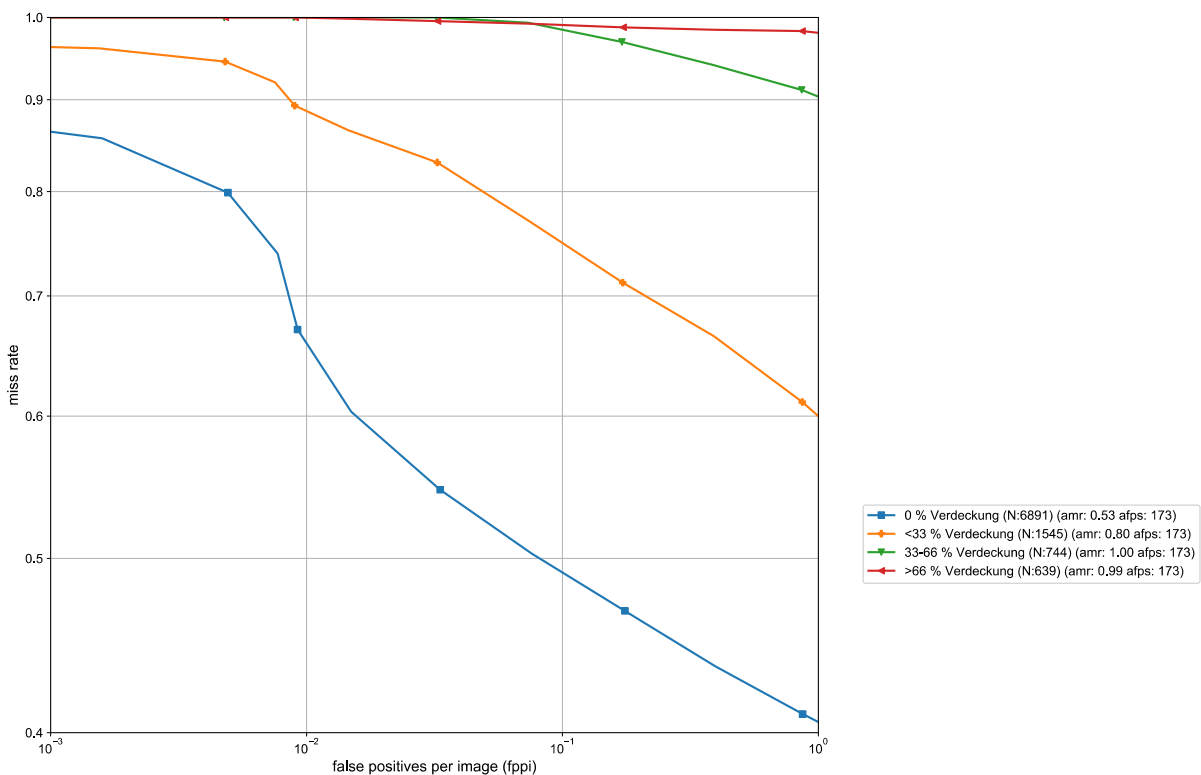


Abbildung 6-17: Erkennungsleistung der SVM in Abhängigkeit der Verdeckung von Personen am Beispiel des Intensitätsbilds.

### Variation der detektierten Personen – Posen (Intensitätsbild)

Für die Personenerkennung wurden in Abschnitt 5.3.1 verschiedene Körperposen definiert, welche typisch für Arbeiter in intralogistischen Umgebungen sind. In den Testdaten wurden diese Posen markiert. Die Markierung erfolgte nur bei vollständig sichtbaren Personen. Demzufolge wurde eine sitzende Person, welche nur zu 50 % sichtbar ist, in dieser Auswertung nicht berücksichtigt. Die Pose „tragend“ wurde aufgrund

der geringen Anzahl von lediglich 23 Markierungen nicht berücksichtigt. Als Referenz dienten die Erkennungen von Markierungen von aufrechten, vollständig sichtbaren Personen („Aufrecht“).

Ähnlich zum Verdeckungsgrad steigt auch bei Posen, welche vom aufrechten Gang abweichen, die amr signifikant. Dabei werden sich beugende oder sitzende Menschen nahezu gleich mit einer amr von über 90 % erkannt. Die enorme Verschlechterung ist nicht nur aufgrund der geringeren Erkennungsleistung, sondern auch auf zu große BB bei Detektionen zurückzuführen.

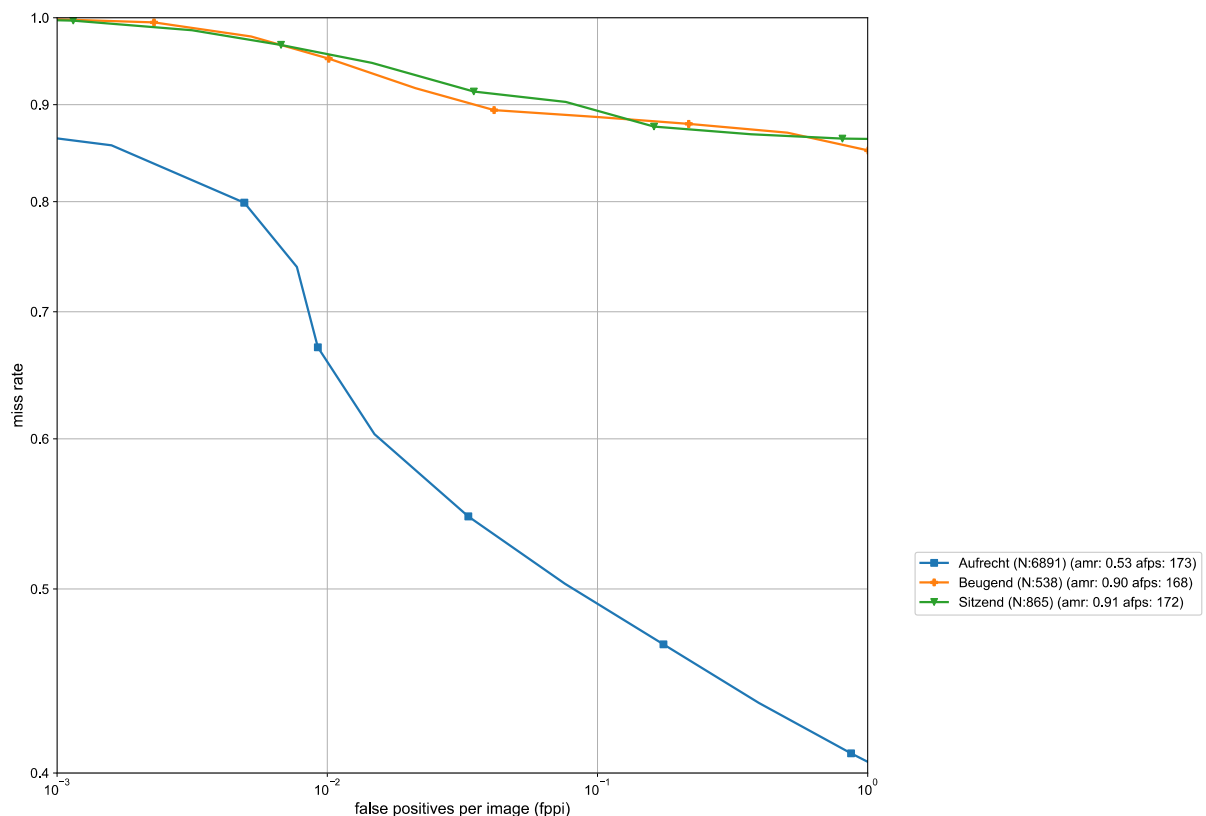


Abbildung 6-18: Erkennungsleistung der SVM in Abhängigkeit der Posen von Personen am Beispiel des Intensitätsbilds.

Ein Beispiel für zu große Erkennungen bei sich beugenden Menschen zeigt Abbildung 6-19. Die Markierungen auf der linken Hälfte des Bildes sind deutlich kleiner als die Detektionsergebnisse. Dadurch werden diese Detektionen als falsch-positiv und zugleich falsch-negativ gewertet. Dieser Fall tritt vor allem bei nicht-aufrechten Menschen häufiger auf, da der Detektor aufgrund des Trainings immer Rechtecke mit dem Verhältnis 2:1 (Höhe zu Breite) liefert. Eine Änderung bezüglich des Verhältnisses wäre aber wiederum für die Erkennung stehender Personen kontraproduktiv. Die Detektion auf der rechten Seite zeigt dagegen eine korrekte falsch-positive Erkennung.



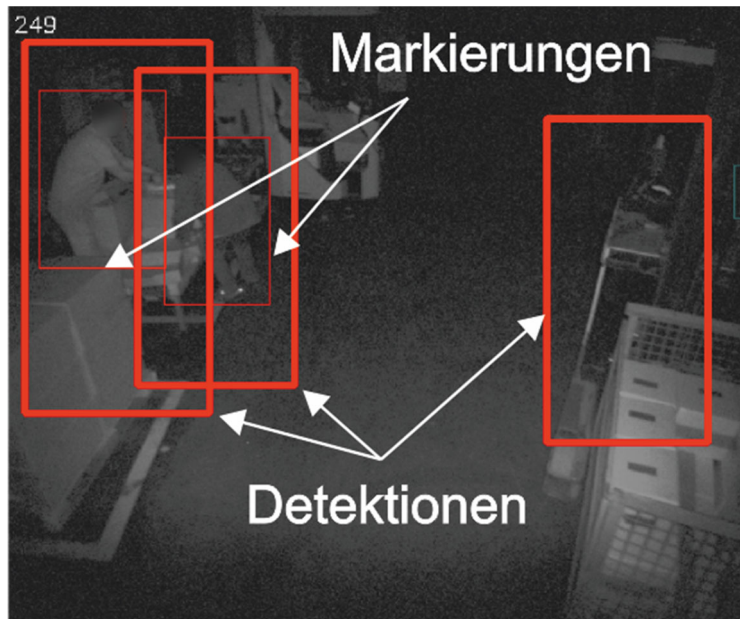


Abbildung 6-19: Beispiel auf der linken Hälfte für als falsch-positiv und falsch-negativ bewertete Erkennungen aufgrund der BB-Größe bei Einsatz einer SVM.

### Variation der detektierten Personen – Sicherheitsweste (Farb- und Intensitätsbild)

Der Einfluss von Sicherheitswesten wurde untersucht, da diese mit stark reflektierenden Flächen ausgestattet sind. Während im Farbbild nur geringe Störungen durch die Reflektionen erzeugt werden, ist die Störung im Intensitätsbild, wie in Abbildung 6-20 zu sehen ist, stärker.



Abbildung 6-20: Bildstörung durch Sicherheitsweste. Links: Intensitätsbild, rechts: Farbbild.

Die Auswertung in Abbildung 6-21 beinhaltet zum Vergleich die Ergebnisse für beide Bildarten. Im Falle des Intensitätsbilds steigt die amr bei Personen mit Sicherheitsweste von 0,53 auf 0,69. Es werden demzufolge 30 % weniger Personen, die eine Sicherheitsweste tragen, erkannt. Dies war aufgrund der starken Störung des Intensitätsbildes zu erwarten, wobei auch im Training derartige Bilder verwendet wurden. Im Farbbild wurden Personen mit Sicherheitsweste besser (amr: 0,59) erkannt als diejenigen ohne Weste (amr: 0,69). Dies kann zufallsbedingt sein oder der Farbbilddetektor hat während des Trainings diese Art von Personen präferiert.

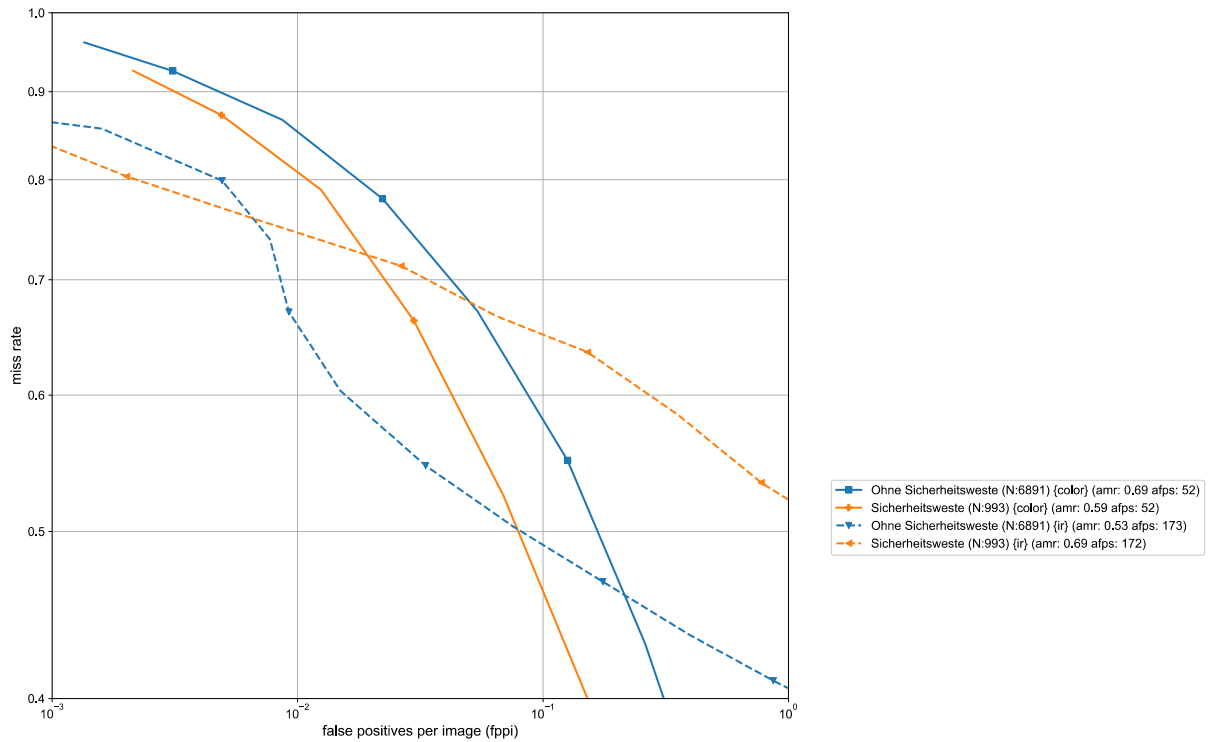


Abbildung 6-21: Erkennungsleistung der SVM in Abhängigkeit des Tragens einer Sicherheitsweste.

### Variation der detektierten Personen – Aktivitäten (Intensitätsbild)

Im Lager gehen Personen während ihrer Arbeit verschiedenen Tätigkeiten nach. Bei den Industrienaufnahmen wurden die in Abbildung 6-22 dargestellten Tätigkeiten im Fahrbereich des ausgestatteten Staplers identifiziert.

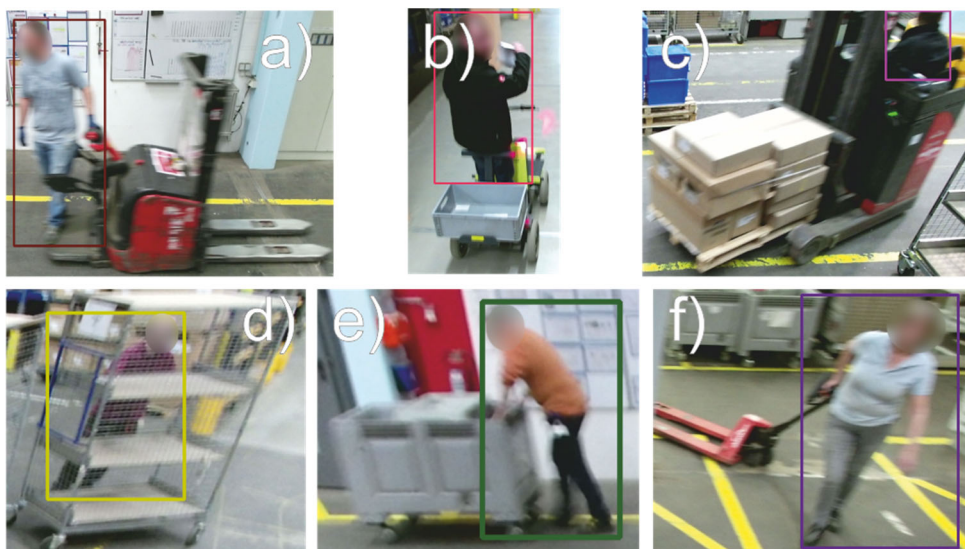


Abbildung 6-22: Beispiele verschiedener Aktivitäten in Lagern: a) Ameise fahren, b) Scooter fahren, c) Gabelstapler fahren, d) Regalwagen ziehen/schieben, e) Rollbehälter ziehen/schieben, f) Hubwagen ziehen/schieben.

Während der Bedienung der in (a-c) gezeigten Flurförderzeuge bewegen sich die Personen nur wenig. Während der Fahrt mit einer Ameise (a) oder mit einem Scooter (b) sind die Personen größtenteils sichtbar und aufrecht. Daher ist von einer guten Erkennung als Mensch auszugehen. Anders verhält es sich bei den Gabelstaplerfahrern (c): Der Verdeckungsgrad ist sehr hoch und die Person befindet sich in einer sitzenden Haltung. Eine Erkennung der Fahrer ist folglich schwieriger.

Beim manuellen Ziehen bzw. Schieben von Flurförder- oder Transporthilfsmitteln (d-f) ist die Person zum größten Teil sehr gut sichtbar und die Haltung ähnelt der einer normalen Laufbewegung. Die Personenerkennung sollte in diesem Fall ähnlich funktionieren wie bei laufenden Menschen. Eine Ausnahme zeigt die Manipulation des Regalwagens (d). Hierbei waren die Personen überwiegend fast vollständig verdeckt.

Die Auswertung der verschiedenen Aktivitäten zeigt Abbildung 6-23 am Beispiel des Intensitätsbilds. Die meisten Personen, welche einer Aktivität nachgehen, wurden schlechter als freistehende, aufrechte Personen (blau) erkannt.

Gabelstaplerfahrer (rosa) und Personen, welche einen Regalwagen bewegten, (rot) wurden nahezu nicht mehr erkannt ( $amr > 0,95$ ). Gabelstaplerfahrer sind für den SVM-Detektor in den meisten Fällen zu klein oder werden aufgrund des abweichenden Seitenverhältnisses nicht erkannt. Zu der Bewegung eines Regalwagens sei angemerkt, dass in diesem Fall nicht nur Sequenzen verwendet wurden, bei denen die Person vollständig sichtbar ist, da auf nahezu allen Sequenzen die Person durch das Regal verdeckt wurde.

Personen, welche auf Ameisen (orange) oder Scootern (dunkelbraun) standen, wurden in über 60 bzw. 64 % der Bilder nicht erkannt. Eine etwas geringere  $amr$  wies die Bewegung des Rollbehälters (lila) auf. Dabei war die  $amr$  ähnlich der von freistehenden Personen. Ähnlich ist die  $amr$  der Hubwagen-ziehenden (grün) Personen. Hierbei lag die  $amr$  mit 50 % leicht unter der  $amr$  mit frei sichtbaren Personen.

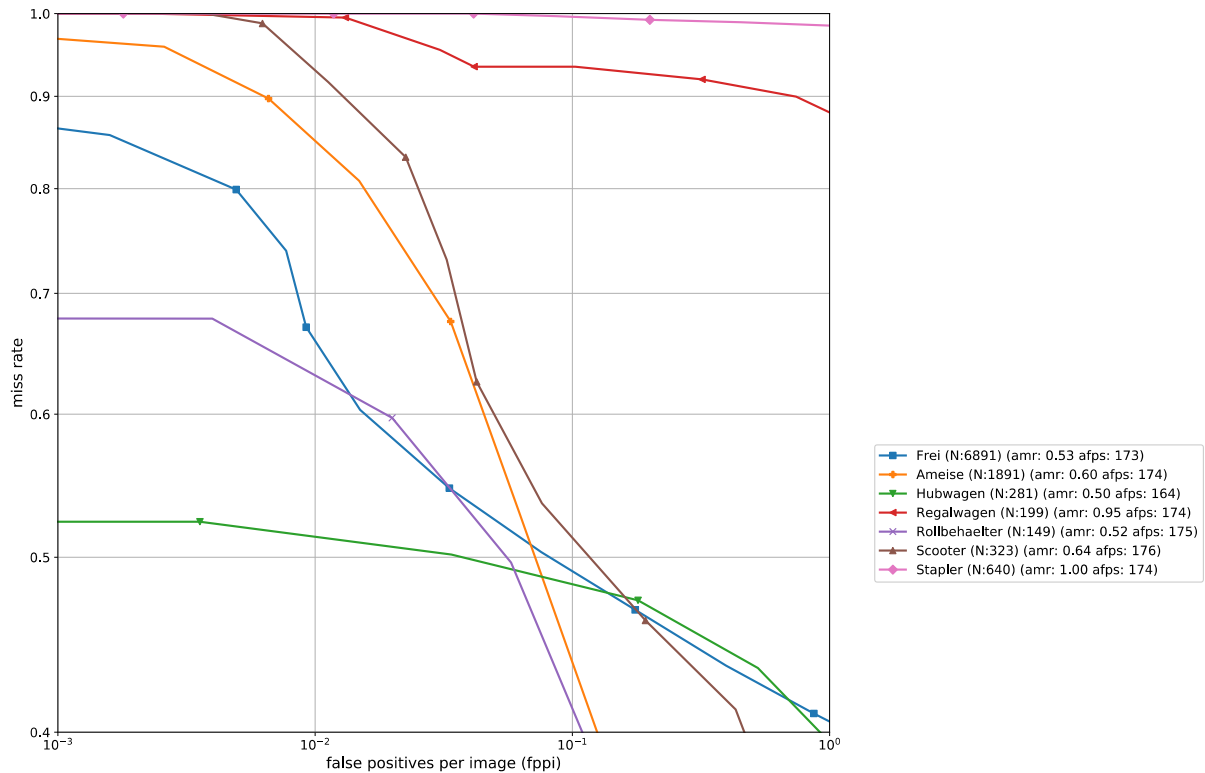


Abbildung 6-23: Erkennungsleistung der SVM in Abhängigkeit der Aktivitäten von Personen am Beispiel des Intensitätsbilds.

### 6.1.3 Ergebnisse – Neuronales Netz

Der große Vorteil moderner neuronaler Netze (NN) gegenüber SVM liegt unter anderem darin, dass sie die zu analysierenden Merkmale automatisch auswählen. In jedem Netz zur Objekterkennung ist eine Vielzahl an möglichen Merkmalen enthalten, wodurch auch die Kombination aus mehreren Merkmalen für die Objekterkennung genutzt werden kann bzw. in der Regel genutzt wird. Neben Merkmalen sind ebenso relevante Filter enthalten und werden im Training zur optimalen Detektion ausgewählt. Aufgrund dessen entfallen bei der Nutzung neuronaler Netze zum größten Teil die Untersuchungen zur Vorverarbeitung der Daten. Die Vorbereitung reduziert sich auf die Konvertierung der Daten in ein Format, welches von dem jeweiligen Netz verarbeitet werden kann. Im Falle der verwendeten Darknet-Architektur ist die Umwandlung aus Abschnitt 4.2.2 notwendig.

Die Möglichkeiten der Parametervariation sind aufgrund der beschriebenen Aspekte bei NN sehr eingeschränkt. Sinnvoll ist nur die Variation der Datenquellen für das Training. Im Falle des Farbbilds gibt es bereits sehr gut trainierte Netze, wodurch das Training für diese Bildart nicht sinnvoll ist. Im Falle der Intensitäts- und Tiefendaten gibt es jedoch keine geeigneten trainierten Netze. Daher wurde der Fokus beim Training von NN auf diese zwei Bildarten gelegt. Die Nutzung dieser Bildarten hat den Zweck die Personenerkennung auch in lichtschwachen Umgebungen, welche im Einsatzgebiet

von Gabelstaplern durchaus häufig sind, zu ermöglichen. Aufgrund mangelnder öffentlich zugänglicher Datenquellen für diese Bildarten konnten nur die eigenen Aufnahmen verwendet werden. Dafür wurde der gleiche Trainingsdatensatz wie bei der SVM verwendet. Hierbei wurde die Anzahl der Trainingsbilder nicht begrenzt, da im eingesetzten Netz Methoden zur Vermeidung von Overfitting enthalten sind. Die Variation der Trainingsdaten bestand in der unterschiedlichen Konvertierung der Bilddaten in das vom Darknet genutzte Format.

### Variation der trainierten Bildarten

Die Darknet-Architektur erfordert für das Training Daten mit drei Bildkanälen. Die dafür möglichen Kombinationen aus Intensitäts- und Tiefenbild (vgl. Abbildung 4-12) wurden zum Training verwendet und anhand der gleichen Testdaten ausgewertet, wie sie für die SVM verwendet wurden. Zusätzlich zu den eigens trainierten Detektoren wurde zum Vergleich der (von *Redmon* und *Fahardi* veröffentlichte) Farbbild-Personendetektor mit in die Auswertung einbezogen [Red-2018], im Folgenden „Yolov3-Detektor“ genannt.

Im Falle des vortrainierten Yolov3-Detektors (lila) liegt die amr bei 0,03 (siehe Abbildung 6-24). Bei den eigens trainierten Detektoren wird eine amr von 0,06 im Intensitätsbild „ir“ (grün) und im Tiefenbild „d(epth)“ (orange) erreicht. Zugleich ist die Bildrate (afps) im Vergleich zum Yolov3-Detektor um 40 % höher. In den zwei Bildarten „ird“ (hellblau) und „ird3x“ (rot), welche Intensitätsbild und Tiefenbild auf unterschiedliche Art und Weise kombinieren, hatten die Detektoren eine etwas schlechtere Leistung als im Intensitätsbild.

Mit Ausnahme des auf Tiefenbilder trainierten Detektors haben alle eigens trainierten Detektoren bei einem ausreichend hohen Schwellwert keine Falscherkennungen mehr. Beispielsweise traten im Intensitätsbild bei einer miss rate von 0,16 (A) keine falschen Erkennungen mehr auf. Das Fehlen von Falscherkennung ist an den horizontalen Linien bei (B) in Abbildung 6-24 erkennbar.<sup>8</sup>

---

<sup>8</sup> Werte von Null sind aufgrund der logarithmischen Skalierung nicht darstellbar.

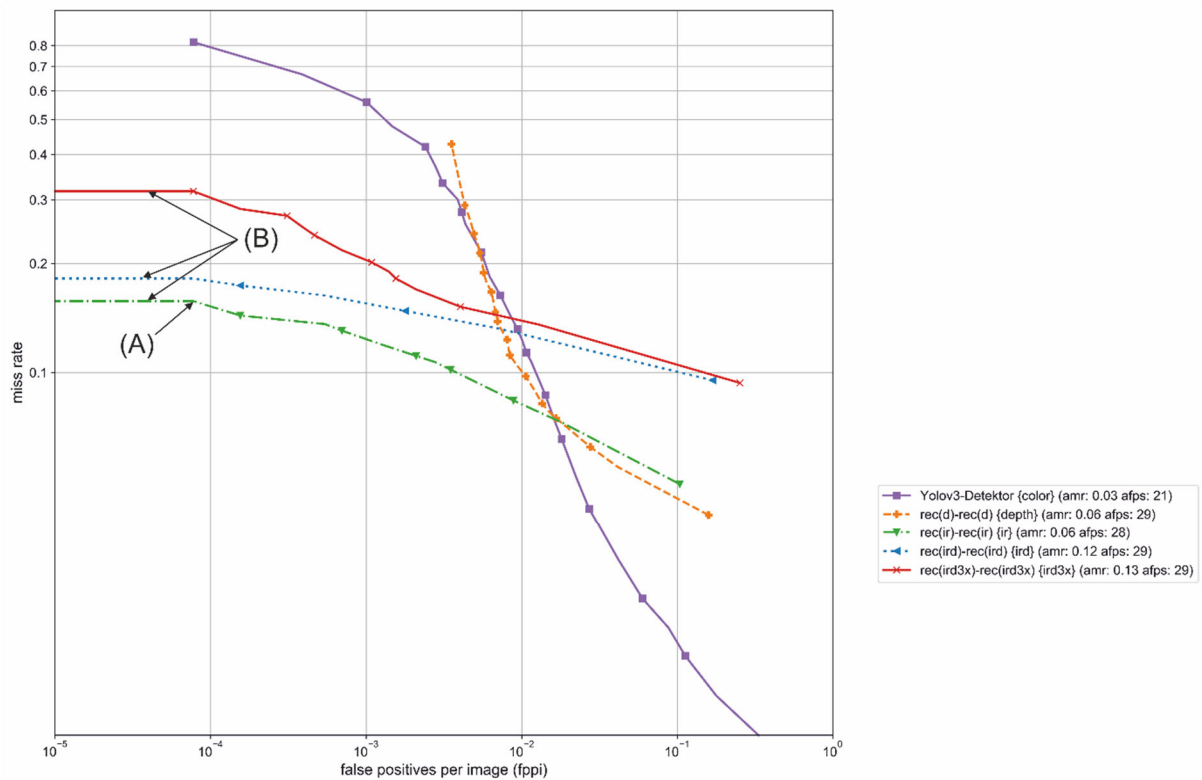


Abbildung 6-24: Ergebnisse der Variation der Bildarten mit einem DL-Detektor für Personen, die maximal zu 33 % verdeckt sind.

### Variation der detektierten Personen – Verdeckung (Intensitätsbild)

DL-Netze sind weitaus besser als SVM darin, zum Teil verdeckte Objekte zu finden, selbst wenn nur vollständig sichtbare Objekte trainiert wurden. Im Gegensatz zur SVM sind daher deutlich bessere Erkennungsraten auch bei nur teils sichtbaren Personen zu erwarten. Die folgend gezeigte Auswertung in Abbildung 6-25 beschränkt sich aus Gründen der Übersichtlichkeit auf das Intensitätsbild, ist aber auf die anderen Bildarten übertragbar.

Vollständig sichtbare Personen (blau) wurden nur in 3 % der Fälle nicht erkannt. Bei den bis zu 33 % (orange) und bis zu 66 % (grün) verdeckten Personen erhöhte sich die amr auf über 18 %. Für Personen, bei denen nur noch maximal ein Körperteil sichtbar war, lag die amr bei über 30 %. Zum Vergleich: Die beste amr lag mit einer SVM bei vollständig sichtbaren Menschen bei 53 % (siehe Abbildung 6-17).

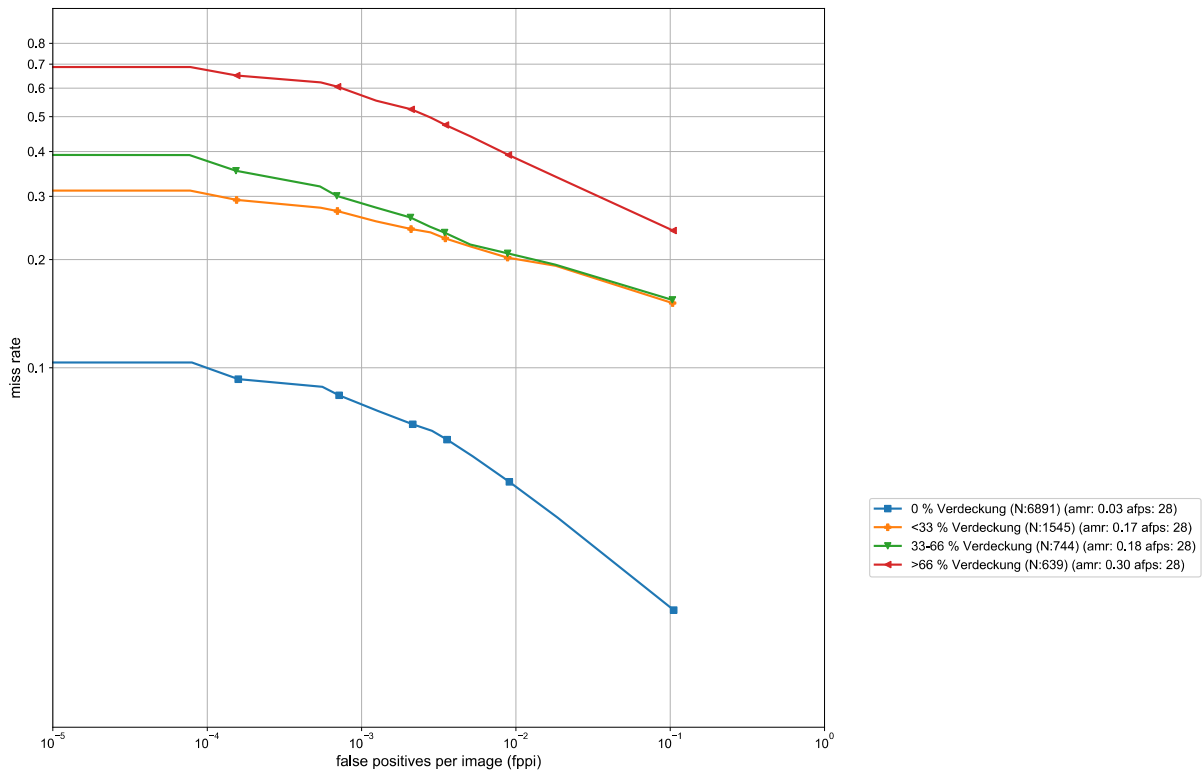


Abbildung 6-25: Unterteilung der DL-Detektionsergebnisse in Abhängigkeit des Verdeckungsgrads markierter Personen.

### Variation der detektierten Personen – Posen (Intensitätsbild)

In dieser Testreihe wurden die gleichen Posen (aufrecht, beugend, sitzend) wie bei der SVM untersucht. Die Markierungen mit aufrechten (laufenden, stehenden) Personen wurden mit einer amr von 3 % erkannt (siehe Abbildung 6-26). Sich beugende Menschen wurden erwartungsgemäß schlechter als aufrechte Menschen erkannt, jedoch lag die amr bei nur 31 %. Eine deutlich höhere amr wiesen sitzende Personen auf, diese wurden im Schnitt zu 76 % nicht erkannt.

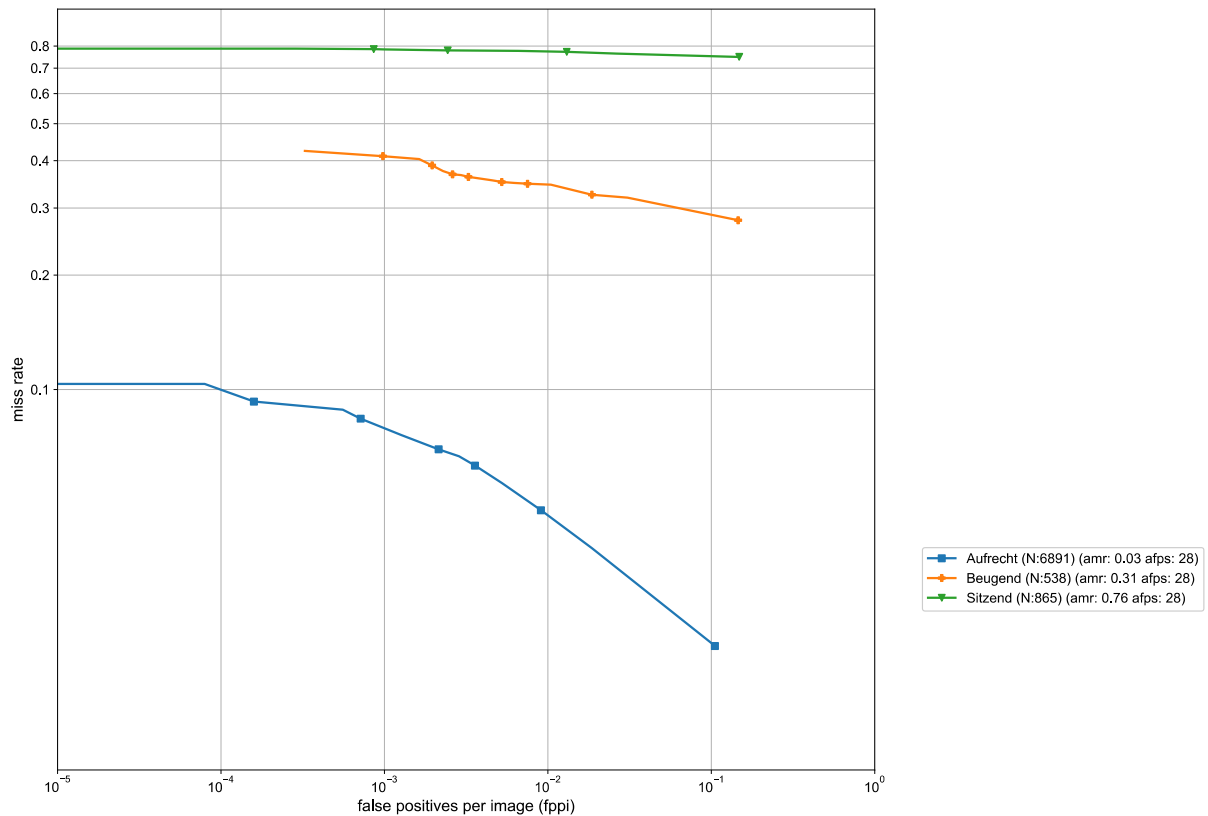


Abbildung 6-26: Unterteilung der DL-Detektionsergebnisse in Abhängigkeit der Pose markierter Personen.

### Variation der detektierten Personen – Sicherheitsweste (Farb- und Intensitätsbild)

Der Einfluss der Sicherheitsweste wird in Abbildung 6-27 analog zur Auswertung der SVM für das Farb- und Intensitätsbild gezeigt, um den Unterschied der Störung durch die Reflektionen in den jeweiligen Bildarten zu verdeutlichen. Im Farbbild (durchgehende Linien) liegt die amr bei Personen mit (orange) und ohne (blau) Sicherheitsweste bei 2 %. Analog zu den Ergebnissen aus der SVM werden auch hier Menschen mit Sicherheitsweste ab einem gewissen Punkt besser erkannt als Menschen ohne Weste. Ebenso hat die Sicherheitsweste im Intensitätsbild (gestrichelte Linien) einen stärkeren Einfluss. Die amr beträgt mehr als das Vierfache (3 % → 13 %), wenn die Person eine Sicherheitsweste trägt.



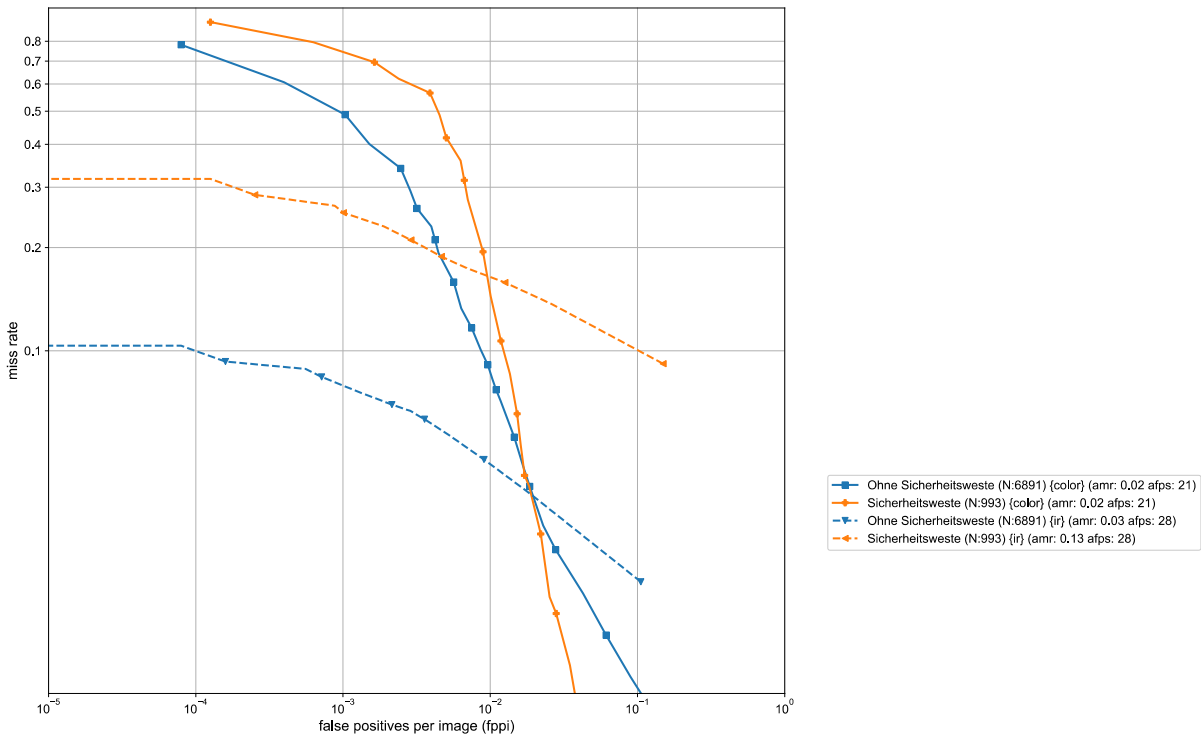


Abbildung 6-27: Einfluss der Sicherheitsweste bei der DL-Detektion im Farb- und Intensitätsbild.

### Variation der detektierten Personen – Aktivitäten (Intensitätsbild)

Bei den verschiedenen Aktivitäten sind im Falle des DL-Netzes geringere Steigerungen der miss rate im Vergleich zu den Ergebnissen der SVM zu erwarten, da bereits gezeigt wurde, dass Verdeckungen (vgl. Abbildung 6-17 und Abbildung 6-25) einen geringeren Einfluss auf die Erkennungsleistung bei DL-Detektoren haben. Ebenso wurde bereits gezeigt, dass die Abweichung von einer aufrechten Pose einen niedrigeren Einfluss (vgl. Abbildung 6-18 und Abbildung 6-26) hat.

Die Auswertung in Abbildung 6-28 bestätigt die vorigen Annahmen. Die höchste amr liegt bei 37 % und wurde bei Personen registriert, welche einen Regalwagen bewegen (rot). Gabelstaplerfahrer (rosa) wurden mit einer amr von 20 % erkannt. Die amr der restlichen Aktivitäten (Ameise, Hubwagen, Rollbehälter, Scooter) waren ähnlich oder kleiner als die der freistehenden, aufrechten Personen.

Bei den Ameisenfahrern wurden nahezu alle (>99 %) als Person erkannt, wobei die Kurve sehr stark fällt. Dies spricht dafür, dass die Streuung der Konfidenz, mit welcher Ameisenfahrer erkannt wurden, relativ gering war. In geringerer Form findet sich dieses Verhalten auch bei den Detektionen der Scooter- und Hubwagenbediener.

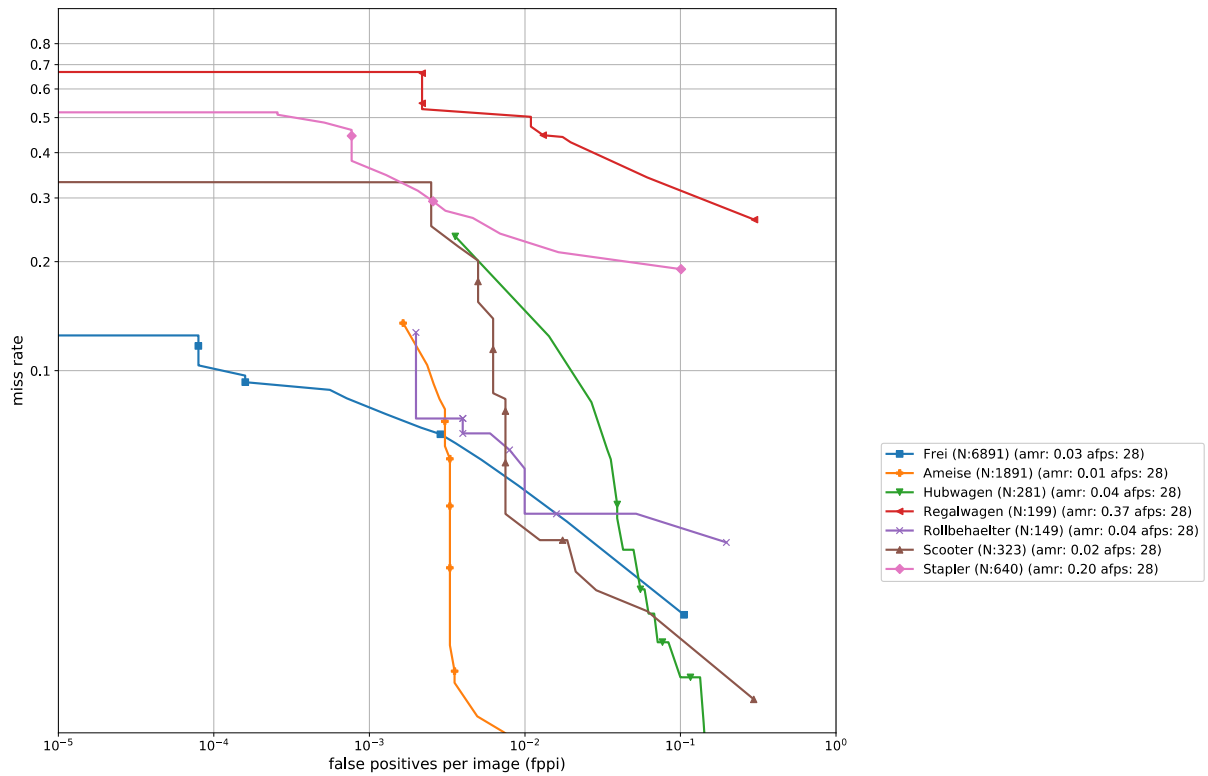


Abbildung 6-28: Unterteilung der DL-Detektionsergebnisse in Abhängigkeit der Aktivität markierter Personen.

## 6.2 Auswertung der Kollisionsgefahrerkennung

Für die Kollisionsgefahrerkennung wurden verschiedene Methoden des optischen Flusses implementiert, welche prinzipiell für den Anwendungsfall in Frage kommen. In diesem Abschnitt wird untersucht, welcher Algorithmus am besten in den definierten Testfällen abschneidet. Zuerst wird die Methode beschrieben, wie die Algorithmen gegenübergestellt werden sollen. Anschließend folgt die Darstellung der Ergebnisse und die Auswahl des besten Algorithmus.

### 6.2.1 Auswertungsmethode

Für die Bewertung der Kollisionsgefahrerkennung wurde aus der Literaturrecherche keine geeignete Methode für das Anwendungsszenario gefunden. Daher wurden Bewertungskriterien identifiziert, mit welchen die Algorithmen gegenübergestellt werden. Diese sind:

1. Warnungsrate: Prozentualer Anteil an Videos, in denen eine Kollisionsgefahr erkannt wurde
2. Rechtzeitig: Prozentualer Anteil an Videos, in welchen die Kollisionsgefahr rechtzeitig erkannt wurde, sodass die Kollision unter Berücksichtigung des maximalen Bremswegs noch zu vermeiden gewesen wäre
3. Permanenz: Verhältnis der Anzahl der Bilder, in welchen gewarnt wurde, und der Gesamtanzahl an Bildern zwischen dem Soll-Bremszeitpunkt und der Kollision
4. Bildrate: Inverse der Berechnungszeit eines Bildes

Pro Testfallszenario wurden üblicherweise zehn Videos aufgenommen. Wenn in acht Videos das System eine Kollision erkannt und eine Warnung ausgegeben hat, ist die Warnungsrate 80 %.

Die Rechtzeitigkeit beschreibt in wie vielen erkannten Kollisionen die Zeit von der ersten Warnung bis zur Kollision größer als die Bremszeit nach Abbildung 3-3 war. Sie ist nur auf Videos bezogen, in welchen eine Kollision erkannt wurde. Wenn in acht Videos eine Kollision erkannt wurde und in vier Videos rechtzeitig, wird als Rechtzeitigkeit der Wert von 50 % angegeben.

Die Permanenz eines Videos wird wie folgt bestimmt: Die Berechnung berücksichtigt den Zeitraum zwischen dem Bremszeitpunkt nach Abbildung 3-3 und dem Zeitpunkt der Kollision. In diesem Zeitraum wird die Anzahl der Bilder gezählt, in welchen das System gewarnt hat und anschließend in das Verhältnis zur Gesamtanzahl der Bilder

gesetzt. Eine Permanenz von 100 % bedeutet demzufolge, dass das System in jedem Bild – vom Bremszeitpunkt bis hin zur Kollision – die Kollisionsgefahr als solche erkannt hat. Eine Permanenz von 100 % ist zwar optimal, aber im Praxisfall nur selten zu erreichen: Nur geringe Lenkeinschläge während der Fahrt oder das Fahren über eine kleine Mulde im Boden können dazu führen, dass eine Kollisionsgefahr eventuell nicht mehr als solche erkannt wird. Der Grund hierfür ist, dass mit dem optischen Fluss in jedem Bild die Bewegungsvektoren berechnet werden. Fährt der Stapler z. B. über eine Unebenheit zwischen zwei Bildern, zeigen die Bewegungsvektoren am Hindernis womöglich in Richtung des Bodens oder in die Decke. In dem Fall würde das System nicht warnen, da nach den berechneten Bewegungsvektoren keine Kollision bevorsteht.

Anschließend wird das arithmetische Mittel dieses Wertes von allen Videos berechnet. Videos, in welchen keine Kollisionsgefahr erkannt wurde, werden in dem arithmetischen Mittel nicht berücksichtigt. Eine Ausnahme bilden die Videos, in welchen der Stapler sehr nahe an einem Objekt vorbeifährt. In diesen Aufnahmen wird die Permanenz nicht mit dem Kollisionszeitpunkt ermittelt, sondern anhand des Zeitpunkts, wenn der Stapler am Hindernis vorbeigefahren ist.

Für die Bildrate wurde die Zeit gemessen, welche zur Berechnung des optischen Flusses zwischen zwei Bildern benötigt wurde. Die angegebene Bildrate ist das arithmetische Mittel der Bildraten aller Videos. Hierbei sei angemerkt, dass die angegebene Bildrate nicht zwangsläufig der maximal erreichbaren entspricht, da bei den Evaluationsdurchläufen nach einiger Zeit aufgrund des Wärmemanagements des verwendeten Notebooks die Leistung der Grafikkarte limitiert wurde. Daher sank die Bildrate von Testfall zu Testfall auch bei Durchläufen mit dem gleichen Algorithmus des optischen Flusses langsam ab.

### 6.2.2 Ergebnisse

Für diese Evaluation wurden die Testfälle 1a bis 2d (siehe Abschnitt 5.3.2) – ohne Menschen – verwendet. Die weiteren Testfälle 3a bis 3c werden gesondert in der Evaluation des Gesamtsystems in Kapitel 7 behandelt. Untersucht wurden die in Abschnitt 4.2.5 erwähnten Methoden des optischen Flusses bzw. „Scene Flows“: Brox (BR), Dual-TVL (D-TVL), Farneback (FB), Lucas-Kanade (LK) und Primal-Dual (PD).

#### **Variation der Methode des optischen Flusses bzw. Scene-Flow**

Die Methoden des optischen Flusses wurden mit den Standardparametern verwendet. Als Eingangsquelle für den optischen Fluss diente das Intensitätsbild. Zur Berechnung des Scene-Flow wurde zudem das Tiefenbild verwendet, wobei kleine Bereiche fehlender Pixel mit der Methode aus Abschnitt 4.2.2 interpoliert wurden.

**Testszenarios mit einer Kollision (1a bis 1c)**

Im Folgenden werden zuerst die Szenarien 1a bis 1c (siehe Abbildung 6-29) behandelt, bei denen eine Kollision auftritt. Im ersten Szenario 1a wurden Frontalzusammenstöße mit einem Hindernis untersucht. Sowohl die Geschwindigkeit des Staplers als auch der horizontale Abstand des Gabelstaplers zur Kante des Hindernisses  $y$  wurden hierbei variiert. Im zweiten Szenario 1b wurden Kollisionen einer ortsfesten Drehung des Staplers mit unterschiedlichem Abstand  $x$  zu einer Wand simuliert. Im letzten Szenario 1c wurden Kollisionen mit einem Hindernis in einer Gasse untersucht; hierbei wurde der Abstand  $x$  des Hindernisses zum Gassenanfang variiert.

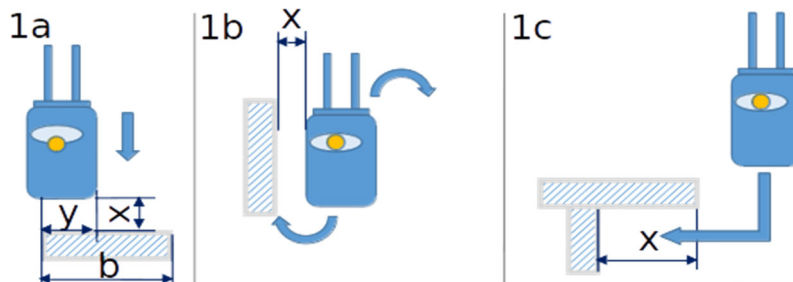


Abbildung 6-29: Schema für die Szenarien 1a bis 1c.

Die Frontalzusammenstöße (Szenario 1a) wurden mit Ausnahme des BR-Algorithmus gut erkannt (siehe Tabelle 6-9). Probleme hatte letzterer mit kleineren Hindernishöhen: Bei 0,26 (Testfall 4) und 0,54 Metern (Testfall 5) Höhe wurde keine der Kollisionen erkannt.

Allgemein traten die größten Defizite bei der ortsfesten Drehung (Szenario 1b) auf, in diesem Fall konnten nur mit dem LK- und PD-Algorithmus die Kollisionen erkannt werden.

Kollisionsgefahren mit einem Hindernis in einer Gasse (Szenario 1c) wurden mit Ausnahme des BR-Algorithmus nahezu alle erkannt.

Tabelle 6-9: Warnungsraten verschiedener Methoden des optischen Flusses für die Testszenarien 1a bis 1c.

		Warnungsrate				
Testfall	Szenario	BR	D-TVL	FB	LK	PD
1	1a	100 %	89 %	100 %	100 %	89 %
2	1a	100 %	100 %	100 %	100 %	100 %
3	1a	100 %	100 %	100 %	100 %	100 %
4	1a	0 %	100 %	100 %	100 %	100 %
5	1a	0 %	90 %	100 %	100 %	100 %
6	1a	100 %	100 %	100 %	100 %	100 %
7	1a	90 %	100 %	100 %	100 %	100 %

## 6.2 Auswertung der Kollisionsgefahrerkennung

<b>8</b> 1a	90 %	100 %	100 %	100 %	90 %
<b>9</b> 1a	100 %	100 %	90 %	100 %	100 %
<b>10</b> 1a	90 %	100 %	100 %	100 %	90 %
<b>11</b> 1b	40 %	100 %	90 %	100 %	60 %
<b>12</b> 1b	0 %	30 %	0 %	100 %	90 %
<b>13</b> 1b	0 %	0 %	0 %	100 %	100 %
<b>14</b> 1b	0 %	0 %	0 %	100 %	100 %
<b>15</b> 1c	100 %	100 %	100 %	100 %	100 %
<b>16</b> 1c	60 %	100 %	100 %	100 %	90 %
<b>17</b> 1c	40 %	100 %	90 %	100 %	100 %
<b>18</b> 1c	40 %	100 %	100 %	100 %	100 %
<b>Ø</b>	58 %	84 %	82 %	100 %	95 %

Mit dem LK- und PD-Algorithmus wurden 100 bzw. 95 % der Kollisionsgefahren erkannt. Allerdings wurden nur mit dem LK- und D-TVL-Algorithmus auch 99 % der Kollisionsgefahren rechtzeitig erkannt (siehe Tabelle 6-10). Mit dem PD-Algorithmus wurden im Schnitt nur 85 % der Kollisionsgefahren rechtzeitig erkannt. Vor allem Kollisionsgefahren mit niedrigen Hindernissen (Testfall 4 bis 5) konnten nicht rechtzeitig erkannt werden.

Mit Ausnahme von LK und D-TVL schnitten alle Algorithmen im Testfall 4 (Hindernishöhe 0,28 m) bezüglich des Warnungszeitpunkts schlecht ab. Die Hauptursache liegt an der Anzahl kollidierender Pixel, welche die Methode des optischen Flusses liefert. Damit eine Kollisionsgefahr als solche gewertet wird, muss der Schwellwert für die distanznormierte Pixelanzahl (siehe S. 80) überschritten werden. Kollidierende Pixel im Bereich des KLT wurden bei allen Algorithmen detektiert, allerdings waren es mit Ausnahme von LK und D-TVL zu wenig.

Eine Auswertung der Testfälle des Szenarios 1b ist bezüglich der Rechtzeitigkeit nicht sinnvoll, da eine Drehung des Gabelstaplers mit konstanter Geschwindigkeit in der Praxis schwer umzusetzen ist und aufgrund dessen kein Brems- bzw. Warnzeitpunkt festgelegt werden konnte.

Die Kurvenfahrtenkollisionen (Szenario 1c) wurden mit Ausnahme des BR- und PD-Algorithmus durchweg rechtzeitig erkannt. Mit dem PD-Algorithmus konnten erst ab einem Abstand von 0,5 m (Testfall 17) des Hindernisses zum Gassenanfang alle Kollisionen rechtzeitig erkannt werden.

## 6 Evaluation einzelner Algorithmen

Tabelle 6-10: Anteil rechtzeitiger Warnungen verschiedener Methoden des optischen Flusses für die Testszenarien 1a bis 1c.

		Rechtzeitige Warnungen				
Testfall	Szenario	BR	D-TVL	FB	LK	PD
1	1a	22 %	100 %	100 %	100 %	100 %
2	1a	0 %	100 %	100 %	100 %	100 %
3	1a	0 %	100 %	100 %	90 %	100 %
4	1a	-	90 %	0 %	90 %	0 %
5	1a	-	100 %	100 %	100 %	60 %
6	1a	0 %	100 %	100 %	100 %	100 %
7	1a	0 %	100 %	100 %	100 %	100 %
8	1a	0 %	100 %	90 %	100 %	78 %
9	1a	0 %	100 %	100 %	100 %	100 %
10	1a	0 %	100 %	100 %	100 %	100 %
11	1b	-	-	-	-	-
12	1b	-	-	-	-	-
13	1b	-	-	-	-	-
14	1b	-	-	-	-	-
15	1c	0 %	100 %	100 %	100 %	90 %
16	1c	0 %	100 %	100 %	100 %	67 %
17	1c	0 %	100 %	100 %	100 %	100 %
18	1c	0 %	100 %	100 %	100 %	100 %
∅		2 %	99 %	92 %	99 %	85 %

Die Permanenz der Kollisionsgefahrerkennung ist zwar für die rechtzeitige Warnung unerheblich, sie dient aber als zusätzliches Maß für die Robustheit der Kollisionsgefahrerkennung. Auch wenn aufgrund der bereits beschriebenen äußeren Faktoren 100 % in der Praxis nicht erreicht werden können, kann sie als Vergleich der Methoden dienen, da die äußeren Einflüsse aufgrund der Auswertung anhand von Aufnahmen für alle Methoden des optischen Flusses gleich sind. Im Szenario 1a bis 1c wies der LK-Algorithmus mit 88 % die höchste Permanenz auf, gefolgt von FB mit 83 %, LK mit 82 % und PD mit 62 % (siehe Tabelle 6-11). Am schlechtesten schnitt der BR-Algorithmus mit 36 % ab, wobei mit diesem ohnehin nur 58 % der Kollision erkannt wurden.

Tabelle 6-11: Durchschnittliche Permanenz der Kollisionsgefahrerkennungen verschiedener Methoden des optischen Flusses für die Testszenarien 1a bis 1c.

		Permanenz				
Testfall	Szenario	BR	D-TVL	FB	LK	PD
1	1a	16 %	82 %	84 %	79 %	68 %
2	1a	31 %	77 %	82 %	96 %	78 %
3	1a	50 %	76 %	80 %	78 %	90 %
4	1a	-	82 %	86 %	99 %	56 %
5	1a	-	79 %	93 %	84 %	65 %
6	1a	47 %	71 %	100 %	94 %	82 %

7 1a	72 %	86 %	92 %	92 %	93 %
8 1a	25 %	81 %	70 %	89 %	64 %
9 1a	42 %	81 %	75 %	85 %	70 %
10 1a	40 %	86 %	82 %	76 %	65 %
11 1b	73 %	80 %	95 %	76 %	28 %
12 1b	-	83 %	-	81 %	33 %
13 1b	-	-	-	91 %	36 %
14 1b	-	-	-	100 %	28 %
15 1c	20 %	98 %	77 %	95 %	65 %
16 1c	30 %	85 %	95 %	84 %	57 %
17 1c	13 %	96 %	75 %	80 %	64 %
18 1c	11 %	73 %	62 %	100 %	79 %
∅	36 %	82 %	83 %	88 %	62 %

### Testszenarios mit keiner Kollision (2a bis 2d)

Die Vermeidung von Falschwarnungen ist ein wichtiger Aspekt, damit das System in der Praxis Akzeptanz findet. In den Szenarios 2a bis 2d wurden Fahrmanöver, in welchen der Stapler nur sehr knapp an Hindernissen vorbeifährt oder dreht (siehe Abbildung 6-30), untersucht. Dabei wurde der Abstand  $x$  von 0,1 m bis 0,4 m erhöht.

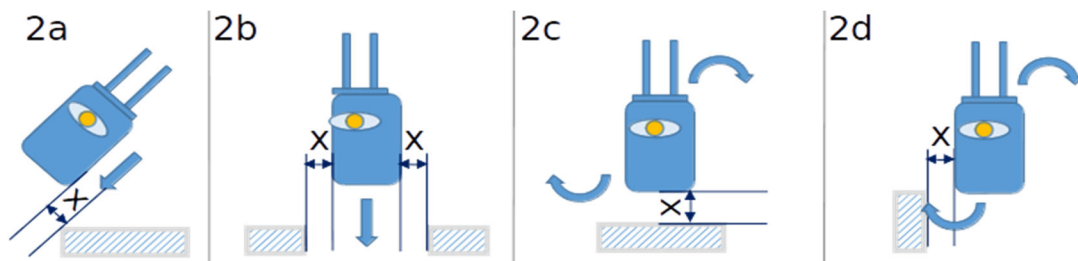


Abbildung 6-30: Schema für die Szenarien 2a bis 2d.

Bei den Durchläufen ohne Kollision wurden trotz der nicht vorhandenen Gefahr mit allen Algorithmen Kollisionsgefahren erkannt. Die niedrigste und damit beste Warnungsrate von 51 % wurde mit dem D-TVL-Algorithmus erreicht (siehe Tabelle 6-12). Das nächstbessere Ergebnis wurde mit dem Einsatz des BR-Algorithmus erreicht. Mit diesem wurde in 79 % der Durchläufe eine Kollisionsgefahr erkannt. Dieser schnitt bei den Testszenarios mit Kollisionen mit einer Warnungsrate von 58 % allerdings am schlechtesten ab. Die beiden besten Algorithmen bei der Erkennung von Kollisionen, LK und PD, schnitten in diesem Test am schlechtesten ab. Mit diesen wurden in über 99 bzw. 96 % der Fahrten eine Kollisionsgefahr erkannt. Beim Einsatz von FB ist die Warnungsrate von 92 % nur geringfügig besser.



## 6 Evaluation einzelner Algorithmen

Tabelle 6-12: Warnungsraten verschiedener Methoden des optischen Flusses für die Testszenarien 2a bis 2d.

Testfall	Szenario	Warnungsrate				
		BR	D-TVL	FB	LK	PD
19	2a	100 %	30 %	80 %	90 %	100 %
20	2a	100 %	0 %	100 %	100 %	100 %
21	2a	70 %	0 %	90 %	100 %	100 %
22	2a	0 %	0 %	60 %	100 %	60 %
23	2b	70 %	0 %	90 %	100 %	100 %
24	2b	80 %	0 %	100 %	100 %	100 %
25	2b	100 %	0 %	100 %	100 %	100 %
26	2b	40 %	0 %	90 %	100 %	100 %
27	2c	100 %	100 %	100 %	100 %	100 %
28	2c	100 %	100 %	90 %	100 %	100 %
29	2c	90 %	100 %	100 %	100 %	80 %
30	2c	50 %	90 %	100 %	100 %	100 %
31	2d	80 %	100 %	90 %	100 %	100 %
32	2d	100 %	100 %	100 %	100 %	100 %
33	2d	90 %	100 %	100 %	100 %	90 %
34	2d	100 %	100 %	80 %	100 %	100 %
∅		79 %	51 %	92 %	99 %	96 %
∅	(Kollisionen)	58%	84%	82%	100%	95%

Ungleich wichtiger als bei richtigen Kollisionsgefahren ist zur Bewertung der Algorithmen die Permanenz bei fälschlicherweise detektierten Kollisionsgefahren. Denn die Warnungsrate wäre auch bei 100 %, wenn nur in einem einzelnen Bild pro Durchlauf eine Kollisionsgefahr erkannt wird. Dies kann auch richtig sein, wenn die Bewegungsvektoren aufgrund eines geringen Lenkeinschlags oder Bodenunebenheit zwischen zwei Bildern eine Kollisionsgefahr vermuten lassen würden. Die Permanenz würde in dem Fall aber zeigen, dass es sich um eine äußerst kurz erkannte Kollisionsgefahr handelt.

Bei allen Algorithmen war eine geringere Permanenz bei den Fahrten ohne Kollisionen im Vergleich zu denen mit Kollision feststellbar. Die größte Minderung wurde beim FB- und PD-Algorithmus beobachtet. So sank die Permanenz von 83 auf 45 % mit dem FB- und von 62 auf 32 % mit dem PD-Algorithmus.

Tabelle 6-13: Durchschnittlicher Permanenz der Kollisionsgefahrerkennungen verschiedener Methoden des optischen Flusses für die Testszzenarien 2a bis 2d.

Testfall	Szenario	Permanenz				
		BR	D-TVL	FB	LK	PD
19	2a	27 %	65 %	59 %	84 %	75 %
20	2a	26 %	-	88 %	95 %	73 %
21	2a	14 %	-	71 %	81 %	53 %
22	2a	-	-	17 %	85 %	6 %
23	2b	20 %	-	51 %	84 %	65 %
24	2b	17 %	-	45 %	98 %	51 %
25	2b	17 %	-	48 %	100 %	27 %
26	2b	10 %	-	18 %	95 %	13 %
27	2c	23 %	78 %	25 %	69 %	28 %
28	2c	17 %	64 %	31 %	69 %	26 %
29	2c	12 %	42 %	32 %	79 %	21 %
30	2c	9 %	43 %	28 %	74 %	13 %
31	2d	45 %	59 %	48 %	78 %	30 %
32	2d	46 %	69 %	54 %	95 %	28 %
33	2d	55 %	93 %	61 %	85 %	19 %
34	2d	47 %	91 %	48 %	77 %	24 %
∅		26 %	67 %	45 %	84 %	35 %
∅	(Kollisionen)	36 %	82 %	83 %	88 %	62 %

Der Rechenaufwand ist zwar sekundär, da die Grafikkartenleistung sehr wahrscheinlich in der Zukunft weiter steigen wird, aber die Berechnung des D-TVL-Algorithmus wird mit einer Bildrate von 0,18 FPS (siehe Tabelle 6-14) auch langfristig weit unter der geforderten Bildrate von 30 FPS liegen. Daher wurde er bei den weiteren Untersuchungen nicht weiter betrachtet. Dies gilt wegen der schlechten Ergebnisse auch für den BR-Algorithmus.

Tabelle 6-14: Bildrate verschiedener Methoden des optischen Flusses in Bildern pro Sekunde (FPS).

∅	Bildrate (FPS)				
	BR	D-TVL	FB	LK	PD
∅	14,60	0,18	64,95	63,62	36,98

### Variation der Bildart

Für die ersten Tests der Kollisionsgefahrerkennung wurde das Intensitätsbild als Quelle für den optischen Fluss verwendet, da es vom gleichen Sensor wie das Tiefenbild generiert wird und demzufolge die beiden Bildarten keinen horizontalen oder vertikalen Versatz bzw. Verzerrung aufweisen. Im Falle des Farbbilds ist die Kalibrierung nicht exakt, daher passen Tiefenwerte zu Farbwerten nicht zwangsläufig überein. Darüber hinaus fehlt im kalibrierten Farbbild durch den unterschiedlichen Öffnungswinkel des Farbsensors ein Teil im oberen und unteren Bildbereich (siehe Abbildung 4-6).

Diese beiden Problematiken können zur fehlerhaften Berechnung des Scene-Flows führen.

Die Annahme, dass die Kollisionsgefahrerkennung mit dem Farbbild schlechter funktioniert, bestätigte sich nur teilweise für die Aufnahmen mit Kollisionen (siehe Tabelle 6-15). Mit dem FB-Algorithmus sank die Warnungsrate moderat und die Kollisionsgefahren wurden später erkannt. Im Falle des LK-Algorithmus traten nur geringfügige Änderungen in den Messwerten auf. Wohingegen sowohl die Warnungsrate, rechtzeitige Warnungen und die Permanenz mit dem PD-Algorithmus im Farbbild höher lag. Für diesen Algorithmus ist demzufolge das Farbbild besser geeignet.

Tabelle 6-15: Vergleich der Warnungsrate, Rechtzeitigkeit und Permanenz verschiedener Methoden des optischen Flusses im Intensitäts- und im Farbbild für die Szenarien 1a bis 1c.

<b>Vergleich Intensitätsbild - Farbbild</b>			
<b>Warnungsrate</b>			
	FB	LK	PD
<b>Intensitätsbild</b>	82 %	100 %	95 %
<b>Farbbild</b>	77 %	99 %	97 %
<b>Rechtzeitige Warnung</b>			
<b>Intensitätsbild</b>	92 %	99 %	85 %
<b>Farbbild</b>	89 %	99 %	88 %
<b>Permanenz</b>			
<b>Intensitätsbild</b>	83 %	88 %	62 %
<b>Farbbild</b>	90 %	92 %	66 %

Für die Aufnahmen mit Beinahe-Kollisionen, ergaben sich ebenfalls nur für den PD-Algorithmus bessere Ergebnisse (siehe Tabelle 6-16). Sowohl die Anzahl der Warnungen als auch die Permanenz der Warnungen sank moderat. Im Falle der anderen Algorithmen wurden fälschlicherweise mehr Kollisionen erkannt und diese zugleich mit höherer Permanenz.

Tabelle 6-16: Vergleich der Warnungsrate und Permanenz verschiedener Methoden des optischen Flusses im Intensitäts- und im Farbbild für die Szenarien 2a bis 2d.

<b>Vergleich Intensitätsbild - Farbbild</b>			
<b>Warnungsrate</b>			
	FB	LK	PD
<b>Intensitätsbild</b>	92 %	99 %	96 %
<b>Farbbild</b>	96 %	100 %	93 %
<b>Permanenz</b>			
<b>Intensitätsbild</b>	45 %	84 %	35 %
<b>Farbbild</b>	48 %	94 %	30 %

Mit Ausnahme des PD-Algorithmus sollte nach den Ergebnissen das Intensitätsbild verwendet werden. Beim PD-Algorithmus traten durch die Verwendung des kalibrierten Farbbilds zwar nur geringfügige, aber positive Änderungen auf.

### **Variation der Vorverarbeitung – Entfernung des Bodens und Hintergrunds**

Der Boden und der Hintergrund wurden in den vorangegangenen Tests im Farb- bzw. Intensitätsbild sowie im Tiefenbild entfernt. Dadurch sollten Fehlwarnungen durch falsche Berechnungen des optischen Flusses reduziert werden. Ein Vergleich mit der Auswertung ohne Anwendung dieser Vorverarbeitungsschritte soll zeigen, ob diese Annahme richtig war.

### **Testszenarios mit einer Kollision (1a bis 1c)**

Der Einfluss der Boden- und Hintergrundentfernung war über fast alle Parameter hinweg positiv (siehe Tabelle 6-17), aber sehr gering. Nur beim PD-Algorithmus war der Anteil rechtzeitiger Warnungen um einen Prozentpunkt geringer, wenn der Boden und Hintergrund entfernt wurde. Ein möglicher Grund für die spätere Erkennung könnte darin liegen, dass durch die Entfernung des Hintergrunds das Kollisionsobjekt geringfügig später im Bild sichtbar ist.

### **Testszenarios mit keiner Kollision (2a bis 2d)**

Analog zu den Szenarien mit Kollision wurden auch bei den Szenarien 2a bis 2d fälschlicherweise mehr Kollisionsgefahren bei zugleich höherer Permanenz erkannt, wenn der Boden und Hintergrund entfernt wurde.

## 6 Evaluation einzelner Algorithmen

Tabelle 6-17: Vergleich der Warnungsrate, rechtzeitigen Warnungen und Permanenz im Intensitätsbild mit und ohne Entfernung des Bodens und des Hintergrunds.

Algorithmus	Videos mit Kollisionen			Videos ohne Kollision		
	FB	LK	PD	FB	LK	PD
<b>Warnungsrate</b>						
ohne Boden & Hintergrund	82 %	100 %	95 %	92 %	99 %	96 %
mit Boden & Hintergrund	78 %	99 %	92 %	85 %	99 %	88 %
<b>Rechtzeitige Warnung</b>						
ohne Boden & Hintergrund	92 %	99 %	85 %			
mit Boden & Hintergrund	95 %	99 %	86 %			
<b>Permanenz</b>						
ohne Boden & Hintergrund	83 %	88 %	62 %	45 %	84 %	35 %
mit Boden & Hintergrund	76 %	84 %	55 %	40 %	77 %	26 %

Die Entfernung des Bodens und des Hintergrunds führt zwar zur Erhöhung von Falschwarnungen, allerdings auch zur Verbesserung der Erkennung von richtigen Kollisionsgefahren. Die Erkennung von Kollisionsgefahren ist als wichtiger einzustufen als die Vermeidung von falschen Erkennungen. Daher gilt für alle verwendeten OF-Algorithmen, dass der Boden und Hintergrund besser entfernt wird.

### Variation der Nachbearbeitung – RANSAC-Filterung

Im Abschnitt 4.2.5 wurde beschrieben, dass der Kollisionsgefahrerkennung eine RANSAC-basierte Filterung hinzugefügt wurde. Damit ist es möglich, Ausreißer aus den Kollisionsgefahrerechnungen, folgend auch als Kollisionsgefahrbilder bezeichnet, zur Vermeidung von falschen Erkennungen zu ignorieren. Außerdem kann auch die Warnungsrate richtiger Kollisionsgefahrerechnungen gesteigert werden, da die erkannten Kollisionspixel durch die Filterung eher einen zusammenhängenden Bereich bilden und somit den Schwellwert der distanznormierten Kollisionsfläche überschreiten. Letztlich wird die Filterung aber zu einer verspäteten Kollisionsgefahrerkennung führen.

Die Auswirkungen der RANSAC-Filterung wurden für das Intervall von drei, vier und fünf Kollisionsberechnungen untersucht. Im Folgenden werden nur zur Übersichtlichkeit die durchschnittlichen Werte aller Testfälle verglichen, eine vollständige Auflistung ist im Anhang A.2 zu finden.

### Test szenarios mit einer Kollision (1a bis 1c)

Für die Aufnahmen mit einer Kollision war zu erwarten, dass sich alle Werte mit Ausnahme der Permanenz verschlechtern, da über den RANSAC-Algorithmus zwar nicht

direkt ein Mittelwert gebildet wird, aber durch die Entfernung von Ausreißern etwaige Kollisionen nicht oder verspätet erkannt werden können. Die Auswertung in Tabelle 6-18 bestätigt diese Vermutung. Mit Ausnahme von FB sanken die Warnungsraten kontinuierlich mit zunehmendem Intervall. Beim FB-Algorithmus stieg die Warnungsrate ab einer Betrachtung von fünf Kollisionsbildern wieder an. Der Grund hierfür ist die erwähnte Konsolidierung der Kollisionspixel.

Die RANSAC-Filterung führte bei FB und PD zu einer Abnahme rechtzeitiger Warnung um 20 bzw. 63 %. Nur beim LK-Algorithmus änderte der Filter den Zeitpunkt der Warnung nur geringfügig.

Die Permanenz blieb mit allen Algorithmen nahezu gleich.

### Testszenarios mit keiner Kollision (2a bis 2d)

Die eigentliche Intention der Filterung ist, Falscherkennungen zu reduzieren. Dies gelang auch beim FB- und PD-Algorithmus. Es sank sowohl die Warnungsrate als auch die Permanenz in Korrelation zum Intervall. Beim LK-Algorithmus trat wie bei den Kollisionsaufnahmen keine signifikante Änderung auf. Im Falle des FB-Algorithmus konnte die Warnungsrate um bis zu 29 %, beim PD-Algorithmus um bis zu 28 % reduziert werden.

Tabelle 6-18: Ergebnisse der Anwendung eines RANSAC-Filters unter Verwendung der letzten drei, vier oder fünf berechneten Kollisionsbilder.

	Videos mit Kollisionen			Videos ohne Kollision		
	FB	LK	PD	FB	LK	PD
<b>Warnungsrate</b>						
<b>Ohne Filterung</b>	82 %	100 %	95 %	92 %	99 %	96 %
<b>Intervall: 3</b>	79 %	99 %	87 %	85 %	99 %	88 %
<b>Intervall: 4</b>	76 %	97 %	82 %	74 %	98 %	75 %
<b>Intervall: 5</b>	81 %	97 %	74 %	65 %	99 %	69 %
<b>Rechtzeitige Warnung</b>						
<b>Ohne Filterung</b>	92 %	99 %	85 %			
<b>Intervall: 3</b>	82 %	99 %	60 %			
<b>Intervall: 4</b>	77 %	99 %	46 %			
<b>Intervall: 5</b>	74 %	96 %	31 %			
<b>Permanenz</b>						
<b>Ohne Filterung</b>	83 %	88 %	62 %	45 %	84 %	35 %
<b>Intervall: 3</b>	78 %	87 %	60 %	40 %	77 %	26 %
<b>Intervall: 4</b>	80 %	84 %	62 %	34 %	79 %	21 %
<b>Intervall: 5</b>	79 %	85 %	62 %	29 %	74 %	20 %

Aufgrund der starken negativen Einflüsse auf die rechtzeitige Erkennung von Kollisionen ist der Einsatz der RANSAC-Filterung mit keinem Algorithmus empfehlenswert. Zwar sank die Warnungsrate beim FB-Algorithmus in den Szenarien 1a bis 1c nicht nennenswert, aber dafür in den Szenarien 2a bis 2d. Jedoch konnten deutlich weniger

Kollisionen rechtzeitig erkannt werden.

Der Einfluss auf den LK-Algorithmus war unerheblich, weshalb die Filterung weder signifikante Vor- noch Nachteile bietet.

Beim PD-Algorithmus verringerten sich die Werte in den Szenarien 1a bis 1c ähnlich wie in den Szenarien 2a bis 2d. Die Filterung ist demzufolge ebenso nicht empfehlenswert.

### **Variation der Auswertung – Mindestanzahl an Erkennungen pro Kollision**

Für erkannte Kollisionsgefahren wird in der Software eine Historie angelegt. Damit ist es möglich, Kollisionsgefahren erst als solche zu werten, wenn sie mehrere Male erkannt wurden. Die Vorgabe der Mindestanzahl kann dazu verwendet werden, Falscherkennungen zu mindern.

Die Auswertung der Verwendung unterschiedlicher Mindestanzahlen zeigt Tabelle 6-19. Die Auswirkung der mehrfachen Kollisionsgefahrerkennung ähnelt derer des RANSAC-Algorithmus, obgleich der RANSAC-Filter auf jegliche Kollisionspixel angewandt wurde und in dieser Versuchsreihe nur erkannte Kollisionscluster gefiltert wurden. Die Warnungsrate sank mit zunehmender Mindestanzahl an Kollisionsgefahren bei den Videos mit Kollisionsgefahr stärker als bei den Videos ohne Kollisionsgefahr. Der Zeitpunkt der Warnung verschlechterte sich mit Ausnahme eines Ausreißers mit dem FB-Algorithmus ebenso. Die Permanenz änderte sich über alle Videos hinweg nur geringfügig.

Tabelle 6-19: Vergleich der Auswirkung einer Mindestanzahl an Erkennungen (mD) für eine Kollision am Beispiel des Intensitätsbilds.

mD	Videos mit Kollisionen			Videos ohne Kollision		
	FB	LK	PD	FB	LK	PD
<b>Warnungsrate</b>						
1	97 %	98 %	95 %	97 %	100 %	96 %
2	82 %	100 %	95 %	92 %	99 %	96 %
3	84 %	98 %	93 %	86 %	100 %	95 %
4	78 %	97 %	92 %	91 %	100 %	93 %
5	72 %	99 %	88 %	88 %	100 %	88 %
<b>Rechtzeitige Warnung</b>						
1	94 %	100 %	92 %			
2	92 %	99 %	85 %			
3	88 %	98 %	85 %			
4	89 %	96 %	79 %			
5	84 %	97 %	71 %			
<b>Permanenz</b>						
1	85 %	90 %	64 %	46 %	89 %	36 %
2	83 %	88 %	62 %	45 %	84 %	35 %
3	85 %	86 %	62 %	49 %	86 %	34 %
4	85 %	83 %	63 %	54 %	88 %	32 %
5	84 %	91 %	61 %	52 %	85 %	29 %

Die Konsequenz der Durchläufe ist die gleiche wie bei der RANSAC-Filterung: Es sollten keine Kollisionsgefahren aufgrund der Anzahl an Erkennungen ignoriert werden. Dies würde zu weniger und späteren richtigen Warnungen bei nur mäßiger Reduzierung der Falschwarnungen führen.

### Vergleich mit statischer Kollisionsgefahrerkennung

Eines der Hauptgründe der Entwicklung waren die häufigen Fehlwarnungen von Kollisionsgefahrerkennungssystemen, welche nur den Abstand messen. Zum Vergleich der dynamischen Kollisionsgefahrerkennung wurde ebenfalls eine statische, abstands-basierte Erkennung implementiert. Wenn Objekte einer bestimmten Größe den Abstand  $r$  unterschreiten, wird es als Kollisionsgefahr interpretiert (siehe Abbildung 6-31).



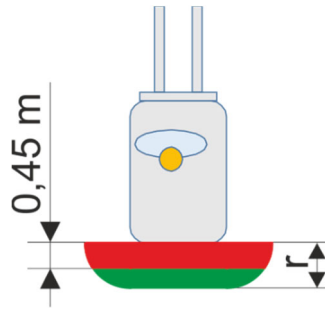


Abbildung 6-31: Schema zur statischen Kollisionsgefahrerkennung. Der grüne Bereich entspricht dem Warnbereich, der rote dem Totwinkelbereich.

Aufgrund der Montageposition und -pose der Kamera kann der Bereich bis zu 45 cm hinter dem Stapler nicht erfasst werden. Dadurch unterscheidet sich die Wirksamkeit bei besonders niedrigen Hindernissen etwas von den in Abschnitt 1.1 vorgestellten Kollisionswarnsystemen.

Da die erhältlichen Kollisionswarnsysteme keine konstanten, sondern konfigurierbare Mindestabstände aufweisen, wurden die Tests mit verschiedenen Abständen  $r$  ([m]) durchgeführt (siehe Tabelle 6-21).

Für die dynamische Kollisionsgefahrerkennung wurden alle Videos unter Berücksichtigung der Erkenntnisse aus den vorherigen Tests nochmals ausgewertet. Die dafür benutzten Konfigurationen sind folgende:

Tabelle 6-20: Optimale Konfigurationen zur Berechnung des optischen Flusses mittels ausgewählter Algorithmen.

	FB	LK	PD
<b>Bildart</b>	Intensitätsbild	Intensitätsbild	Farbbild
<b>Hinderung- &amp; Bodenentfernung</b>	An	An	An
<b>RANSAC-Filterung</b>	Aus	Aus	Aus
<b>Mindestanzahl an Kollisionen</b>	1	1	1

### Test szenarios mit einer Kollision (1a bis 1c)

Mit der statischen Kollisionsgefahrerkennung wurden nahezu alle Kollisionen erkannt. Nur im Fall von  $r = 0,5$  m wurden 15 % nicht erkannt. Der Grund hierfür ist der Totwinkelbereich. Vor allem bei niedrigen Hindernishöhen konnte hierbei keine Kollisionsgefahr erkannt werden, da der KLT beim Abstand von unter 0,5 m nur noch zu einem sehr kleinen Anteil im Sichtbereich der Kamera war.

Vor allem zum Zeitpunkt der Warnungen zeigte die statische Kollisionsgefahrerkennung schlechte Ergebnisse, da sie unabhängig von der Geschwindigkeit ist und daher die meisten Kollisionsgefahren zu spät erkannt wurden. Ein Faktor, welcher die rechtzeitige Erkennung über die Distanz zusätzlich verschlechtert, ist ein hardwaretechnisches Problem der Kinect v2: Die Kamera liefert nicht zuverlässig jedes Bild, d. h. gelegentlich werden Bilder ausgelassen, wodurch die Erkennung der Abstandsunterschreitung verspätet auftreten kann. Erst mit  $r = 3,0$  m war der Zeitpunkt der Warnung annähernd im Bereich der Methoden des optischen Flusses.

Die Permanenz der Kollisionsgefahrerkennung war erwartungsgemäß bei der statischen Messung höher. Wenn ein Objekt den Mindestabstand unterschritten hat, wird es, mit Ausnahme sehr kleiner Hindernisse, in jedem Bild erkannt.

### **Testszenarios mit keiner Kollision (2a bis 2d)**

Die Fahrten der Szenarien 2a bis 2d wurden maximal mit einem Abstand von 40 cm zum Hindernis durchgeführt. Daher wurden ab einem Warnungsabstand  $r = 1,0$  m nahezu alle Fahrten als Kollision erkannt. Bei einem Warnungsabstand von  $r = 0,5$  m wurden aufgrund des Totwinkelbereichs seitlich des Staplers nicht alle Kollisionen erkannt. Die Permanenz war bei allen Warnungsabständen ähnlich hoch wie in den Szenarien mit Kollisionsgefahren.

Tabelle 6-21: Vergleich der dynamischen und statischen (S) Kollisionsgefahrerkennung am Beispiel des Intensitätsbilds bzw. Tiefenbilds (statische Kollisionsgefahrerkennung). Die Variable  $r$  gibt bei der statischen Kollisionsgefahrerkennung den Warnungsabstand vom Stapler in Metern an.

Vergleich der dynamischen und statischen Kollisionsgefahrerkennung									
Videos mit Kollisionen (Szenarien 1a bis 1c)									
	FB	LK	PD	S( $r = 0,5$ m)	S( $r = 1,0$ m)	S( $r = 1,5$ m)	S( $r = 2,0$ m)	S( $r = 2,5$ m)	S( $r = 3,0$ m)
<b>War- nungs- rate</b>	82 %	100 %	97 %	85 %	100 %	100 %	100 %	100 %	100 %
<b>Rechtzei- tige War- nung</b>	92 %	99 %	88 %	0 %	6 %	51 %	66 %	80 %	87 %
<b>Per- manenz</b>	83 %	88 %	66 %	85 %	96 %	90 %	93 %	93 %	93 %
Videos ohne Kollisionen (Szenarien 2a bis 2d)									
	FB	LK	PD	S( $r = 0,5$ m)	S( $r = 1,0$ m)	S( $r = 1,5$ m)	S( $r = 2,0$ m)	S( $r = 2,5$ m)	S( $r = 3,0$ m)
<b>War- nungs- rate</b>	92 %	99 %	93 %	56 %	96 %	100 %	100 %	100 %	100 %
<b>Per- manenz</b>	45 %	84 %	30 %	74 %	93 %	88 %	94 %	97 %	99 %

Um eine ähnliche Qualität bei der statischen wie bei der dynamischen Kollisionsgefahrerkennung zu erreichen, wäre ein Warnungsabstand von 3 m notwendig. Dies würde in einer sehr hohen Anzahl in Falscherkennungen resultieren. Darüber hinaus würde sich der notwendige Mindestabstand weiter deutlich erhöhen, wenn der Stapler schneller als die betrachteten 6 km/h fährt.

### 6.3 Fazit

In diesem Kapitel wurden die Personen- und die Kollisionsgefahrerkennung separat evaluiert. Mit der besten SVM-Konfiguration konnte eine mittlere miss rate von 59 % erreicht werden. Darüber hinaus wurde die Abhängigkeit der Erkennungsleistung anhand der Verdeckung, Pose und Aktivität von Personen untersucht. Mit der SVM wurden vor allem nur bis zu 33 % verdeckte Personen, welche sich aufrecht bewegten, entdeckt. Während Personen, welche eine Ameise, einen Hubwagen, einen Rollbehälter oder einen Scooter bewegten bzw. fuhren, mit einer ähnlichen mittleren miss rate wie aufrechte Personen erkannt wurden, konnten Staplerfahrer oder Personen, die ein Regalwagen bewegten, kaum erkannt werden.

Es wurde auch eine Personenerkennung mittels neuronalen Netzen evaluiert. Das beste Resultat wurde mit einem auf Intensitätsbilder trainierten Detektor erreicht. Damit wurde eine miss rate von unter 10 % bei weniger als 0,01 Falscherkennungen (fppi)

pro Bild erreicht. Selbst Personen, welche über 66 % verdeckt waren, wurden zu 41 % nicht erkannt. Nur sitzende Personen wurden relativ schlecht (mittlere miss rate über 71 %) erkannt. Bei den Aktivitäten wurden Personen, die ein Regalwagen bewegten, mit einer miss rate von 53 % am schlechtesten erkannt. Die restlichen Aktivitäten wurden ähnlich oder teils auch besser als aufrechte Personen erkannt.

Zusammenfassend erkannte das neuronale Netz Personen deutlich besser als eine SVM und das bei minimalem Optimierungsaufwand. Allerdings lag die Bildrate bei 30 FPS, wohingegen über 170 FPS mit der SVM erreicht wurden. Mit zunehmender Grafikkartenleistung sollte dies in der Zukunft allerdings kein Hindernis mehr darstellen.

Für die Kollisionsgefahrerkennung wurden verschiedene Methoden des optischen Flusses evaluiert. Es wurden Aufnahmen ausgewertet, welche eine Kollision mit einem Hindernis aufweisen und welche bei denen der Stapler nah an einem Hindernis vorbeifährt.

Die Kollisionsgefahren wurden je nach eingesetztem Algorithmus bis zu 82 % - 100 % erkannt, wobei die Erkennung in bis zu 88 % bis 99 % rechtzeitig war. Rechtzeitig bedeutet, dass die Kollisionsgefahr zu dem Zeitpunkt das erste Mal erkannt wurde, dass der Stapler nach der Norm „DIN ISO 6292“ nach der ersten Kollisionsgefahrerkennung vor dem Hindernis zum Stillstand kommen kann. Allerdings wurden mit den Algorithmen auch die Fahrten vorbei am Hindernis meist als Kollision gewertet. Diese konnten weder mit einer RANSAC-Filterung noch mit einer Mindestanzahl an Erkennungen pro Kollision ohne eine Minderung richtiger Kollisionsgefahrerkennungen gesenkt werden. Der abschließende Vergleich mit einer distanzbasierten Kollisionsgefahrerkennung, welche der Funktionsweise bereits erhältlicher Systeme widerspiegelt, zeigte aber, dass die entwickelt Variante Kollisionsgefahren besser vorhersagen kann.



## 7 Evaluation des Gesamtsystems

---

Im vorigen Kapitel wurden verschiedene Methoden und Optimierungsmöglichkeiten für die Personen- und Kollisionsgefahrerkennung getrennt voneinander untersucht. In diesem Kapitel werden die daraus gewonnenen Erkenntnisse für das Gesamtsystem, also die kombinierte Anwendung der entwickelten Personen- und Kollisionsgefahrerkennung, genutzt. Dazu werden in einem ersten Schritt die aufgenommenen Szenarien 3a bis 3c, welche Kollisionen mit einer Schaufensterpuppe beinhalten, ausgewertet. Darauf folgt der Einsatz des Systems in Aufnahmen, welche über mehrere Tage hinweg im Praxisbetrieb eines Gabelstaplers gemacht wurden. Damit soll vor allem untersucht werden, inwieweit Falscherkennungen in der Praxis auftreten. Analog zu den vorherigen Testreihen wird auch in diesem Kapitel ein Vergleich zu einer statischen Kollisionsgefahrerkennung hergestellt.

### 7.1 Kollisionen mit Personenschaden

Bisher wurde die Kollisionsgefahrerkennung ohne aktivierte Kategorisierung (Kollisionsgefahr mit oder ohne Menschen) untersucht. Grundsätzlich ist die Funktionsweise der Erkennung von Kollisionsgefahren mit Menschen die gleiche, allerdings wird die notwendige Bremszeit um einen Sicherheitsaufschlag von 50 % erhöht. Dementsprechend muss das System die Kollision mit einem Menschen deutlich früher kennen. Daraus folgt die Anforderung, dass das System den Menschen mindestens einmal vor dieser Zeit erkannt hat, sodass auch eine Warnung entsprechend rechtzeitig ausgelöst werden kann.

In den Aufnahmen wurde eine Schaufensterpuppe als Repräsentant von Menschen verwendet. Im Folgenden sind Kollisionen mit einem Menschen demzufolge als eine Kollision mit dieser Puppe zu verstehen.

Im Testszenario 3a (siehe Abbildung 7-1) wurden Kollisionen, in welchen der Gabelstapler frontal auf den Menschen zu fuhr, mit verschiedenen Staplergeschwindigkeiten untersucht. Im zweiten Szenario 3b wurden ebenso frontale Zusammenstöße untersucht. Allerdings befand sich bei diesen ein Hindernis (KLT) zwischen dem Menschen und dem Stapler. Dabei wurde die Höhe des Hindernisses von 0,56 bis 2,24 m variiert. Anhand dieses Szenarios wurde in Abhängigkeit der Höhe des Hindernisses untersucht, wie das System derartige Kollisionsgefahren (mit/ohne Menschen) kategorisiert. Im letzten Szenario 3c wurden Fahrten in eine Gasse, in welcher sich ein Mensch befindet, geprüft.

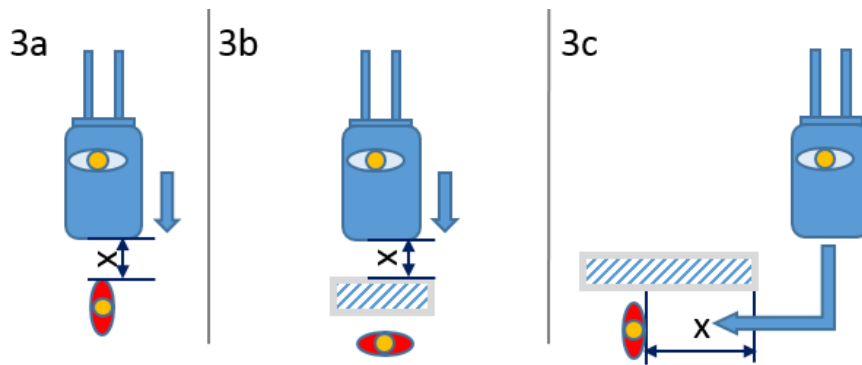


Abbildung 7-1: Testszenarien 3a bis 3c: a) Frontalzusammenstoß mit einem Menschen, b) Frontalzusammenstoß mit einem Hindernis unterschiedlicher Höhe, wobei sich ein Mensch hinter dem Hindernis befindet, c) Zusammenstoß mit einem Menschen, der sich in einer Gasse befindet.

Bei der Personenerkennung wurde im vorigen Kapitel angenommen, dass das System in einem Bereich von 0,001 bis 0,01 Falscherkennungen pro Bild (fppi) ausreichend funktioniert. Diese Annahme wurde bei den Kollisionen mit einem Menschen überprüft. Bei der Verwendung des DL-Algorithmus zur Personenerkennung wurde außerdem deren Funktionsfähigkeit getestet, wenn der Schwellwert für die Konfidenz so hoch gesetzt wird, dass keine Falscherkennungen mehr auftreten. Für die SVM war diese Untersuchung nicht möglich, da mit dieser immer Falscherkennungen zu verzeichnen waren.

Die Auswertung des Gesamtsystems in Tabelle 7-1 zeigt (detaillierte Auswertung ist im Anhang B), dass mit der SVM nahezu alle und mit DL alle Kollisionsgefahren unabhängig vom optischen Fluss Verfahren (FB, LK, PD) erkannt und richtig kategorisiert wurden.

Im Falle der Verwendung der SVM wurde in einem Video die Kollisionsgefahr nicht erkannt. Dabei handelte es sich um eine Kurvenfahrt (Szenario 3c), bei der erst nach dem Kontakt des Staplers mit der Schaufensterpuppe eine Kollisionsgefahr erkannt wurde. Dies wirkt sich analog auch auf den Zeitpunkt der Warnung aus. Daher wurden nur 94 bzw. 98 % der Kollisionsgefahren mit der SVM rechtzeitig erkannt. Der Wert für den Konfidenzschwellwert, welcher sich in den fppi widerspiegelt, hatte also nur einen geringen Einfluss auf den Erfolg der Kollisionsgefahrkategorisierung.

Bei Verwendung des DL-Algorithmus zur Personenerkennung wurden alle Kollisionen rechtzeitig erkannt, unabhängig von der Einstellung des Schwellwerts für Konfidenz. Selbst bei einem sehr hohen Schwellwert, bei dem in den Testreihen des letzten Kapitels keinerlei Falscherkennungen mehr auftraten, wurden keine Kollisionen falsch kategorisiert.

Tabelle 7-1: Konsolidierte Ergebnisse der Evaluation des Gesamtsystems für die Testszenarien 3a bis 3c.

		FB	LK	PD
<b>Personenerkennung</b>	fppi	Warnungsrate		
<b>SVM</b>	0,01	99 %	100 %	99 %
<b>SVM</b>	0,001	99 %	100 %	99 %
<b>DL</b>	0,01	100 %	100 %	100 %
<b>DL</b>	0,001	100 %	100 %	100 %
<b>DL</b>	0	100 %	100 %	100 %
		Rechtzeitig (Szenario 3a und 3b)		
<b>SVM</b>	0,01	94 %	100 %	100 %
<b>SVM</b>	0,001	98 %	99 %	99 %
<b>DL</b>	0,01	100 %	100 %	100 %
<b>DL</b>	0,001	100 %	100 %	100 %
<b>DL</b>	0	100 %	100 %	100 %
		Permanenz		
<b>SVM</b>	0,01	67 %	91 %	79 %
<b>SVM</b>	0,001	70 %	80 %	75 %
<b>DL</b>	0,01	81 %	100 %	84 %
<b>DL</b>	0,001	81 %	100 %	84 %
<b>DL</b>	0	81 %	100 %	84 %

Für die Testszenarios wurde außerdem ausgewertet, wie viele Gefahrensituationen das System als Kollisionsgefahr mit und ohne Menschen eingestuft hat. Im entwickelten System werden Kollisionsgefahren, welche als gefährlich für Menschen eingestuft wurden, eigentlich zu Objektkollisionen heruntergestuft, wenn über mehrere Bilder hinweg keine Person mehr in Kollisionsgefahrbereich erkannt wurde. Zu Auswertungszwecken wurde diese Funktion ausgeschaltet. Das heißt: In jedem einzelnen Bild wurde die Kollisionsgefahr ohne die Betrachtung der angelegten Kollisionsgefahrhistorie (siehe S.81 „Bestimmung der Historie“) neu bewertet.

Die Auswertung in Tabelle 7-2 zeigt die prozentuale Verteilung der Kollisionskategorien. Die Summe aus Mensch- und Objektkollisionsgefahren kann aufgrund von Rundungsfehlern von 100 % abweichen.

Erwartungsgemäß ist die Kategorisierung als Menschekollision unter Verwendung der SVM durch die geringere Erkennungsleistung deutlich niedriger als es bei der Verwendung von DL der Fall ist. Der größte Unterschied tritt bei fppi = 0,001 und dem PD-Algorithmus auf: Der Anteil der Kategorisierung als Menschekollision beträgt mit DL mehr als das 3,5-fache im Vergleich zum Ergebnis mit der SVM.



## 7 Evaluation des Gesamtsystems

Tabelle 7-2: Auswertung der Testszenarien 3a bis 3c bezüglich der Kategorisierung als Kollisionen mit und ohne Menschgefährdung.

Personenerkennung	Warnungen vor Mensch				Warnungen vor Objekt		
	fppi	FB	LK	PD	FB	LK	PD
<b>SVM</b>	0,01	25 %	34 %	29 %	74 %	66 %	70 %
<b>SVM</b>	0,001	20 %	24 %	22 %	79 %	76 %	77 %
<b>DL</b>	0,01	69 %	63 %	82 %	31 %	37 %	18 %
<b>DL</b>	0,001	67 %	62 %	80 %	33 %	38 %	20 %
<b>DL</b>	0	48 %	48 %	52 %	52 %	52 %	48 %

Bei den Frontalzusammenstößen (Szenario 3a) wurden mit dem DL-Algorithmus nahezu 100 % aller Bilder in Kollisionssituationen als Menschkollisionen kategorisiert.

In Szenario 3b sank die Kategorisierung als Menschkollision sowohl unter Verwendung einer SVM als auch unter Verwendung von DL ab einer Hindernishöhe von 1,5 m signifikant ab (siehe Tabelle 7-3). Durch die Montagehöhe der Kamera war die Schaufensterpuppe hierbei noch zu einem Drittel sichtbar. Bei 2,24 m Hindernishöhe war nur noch ein Teil des Kopfes sichtbar. Trotzdem wurden in diesem Fall 27 % der Kollisionenbilder als Gefährdung eines Menschen mit DL kategorisiert.

Im letzten Szenario 3c wurden unabhängig vom Personenerkennungsalgorithmus ca. die Hälfte aller Kollisionenbilder als Menschkollisionen kategorisiert. Das lag hauptsächlich daran, dass die Schaufensterpuppe erst sehr spät im Bildbereich der Kamera zu sehen war.

Tabelle 7-3: Auswirkung der Hindernishöhe in Szenario 3b auf die Kategorisierung der Kollisionen als Mensch am Beispiel von fppi = 0,001 und dem PD-Algorithmus.

Testfall	Hindernishöhe	SVM	DL
<b>38</b>	0,28 m	38 %	99 %
<b>39</b>	0,56 m	42 %	96 %
<b>40</b>	1,12 m	2 %	40 %
<b>41</b>	2,24 m	1 %	27 %

Der Vergleich der statischen mit der dynamischen Kollisionsgefahrerkennung in Tabelle 7-4 zeigt, dass mit der statischen Variante erst ab einem Warnabstand von  $r = 2$  m alle Kollisionen rechtzeitig erkannt werden. Die Begründung für die Warnungsrate von unter 100 % bei der statischen Kollisionsgefahrerkennung mit einem Warnabstand von  $r = 0,5$  m ist die Gleiche wie in Abschnitt 6.2.2 (S. 157): Bei niedrigen Hindernissen (Szenario 3b) befindet sich das Hindernis im Totwinkelbereich der Kamera.

Tabelle 7-4: Vergleich der statischen (S) mit verschiedenen Warnabständen  $r$  [m] und der dynamischen Kollisionsgefahrerkennung (DL, niedriger Schwellwert) für Kollisionen mit Menschen (Szenarien 3a bis 3c).

	FB	LK	PD	S( $r = 0,5$ )	S( $r = 1,0$ )	S( $r = 1,5$ )	S( $r = 2,0$ )	S( $r = 2,5$ )	S( $r = 3,0$ )
<b>Warnungsrate</b>	100 %	100 %	100 %	78 %	100 %	100 %	100 %	100 %	100 %
<b>Rechtzeitige Warnung</b>	100 %	100 %	100 %	0 %	0 %	75 %	100 %	100 %	100 %
<b>Permanenz</b>	81 %	100 %	84 %	98 %	100 %	71 %	96 %	97 %	98 %

## 7.2 Durchführung im Industrieunternehmen

Die Kollisionstests im universitären Umfeld haben gezeigt, dass in den festgelegten Szenarien je nach OF-Algorithmus alle Kollisionen von dem System rechtzeitig erkannt werden können. Eine eindeutige Auswahl, welcher OF-Algorithmus am besten für ein Kollisionsgefahrerkennungssystem geeignet ist, konnte aufgrund der marginalen Unterschiede in der Erkennung nicht abschließend getroffen werden. Denn nicht nur die richtige Erkennung ist für die Praktikabilität des Kollisionsgefahrerkennungssystems entscheidend, sondern auch eine möglichst geringe Anzahl an Falscherkennungen. Aber auch bezüglich der Personenerkennung wurde nicht abschließend geklärt, ob z. B. die Erkennungsrate der SVM ausreicht, damit die Funktionsfähigkeit des Systems auch in der Praxis gewährleistet ist. Die höhere Rate an falschen Personenerkennungen der SVM im Vergleich zu DL könnte zur Erhöhung von falschen Kollisionsgefahrerkennungen führen. Diese zwei Punkte werden im Folgenden anhand von Aufnahmen eines Gabelstaplers im industriellen Betrieb behandelt.

### 7.2.1 Randbedingungen

Im Gegensatz zu den Versuchen im universitären Umfeld wurden die Aufnahmen im Realbetrieb nicht mit einem Gegengewichtsstapler, sondern mit einem Schubmaststapler durchgeführt. Dieser hatte andere Abmessungen, wodurch auch die Montagehöhe der Kamera nicht beibehalten werden konnte. Die Abmessungen wurden in der Konfiguration entsprechend angepasst, die Höhe der Kamera wird ohnehin automatisch vom System ermittelt.

Mit Ausnahme der Staplerabmessungen wurden alle Parameter aus den Ergebnissen der Untersuchungen in Kapitel 6 übernommen.

Im Lager, in welchem das System eingesetzt wurde, findet der Hauptverkehr der Gabelstapler zwischen einem Hub (siehe Abbildung 7-2 (1)) und einem Verschieberegalager (VSR) (2) statt. Die Route ist durch einen gemischten Betrieb gekennzeichnet, d. h. auch Menschen ohne Flurförderzeug sind auf dieser Strecke unterwegs. An dem Hub werden Paletten manuell beladen und anschließend in eines der VSR mit Staplern

transportiert. Der größte Anteil an Menschen befindet sich entlang der Hauptverkehrsroute am beschriebenen Hub.

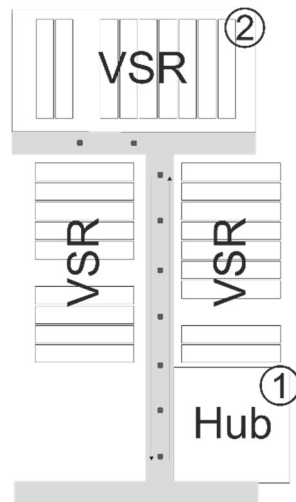


Abbildung 7-2: Hauptverkehrsroute des Gabelstaplerverkehrs im betrachteten Lager.

Im gesamten Lager war die Geschwindigkeit aller Flurförderzeuge auf maximal 8 km/h begrenzt. Demzufolge darf nach Abbildung 3-3 der Bremsweg eines Flurförderzeugs maximal 2,12 m bzw. die Bremszeit maximal 1,91 s betragen.

Der Boden des Lagers wies erhebliche Unebenheiten auf, dadurch waren vor allem die Farbbilder von einer deutlichen Unschärfe gekennzeichnet.

### 7.2.2 Systemaufbau

Wie bereits erwähnt, war in dem Lager kein Gegengewichtsstapler vorhanden. Die Montage des Systems wich daher von der in Abschnitt 4.1 beschriebenen ab. Für die Befestigung von Transformator, Laptop und Kamera wurde ein Gehäuse, welches auf dem Dach des Staplers (siehe Abbildung 7-3) befestigt wurde, gebaut. Die Montagehöhe der Kamera lag bei 3,38 m.

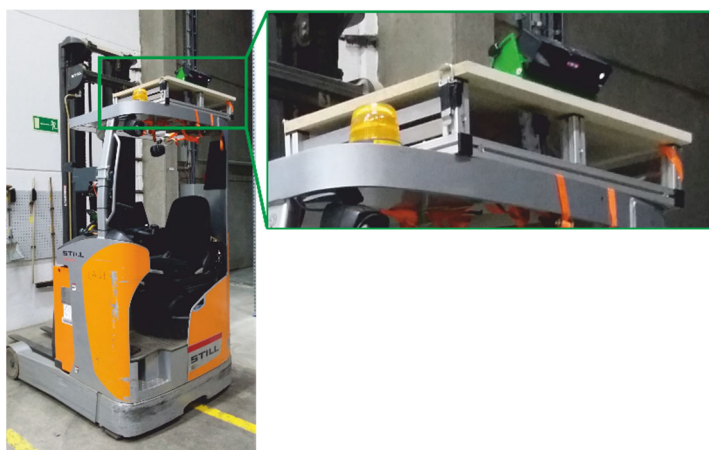


Abbildung 7-3: Montage des Systems auf dem Schubmaststapler „Still FX-17“.

### **7.2.3 Auswertungsmethode**

Für die Auswertung wurden zwei halbe und eine ganze Schicht eines Gabelstaplerfahrers aufgenommen. Es entstand 16 Stunden Datenmaterial. Allerdings war der Stapler nicht durchgehend im Einsatz. Nach der Entfernung der Standszenen verblieben 4,2 Stunden, in welchen der Stapler tatsächlich verwendet wurde. Kurze Szenen, in welchen der Stapler weniger als fünf Minuten stand, wurden beibehalten.

Aufgrund der Beschränkung der Staplergeschwindigkeit wurde der Grenzwert zur Auslösung einer Kollision auf maximal 1,91 s gesetzt. Bei Menschkollisionen wurde das System so eingestellt, dass es bis zu einer Kollisionszeit von 2,9 s (50 % Sicherheitszuschlag) warnt. Für die statische Kollisionsgefahrerkennung wurde der minimale Abstand des Staplers zu anderen Objekten auf 2,12 m eingestellt. Mit diesen Einstellungen ist nach Abbildung 3-3 ein Abbremsen bis zum Stillstand vor einem Zusammenstoß möglich.

Damit eine Aussage über die Qualität der Warnungen möglich ist, wurde jede detektierte Kollision einzeln betrachtet und kategorisiert. Entscheidende Kriterien hierfür waren: der Fahrzustand des Gabelstaplers, die Kollisionsursache und die Güte der Gefahr. Folgende Fahrzustände konnten beobachtet werden:

- Stapler steht
- Stapler fährt geradeaus
- Stapler fährt um eine Kurve
- Stapler wendet auf der Stelle

Als mögliche Ursachen wurden in der Auswertung folgende identifiziert:

- Stapler fährt auf Hindernis/Mensch zu (siehe Abbildung 7-4): Der Stapler fährt zeitweise auf ein Hindernis (geradeaus) zu. Das Hindernis war meist eine Wand, ein Regal oder eine Säule

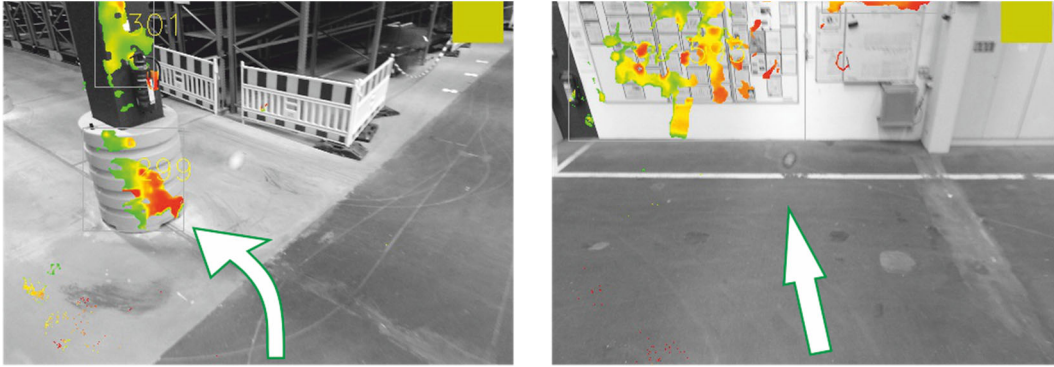


Abbildung 7-4: Beispiel für Fahrt in Richtung eines Hindernisses. Links: Kurvenfahrt in die Richtung einer Säule, rechts: Fahrt in die Richtung einer Wand.

- Stapler passiert Hindernis sehr nahe (siehe Abbildung 7-5): Vorbeifahren an einem Hindernis mit geringem Abstand ( $< 0,5$  m)

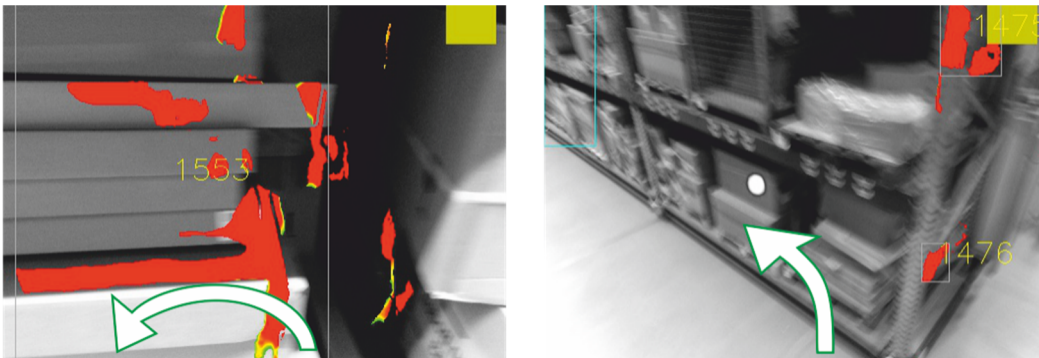


Abbildung 7-5: Beispiel für enges passieren eines Hindernisses: Links: Wenden in der Gasse vor einem Regal, rechts: Einfahrt in eine Gasse.

- Objekt/anderes Fahrzeug/Mensch bewegt sich auf Stapler zu (siehe Abbildung 7-6): Die Kollision entstand nicht durch die Bewegung des Staplers, sondern durch andere Objekte, welche sich auf den Stapler zubewegten. Beispiele für bewegte Objekte sind wackelnde Schilder oder ein bewegtes Verschieberegale



Abbildung 7-6: Beispiele für Hindernisse, welche sich auf den stehenden Stapler zubewegen. Links: Anderer Stapler fährt heran, mittig: Wackelndes Schild, rechts: Person bewegt sich auf Stapler zu.

- Anderes Fahrzeug passiert Stapler (siehe Abbildung 7-7): Eine Ameise fährt bspw. an dem Stapler vorbei oder ein anderer Stapler überholt

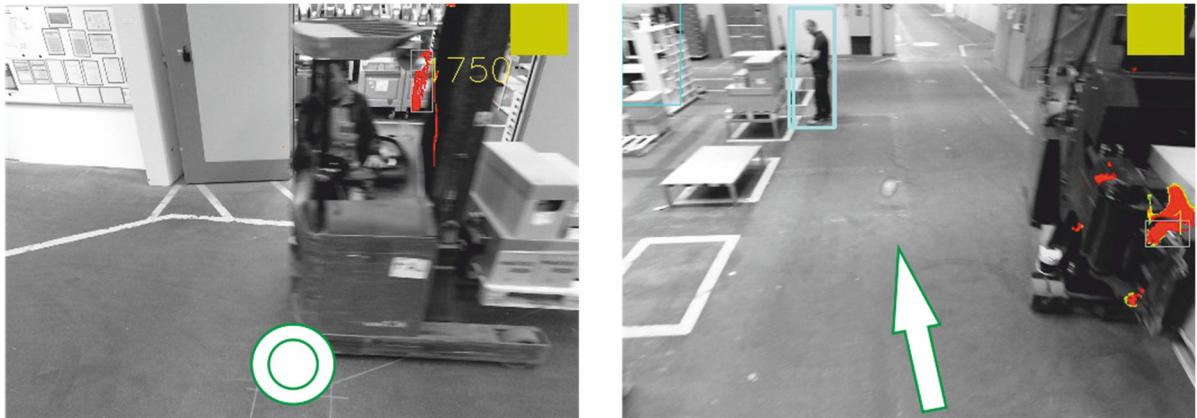


Abbildung 7-7: Beispiele für Kollisionsgefahrerkennungen durch vorbeifahrende Fahrzeuge. Links: Anderer Stapler fährt während dem Stand vorbei, rechts: Anderer Stapler fährt während der Fahrt vorbei.

- Bodenkollision (siehe Abbildung 7-8): Hervorgerufen durch Unebenheiten im Boden, wodurch der Stapler aufgrund der Bewegungsvektoren scheinbar mit dem Boden kollidiert

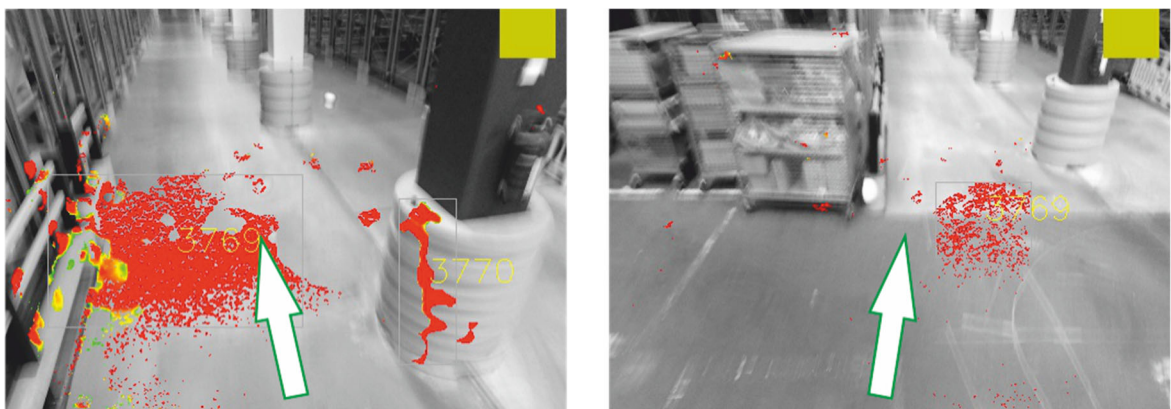


Abbildung 7-8: Beispiel für erkannte Kollisionen mit dem Boden.

- Sensorrauschen (siehe Abbildung 7-9): Hauptsächlich hervorgerufen durch nahe stationäre metallische Objekte wie z. B. Regale oder der Käfig eines anderen Staplers



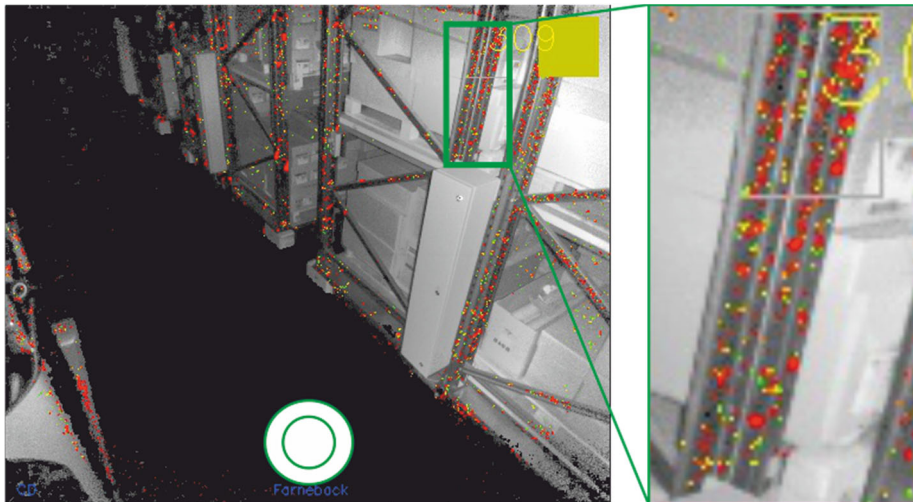


Abbildung 7-9: Beispiel für erkannte Kollisionen durch Sensorrauschen.

- Keine erkennbare Gefahr (siehe Abbildung 7-10): Meist durch Fehlberechnungen des optischen Flusses beim Drehen auf der Stelle



Abbildung 7-10: Beispiel für erkannte Kollisionen ohne ersichtlichen Grund.

Die erkannten Kollisionen wurden darüber hinaus eingeteilt in:

- Warnung richtig: Es kam nur zu keiner Kollision, weil der Gabelstaplerfahrer abbremste oder dem Hindernis auswich bzw. abbog
- Warnung richtig, aber unnötig: Die Kollisionsgefahr wurde richtig berechnet, allerdings ging von der Situation keine Gefahr aus. Hauptsächlich waren dies Kollisionsgefahren, in welchen der Stapler stand und sich andere Objekte auf diesen zubewegten
- Warnung falsch: Entweder durch Sensorrauschen verursachte Kollisionsgefahrerkennungen oder ohne ersichtlichen Grund. Letzteres trat vor allem bei der Rückwärtsfahrt auf

Die Auswertung wurde in den gleichen Variationen wie in den Versuchen aus Abschnitt 7.1 durchgeführt. Dementsprechend wurden alle Methoden des optischen Flusses als auch der Personenerkennung untersucht.

### **7.2.4 Ergebnisse**

Folgend werden die Ergebnisse der Anwendung des Systems auf Aufnahmen aus der Industrie dargelegt. Zuerst werden die verschiedenen Algorithmen des optischen Flusses gegenübergestellt. Anschließend folgt die Auswirkung der Nutzung unterschiedlicher Konfigurationen der Personenerkennung auf die Qualität der Kollisionsgefahrerkennung.

#### **Vergleich der OF-Algorithmen und statischen Kollisionsgefahrerkennung**

Für den Vergleich der OF-Algorithmen wurde als Personenerkennungsalgorithmus DL verwendet. Dabei wurde der höchste ermittelte Konfidenzschwellwert für Personen verwendet, welcher in den Testdurchläufen in Abschnitt 6.1.3 keine Falscherkennungen ( $f_{ppi} = 0$ ) aufwies.

Der Vergleich der Kategorie richtiger Warnungen in Tabelle 7-5 zeigt, dass die Anzahl erkannter Kollisionsgefahren im Gegensatz zu den Versuchen in Abschnitt 7.1 stark vom verwendeten OF-Algorithmus abhängig ist. Mit dem LK-Algorithmus wurden nur 61, mit dem FB-Algorithmus nur 236 Kollisionsgefahren erkannt. Dagegen wurden mit dem PD-Algorithmus 576 richtige Kollisionsgefahren erkannt. Neben den vom PD-Algorithmus erkannten Kollisionsgefahren konnten keine zusätzlichen Kollisionsgefahren in den Aufnahmen erkannt werden, daher stellt dieser Wert das Maximum dar. Vor allem aber die 14 kritischen Kollisionsgefahren mit einem Menschen wurden nur mit dem PD-Algorithmus zuverlässig erkannt.



Tabelle 7-5: Vergleich der OF-Algorithmen bei richtigen Warnungen.

Warnungsursache	Stapler steht	Stapler fährt geradeaus	Stapler fährt eine Kurve	Stapler dreht
<b>FB (N: 236)</b>				
Stapler fährt auf Hindernis zu	-	164	0	0
Stapler passiert Hindernis	-	21	21	30
Stapler fährt auf Mensch zu	-	0	0	0
<b>LK (N: 61)</b>				
Stapler fährt auf Hindernis zu	-	56	0	0
Stapler passiert Hindernis	-	0	4	1
Stapler fährt auf Mensch zu	-	0	0	0
<b>PD (N: 576)</b>				
Stapler fährt auf Hindernis zu	-	299	0	0
Stapler passiert Hindernis	-	137	31	95
Stapler fährt auf Mensch zu	-	14	0	0

Die nächste Kategorie umfasst erkannte Kollisionsgefahren, welche zwar richtig berechnet wurden, aber aufgrund der Situation keine akute Kollisionsgefahren festgestellt werden konnte. Der Tabelle 7-6 kann entnommen werden, dass es sich hierbei mit allen Algorithmen zum größten Teil um Kollisionsgefahren handelt, in welchen der Stapler stand und sich ein anderes Objekt auf den Stapler zubewegte. Eine Warnung wäre aufgrund des stehenden Staplers nicht notwendig.

Die einzigen erkannten und unnötigen Kollisionsgefahren, welche während der Fahrt auftraten, waren Bodenkollisionen. Diese traten vor allem mit dem PD-Algorithmus auf. Allgemein war die Anzahl richtig berechneter, aber unnötiger Warnungen bei allen Algorithmen des OF im Vergleich zu den als notwendig befundenen Warnungen nur unwesentlich geringer. Allerdings können die meisten dieser Warnungen – wie später in diesem Abschnitt beschrieben – mit bereits implementierten Filtern reduziert werden.

Tabelle 7-6: Vergleich der OF-Algorithmen bei richtigen, aber unnötigen Warnungen.

Warnungsursache	Stapler steht	Stapler fährt geradeaus	Stapler fährt eine Kurve	Stapler dreht
<b>FB (N: 179)</b>				
Objekt bewegt sich auf Stapler zu	50	0	0	0
Mensch bewegt sich in Richtung Stapler	55	0	0	0
Anderes Fahrzeug fährt an Stapler vorbei	55	0	0	0
Bodenkollision	0	19		0
<b>LK (N: 99)</b>				
Objekt bewegt sich auf Stapler zu	24	0	0	0
Mensch bewegt sich in Richtung Stapler	21	0	0	0
Anderes Fahrzeug fährt an Stapler vorbei	54	0	0	0
Bodenkollision	0	0		0
<b>PD (N: 336)</b>				
Objekt bewegt sich auf Stapler zu	80	0	0	0
Mensch bewegt sich in Richtung Stapler	115	0	0	0
Anderes Fahrzeug fährt an Stapler vorbei	62	0	0	0
Bodenkollision	0	79		0

Falsche Warnungen traten entweder aufgrund der falschen Berechnung der Bewegungsvektoren durch den OF-Algorithmus oder durch Sensorfehler auf. Bezüglich der Falschwarnungen schnitt der LK-Algorithmus am besten ab (siehe Tabelle 7-7). Über 90 % der falschen Warnungen dieses Algorithmus war durch Sensorrauschen bedingt. Das Sensorrauschen war auch die Hauptursache für Falschwarnungen bei den anderen OF-Algorithmen, wenn der Stapler stand. In Relation zu den richtigen Warnungen, sind die Warnungen bedingt durch Sensorrauschen beim FB-Algorithmus auffällig hoch. Dieser kompensiert die falschen Signale am schlechtesten. Für die restlichen Falschwarnungen des stehenden Staplers konnte keine Ursache für die jeweilige Meldung identifiziert werden.

Unter der Fahrt verminderte sich die Anzahl an Falschwarnungen durch Sensorrauschen bei allen OF-Algorithmen deutlich. In dieser Fahrsituation waren hauptsächlich andere Verkehrsteilnehmer ausschlaggebend für falsche Warnungen. Die Warnungen wurden überwiegend durch Objekte bzw. Menschen, welche sich auf den Stapler zubewegten, hervorgerufen. Diese Warnungen waren zwar prinzipiell richtig, allerdings traten diese Kollisionsgefahren nur in Einzelbildern auf, d. h. nur in einem einzigen Moment bestand diese Gefahr. Daher wurden diese Situationen als Falschwarnung eingestuft. Ein anderes Problem zeigt die Warnungsursache „Anderes Fahrzeug fährt an Stapler vorbei“: Die Ursache der falschen Warnungen war zumeist nicht die falsche

Berechnung des Bewegungsvektors des vorbeifahrenden Fahrzeugs, sondern eine Störung des Sensors durch den metallischen Rahmen.

Eine weitere Quelle für Falschwarnungen waren Drehungen des Staplers um die eigene Achse. In dieser Fahrsituation wurden der OF und demzufolge die Bewegungsvektoren falsch berechnet. Dieses Problem trat sowohl beim FB- als auch beim PD-Algorithmus vermehrt auf.

Tabelle 7-7: Vergleich der OF-Algorithmen bei falschen Warnungen.

Warnungsursache	Stapler steht	Stapler fährt geradeaus	Stapler fährt eine Kurve	Stapler dreht
<b>FB (N: 479)</b>				
Objekt bewegt sich auf Stapler zu	0	10	0	0
Mensch bewegt sich in Richtung Stapler	0	6	0	0
Anderes Fahrzeug fährt an Stapler vorbei	0	13	0	0
Sensorrauschen	351		4	
Keine erkennbare Gefahr	20	29	0	46
<b>LK (N: 120)</b>				
Objekt bewegt sich auf Stapler zu	0	1	0	0
Mensch bewegt sich in Richtung Stapler	0	0	0	0
Anderes Fahrzeug fährt an Stapler vorbei	0	2	0	0
Sensorrauschen	108		0	
Keine erkennbare Gefahr	3	5	0	1
<b>PD (N: 270)</b>				
Objekt bewegt sich auf Stapler zu	0	17	0	0
Mensch bewegt sich in Richtung Stapler	0	1	0	0
Anderes Fahrzeug fährt an Stapler vorbei	0	12	0	0
Sensorrauschen	164		1	
Keine erkennbare Gefahr	1	18	0	56

Zusammengefasst schnitt der PD-Algorithmus am besten ab, da mit diesem alle gefährlichen Situationen erkannt wurden. Insbesondere die gefährlichen Situationen, in welche Menschen involviert waren, wurden zuverlässig erkannt. Die anderen zwei Algorithmen haben in dem Industrierversuch deutlich schlechter abgeschnitten. Mit dem FB-Algorithmus wurde etwa die Hälfte aller gefährlichen Situationen erkannt, allerdings wurden auch mit großem Abstand am meisten Falschwarnungen erzeugt. Mit dem LK-

Algorithmen wurden zwar am wenigsten unnötige und falsche Warnungen produziert, allerdings wurden auch am wenigsten richtige Warnungen ausgelöst.

Trotz der guten Ergebnisse des PD-Algorithmus bei den richtigen Warnungen ist die Anzahl an unnötigen oder falschen Warnungen deutlich zu hoch. Viele dieser Falschwarnungen können wie bereits angedeutet softwaretechnisch reduziert werden. Die Möglichkeit zur Reduktion wurde bereits in dem System implementiert, aber für den Versuch bewusst deaktiviert, um die Qualität der OF-Algorithmen ohne Verfälschung zu vergleichen. Die unnötigen und falschen Warnungen mit den Ursachen „Sensorrauschen, Bodenkollision“ und in der Fahrsituation „Stapler steht“ können wie folgt gefiltert werden:

- Sensorrauschen: RANSAC-Filterung (siehe S. 79) und/oder Erhöhung der Mindestanzahl an Kollisionen zur Warnungsauslösung (siehe S. 154)
- Bodenkollision: Entfernung des Bodens und Erhöhung des Schwellwerts für die Höhe des Bodens (siehe S. 70)
- Stapler steht: Ignorieren von Kollisionen, wenn der Median-Bewegungsvektor des OF nahe Null ist

Werden diese Filter angewandt, so werden nach Tabelle 7-8 alle unnötigen Warnungen entfernt allerdings werden auch richtige Warnungen gemindert. Bei den Falschwarnungen verbleiben mit dem PD-Algorithmus 88, mit dem FB-Algorithmus 86.

Tabelle 7-8: Reduktion der Falschwarnungen durch implementierte Filter.

<b>Ohne Filter</b>			
	FB	LK	PD
<b>Richtige Warnungen</b>	236	61	576
<b>Richtige Warnungen, aber unnötig</b>	179	99	336
<b>Falschwarnungen</b>	479	120	270
<b>Mit Filter</b>			
	FB	LK	PD
<b>Richtige Warnungen, aber unnötig</b>	0	0	0
<b>Falschwarnungen</b>	88	8	86

Im Durchschnitt wird demnach mit Bezug auf die Aufnahmezeit von 4,2 Stunden alle drei Minuten eine Falschwarnung ausgelöst. Zusätzlich zu diesen würde beim PD-Algorithmus im Durchschnitt alle 26 Sekunden vor einer etwaigen „richtigen“ Kollisionsgefahr gewarnt.

Für die praktische Anwendung ist die Anzahl der Warnungen trotz Filterung zu hoch. Es kann als unzumutbar angesehen werden, dass jede halbe Minute der Fahrer gewarnt wird. Daher ist das System nur in dem Fall praktisch verwendbar, wenn es derart konfiguriert ist, dass nur vor Kollisionen mit Personen gewarnt wird. In diesem Fall wären 14 Warnungen aufgetreten. Im Durchschnitt ergäbe dies alle 18 Minuten eine Warnung.

Die Bemühungen um eine Reduktion von Falschwarnungen bei Kollisionsgefahrerkennungssystemen beim Gabelstapler sind im Vergleich zu konventionellen Lösungen trotz alledem gelungen. Die Auswertung wurde auch mit einer statischen Kollisionsgefahrerkennung mit einem minimalen Abstand von 2,12 m durchgeführt. In diesem Fall wäre in über 70 % der gesamten Zeit gewarnt worden. Aufgrund der hohen Dauer der Warnungen war eine detaillierte Auswertung, wie sie für die dynamische Kollisionsgefahrerkennung gemacht wurde, nicht sinnvoll.

### **Vergleich der Personenerkennungsalgorithmen**

In den letzten beschriebenen Durchläufen wurde DL zur Personenerkennung mit dem höchsten Schwellwert, bei dem in den Auswertungen aus 6.1.3 keine Falscherkennungen mehr auftraten, verwendet. In den Testszenarien aus Abschnitt 7.1 wurde gezeigt, dass die SVM trotz der schlechteren Leistung bei der Erkennung von Personen (siehe Abschnitt 6.1.2) zur Kollisionskategorisierung nur geringfügig schlechter funktioniert als DL. Allerdings blieb nach der Untersuchung die Frage offen, ob durch den Einsatz einer SVM aufgrund der deutlich höheren Rate an Falscherkennungen (siehe Abschnitt 6.1.2) nicht zugleich auch mehr Kollisionsgefahren falsch erkannt werden. Diese Fragestellung sollte anhand von Industriefaufnahmen unter Verwendung des PD-Algorithmus geklärt werden.

Der Vergleich in Tabelle 7-9 zeigt analog zum Vergleich der OF-Algorithmen richtige, unnötige und falsche Warnungen in Abhängigkeit des verwendeten Personenerkennungsalgorithmus. Diese Ergebnisse sind auf Kollisionsgefahren beschränkt, welche einen Menschen als Kollisionsobjekt beinhalten; die restlichen Kollisionsgefahren unterscheiden sich nicht von den bereits in diesem Abschnitt besprochenen Ergebnissen. Im Falle einer „falschen Warnung“ wurde eine Kollisionsgefahr als Menschekollision kategorisiert, obwohl kein Mensch in Gefahr war. Für die Verwendung der SVM wurden zwei Schwellwerte für die Akzeptanz als Menschekennung, welche in Abschnitt 6.1.2 zu  $f_{ppi} = 0,001$  bzw.  $f_{ppi} = 0,01$  führten, verwendet.

Mit der SVM konnten bei  $f_{ppi} = 0,001$  nicht alle Mensch-Kollisionsgefahren erkannt werden. Lediglich neun von den 14 Gefahren wurden hierbei erkannt. Bei  $f_{ppi} = 0,01$  hingegen wurden alle Gefahren zuverlässig erkannt. Allerdings wird bei den unnötigen

Warnungen (Mensch bewegt sich auf den stehenden Stapler zu) ersichtlich, dass in diesen Situationen nur weniger als die Hälfte der Gefahren im Vergleich zu DL erkannt wurden. Das lag hauptsächlich daran, dass in diesen Situationen der jeweilige Mensch nur zu einem kleinen Teil sichtbar war und die SVM diese nur schlecht erkennt.

Mit der SVM wurden sechs (fppi = 0,001) bzw. 77 (fppi = 0,01) Kollisionsgefahren als Menschkollisionen eingestuft, obwohl kein Mensch in dieser Situation in Gefahr war. Diese Anzahl konnte durch den Einsatz von Filtern auf zwei (fppi = 0,001) bzw. 34 (fppi = 0,01) und die Anzahl unnötiger Warnungen auf null reduziert werden.

*Tabelle 7-9: Vergleich der Erkennung von Menschkollisionen in Abhängigkeit des Personenerkennungsalgorithmus (OF: PD). Falschwarnungen: Kategorisierung als Mensch- statt Objektkollision.*

<b>Ohne Filter</b>			
	DL (fppi = 0)	SVM (fppi = 0,001)	SVM (fppi = 0,01)
<b>Richtige Warnungen</b>	14	9	14
<b>Richtige Warnungen, aber unnötig</b>	115	45	60
<b>Falschwarnungen</b>	0	6	77
<b>Mit Filter</b>			
	DL (fppi = 0)	SVM (fppi = 0,001)	SVM (fppi = 0,01)
<b>Richtige Warnungen, aber unnötig</b>	0	0	0
<b>Falschwarnungen</b>	0	2	34

Der Einsatz der SVM als Personenerkennungsalgorithmus ist nach den Ergebnissen zu vermeiden, da entweder nicht alle Kollisionsgefahren mit Menschen zuverlässig erkannt werden oder aber die Anzahl an Warnungen durch falsche Warnungen verdreifacht wird. Grundsätzlich ist der Einsatz einer SVM allerdings möglich, sofern die Leistung des Systems für den Einsatz von DL nicht ausreicht.

## 7.3 Fazit

In diesem Kapitel wurde das Gesamtsystem auf die korrekte Funktionsweise in Test-szenarios im universitären und industriellen Umfeld überprüft.

Die Testszenarios im universitären Umfeld bestanden aus Szenen, in welchen Kollisionen mit einer Schaufensterpuppe in der Versuchshalle des Lehrstuhls fml nachgestellt wurden. Die Szenarien beinhalteten sowohl Frontalkollisionen als auch Kollisionen nach der Einfahrt in eine Gasse.

Es wurden alle Kombinationen aus den implementierten OF-Algorithmen und Personenerkennungsalgorithmen auf die Auslösung einer Warnung, die rechtzeitige Auslösung und der Permanenz der Warnung hin untersucht.

In nahezu allen Konstellationen wurde die Kollision rechtzeitig erkannt. Demzufolge

ging kein OF-Algorithmus als beste Lösung hervor. Auch bei den zwei Personenerkennungsalgorithmen war kein signifikanter Unterschied feststellbar, im schlechtesten Fall war die Kollisionsgefahrerkennung um einen Prozentpunkt bei der Verwendung der SVM im Vergleich zu DL schlechter. Der Vergleich mit einer statischen Kollisionsgefahrerkennung (d. h. eine Kollision wird durch das Unterschreiten eines Mindestabstands definiert) hat gezeigt, dass für dieselbe Qualität an Warnungen ein Mindestabstand von 2 m notwendig wäre.

In den universitären Versuchen wurde zwar die richtige Erkennung von Kollisionen untersucht, allerdings konnte mit diesen Versuchen keine Aussage darüber getroffen werden, inwieweit Falschwarnungen durch die Kombination der verschiedenen Algorithmen hervorgerufen werden. Da die Senkung an Falschwarnungen ein Primärziel des Systems ist, wurden Aufnahmen im Praxisbetrieb eines Gabelstaplers im Werk eines Unternehmens gemacht. Mithilfe dieser Aufnahmen wurden sowohl richtige Warnungen als auch falsche Warnungen identifiziert. Kollisionen gab es während der Aufnahmen keine, es wurden nur Kollisionsgefahren identifiziert und bewertet.

Bei den OF-Algorithmen fiel auf, dass im Vergleich zu den universitären Kollisionstests deutliche Unterschiede in der Erkennung von Kollisionsgefahren bestand. Der PD-Algorithmus funktionierte deutlich besser als die restlichen OF-Algorithmen. Mit diesem wurden alle Kollisionsgefahren erkannt, welche in den Aufnahmen vorhanden waren. Allerdings wurden auch mit dem PD-Algorithmus viele Warnungen ausgelöst, welche nicht notwendig gewesen wären, da der Stapler in der Situation nicht fuhr oder die Kollisionsgefahr nur in einem einzigen Moment bestand. Ebenso traten sehr viele Falschwarnungen auf, welche hauptsächlich durch Sensorrauschen hervorgerufen wurden. Mittels verschiedener vorgestellter Filterverfahren können die unnötigen und falschen Warnungen deutlich reduziert werden, allerdings würde trotzdem im Durchschnitt fast jede halbe Minute eine Warnung ausgelöst werden. Die meisten dieser „richtigen“ Warnungen waren aufgrund der engen Situation im Lager unvermeidbar. Daher ist der Einsatz des Systems mit der Konfiguration einer zweistufigen Warnung nicht empfehlenswert, da es zu viele Falschwarnungen gäbe. Für die Praxis wäre daher sinnvoller nur vor Personenschäden zu warnen. In diesem Fall hätte das System im Schnitt alle 18 Minuten eine Warnung ausgelöst.

Der Industrieversuch hat darüber hinaus gezeigt, dass der Einsatz einer SVM im Vergleich zu DL entweder dazu führt, dass Kollisionsgefahren mit Menschen weniger zuverlässig erkannt oder aber die Anzahl an falschen Kollisionswarnungen deutlich erhöht werden.

## Ergebnisse bezüglich der gestellten Forschungsfragen

Die Forschungsfragen konnten im Laufe dieser Arbeit wie folgt geklärt werden:

1. Welche Methoden der Bildverarbeitung können zur Kollisionsgefahrerkennung am Gabelstapler genutzt werden?

*Für die reine Kollisionsgefahrerkennung können Methoden des optischen Flusses bzw. Scene-Flows genutzt werden. Diese liefern die Bewegungsvektoren im Bildbereich, welche sich in das Welt-Koordinatensystem des Staplers umrechnen lassen.*

2. Wie können Kollisionsgefahren relativiert bzw. kategorisiert werden, sodass möglichst keine Fehlwarnungen entstehen?

*Die Relativierung von Kollisionsgefahren geschieht einerseits durch die Nutzung von Bewegungsvektoren; dadurch werden z. B. langsame Fahrgeschwindigkeiten des Staplers in der Bewertung der Kollisionsgefahr berücksichtigt. Andererseits kann das System über eine integrierte Personenerkennung unterscheiden, ob eine Kollision mit einem Menschen oder einem Objekt bevorsteht.*

3. Wie lassen sich die eingesetzten Methoden für den Anwendungsfall im intralogistischen Transport optimieren?

*Zum einen betrifft die Optimierung die Kollisionsgefahrerkennung. Erstens ist die Wahl des richtigen OF-Algorithmus entscheidend. Zweitens kann die Aufbereitung der Eingangsdaten (wie bspw. die Entfernung des Bodens und des Hintergrunds) die Qualität der Kollisionsgefahrerkennung erhöhen. Zum anderen wurden Wege gezeigt, wie die Personenerkennung am besten im Anwendungsszenario funktioniert. Während bei der SVM mehrere Optimierungsschritte notwendig sind, bedarf es für den DL-Detektor nur an geeigneten Trainingsdaten.*

4. Welche Fahrsituationen können mit einem kamerabasierten Kollisionswarnsystem abgedeckt werden?

*Die meisten identifizierten Fahrsituationen wie Geradeaus- oder Kurvenfahrten konnten mit dem entwickelten System abgedeckt werden. Eine Ausnahme bildet die Fahrsituation, wenn der Stapler auf der Stelle wendet; in dieser Situation wurden sowohl etwaige Kollisionen nur teilweise vorhergesagt und es traten vermehrt falsche Kollisionsgefahrerkennungen auf.*



## 7 Evaluation des Gesamtsystems

---

Mit der Klärung der Forschungsfragen wurden auch die festgelegten Forschungsziele erreicht. Das entwickelte System kann Kollisionsgefahren erkennen und bezieht die relative Geschwindigkeit zwischen Stapler und Hindernissen mit ein. Darüber hinaus ist es dazu in der Lage, Kollisionsgefahren zu kategorisieren (Kollisionsgefahren mit oder ohne Menschen).

## 8 Zusammenfassung und Ausblick

---

Der Fokus dieser Arbeit stand in der Entwicklung eines Kollisionsgefahrerkennungssystems, welches speziell für den Einsatz am Gabelstapler konzipiert ist. Die Recherche zum Stand der Technik ergab, dass die Verwendung neuer Algorithmen aus dem Bereich des maschinellen Sehens (engl.: „computer vision“) und Lernens (engl.: „machine learning“) in Verbindung mit 3D-Kameras das größte Potenzial zur Erkennung von Kollisionsgefahren bietet.

Auf Basis der Recherche wurden Anforderungen an ein Kollisionsgefahrerkennungssystem für Gabelstapler erstellt und entsprechend ein Konzept abgeleitet. Das Hardwarekonzept beinhaltet eine Recheneinheit und eine 3D-Kamera. Aus Gründen der Flexibilität und Portabilität wurde ein Laptop als Recheneinheit verwendet.

Das Softwarekonzept beinhaltet die Algorithmen, welche zur Erkennung von Kollisionsgefahren notwendig sind. Der zusätzliche Einsatz einer Personenerkennung ermöglicht es vor Kollisionsgefahren mit Hindernissen auf eine andere Art und Weise zu warnen als vor Kollisionsgefahren mit Menschen.

Für die Kollisionsgefahrerkennung wurden Methoden des optischen Flusses (OF) zur Berechnung der Bewegungsvektoren zwischen Stapler und Umgebung genutzt. Anhand dieser Bewegungsvektoren wird die Kollisionszeit jedes Pixels der Umgebung mit dem Stapler berechnet. Anschließend wird auf Basis der zusammengefassten Kollisionspixel entschieden, ob eine Kollision bevorsteht.

Die Personenerkennung nutzt Methoden des maschinellen Lernens für die Lokalisierung von Personen im Sichtfeld bzw. Fahrweg des Staplers. Im Detail wurden die „Support Vector Machine“ (SVM) und der leistungsoptimierte „Deep Learning“ (DL)-Detektor „YoloV3“ implementiert.

Für die Entscheidung, ob eine Kollision mit einem Menschen vorliegt oder mit einem anderen Hindernis, werden die Positionen erkannter Menschen mit den Positionen erkannter Kollisionscluster verglichen. Für Kollisionsgefahren mit der Beteiligung eines Menschen wird ein höherer Schwellwert für die maximale Kollisionszeit verwendet als jene mit einem Hindernis. Dadurch warnt das System früher, wenn ein Mensch in Gefahr ist. Die Warnung wird über ein akustisches Signal ausgegeben, wobei die Warnungsmodalität nicht im Fokus der Forschungsarbeit stand und daher nicht näher betrachtet wurde.

Nach der Fertigstellung der Implementierung und des Trainings der ML-Algorithmen wurden die Kollisions- und Personenerkennung einzeln evaluiert. Für die Kollisionsgefahrerkennung wurden mehrere Kollisionen und enge Fahrten mit stationären Hindernissen in verschiedenen Fahrsituationen (Geradausfahrt, Fahrt in einer Kurve, Drehen um die eigene Achse) aufgenommen und ausgewertet. Drei der verwendeten OF-Methoden erzielten sehr gute Ergebnisse bei den Testszenarien. Daher wurden sie für die spätere Evaluation des Gesamtsystems ausgewählt.

Die Personenerkennungsalgorithmen wurden in verschiedenen Bildarten (Farb-, Intensitäts- und Tiefenbild), auf ihre Funktionalität hin überprüft. Dabei wurde nicht nur betrachtet, wie gut Personen auf den Testdaten erkannt wurden, sondern auch, wie viele Falscherkennungen auftraten. In diesen Versuchen lieferte der eingesetzte DL-Detektor erwartungsgemäß deutlich bessere Ergebnisse als die beste SVM, jedoch war dafür mehr die als zehnfache Rechenleistung notwendig. Mit dem DL-Detektor wurden gemittelt 97 % der Personen (maximal zu 33 % verdeckt) erkannt, mit der SVM bestenfalls 41 %.

Mit der darauffolgenden Evaluation des Gesamtsystems wurde überprüft, ob die 41 %-ige Erkennungsrate der SVM für eine zuverlässige Kategorisierung von Kollisionsgefahren ausreicht. Für diesen Zweck wurden Kollisionen mit einer Schaufensterpuppe im universitären Umfeld aufgenommen. In 99 % der Testszenarien wurden mit der SVM die Kollisionen unabhängig vom OF-Algorithmus rechtzeitig erkannt und kategorisiert, im Fall von DL waren es durchweg 100 %. Mit diesen Ergebnissen wurde zwar die richtige Erkennung der Kollisionen in den Testszenarien bestätigt, allerdings ließ diese Untersuchung keine Aussage darüber zu, inwieweit fälschlicherweise nicht vorhandene Kollisionsgefahren vom System erkannt werden. Um dies zu überprüfen, wurden Aufnahmen im täglichen Betrieb eines Gabelstaplers in der Industrie gemacht. Einerseits zeigte die Auswertung mit diesen Aufnahmen, dass mit den unterschiedlichen OF-Algorithmen in der Praxis nicht alle Kollisionsgefahren mit derselben Robustheit erkannt wurden. Nur mit einem Algorithmus wurden alle Kollisionsgefahren zuverlässig erkannt. Andererseits traten sehr viele Falschwarnungen, hauptsächlich verursacht durch Sensorrauschen, auf. In der besten Konfiguration bedeutete dies, dass im Durchschnitt alle drei Minuten eine falsche Warnung und jede halbe Minute eine korrekte Kollisionsgefahr ausgegeben worden wäre. Vor allem die korrekten Kollisionsgefahren, z. B. wenn der Gabelstaplerfahrer in die Richtung einer Wand fuhr und erst kurz davor abbog oder anhielt, ließen sich nur schwer softwaretechnisch reduzieren. Auch der Einsatz der SVM ist in der Praxis nur als zweite Wahl anzusehen, wenn die Leistung der Recheneinheit nicht für DL ausreicht. Entweder wurden mit der SVM nicht alle Kollisionsgefahren mit Personen zuverlässig erkannt oder es traten vermehrt falsche Kollisionsgefahrwarnungen auf.

Zusammenfassend ist die Erkenntnis aus der Anwendung im industriellen Umfeld,

dass das System im zweistufigen Warnmodus – Warnung bei Hindernissen und Menschen – aufgrund vieler Falschmeldungen nicht praktikabel ist. Die einzig sinnvollen Konfigurationen für einen praktischen Einsatz sind, dass entweder nur vor Kollisionsgefahren mit Menschen gewarnt wird oder der Grenzwert für Kollisionen mit Objekten derart eingestellt wird, dass das System erst später warnt. Dadurch würden zwar Kollisionen mit Objekten nicht gänzlich vermieden werden, aber etwaige Schäden könnten gemindert und Falschmeldungen reduziert werden.

Der abschließende Vergleich mit herkömmlichen Systemen bekräftigte die Berechtigung der Forschung. Ein konventionelles (distanzbasiertes) System hätte in den Daten, welche in der Industrie aufgenommen wurden, in über 70 % der gesamten Fahrzeit gewarnt. Das entwickelte System lag bezüglich des Anteils von Warnungen gemessen an der Fahrzeit bei unter 3 % im zweistufigen Modus und bei 0,5 % im einstufigen (nur Warnung vor Kollisionsgefahren mit Menschen) Modus.

Die Forschung ist in diesem Thema aufgrund der großen Anzahl an falschen Warnungen noch nicht abgeschlossen. Eine Möglichkeit falsche Warnungen zu reduzieren wäre, z. B. einen ML-Algorithmus mit speziellen Fahrsituationen wie „der Stapler dreht sich um die eigene Achse“ zu trainieren. Diese Fahrsituationen könnten somit während der Fahrt kategorisiert und etwaige Kollisionsgefahren ignoriert werden. Dies könnte soweit reichen, dass der ML-Algorithmus die Fahrweise eines jeweiligen Gabelstaplerfahrers lernt und in Abhängigkeit von ihm und der Fahrsituation entscheiden kann, wann tatsächlich eine Gefahr vorliegt.

Neben den falschen Warnungen ist auch die Anbringung des Sensors in Abhängigkeit vom Staplertyp (Gegengewichtsstapler, Schubmaststapler, ...) ein weiteres Forschungsgebiet. Nach den analysierten Unfalldaten treten auch bei der Vorwärtsfahrt (in Richtung der Gabeln) häufig Unfälle auf. Im Falle des Gegengewichtsstaplers wurden im Rahmen der Arbeit aber nur Rückwärtsfahrten analysiert. Für diese Situationen ist die Anbringung einer Kamera nicht trivial und bedarf einer Konstruktion, welche die Kamera in eine Position verfährt, in der sie freie Sicht hat.

Die Anzahl der notwendigen Sensoren wurde in dieser Arbeit ebenso nicht geklärt; da die Verarbeitung der Daten sehr rechenaufwändig ist, können nicht unbegrenzt Sensoren verwendet werden. Eine Möglichkeit wäre bspw. ein drehbares Stativ, auf dem die Kamera montiert ist. Die Drehung könnte damit simultan zur Lenkbewegung gesteuert werden. Zusätzlich könnte die Kamera nach unten schwenken, wenn der Stapler steht und höher, wenn der Stapler fährt. Diese Bewegungen müssten folglich auch in der Berechnung der Bewegungsvektoren berücksichtigt werden.



## Literaturverzeichnis

---

- [Abr-2012] Abramov, A.; Pauwels, K.; Papon, J.; Worgotter, F.; Dellen, B.: Depth-supported real-time video segmentation with the Kinect. In: IEEE (Hrsg.): 2012 IEEE Workshop on Applications of Computer Vision. IEEE, [Place of publication not identified], 2012, S. 457–464.
- [Acu-2017] Acure: Blaxtair. <https://blaxtair.com>, Aufruf am 02.06.2017.
- [Awa-2015] Awad, M.; Khanna, R.: Efficient learning machines – Theories, concepts, and applications for engineers and system designers. Apress, [New York, NY], 2015.
- [Bab-2009] Babenko, B.; Yang, M.-H.; Belongie, S.: Visual tracking with online multiple instance learning. *Computer Vision and Pattern Recognition*, 2009, S. 983–990.
- [Bar-2012] Barto, A. G.; Dietterich, T. G.: Reinforcement Learning and Its Relationship to Supervised Learning, 2012.
- [Bas-2017] Basler: Datenblatt Basler Time-of-Flight Camera, Aufruf am 22.01.2019.
- [Bay-2006] Bay, H.; Tuytelaars, T.; van Gool, L.: SURF. In: Leonardis, A.; Bischof, H.; Pinz, A. (Hrsg.): *Computer Vision - ECCV 2006*, Berlin, Heidelberg, 2006, S. 404–417.
- [Ben-2015] Benenson, R.; Omran, M.; Hosang, J.; Schiele, B.: Ten Years of Pedestrian Detection, What Have We Learned? In: Agapito, L.; Bronstein, M. M.; Rother, C. (Hrsg.): *Computer Vision - ECCV 2014 Workshops: Zurich, Switzerland, September 6-7 and 12, 2014, Proceedings, Part II*. Springer International Publishing, Cham, 2015, S. 613–627.
- [Ber-2014] Bernard Schmidt; Lihui Wang: Depth camera based collision avoidance via active robot control. In: *Journal of Manufacturing Systems*, Jg. 33 (2014) Nr. 4, S. 711–718.
- [Bli-2018] Blickfeld: Datenblatt Blickfeld Cube. <https://www.blickfeld.com/technology/>, Aufruf am 22.01.2019.

- [Bör-2018] Börold, A.; Lang, A.; Sherman, A.; Tödter, J., et al.: Forschungsbericht – PräVISION - Methodenentwicklung zur präventiven Steigerung der Arbeitssicherheit an Flurförderzeugen mit Umsetzung eines Assistenzsystems durch Fusion und Analyse von 2D- und 3D-Bilddaten, 2018.
- [Bör-2019] Börold, A.; Freitag, M.: Real-time environmental analysis for industrial vehicles based on synthetic sensor data and deep learning. In: Procedia CIRP, Jg. 81 (2019), S. 252–257.
- [Bra-2011] Bradski, G. R.; Kaehler, A.: Learning OpenCV – [computer vision with the OpenCV library]. O'Reilly, Beijing, 2011.
- [Bro-2004] Brox, T.; Bruhn, A.; Papenber, N.; Weickert, J.: High Accuracy Optical Flow Estimation Based on a Theory for Warping ECCV, 2004.
- [Car-2002] Carlson, B. S.: Comparison of modern CCD and CMOS image sensor technologies and systems for low resolution imaging SENSORS, 2002 IEEE, 2002, S. 171-176 vol.1.
- [Cio-2015] Cioran, A.: System Integration Testing of Advanced Driver Assistance Systems, KTH, Automatic Control, 2015.
- [Com-2016] Comnovo GmbH: Beeper; Keeper Static. <https://www.comnovo.de/>, Aufruf am 05.11.2018.
- [Dag-2004] Dagan, E.; Mano, O.; Stein, G. P.; Shashua, A.: Forward collision warning with a single camera IEEE Intelligent Vehicles Symposium, 2004, 2004, S. 37–42.
- [Dal-2005] Dalal, N.; Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: Schmid, C. (Hrsg.): CVPR. Conference on Computer Vision and Pattern Recognition; CVPR 2005; IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Los Alamitos, Calif. [u.a.], 2005, S. 886–893.
- [Dal-2006] Dalal, N.; Triggs, B.; Schmid, C.: Human Detection Using Oriented Histograms of Flow and Appearance. In: Leonardis, A.; Bischof, H.; Pinz, A. (Hrsg.): Computer Vision – ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part II. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, S. 428–441.

- [Dan-2014] Danelljan, M.; Khan, F. S.; Felsberg, M.; van de Weijer, J.: Adaptive Color Attributes for Real-Time Visual Tracking. *Computer Vision and Pattern Recognition - IEEE Conference on*, 2014, S. 1090–1097.
- [Deu-2011] Deutsche Gesetzliche Unfallversicherung (DGUV): Arbeitsunfallgeschehen 2010 – Statistik. Referat Statistik, 2011.
- [Deu-2012] Deutsche Gesetzliche Unfallversicherung (DGUV): Arbeitsunfallgeschehen 2011 – Statistik. Referat Statistik, 2012.
- [Deu-2013] Deutsche Gesetzliche Unfallversicherung (DGUV): Arbeitsunfallgeschehen 2012 – Statistik. Referat Statistik, 2013.
- [Deu-2014] Deutsche Gesetzliche Unfallversicherung (DGUV): Arbeitsunfallgeschehen 2013 – Statistik. Referat Statistik, 2014.
- [Deu-2015] Deutsche Gesetzliche Unfallversicherung (DGUV): Arbeitsunfallgeschehen 2014 – Statistik. Referat Statistik, 2015.
- [Deu-2016] Deutsche Gesetzliche Unfallversicherung (DGUV): Arbeitsunfallgeschehen 2015 – Statistik. Referat Statistik, 2016.
- [Deu-2017] Deutsche Gesetzliche Unfallversicherung (DGUV): Arbeitsunfallgeschehen 2016 – Statistik. Referat Statistik, 2017.
- [Deu-2018] Deutsche Gesetzliche Unfallversicherung (DGUV): Arbeitsunfallgeschehen 2017 – Statistik. Referat Statistik, 2018.
- [DIN 15185-2]: Flurförderzeuge – Sicherheitsanforderungen - Teil 2: Einsatz in Schmalgängen. DIN Nr. 15185-2, 2013.
- [DIN 33402-2]: Ergonomie - Körpermaße des Menschen - Teil 2: Werte. DIN Nr. 33402-2, 2005.
- [DIN ISO 6292]: Kraftbetriebene Flurförderzeuge und Schlepper - Bremsleistung und Komponentenfestigkeit. DIN ISO Nr. 6292, 2013.
- [Dol-2009] Dollar, P.; Wojek, C.; Schiele, B.; Perona, P.: Pedestrian detection. *CVPR 2009*, Los Alamitos, California, 2009, S. 304–311.
- [Dol-2012] Dollar, P.; Wojek, C.; Schiele, B.; Perona, P.: Pedestrian Detection: An Evaluation of the State of the Art. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Jg. 34 (2012) Nr. 4, S. 743–761.



- [ELO-2018] ELOKON GmbH: ELOprotect; ELOshield. <http://www.elokon.com>, Aufruf am 06.11.2018.
- [Ema-2012] Emaduddin, M.; Al-mutib, K.; Alsulaiman, M.; Hedjar, R.; Mattar, E.: Accurate floor detection and segmentation for indoor navigation using RGB+D and stereo cameras, 2012.
- [Enz-2009] Enzweiler, M.; Gavrilu, D. M.: Monocular pedestrian detection: survey and experiments. In: IEEE transactions on pattern analysis and machine intelligence, Jg. 31 (2009) Nr. 12, S. 2179–2195.
- [EUR-2019] EUROPEAN MATERIALS HANDLING FEDERATION: WORLD INDUSTRIAL TRUCK STATISTICS – INFORMATION SHEET Q4 – 2018. <https://www.fem-eur.com/wp-content/uploads/2019/03/IT-WITS-2018-Q4.pdf>, Aufruf am 07.05.2019.
- [Eve-2001] Everett, H. R.: Sensors for mobile robots – Theory and application. Peters, Wellesley, Mass., 2001.
- [Far-2003] Farneback, G.: Two-frame Motion Estimation Based on Polynomial Expansion Proceedings of the 13th Scandinavian Conference on Image Analysis, Berlin, Heidelberg, 2003, S. 363–370.
- [Fis-1999] Fischer, P.: Algorithmisches Lernen. Vieweg+Teubner Verlag; Imprint, Wiesbaden, 1999.
- [For-2008] Forsyth, D.; Torr, P.; Zisserman, A.; Grabner, H., et al. (Hrsg.): Semi-supervised On-Line Boosting for Robust Tracking – Computer Vision – ECCV 2008, 2008.
- [Gau-2015] Gaub, M.: Assistenzsysteme für Stapler – Was bringt's ? Vortrag, 6.9.2015.
- [Geh-2007] Gehrig, S. K.; Stein, F. J.: Collision Avoidance for Vehicle-Following Systems. In: IEEE Transactions on Intelligent Transportation Systems, Jg. 8 (2007) Nr. 2, S. 233–244.
- [Gel-2002] Gelbart, A.; Redman, B. C.; Light, R. S.; Schwartzlow, C. A.; Griffis, A. J.: Flash lidar based on multiple-slit streak tube imaging lidar. In: Kamerman, G. W., 2002, S. 9.
- [Gen-2011] Geng, J.: Structured-light 3D surface imaging – A tutorial. In: Adv. Opt. Photon., Jg. 3 (2011) Nr. 2, S. 128–160.

- [Gir-2015] Girshick, R.: Fast R-CNN 2015 IEEE International Conference on Computer Vision. IEEE International Conference on Computer Vision; ICCV, Piscataway, NJ, 2015, S. 1440–1448.
- [Gra-2006] Grabner, H.; Grabner, M.; Bischof, H.: Real-Time Tracking via Online Boosting. In: Chantler, M.; Fisher, B.; Trucco, M. (Hrsg.): British Machine Vision Conference 2006, 2006, S. 6.1-6.10.
- [Gup-2020] Gupta, S.; Chand, D.; Kavati, I.: Computer Vision based Animal Collision Avoidance Framework for Autonomous Vehicles, 2020.
- [Hag-2007] Hagebecker, D.-I. B.; Marketing, P.: A 3D time of flight camera for object detection. In: PMD Technologies GmbH (2007)
- [Han-2016] Handelsblatt: Gabelstapler-Hersteller steigert Umsatz. <https://www.handelsblatt.com/unternehmen/industrie/kion-gabelstapler-hersteller-steigert-umsatz/13511622.html?ticket=ST-3182512-ljWs7Zma5zxndqC4D7t5-ap2>, Aufruf am 28.08.2019.
- [Har-1988] Harris, C.; Stephens, M.: A combined corner and edge detector In Proc. of Fourth Alvey Vision Conference, 1988, S. 147–151.
- [Hay-2001] Hayashi, F.: Econometrics. Princeton University Press, Princeton, 2001.
- [He-2016] He, K.; Zhang, X.; Ren, S.; Sun, J.: Deep Residual Learning for Image Recognition 29th IEEE Conference on Computer Vision and Pattern Recognition. IEEE Conference on Computer Vision and Pattern Recognition; CVPR, Piscataway, NJ, 2016, S. 770–778.
- [He-2017] He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. B.: Mask R-CNN. In: CoRR, Jg. abs/1703.06870 (2017)
- [Hol-2011] Holz, D.; Holzer, S.; Rusu, R. B.; Behnke, S.: Real-Time Plane Segmentation Using RGB-D Cameras RoboCup, 2011.
- [Hor-1981] Horn, B. K.; Schunck, B. G.: Determining optical flow. In: Artificial Intelligence, Jg. 17 (1981) Nr. 1-3, S. 185–203.
- [Inn-2016] InnoSent: Datenblatt iSYS-3106. [https://www.innosent.de/fileadmin/media/dokumente/DATASHEETS\\_2016/170921\\_DataSheet\\_iSYS\\_3106.pdf](https://www.innosent.de/fileadmin/media/dokumente/DATASHEETS_2016/170921_DataSheet_iSYS_3106.pdf), Aufruf am 22.01.2019.

- [Int-2018] Intel: Datenblatt Realsense Depth Camera D435. [https://simple-core.intel.com/realsensehub/wp-content/uploads/sites/63/D435\\_Series\\_ProductBrief\\_010718.pdf](https://simple-core.intel.com/realsensehub/wp-content/uploads/sites/63/D435_Series_ProductBrief_010718.pdf), Aufruf am 22.01.2019.
- [Jai-2015] Jaimez, M.; Souiai, M.; Gonzalez-Jimenez, J.; Cremers, D.: A Primal-Dual Framework for Real-Time Dense RGB-D Scene FlowProc. of the IEEE Int. Conf. on Robotics and Automation (ICRA), 2015.
- [Jun-2018] Jungheinrich AG: Technische Daten "EFG 2"-Serie. <https://www.jungheinrich.de/produkte/neufahrzeuge/gabelstapler/elektro-gabelstapler/efg-213-215-216k-216-218k-218-220-102526>, Aufruf am 23.04.2019.
- [Kal-2010] Kalal, Z.; Mikolajczyk, K.; Matas, J.: Forward-backward error: Automatic detection of tracking failuresPattern recognition (ICPR), 2010 20th international conference on. IEEE, 2010, S. 2756–2759.
- [Kal-2012] Kalal, Z.; Mikolajczyk, K.; Matas, J.: Tracking-learning-detection. In: IEEE transactions on pattern analysis and machine intelligence, Jg. 34 (2012) Nr. 7, S. 1409–1422.
- [Kri-2012] Krizhevsky, A.; Sutskever, I.; Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural NetworksProceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, USA, 2012, S. 1097–1105.
- [Kru-2015] Kruse, R.; Borgelt, C.; Braune, C.; Klawonn, F., et al.: Computational Intelligence – Eine methodische Einführung in Künstliche Neuronale Netze, Evolutionäre Algorithmen, Fuzzy-Systeme und Bayes-Netze. Springer Vieweg, Wiesbaden, 2015.
- [Kub-2017] Kubat, M.: An Introduction to Machine Learning. Springer, Cham, 2017.
- [Lan-2017] Lang, A.; Fottner, J.: Konzeption eines kamerabasierten Kollisionswarnsystems zur Prävention von Arbeitsunfällen an Gabelstaplern. In: WGTl (Hrsg.): Tagungsband zum 13. Fachkolloquium der Wissenschaftlichen Gesellschaft für Technische Logistik e. V., 2017, S. 83–88.
- [Lea-2016] Leal-Taixe, Laura: Garching, Deutschland. Expertengespräch am 30.08.2016.

- [Lin-2014] Lin, T.-Y.; Maire, M.; Belongie, S. J.; Bourdev, L. D., et al.: Microsoft COCO – Common Objects in Context. In: CoRR, Jg. abs/1405.0312 (2014)
- [Lin-2018] Linde Material Handling AG: BlueSpot™. <https://www.linde-mh.de/de/Ueber-uns/Innovationen-von-Linde/Linde-BlueSpot%E2%84%A2.html>, Aufruf am 06.11.2018.
- [Liu-2016] Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C., et al.: SSD – Single Shot MultiBox Detector. In: Leibe, B., et al. (Hrsg.): Computer Vision - ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part I. Springer International Publishing, Cham, 2016, S. 21–37.
- [Llo-2011] Llorca, D. F.; Milanés, V.; Alonso, I. P.; Gavilan, M., et al.: Autonomous Pedestrian Collision Avoidance Using a Fuzzy Steering Controller. In: IEEE Transactions on Intelligent Transportation Systems, Jg. 12 (2011) Nr. 2, S. 390–401.
- [Low-1999] Lowe, D. G.: Object recognition from local scale-invariant features The proceedings of the seventh IEEE International Conference on Computer Vision, September 20-27, 1999, Kerkyra, Greece, Los Alamitos, Calif., 1999, S. 1150-1157 vol.2.
- [Luc-1981] Lucas, B. D.; Kanade, T. (Hrsg.): An Iterative Image Registration Technique with an Application to Stereo Vision (DARPA), 1981.
- [Moh-2017] Mohammed, A.; Schmidt, B.; Wang, L.: Active collision avoidance for human–robot collaboration driven by vision sensors. In: International Journal of Computer Integrated Manufacturing, Jg. 30 (2017) Nr. 9, S. 970–980.
- [Nix-2010] Nixon, M. S.; Aguado, A. S.: Feature extraction and image processing. Acad. Press, Amsterdam, 2010.
- [Occ-2015] Occupational Safety and Health Administration; Industrial Truck Association: Forklift Safety. <https://www.slideshare.net/oshasa-fetymanagement/opt-infographicforkliff>, Aufruf am 07.05.2019.
- [Ocu-2015] Ocular Robotics: RobotEye RE08 3D LIDAR. <http://www.ocularrobotics.com/wp/wp-content/uploads/2018/08/RobotEye-RE08-3D-LI-DAR-Datasheet.pdf>, Aufruf am 22.01.2019.

- [Pan-2017] Panasonic: Technische Daten 3D LiDAR. <https://industrial.panasonic.com/ww/products-rd/3dlidar>, Aufruf am 23.01.2019.
- [Par-2011] Parker, J. R.: Algorithms for image processing and computer vision – Previous ed.: 1997. Wiley Pub, Indianapolis, IN, 2011.
- [Pér-2013] Pérez, J. S.; Meinhardt-Llopis, E.; Facciolo, G.: TV-L1 optical flow estimation. In: Image Processing On Line, Jg. 2013 (2013), S. 137–150.
- [Rah-2018] Rahman, M.: Googles Night Sight. <https://www.flickr.com/photos/142977595@N02/>, Aufruf am 12.11.2018.
- [Ram-2017] Ramachandran, P.; Zoph, B.; Le V, Q.: Searching for Activation Functions. In: CoRR, Jg. abs/1710.05941 (2017)
- [Red-2016a] Redmon, J.; Farhadi, A.: YOLO9000 – Better, Faster, Stronger. In: ArXiv e-prints (2016)
- [Red-2016b] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A.: You Only Look Once The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
- [Red-2018] Redmon, J.; Farhadi, A.: YOLOv3 – An Incremental Improvement (2018)
- [Ren-2015] Ren, S.; He, K.; Girshick, R. B.; Sun, J.: Faster R-CNN – Towards Real-Time Object Detection with Region Proposal Networks. In: CoRR, Jg. abs/1506.01497 (2015)
- [Rob-1986] Robert K. McConnell: Method of and apparatus for pattern recognition. Patentschrift, 1986.
- [Sam-1959] Samuel, A. L.: Some Studies in Machine Learning Using the Game of Checkers. In: IBM Journal of Research and Development, Jg. 3 (1959) Nr. 3, S. 210–229.
- [Sch-1993] Schnell, G.: Sensoren in der Automatisierungstechnik. Vieweg+Teubner Verlag; Imprint, Wiesbaden, 1993.
- [Sch-2015] Schmidhuber, J.: Deep learning in neural networks – An overview. In: Neural networks : the official journal of the International Neural Network Society, Jg. 61 (2015), S. 85–117.

- [Sch-2017] Schaeffer, C.: Die FTS-Branche boomt. In: Logistik für Unternehmen (2017) Nr. 11-12
- [Sed-2017] Sedaghat-Pisheh, H.; Rivera, A. R.; Biaz, S.; Chapman, R.: Collision Avoidance Algorithms for Unmanned Aerial Vehicles Using Computer Vision. In: J. Comput. Sci. Coll., Jg. 33 (2017) Nr. 2, S. 191–197.
- [Sha-2014] Shankar, A.; Vatsa, M.; Sujit, P. B.: Collision avoidance for a low-cost robot using SVM-based monocular vision 2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014), 2014, S. 277–282.
- [Sie-2011] Siegwart, R.; Nourbakhsh, I. R.; Scaramuzza, D.: Introduction to autonomous mobile robots. MIT Press, Cambridge, Mass., 2011.
- [Sny-2017] Snyder, W. E.; Qi, H.: Fundamentals of computer vision, 2017.
- [Ste-2018] Stereolabs: Technische Daten ZED. <https://www.stereolabs.com/zed/>, Aufruf am 22.01.2019.
- [STI-2019a] STILL GmbH: Broschüre Elektrostapler. [https://data.still.de/assets/products/Vehicles/Electric\\_Forklift\\_Trucks/E-Stapler\\_HHd.pdf?mod=1553083161&download=1&s=eb3268f1bda0331df78db0e5b9fae1b5](https://data.still.de/assets/products/Vehicles/Electric_Forklift_Trucks/E-Stapler_HHd.pdf?mod=1553083161&download=1&s=eb3268f1bda0331df78db0e5b9fae1b5), Aufruf am 23.04.2019.
- [STI-2019b] STILL GmbH: Broschüre Schubmaststapler. [https://data.still.de/assets/products/Vehicles/Reach\\_Trucks/Lagertechnik\\_DE.pdf?mod=1555410271&download=1&s=34efef74a49f339e01b257234177169b](https://data.still.de/assets/products/Vehicles/Reach_Trucks/Lagertechnik_DE.pdf?mod=1555410271&download=1&s=34efef74a49f339e01b257234177169b), Aufruf am 23.04.2019.
- [STI-2020a] STILL GmbH: Broschüre Elektrogabelstapler. <https://www.still.de/fahrzeuge/gabelstapler-und-lagertechnik/elektrostapler.html>, Aufruf am 10.01.2021.
- [STI-2020b] STILL GmbH: Datenblatt OPX iGo neo. <https://www.still.de/intralogistik-systeme/automatisierungsloesungen/igo-neo.html>, Aufruf am 10.01.2021.
- [Str-2018] Strömgren, O.: Deep Learning for Autonomous Collision Avoidance, Linköping University, Computer Vision, 2018.

- [Suz-1985] Suzuki, S.; Abe, K.: Topological structural analysis of digitized binary images by border following. In: Computer Vision, Graphics, and Image Processing, Jg. 30 (1985), S. 32–46.
- [Tan-2013] Tan, Y.; Hassan, B.: A method for testing camera based advanced driving assistance systems. In: Hong, J.; Wang, M. Y. (Hrsg.): 2013 IEEE International Symposium on Assembly and Manufacturing (I-SAM), Piscataway, NJ, 2013, S. 151–154.
- [tbm-2017] tbm hightech control GmbH: RRW-207/3D; RAM-107; RRW-107plus. <http://www.tbm.biz>, Aufruf am 02.06.2017.
- [Tha-2018] Thamer, H.; Börold, A.; Benggolo, A.; Freitag, M.: Artificial intelligence in warehouse automation for flexible material handling, 2018.
- [U-T-2017] U-Tech GmbH: U-Tech. <http://www.u-tech-gmbh.de>, Aufruf am 02.06.2017.
- [Uwe-2007] Uwe Franke; Clemens Rabe; Stefan K. Gehrig: Kollisionsvermeidung durch raum-zeitliche Bildanalyse (Collision Avoidance based on Space-Time Image Analysis). In: it - Information Technology, Jg. 49 (2007) Nr. 1, S. 25–32.
- [van-2017] van Crombrugge, I.; Mertens, L.; Penne, R.: Fast Free Floor Detection for Range Cameras. In: Imai, F.; Tremeau, A.; Braz, J. (Hrsg.): VISAPP. International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, et al., [Setúbal], 2017, S. 509–516.
- [Vap-1963] Vapnik, V.; Lerner, A.: Pattern recognition using generalized portrait methodAutomation and Remote Control, 1963, S. 774–780.
- [Vom-2018] Vom Stein, A. M.; Dorofeev, A.; Fottner, J.: Visual collision warning displays in industrial trucks2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES), 2018, S. 1–7.
- [Wil-2005] Williams, A.; Barrus, S.; Morley, R. K.; Shirley, P.: An Efficient and Robust Ray-Box Intersection Algorithm. In: Journal of Graphics Tools, Jg. 10 (2005) Nr. 1, S. 49–54.
- [Win-2015] Winner, H.; Hakuli, S.; Lotz, F.; Singer, C.: Handbuch Fahrerassistenzsysteme – Grundlagen, Komponenten und Systeme für aktive Sicherheit und Komfort. Springer Vieweg, Wiesbaden, 2015.

- [Xue-2016] Xueming Wang; Jinhui Tang; Jianwei Niu; Xiaoke Zhao: Vision-based two-step brake detection method for vehicle collision avoidance. In: Neurocomputing, Jg. 173 (2016), S. 450–461.
- [Yu-2010] Yu, H.; Beard, R.; Byrne, J.: Vision-based navigation frame mapping and planning for collision avoidance for miniature air vehicles. In: Control Engineering Practice, Jg. 18 (2010) Nr. 7, S. 824–836.



## Verzeichnis der betreuten Studienarbeiten

---

Im Rahmen der durch diese Dissertation aufgegriffenen Forschungsthematik wurden nachfolgende Studienarbeiten erstellt. Die Anfertigung der Arbeiten wurde durch den Autor wissenschaftlich und inhaltlich angeleitet. Ein herzlicher Dank geht an die betreuten Studentinnen und Studenten für ihren Einsatz.

- [Bam-2017] Bamopoulos, F.: Computergestützte Erzeugung von perspektivischen Trainingsdaten für maschinelle Personenerkennungsalgorithmen. Masterarbeit. Lehrstuhl für Fördertechnik Materialfluss und Logistik, Technische Universität München, München, 2017.
- [Bar-2016] Bartneck, N.: Implementierung und Analyse von Verfahren des Maschinellen Lernens zur Echtzeit-Personenerkennung für die Fahrerassistenz. Semesterarbeit. Lehrstuhl für Fördertechnik Materialfluss und Logistik, Technische Universität München, München, 2016.
- [Ber-2017] Bernhardt, L.: Adaption von Deep Learning Algorithmen zur echtzeitfähigen Personenerkennung für sicherheitsrelevante Fahrerassistenzsysteme. Bachelorarbeit. Lehrstuhl für Fördertechnik Materialfluss und Logistik, Technische Universität München, München, 2017.
- [Daß-2018] Daßler, T.: Entwicklung einer auf Convolutional Neural Networks basierenden Personenerkennung für ein Kollisionswarnsystem am Gabelstapler. Masterarbeit. Lehrstuhl für Fördertechnik Materialfluss und Logistik, Technische Universität München, München, 2018.
- [Gil-2017] Gilch, T.: Bewegungsdatenerfassung für ein Gabelstapler-Assistenzsystem durch Reverse-Engineering der CAN-Kommunikation und Einsatz einer IMU. Masterarbeit. Lehrstuhl für Fördertechnik Materialfluss und Logistik, Technische Universität München, München, 2017.
- [Kai-2017] Kaiser, P.: Segmentierung und Visualisierung relativer Bewegungen aus 2D und 3D Bilddaten zur Warnung des Staplerfahrers vor Kollisionen. Interdisziplinäres Projekt. Lehrstuhl für Fördertechnik Materialfluss und Logistik, Technische Universität München, München, 2017.
- [Lou-2015] Lou, T.: Kollisionserkennung für Gabelstapler mittels TOF-Kameras und Lokalisierungsinformationen. Interdisziplinäres Projekt. Lehrstuhl

für Fördertechnik Materialfluss und Logistik, Technische Universität München, München, 2015.

- [Mai-2017] Maier, T.: Aufbau einer Echtzeit-Personenerkennung in einer Lagerumgebung unter dem Einsatz von neuronalen Netzen zur Bildverarbeitung. Masterarbeit. Lehrstuhl für Fördertechnik Materialfluss und Logistik, Technische Universität München, München, 2017.
- [Möl-2018] Möller, S.: Parameteroptimierung von Support Vector Machines mit neuronalen Netzen. Interdisziplinäres Projekt. Lehrstuhl für Fördertechnik Materialfluss und Logistik, Technische Universität München, München, 2018.
- [Tod-2017] Todorovska, T.: Evaluation eines kamerabasierten Sicherheitsassistenzsystems für Gabelstaplerfahrer. Interdisziplinäres Projekt. Lehrstuhl für Fördertechnik Materialfluss und Logistik, Technische Universität München, München, 2017.
- [Zwe-2017] Zwenig, A.: Entwicklung eines Algorithmus zur kamerabasierten Personenerkennung anhand von Tiefendaten. Bachelorarbeit. Lehrstuhl für Fördertechnik Materialfluss und Logistik, Technische Universität München, München, 2017.

# Abkürzungsverzeichnis

---

amr	mittlere Fehlerkennungsrate (engl.: „average miss rate“)
BB	Bounding Box
BGHW	Berufsgenossenschaft Handel und Warenlogistik
CPU	Central Processing Unit
DGUV	Deutsche Gesetzliche Unfallversicherung
DL	Deep-Learning
DNN	Deep-Neural-Network
FB	Farneback (Methode des optischen Flusses)
fml	(Lehrstuhl für) Fördertechnik Materialfluss Logistik (an der Technischen Universität München)
FTS	Fahrerloses Transportsystem
GPU	Graphical Processing Unit
IR-Bilder	Intensitätsbilder (oft als Infrarotbilder bezeichnet)
LK	Lucas-Kanade (Methode des optischen Flusses)
KOS	Koordinatensystem
ML	Maschinelles Lernen (engl.: „machine learning“)
NN	Neuronales Netz
PD	Primal-Dual (SF-Methode)

OF	Optischer Fluss (engl.: "optical flow")
ToF	Time-of-Flight
SF	Scene-Flow
SL-Kamera	Structured-Light-Kamera
SVM	Support Vector Machine



# Abbildungsverzeichnis

---

Abbildung 1-1: Entwicklung des Ordervolumens von manuell bedienten Gabelstaplern (Flurförderzeug Klasse 1) nach [EUR-2019].	2
Abbildung 2-1: Technologische Einordnung von Sensoren zur Distanz- und Texturerkennung.	8
Abbildung 2-2: a) Mögliche Diskriminanten zweier Klassen b) Optimales Grenzband mit größtmöglicher Breite. [Par-2011, S. 310].	18
Abbildung 2-3: Hülle der Feature-Vektoren und Support-Vektoren [Par-2011, S. 310].	19
Abbildung 2-4: Schematische Darstellung eines r-schichtigen Perzeptrons [Kru-2015, S. 44].	20
Abbildung 2-5: Vergleich verschiedener Aktivierungsfunktionen neuronaler Netze.	21
Abbildung 2-6: Anwendung des "Night Sight"-Filters von Google auf ein Bild [Rah-2018].	23
Abbildung 2-7: Beispiel für Schwellwertfilterung (H:217 ± 10, S: 80 ± 20).	24
Abbildung 2-8: Originalbild, normalisiertes Bild, egalisiertes Bild, jeweils mit Histogramm.	25
Abbildung 2-9: Beispiele für Weichzeichnen. (links nach rechts): Originalbild, Gaußsches Weichzeichnen, Weichzeichnen mit bilateralem Filter.	26
Abbildung 2-10: Beispiel zur Extraktion von Kanten.	27
Abbildung 2-11: Beispiel für Eckpunkte mittels Harris-Detektor [Har-1988].	27
Abbildung 2-12: Beispiel für Textur-Matching mittels SIFT-Merkmalen [Low-1999].	28
Abbildung 2-13: Differenz zweier aufeinander folgender Bilder (von links nach rechts): Bild 1, Bild 2, Differenzbild.	29
Abbildung 2-14: Darstellung des optischen Flusses, berechnet anhand von zwei konsekutiven Bildern [Abr-2012].	30
Abbildung 2-15: a) durchschnittliche Kanten des Trainingssets b) Positive Gewichtungen der SVM c) Negative Gewichtungen der SVM d) Beispielbild e) HOG-Merkmale des Beispielbilds f) durch SVM positiv gewichtete HOG-Merkmale des Beispielbilds g) durch SVM negativ gewichtete HOG-Merkmale des Beispielbilds [Dal-2005].	31
Abbildung 2-16: Architektur des AlexNet [Kri-2012].	32
Abbildung 2-17: Beispiel eines „residual layer“ [He-2016].	33

Abbildung 3-1: Entwicklung der Unfälle mit Gabelstaplern nach [Deu-2011; Deu-2012; Deu-2013; Deu-2014; Deu-2015; Deu-2016; Deu-2017; Deu-2018] [Deu-2013; Deu-2011; Deu-2014; Deu-2012; Deu-2015; Deu-2016; Deu-2017; Deu-2018].	42
Abbildung 3-2: Verteilung der Unfälle nach Aufenthaltsbereich des Geschädigten bei Rückwärts- bzw. Vorwärtsfahrt nach [Bör-2018, S. 7ff].	43
Abbildung 3-3: Maximaler Bremsweg und -zeit eines Staplers mit Berücksichtigung der Reaktionszeit.	44
Abbildung 3-4: Hardwarekonzept für ein Kollisionsgefahrerkennungssystem.	50
Abbildung 3-5: Module des Kollisionsgefahrerkennungssystems.	51
Abbildung 4-1: Beispielhafte Befestigung des Sensors an einem Gegengewichtsstapler.	56
Abbildung 4-2: Schema zur Kameraausrichtung.	57
Abbildung 4-3: Aufteilung der Berechnung auf CPU & GPU.	60
Abbildung 4-4: Klassendiagramm für Input.	61
Abbildung 4-5: Klassendiagramm für Preprocessing.	63
Abbildung 4-6: Vergleich der Farbbilder von links nach rechts: Originalbild, registriertes Farbbild (libfreenect2), kalibriertes Farbbild mit eigener Methode.	64
Abbildung 4-7: Umwandlung von Intensitätsbildern in den RGB-8-Bit-Bereich durch lineare Skalierung. Links: Linear konvertiertes Bild, rechts: Normalisiertes, linear konvertiertes Bild.	64
Abbildung 4-8: Links: Ergebnis der nichtlinearen Konvertierungsmethode. Rechts: Zusätzliche Entfernung der Vignetten-Abschattung.	66
Abbildung 4-9: Interpolation fehlender Daten. Links: Originaltiefenbild, rechts: interpoliertes Tiefenbild.	67
Abbildung 4-10: Anwendung des implementierten bilateralen Weichzeichnen-Filters auf egalisiertes Bild. Links: Originalbild, rechts: stark bilateral, weichgezeichnetes Bild.	67
Abbildung 4-11: Veranschaulichung des DL-Formats anhand der Umwandlung eines Farbbilds.	68
Abbildung 4-12: DL-konvertierte Bilder von links nach rechts: IR: Intensitätsbild (1 Kanal), IR3: Intensitätsbild (alle Kanäle), D: Tiefenbild (1 Kanal), D3: Tiefenbild (3 Kanäle), IRD: Infrarot (1. Kanal) & Tiefe (2. Kanal), IRDx: Infrarot & Tiefe kombiniert (1 Kanal), IRD3x: Infrarot & Tiefe kombiniert (3 Kanäle).	68
Abbildung 4-13: Schematische Darstellung unterschiedlicher Koordinatensysteme.	71
Abbildung 4-14: Schematische Darstellung der ausgewählten Linien für die Geradenschätzung [Lan-2017].	72

Abbildung 4-15: Ableitung des Kamerawinkels und -höhe aus den interpolierten Geraden [Lan-2017].	72
Abbildung 4-16: Klassendiagramm für Winkel- und Höhenerkennung.	73
Abbildung 4-17: Klassendiagramm zur Personenerkennung.	74
Abbildung 4-18: Personenerkennung mit einer SVM im Farbbild.	75
Abbildung 4-19: Personenerkennung mit einem Yolov3-Detektor im Farbbild.	75
Abbildung 4-20: Klassendiagramm der Kollisionsgefahrerkennung.	77
Abbildung 4-21: Veranschaulichung der Nutzung eines Quaders als Kollisionsobjekt zur Berechnung der Kollisionszeit einzelner Pixel.	78
Abbildung 4-22: Beispielhafte Situation mit mehreren Kollisionsquellen. Kollidierende Pixel sind grün (Kollision < 5 s) bis rot (Kollision < 1 s) eingefärbt.	79
Abbildung 4-23: Links: Clustering mit Konturensuche im Kollisionsbild, Cluster farblich unterschiedlich. Rechts: Überlagerung mit Intensitätsbild und Darstellung der validen Cluster.	80
Abbildung 4-24: Klassendiagramm der an der Ausgabe beteiligten Klassen.	82
Abbildung 5-1: Darstellung möglicher Bewegungskonstellationen im intralogistischen Transport.	88
Abbildung 5-2: Übersicht zu den Testszenarien der Kollisionsgefahrerkennung.	90
Abbildung 5-3: Beispiel für Bilderkategorien, von links nach rechts: Negativbild, Positivbild, markiertes Bild.	93
Abbildung 5-4: Screenshots des Markierungstools. Links: Verwendung von Kategorien – Mensch (grün) und Gabelstaplerfahrer (pink), rechts: Zuweisung der Verdeckung – 33 % Verdeckung → grau.	98
Abbildung 5-5: Ergebnis der automatischen Markierungsanpassung für das Intensitäts- bzw. Tiefenbild. Dargestellt sind die Bilder eines Aufnahmezeitpunktes. Oben: Farbbild, unten links: Intensitätsbild, unten rechts: Tiefenbild.	99
Abbildung 6-1: Grafische Auswertung nach Dollar et al.	106
Abbildung 6-2: Darstellung der Bildung der Rechtecke aus Schnitt- und Vereinigungsmenge erkannter und markierter Rechtecke. Links: Markierung (blau) und Erkennung (grün), rechts: Rechtecke der Schnitt- (rot) und Vereinigungsmenge (gelb).	107
Abbildung 6-3: Ergebnis der Variation der Detektorgröße.	111
Abbildung 6-4: Ergebnis der Variation der Bildquellen für die Positivtrainingsbilder.	114
Abbildung 6-5: Ergebnis der Variation der Bildquellen für die Negativtrainingsbilder.	115
Abbildung 6-6: Ergebnis der Variation der Anzahl an Positivtrainingsbildern.	117



Abbildung 6-7: Ergebnis der Variation des Verhältnisses trainierter Negativ- zu Positivbildern ( $np_*$ ).	118
Abbildung 6-8: Ergebnis der Variation der Skalierung des Eingangsbildes.	120
Abbildung 6-9: Ergebnis der Variation der Weichzeichnung des Eingangsbildes.	122
Abbildung 6-10: Ergebnis der Variation der IR-Konvertierungsmethode.	123
Abbildung 6-11: Ergebnis der Erkennungsleistung bei Vorverarbeitung des Intensitätsbildes durch Normalisierung und Egalisierung.	124
Abbildung 6-12: Ergebnis der Variation des SVM-Parameters $h_{scale}$ .	125
Abbildung 6-13: Ergebnis der Variation des SVM-Parameters $n_{levels}$ .	126
Abbildung 6-14: Ergebnis der Variation des SVM-Parameters „group threshold“.	127
Abbildung 6-15: Unterschied der Erkennungsleistungen der SVM nach Optimierung des Eingangsbildes und SVM-Parameter.	129
Abbildung 6-16: Beispiele für unterschiedliche Verdeckungsgrade von Personen.	129
Abbildung 6-17: Erkennungsleistung der SVM in Abhängigkeit der Verdeckung von Personen am Beispiel des Intensitätsbilds.	130
Abbildung 6-18: Erkennungsleistung der SVM in Abhängigkeit der Posen von Personen am Beispiel des Intensitätsbilds.	131
Abbildung 6-19: Beispiel auf der linken Hälfte für als falsch-positiv und falsch-negativ bewertete Erkennungen aufgrund der BB-Größe bei Einsatz einer SVM.	132
Abbildung 6-20: Bildstörung durch Sicherheitsweste. Links: Intensitätsbild, rechts: Farbbild.	132
Abbildung 6-21: Erkennungsleistung der SVM in Abhängigkeit des Tragens einer Sicherheitsweste.	133
Abbildung 6-22: Beispiele verschiedener Aktivitäten in Lagern: a) Ameise fahren, b) Scooter fahren, c) Gabelstapler fahren, d) Regalwagen ziehen/schieben, e) Rollbehälter ziehen/schieben, f) Hubwagen ziehen/schieben.	133
Abbildung 6-23: Erkennungsleistung der SVM in Abhängigkeit der Aktivitäten von Personen am Beispiel des Intensitätsbilds.	135
Abbildung 6-24: Ergebnisse der Variation der Bildarten mit einem DL-Detektor für Personen, die maximal zu 33 % verdeckt sind.	137
Abbildung 6-25: Unterteilung der DL-Detektionsergebnisse in Abhängigkeit des Verdeckungsgrads markierter Personen.	138
Abbildung 6-26: Unterteilung der DL-Detektionsergebnisse in Abhängigkeit der Pose markierter Personen.	139
Abbildung 6-27: Einfluss der Sicherheitsweste bei der DL-Detektion im Farb- und Intensitätsbild.	140

---

Abbildung 6-28: Unterteilung der DL-Detektionsergebnisse in Abhängigkeit der Aktivität markierter Personen.	141
Abbildung 6-29: Schema für die Szenarien 1a bis 1c.	144
Abbildung 6-30: Schema für die Szenarien 2a bis 2d.	147
Abbildung 6-31: Schema zur statischen Kollisionsgefahrerkennung. Der grüne Bereich entspricht dem Warnbereich, der rote dem Totwinkelbereich.	156
Abbildung 7-1: Testszenarios 3a bis 3c: a) Frontalzusammenstoß mit einem Menschen, b) Frontalzusammenstoß mit einem Hindernis unterschiedlicher Höhe, wobei sich ein Mensch hinter dem Hindernis befindet, c) Zusammenstoß mit einem Menschen, der sich in einer Gasse befindet.	162
Abbildung 7-2: Hauptverkehrsrouten des Gabelstaplerverkehrs im betrachteten Lager.	166
Abbildung 7-3: Montage des Systems auf dem Schubmaststapler „Still FX-17“.	166
Abbildung 7-4: Beispiel für Fahrt in Richtung eines Hindernisses. Links: Kurvenfahrt in die Richtung einer Säule, rechts: Fahrt in die Richtung einer Wand.	168
Abbildung 7-5: Beispiel für enges passieren eines Hindernisses: Links: Wenden in der Gasse vor einem Regal, rechts: Einfahrt in eine Gasse.	168
Abbildung 7-6: Beispiele für Hindernisse, welche sich auf den stehenden Stapler zubewegen. Links: Anderer Stapler fährt heran, mittig: Wackelndes Schild, rechts: Person bewegt sich auf Stapler zu.	168
Abbildung 7-7: Beispiele für Kollisionsgefahrerkennungen durch vorbeifahrende Fahrzeuge. Links: Anderer Stapler fährt während dem Stand vorbei, rechts: Anderer Stapler fährt während der Fahrt vorbei.	169
Abbildung 7-8: Beispiel für erkannte Kollisionen mit dem Boden.	169
Abbildung 7-9: Beispiel für erkannte Kollisionen durch Sensorrauschen.	170
Abbildung 7-10: Beispiel für erkannte Kollisionen ohne ersichtlichen Grund.	170
Abbildung A.1-1: Ergebnis der Variation der Detektorgröße.	A-1
Abbildung A.1-2: Ergebnis der Variation der Bildquellen für die Positivtrainingsbilder.	A-3
Abbildung A.1-3: Ergebnis der Variation der Bildquellen für die Positivtrainingsbilder.	A-5
Abbildung A.1-4: Ergebnis der Variation der Anzahl an Positivtrainingsbildern.	A-6

Abbildung A.1-5: Ergebnis der Variation des Verhältnisses trainierter Negativ- zu Positivbildern ( $np_*$ ).	A-7
Abbildung A.1-6: Ergebnis der Variation der Skalierung des Eingangsbildes.	A-9
Abbildung A.1-7: Ergebnis der Variation der Weichzeichnung des Eingangsbildes.	A-12
Abbildung A.1-8: Ergebnis der Variation der IR-Konvertierungsmethode.	A-13
Abbildung A.1-9: Ergebnis der Erkennungsleistung bei Vorverarbeitung des Intensitätsbildes durch Normalisierung.	A-14
Abbildung A.1-10: Ergebnis der Erkennungsleistung bei Vorverarbeitung des Intensitätsbildes durch Egalisierung.	A-15
Abbildung A.1-11: Ergebnis der Variation des SVM-Parameters $h_{scale}$ .	A-16
Abbildung A.1-12: Ergebnis der Variation des SVM-Parameters $n_{levels}$ .	A-20
Abbildung A.1-13: Ergebnis der Variation des SVM-Parameters „group threshold“.	A-23
Abbildung A.1-14: Ergebnis der Anwendung der optimierten SVM-Konfiguration.	A-26
Abbildung A.1-15: Erkennungsleistung der SVM in Abhängigkeit der Verdeckung von Personen am Beispiel des Intensitätsbilds.	A-27
Abbildung A.1-16: Erkennungsleistung der SVM in Abhängigkeit der Posen von Personen am Beispiel des Intensitätsbilds.	A-28
Abbildung A.1-17: Erkennungsleistung der SVM in Abhängigkeit des Tagens einer Sicherheitsweste.	A-29
Abbildung A.1-18: Erkennungsleistung der SVM in Abhängigkeit der Aktivitäten von Personen am Beispiel des Intensitätsbilds.	A-30
Abbildung A.1-19: Ergebnisse der Variation der Bildarten mit einem DL-Detektor für Personen, die maximal zu 33 % verdeckt sind.	A-1
Abbildung A.1-20: Unterteilung der DL-Detektionsergebnisse in Abhängigkeit des Verdeckungsgrads markierter Personen.	A-3
Abbildung A.1-21: Unterteilung der DL-Detektionsergebnisse in Abhängigkeit der Pose markierter Personen.	A-5
Abbildung A.1-22: Einfluss der Sicherheitsweste bei der DL-Detektion im Farb- und Intensitätsbild.	A-7
Abbildung A.1-23: Unterteilung der DL-Detektionsergebnisse in Abhängigkeit der Aktivität markierter Personen.	A-9

# Tabellenverzeichnis

---

Tabelle 2-1: Ausgewählte technische Daten verschiedener Sensortechnologien zur Erfassung der Umwelt.	11
Tabelle 2-2: Übersicht erhältlicher Kollisionswarnsysteme für Gabelstapler.	14
Tabelle 2-3: Ergebnisse der Nutzerwertanalyse über die Eignung für ein Kollisionsgefahrerkennungssystem. (1 – sehr schlecht, 3 – neutral, 5 – sehr gut)	38
Tabelle 3-1 Ausgewählte Körpermaßperzentile zur Beschreibung der minimalen bzw. maximalen Körperkontur [DIN 33402-2]:	45
Tabelle 3-2: Auflistung der Anforderungen an das Kollisionsgefahrerkennungssystem.	47
Tabelle 4-1: Beispielrechnung von $r_{\text{klein}}$ mit $\beta = 60^\circ$ , $x_{\text{klein}}=0,35$ m, $y_{\text{klein}}=1,54$ m, $y_{\text{groß}}=1,86$ m, $r_{\text{groß}}=0,95$ , $y_{\text{sensor}}=2,4$ m.	58
Tabelle 4-2: Beispielrechnung von $r_{\text{klein}}$ mit einem alternativen Sensor mit $\beta = 70^\circ$ , $x_{\text{klein}}=0,35$ m, $y_{\text{klein}}=1,54$ m, $y_{\text{groß}}=1,86$ m, $r_{\text{groß}}=0,95$ , $y_{\text{sensor}}=2,4$ m.	59
Tabelle 4-3: Parameter zur nichtlinearen Konvertierung eines 16-Bit-Intensitätswerts in einen 8-Bit-Farbwert.	65
Tabelle 5-1: Versuchsplan für die Kollisionstests (n. b.: nicht bestimmt).	91
Tabelle 5-2: Verwendete Bilddatenquellen für das Training und teilweise für die Evaluation der ML-Algorithmen.	95
Tabelle 6-1: Standard-Parameter für SVM-Training.	108
Tabelle 6-2: Standardeinstellungen des Systems für die Parametervariation.	109
Tabelle 6-3: Trainingskonfigurationen für Variation der Detektorgröße.	110
Tabelle 6-4: Trainingskonfigurationen für Variation der Positivbilder.	113
Tabelle 6-5: Trainingskonfigurationen für Variation der Negativbilder.	115
Tabelle 6-6: Trainingskonfigurationen für Variation der Anzahl der Trainingsbilder.	116
Tabelle 6-7: Trainingskonfigurationen für Variation des Verhältnisses zwischen Negativ- zu Positivbildern.	117
Tabelle 6-8: Zusammenfassung der Ergebnisse für das Farb- und Intensitätsbild zur Personenerkennung mit einer SVM.	128
Tabelle 6-9: Warnungsraten verschiedener Methoden des optischen Flusses für die Testszenarien 1a bis 1c.	144
Tabelle 6-10: Anteil rechtzeitiger Warnungen verschiedener Methoden des optischen Flusses für die Testszenarien 1a bis 1c.	146
Tabelle 6-11: Durchschnittliche Permanenz der Kollisionsgefahrerkennungen verschiedener Methoden des optischen Flusses für die Testszenarien 1a bis 1c.	146

Tabelle 6-12: Warnungsraten verschiedener Methoden des optischen Flusses für die Testszenarien 2a bis 2d.	148
Tabelle 6-13: Durchschnittlicher Permanenz der Kollisionsgefahrerkennungen verschiedener Methoden des optischen Flusses für die Testszenarien 2a bis 2d.	149
Tabelle 6-14: Bildrate verschiedener Methoden des optischen Flusses in Bildern pro Sekunde (FPS).	149
Tabelle 6-15: Vergleich der Warnungsrate, Rechtzeitigkeit und Permanenz verschiedener Methoden des optischen Flusses im Intensitäts- und im Farbbild für die Szenarien 1a bis 1c.	150
Tabelle 6-16: Vergleich der Warnungsrate und Permanenz verschiedener Methoden des optischen Flusses im Intensitäts- und im Farbbild für die Szenarien 2a bis 2d.	150
Tabelle 6-17: Vergleich der Warnungsrate, rechtzeitigen Warnungen und Permanenz im Intensitätsbild mit und ohne Entfernung des Bodens und des Hintergrunds.	152
Tabelle 6-18: Ergebnisse der Anwendung eines RANSAC-Filters unter Verwendung der letzten drei, vier oder fünf berechneten Kollisionsbilder.	153
Tabelle 6-19: Vergleich der Auswirkung einer Mindestanzahl an Erkennungen (mD) für eine Kollision am Beispiel des Intensitätsbilds.	155
Tabelle 6-20: Optimale Konfigurationen zur Berechnung des optischen Flusses mittels ausgewählter Algorithmen.	156
Tabelle 6-21: Vergleich der dynamischen und statischen (S) Kollisionsgefahrerkennung am Beispiel des Intensitätsbilds bzw. Tiefenbilds (statische Kollisionsgefahrerkennung). Die Variable $r$ gibt bei der statischen Kollisionsgefahrerkennung den Warnungsabstand vom Stapler in Metern an.	158
Tabelle 7-1: Konsolidierte Ergebnisse der Evaluation des Gesamtsystems für die Testszenarien 3a bis 3c.	163
Tabelle 7-2: Auswertung der Testszenarien 3a bis 3c bezüglich der Kategorisierung als Kollisionen mit und ohne Menschgefährdung.	164
Tabelle 7-3: Auswirkung der Hindernishöhe in Szenario 3b auf die Kategorisierung der Kollisionen als Mensch am Beispiel von $f_{ppi} = 0,001$ und dem PD-Algorithmus.	164
Tabelle 7-4: Vergleich der statischen (S) mit verschiedenen Warnabständen $r$ [m] und der dynamischen Kollisionsgefahrerkennung (DL, niedriger Schwellwert) für Kollisionen mit Menschen (Szenarien 3a bis 3c).	165
Tabelle 7-5: Vergleich der OF-Algorithmen bei richtigen Warnungen.	172
Tabelle 7-6: Vergleich der OF-Algorithmen bei richtigen, aber unnötigen Warnungen.	173

Tabelle 7-7: Vergleich der OF-Algorithmen bei falschen Warnungen.	174
Tabelle 7-8: Reduktion der Falschwarnungen durch implementierte Filter.	175
Tabelle 7-9: Vergleich der Erkennung von Menschkollisionen in Abhängigkeit des Personenerkennungsalgorithmus (OF: PD). Falschwarnungen: Kategorisierung als Mensch- statt Objektkollision.	177
Tabelle A.1-1: Rohdaten zur Variation der Detektorgröße.	A-1
Tabelle A.1-2: Rohdaten zur Variation der Bildquellen für die Positivtrainingsbilder.	A-3
Tabelle A.1-3: Rohdaten zur Variation der Bildquellen für die Positivtrainingsbilder.	A-5
Tabelle A.1-4: Rohdaten zur Variation der Anzahl an Positivtrainingsbildern.	A-6
Tabelle A.1-5: Rohdaten zur Variation des Verhältnisses trainierter Negativ- zu Positivbildern (np_*).	A-7
Tabelle A.1-6: Rohdaten zur Variation der Skalierung des Eingangsbildes.	A-9
Tabelle A.1-7: Rohdaten zur Variation der Weichzeichnung des Eingangsbildes.	A-12
Tabelle A.1-8: Rohdaten zur Variation der IR-Konvertierungsmethode.	A-14
Tabelle A.1-9: Rohdaten zur Erkennungsleistung bei Vorverarbeitung des Intensitätsbildes durch Normalisierung.	A-15
Tabelle A.1-10: Rohdaten zur Erkennungsleistung bei Vorverarbeitung des Intensitätsbildes durch Egalisierung.	A-16
Tabelle A.1-11: Rohdaten zur Variation des SVM-Parameters $h_{scale}$ .	A-16
Tabelle A.1-12: Rohdaten zur Variation des SVM-Parameters $n_{levels}$ .	A-20
Tabelle A.1-13: Rohdaten zur Variation des SVM-Parameters „group threshold“.	A-23
Tabelle A.1-14: Rohdaten zur Anwendung der optimierten SVM-Konfiguration.	A-26
Tabelle A.1-15: Rohdaten zur Erkennungsleistung der SVM in Abhängigkeit der Verdeckung von Personen am Beispiel des Intensitätsbildes.	A-27
Tabelle A.1-16: Rohdaten zur Erkennungsleistung der SVM in Abhängigkeit der Posen von Personen am Beispiel des Intensitätsbildes.	A-28
Tabelle A.1-17: Rohdaten zur Erkennungsleistung der SVM in Abhängigkeit des Tagens einer Sicherheitsweste.	A-29
Tabelle A.1-18: Rohdaten zur Erkennungsleistung der SVM in Abhängigkeit der Aktivitäten von Personen am Beispiel des Intensitätsbildes.	A-30
Tabelle A.1-19: Rohdaten zur Variation der Bildarten mit einem DL-Detektor für Personen, die maximal zu 33 % verdeckt sind.	A-1

Tabelle A.1-20: Rohdaten zur Unterteilung der DL-Detektionsergebnisse in Abhängigkeit des Verdeckungsgrads markierter Personen.	A-3
Tabelle A.1-21: Rohdaten zur Unterteilung der DL-Detektionsergebnisse in Abhängigkeit der Pose markierter Personen.	A-5
Tabelle A.1-22: Rohdaten zum Einfluss der Sicherheitsweste bei der DL-Detektion im Farb- und Intensitätsbild.	A-7
Tabelle A.1-23: Rohdaten zur Unterteilung der DL-Detektionsergebnisse in Abhängigkeit der Aktivität markierter Personen.	A-9
Tabelle A.2-1: Ergebnisse der Kollisionstests ohne Personen, Vergleich der OF-Algorithmen im Intensitätsbild (RANASC-Intervall: 0, Entfernung des Bodens & Hintergrunds: An, Mindestanzahl an Kollisionen: 2).	A-1
Tabelle A.2-2: Ergebnisse der Kollisionstests ohne Personen, Vergleich der OF-Algorithmen im Farbbild. (RANASC-Intervall: 0, Entfernung des Bodens & Hintergrunds: An, Mindestanzahl an Kollisionen: 2).	A-3
Tabelle A.2-3: Ergebnisse der Kollisionstests ohne Personen, ohne Entfernung des Bodens und des Hintergrunds.	A-5
Tabelle A.2-4: Ergebnisse der Kollisionstests ohne Personen, RANSAC-Intervall = 3.	A-6
Tabelle A.2-5: Ergebnisse der Kollisionstests ohne Personen, RANSAC-Intervall = 4.	A-6
Tabelle A.2-6: Ergebnisse der Kollisionstests ohne Personen, RANSAC-Intervall = 5.	A-7
Tabelle A.2-7: Ergebnisse der Kollisionstests ohne Personenerkennung, Mindestanzahl an Kollisionen = 1.	A-8
Tabelle A.2-8: Ergebnisse der Kollisionstests ohne Personenerkennung, Mindestanzahl an Kollisionen = 3.	A-9
Tabelle A.2-9: Ergebnisse der Kollisionstests ohne Personenerkennung, Mindestanzahl an Kollisionen = 4.	A-10
Tabelle A.2-10: Ergebnisse der Kollisionstests ohne Personenerkennung, Mindestanzahl an Kollisionen = 5.	A-11
Tabelle A.2-11: Ergebnisse der Kollisionstests ohne Personenerkennung, optimierte Konfiguration.	A-12
Tabelle A.2-12: Ergebnisse der Kollisionstests ohne Personenerkennung mit der statischen Kollisionsgefahrerkennung in Abhängigkeit der minimalen Distanz $r$ [m].	A-13
Tabelle B.1-1: Rohdaten zu den Kollisionstests mit Personen, Vergleich der OF-Algorithmen in Verbindung mit einem SVM bei $f_{ppi}=0,01$ .	B-1
Tabelle B.1-2: Rohdaten zu den Kollisionstests mit Personen, Vergleich der OF-Algorithmen in Verbindung mit einem SVM bei $f_{ppi}=0,1$ .	B-1
Tabelle B.2-1: Rohdaten zu den Kollisionstests mit Personen, Vergleich der OF-Algorithmen in Verbindung mit einem DL-Detektor bei $f_{ppi}=0$ .	B-2

- Tabelle B.2-2: Rohdaten zu den Kollisionstests mit Personen, Vergleich der OF-  
Algorithmen in Verbindung mit einem DL-Detektor bei  $f_{ppi}=0,01$ .  
B-2
- Tabelle B.2-3: Rohdaten zu den Kollisionstests mit Personen, Vergleich der OF-  
Algorithmen in Verbindung mit einem DL-Detektor bei  $f_{ppi}=0,1$ . B-  
2
- Tabelle B.3-1: Rohdaten der Auswertung der statischen  
Kollisionsgefahrerkennung bei Kollisionen mit Personen in  
Abhängigkeit der minimalen Distanz  $r$  [m]. B-3





# Anhang A Ergebnisse der Evaluation einzelner Algorithmen

## A.1 Personenerkennung

### A.1.1 Support-Vector-Machine

#### Variation des Trainings – Detektorgröße

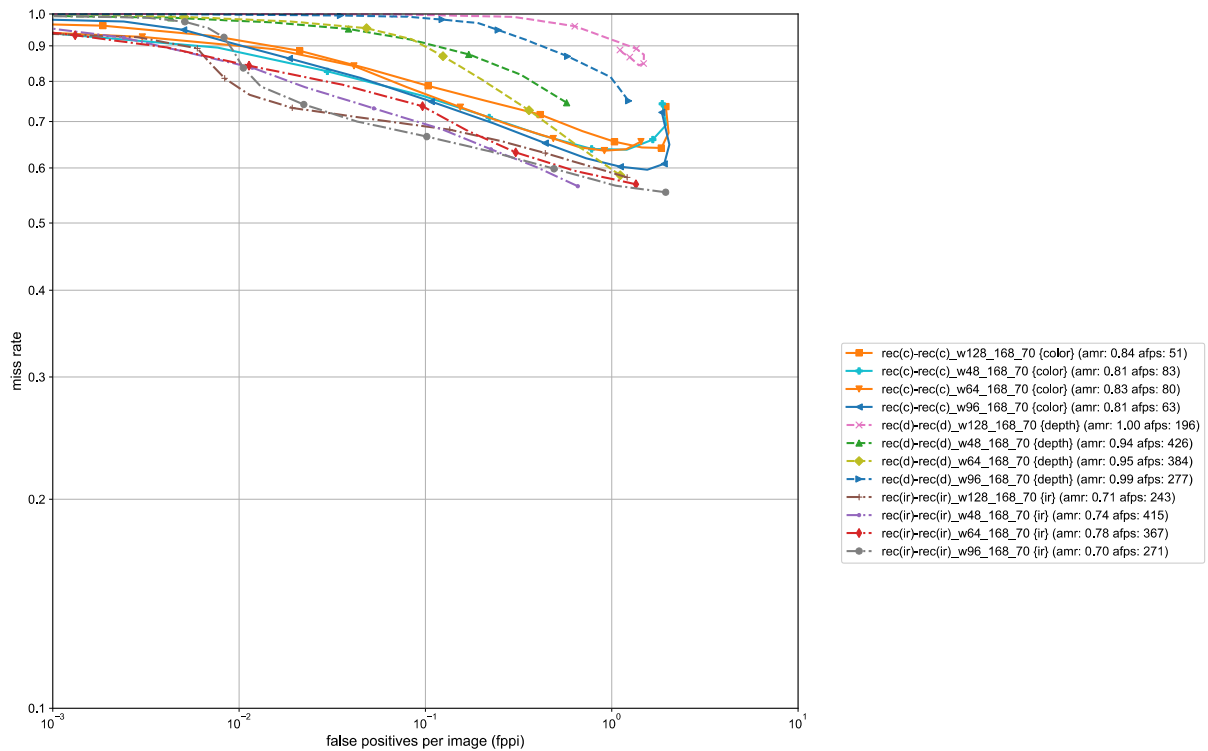


Abbildung A.1-1: Ergebnis der Variation der Detektorgröße.

Tabelle A.1-1: Rohdaten zur Variation der Detektorgröße.

rec(c)-rec(c)_w128_168_70 {color} (amr: 0.84)																		
fps	44,70	46,42	48,24	48,04	48,34	48,81	48,63	48,72	48,97	49,17	48,68	48,51	50,65	56,10	57,96	60,23	62,74	65,96
precision	0,09	0,11	0,13	0,15	0,20	0,26	0,34	0,47	0,60	0,71	0,80	0,88	0,94	0,99	1,00	1,00	1,00	1,00
recall	0,26	0,33	0,36	0,36	0,35	0,32	0,28	0,25	0,21	0,17	0,11	0,07	0,04	0,02	0,01	0,00	0,00	0,00
missrate	0,74	0,67	0,64	0,64	0,65	0,68	0,72	0,75	0,79	0,83	0,89	0,93	0,96	0,98	0,99	1,00	1,00	1,00
fppi	1,95	2,02	1,85	1,45	1,04	0,70	0,41	0,21	0,10	0,05	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00
rec(d)-rec(d)_w128_168_70 {depth} (amr: 1.00)																		
fps	39,72	52,89	76,42	111,39	155,48	193,30	219,63	236,40	242,93	245,48	244,11	244,72	244,12	244,85	244,38	244,93	244,29	243,98
precision	0,06	0,06	0,06	0,06	0,06	0,05	0,05	0,05	0,04	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00
recall	0,11	0,12	0,13	0,16	0,15	0,13	0,11	0,08	0,04	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
missrate	0,89	0,88	0,87	0,84	0,85	0,87	0,89	0,92	0,96	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
fppi	1,11	1,15	1,25	1,40	1,49	1,49	1,36	1,00	0,63	0,30	0,07	0,01	0,00	0,00	0,00	0,00	0,00	0,00

## A Ergebnisse der Evaluation einzelner Algorithmen

rec(ir)-rec(ir)_w128_168_70 {ir} (amr: 0.71)																		
fps	234,27	238,02	240,29	242,35	243,77	245,16	245,43	245,33	245,75	245,04	246,57	244,38	246,84	246,24	244,82	245,61	244,70	244,89
precision	0,20	0,29	0,38	0,50	0,64	0,81	0,91	0,94	0,94	0,94	0,93	0,95	0,99	1,00	1,00	1,00	1,00	1,00
recall	0,42	0,39	0,37	0,34	0,32	0,29	0,27	0,24	0,19	0,15	0,11	0,07	0,05	0,03	0,02	0,01	0,00	0,00
missrate	0,58	0,61	0,63	0,66	0,68	0,71	0,73	0,76	0,81	0,85	0,89	0,93	0,95	0,97	0,98	0,99	1,00	1,00
fppi	1,21	0,73	0,44	0,26	0,13	0,05	0,02	0,01	0,01	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00
rec(c)-rec(c)_w48_168_70 {color} (amr: 0.81)																		
fps	81,07	81,70	82,44	82,86	83,00	83,25	83,69	83,56	83,53	83,22	83,59	84,09	84,12	83,91	83,95	82,99	83,71	84,16
precision	0,09	0,10	0,13	0,18	0,26	0,37	0,49	0,65	0,81	0,91	0,98	1,00	1,00	1,00	1,00	1,00	1,00	0,00
recall	0,26	0,31	0,34	0,36	0,36	0,33	0,29	0,24	0,17	0,11	0,05	0,03	0,01	0,01	0,00	0,00	0,00	0,00
missrate	0,74	0,69	0,66	0,64	0,64	0,67	0,71	0,76	0,83	0,89	0,95	0,97	0,99	0,99	1,00	1,00	1,00	1,00
fppi	1,87	1,96	1,66	1,21	0,78	0,42	0,22	0,10	0,03	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
rec(d)-rec(d)_w48_168_70 {depth} (amr: 0.94)																		
fps	426,16	423,72	426,20	430,72	428,60	426,22	424,09	427,63	425,44	426,45	424,25	426,89	424,69	427,80	424,62	424,38	430,89	428,50
precision	0,22	0,26	0,31	0,38	0,44	0,53	0,63	0,81	0,90	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00
recall	0,25	0,18	0,13	0,08	0,05	0,03	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
missrate	0,75	0,82	0,87	0,92	0,95	0,97	0,99	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
fppi	0,57	0,33	0,17	0,08	0,04	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
rec(ir)-rec(ir)_w48_168_70 {ir} (amr: 0.74)																		
fps	410,62	411,49	414,29	415,38	413,76	414,04	413,72	422,16	416,82	415,16	419,18	415,23	416,80	414,57	415,65	415,99	413,68	411,93
precision	0,33	0,43	0,54	0,66	0,79	0,88	0,91	0,95	0,96	0,97	0,97	0,99	1,00	1,00	1,00	0,00	0,00	0,00
recall	0,44	0,40	0,36	0,32	0,27	0,22	0,16	0,12	0,08	0,05	0,03	0,01	0,00	0,00	0,00	0,00	0,00	0,00
missrate	0,56	0,60	0,64	0,68	0,73	0,78	0,84	0,88	0,92	0,95	0,97	0,99	1,00	1,00	1,00	1,00	1,00	1,00
fppi	0,66	0,39	0,22	0,12	0,05	0,02	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
rec(c)-rec(c)_w64_168_70 {color} (amr: 0.83)																		
fps	77,62	78,28	78,70	78,37	78,68	79,23	79,04	78,91	79,00	79,10	79,55	80,69	82,11	82,81	83,97	84,49	85,48	85,42
precision	0,15	0,18	0,23	0,28	0,34	0,44	0,56	0,67	0,74	0,84	0,95	0,99	1,00	1,00	1,00	1,00	1,00	1,00
recall	0,34	0,36	0,36	0,36	0,34	0,31	0,27	0,21	0,16	0,11	0,07	0,04	0,02	0,01	0,01	0,00	0,00	0,00
missrate	0,66	0,64	0,64	0,64	0,66	0,69	0,73	0,79	0,84	0,89	0,93	0,96	0,98	0,99	0,99	1,00	1,00	1,00
fppi	1,44	1,20	0,91	0,67	0,49	0,29	0,15	0,08	0,04	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
rec(d)-rec(d)_w64_168_70 {depth} (amr: 0.95)																		
fps	378,70	384,45	381,29	383,18	384,45	384,27	384,94	383,45	384,61	384,24	384,19	385,31	383,79	382,13	386,69	386,09	385,46	384,87
precision	0,19	0,25	0,32	0,38	0,39	0,37	0,37	0,45	0,53	0,51	0,25	0,00	0,00	0,00	0,00	0,00	0,00	0,00
recall	0,41	0,36	0,27	0,20	0,13	0,08	0,05	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
missrate	0,59	0,64	0,73	0,80	0,87	0,92	0,95	0,98	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
fppi	1,11	0,67	0,36	0,20	0,12	0,08	0,05	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
rec(ir)-rec(ir)_w64_168_70 {ir} (amr: 0.78)																		
fps	361,32	365,47	367,80	369,28	367,56	368,93	369,06	369,29	366,37	368,73	367,98	369,04	368,77	369,23	368,46	370,46	368,51	363,29
precision	0,19	0,32	0,47	0,59	0,67	0,81	0,91	0,95	0,97	0,98	1,00	1,00	1,00	1,00	1,00	0,00	0,00	0,00
recall	0,43	0,41	0,37	0,32	0,26	0,21	0,16	0,11	0,07	0,04	0,02	0,01	0,01	0,00	0,00	0,00	0,00	0,00
missrate	0,57	0,59	0,63	0,68	0,74	0,79	0,84	0,89	0,93	0,96	0,98	0,99	0,99	1,00	1,00	1,00	1,00	1,00
fppi	1,35	0,63	0,31	0,17	0,10	0,04	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
rec(c)-rec(c)_w96_168_70 {color} (amr: 0.81)																		
fps	57,46	58,98	60,32	61,02	61,52	61,38	61,63	61,80	61,70	61,91	61,82	61,65	62,38	64,94	69,77	73,02	75,12	75,30
precision	0,10	0,11	0,13	0,16	0,21	0,28	0,37	0,49	0,64	0,76	0,85	0,88	0,88	0,88	0,96	1,00	1,00	1,00

recall	0,28	0,35	0,39	0,40	0,40	0,38	0,35	0,30	0,25	0,19	0,14	0,09	0,05	0,02	0,01	0,00	0,00	0,00
missrate	0,72	0,65	0,61	0,60	0,60	0,62	0,65	0,70	0,75	0,81	0,86	0,91	0,95	0,98	0,99	1,00	1,00	1,00
fppi	1,86	2,04	1,91	1,55	1,11	0,73	0,44	0,23	0,11	0,04	0,02	0,01	0,01	0,00	0,00	0,00	0,00	0,00
<b>rec(d)-rec(d)_w96_168_70 {depth} (amr: 0.99)</b>																		
fps	267,0	272,5	274,3	275,7	277,6	277,9	278,9	278,1	278,1	277,3	277,6	279,7	279,6	279,3	278,9	278,0	278,9	279,1
precision	0,11	0,11	0,12	0,13	0,12	0,09	0,08	0,07	0,08	0,11	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
recall	0,25	0,19	0,13	0,08	0,05	0,03	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
missrate	0,75	0,81	0,87	0,92	0,95	0,97	0,98	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
fppi	1,23	0,99	0,58	0,35	0,25	0,19	0,12	0,08	0,04	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
<b>rec(ir)-rec(ir)_w96_168_70 {ir} (amr: 0.70)</b>																		
fps	260,1	267,0	269,2	268,7	269,0	270,6	271,3	273,2	273,8	273,2	272,8	274,7	273,1	273,4	273,8	270,7	271,8	273,2
precision	0,15	0,23	0,38	0,55	0,71	0,84	0,90	0,92	0,92	0,90	0,87	0,83	0,79	0,74	0,84	1,00	0,00	0,00
recall	0,45	0,43	0,40	0,37	0,33	0,30	0,26	0,21	0,16	0,11	0,07	0,05	0,03	0,01	0,00	0,00	0,00	0,00
missrate	0,55	0,57	0,60	0,63	0,67	0,70	0,74	0,79	0,84	0,89	0,93	0,95	0,97	0,99	1,00	1,00	1,00	1,00
Fppi	1,95	1,05	0,49	0,23	0,10	0,04	0,02	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,00	0,00	0,00	0,00

Variation des Trainings – Bildquellen für Positivbilder

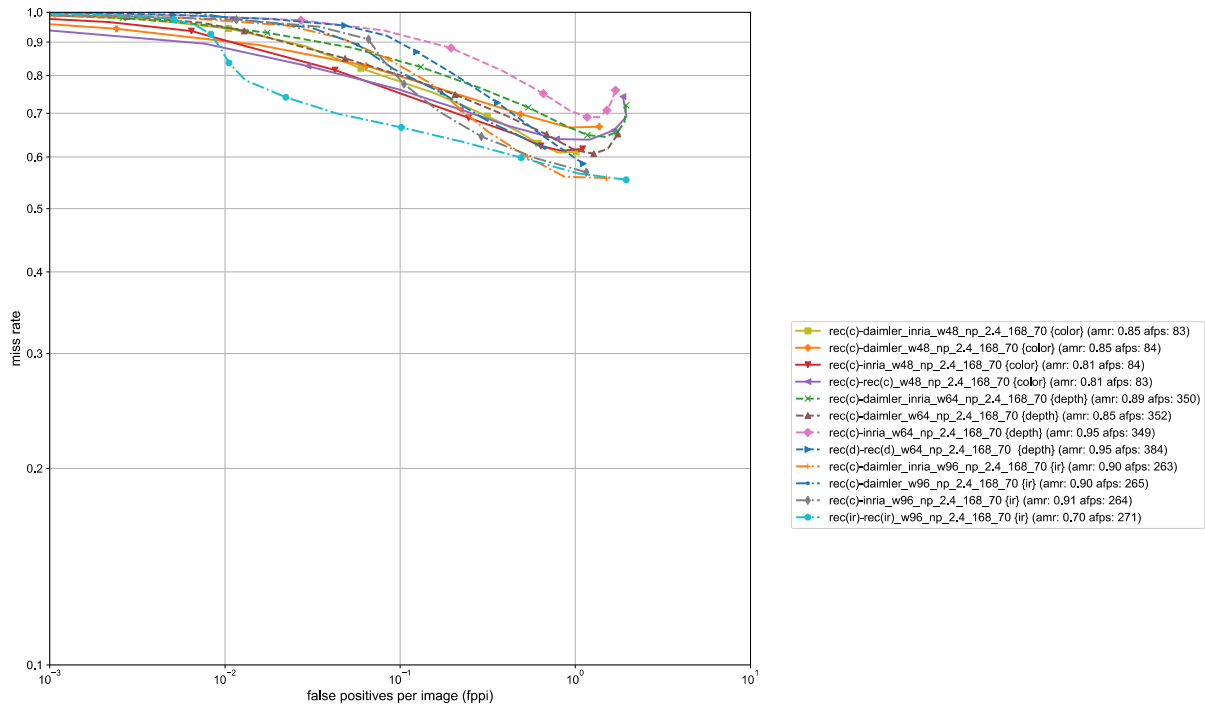


Abbildung A.1-2: Ergebnis der Variation der Bildquellen für die Positivtrainingsbilder.

Tabelle A.1-2: Rohdaten zur Variation der Bildquellen für die Positivtrainingsbilder.

<b>fml-daimler_inria_w48_EL_np_2.4 (amr: 0.85)</b>																		
fps	83,84	84,41	84,13	83,64	83,57	83,88	84,69	84,53	84,23	84,18	83,51	83,74	83,60	83,44	83,17	83,51	83,80	83,90
precision	0,22	0,27	0,31	0,35	0,42	0,54	0,69	0,74	0,80	0,84	0,90	1,00	1,00	1,00	0,00	0,00	0,00	0,00
recall	0,39	0,39	0,37	0,34	0,31	0,25	0,18	0,11	0,06	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00
missrate	0,61	0,61	0,63	0,66	0,69	0,75	0,82	0,89	0,94	0,98	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00
fppi	1,01	0,80	0,61	0,47	0,31	0,16	0,06	0,03	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
<b>fml-daimler_inria_w64_EL_np_2.4 (amr: 0.89)</b>																		
fps	325,64	336,42	349,59	352,89	359,0	359,4	356,9	356,0	347,3	353,7	352,2	345,5	346,4	348,0	353,4	355,7	355,7	355,3
precision	0,08	0,09	0,11	0,13	0,16	0,19	0,25	0,34	0,45	0,58	0,71	0,79	0,83	0,87	0,87	0,88	0,00	0,00
recall	0,28	0,32	0,34	0,36	0,35	0,33	0,29	0,23	0,18	0,12	0,07	0,04	0,02	0,01	0,00	0,00	0,00	0,00
missrate	0,72	0,68	0,66	0,64	0,65	0,67	0,71	0,77	0,82	0,88	0,93	0,96	0,98	0,99	1,00	1,00	1,00	1,00
fppi	1,95	1,91	1,74	1,48	1,17	0,86	0,54	0,29	0,13	0,05	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00

# A Ergebnisse der Evaluation einzelner Algorithmen

<b>fmi-daimler_inria_w96_EL_np_2.4 (amr: 0.90)</b>																		
fps	246,57	256,44	260,70	264,20	264,34	264,16	265,39	265,60	265,19	265,79	263,60	266,01	265,45	264,62	265,58	266,61	266,38	266,96
precision	0,18	0,27	0,36	0,45	0,49	0,52	0,57	0,58	0,60	0,71	0,94	1,00	1,00	0,00	0,00	0,00	0,00	0,00
recall	0,44	0,44	0,40	0,34	0,29	0,22	0,15	0,09	0,05	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
missrate	0,56	0,56	0,60	0,66	0,71	0,78	0,85	0,91	0,95	0,98	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
fppi	1,51	0,87	0,53	0,32	0,22	0,15	0,09	0,05	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
<b>fmi-daimler_w48_EL_np_2.4 (amr: 0.85)</b>																		
fps	83,84	84,09	84,44	84,69	84,62	84,02	84,73	84,59	84,41	84,25	84,32	84,46	84,20	84,61	84,47	84,78	84,37	83,88
precision	0,15	0,21	0,32	0,48	0,67	0,84	0,95	0,98	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
recall	0,33	0,33	0,30	0,25	0,17	0,11	0,06	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
missrate	0,67	0,67	0,70	0,75	0,83	0,89	0,94	0,98	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
fppi	1,37	0,92	0,49	0,20	0,06	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
<b>fmi-daimler_w64_EL_np_2.4 (amr: 0.85)</b>																		
fps	332,34	345,42	349,33	349,77	355,55	359,47	357,03	358,85	360,43	357,38	355,56	356,78	352,37	346,64	342,62	357,93	355,53	360,41
precision	0,11	0,13	0,16	0,19	0,24	0,32	0,43	0,55	0,66	0,74	0,75	0,75	0,77	0,90	1,00	1,00	0,00	0,00
recall	0,35	0,38	0,39	0,39	0,35	0,31	0,25	0,20	0,15	0,10	0,06	0,04	0,01	0,01	0,00	0,00	0,00	0,00
missrate	0,65	0,62	0,61	0,61	0,65	0,69	0,75	0,80	0,85	0,90	0,94	0,96	0,99	0,99	1,00	1,00	1,00	1,00
fppi	1,75	1,53	1,27	0,99	0,69	0,40	0,21	0,10	0,05	0,02	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00
<b>fmi-daimler_w96_EL_np_2.4 (amr: 0.90)</b>																		
fps	262,30	264,29	264,81	264,18	265,74	265,26	266,06	266,81	267,41	265,55	264,76	266,42	267,46	267,49	264,90	266,53	265,84	265,44
precision	0,30	0,42	0,52	0,60	0,58	0,56	0,60	0,43	0,14	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
recall	0,38	0,32	0,25	0,18	0,11	0,05	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
missrate	0,62	0,68	0,75	0,82	0,89	0,95	0,98	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
fppi	0,67	0,32	0,17	0,09	0,06	0,03	0,01	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
<b>fmi-inria_w48_EL_np_2.4 (amr: 0.81)</b>																		
fps	83,17	83,58	84,54	83,73	83,68	84,24	84,19	84,45	84,77	84,69	84,45	85,02	84,52	84,30	83,92	83,78	83,99	84,29
precision	0,21	0,26	0,31	0,36	0,48	0,63	0,76	0,86	0,88	0,92	0,95	0,97	1,00	0,00	0,00	0,00	0,00	0,00
recall	0,38	0,39	0,38	0,35	0,31	0,26	0,18	0,11	0,06	0,03	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00
missrate	0,62	0,61	0,62	0,65	0,69	0,74	0,82	0,89	0,94	0,97	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00
fppi	1,10	0,84	0,63	0,46	0,25	0,12	0,04	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
<b>fmi-inria_w64_EL_np_2.4 (amr: 0.95)</b>																		
fps	311,06	325,48	331,02	344,50	339,40	348,36	347,29	360,56	360,70	358,83	358,50	360,82	359,59	356,90	357,62	358,19	358,75	355,19
precision	0,08	0,09	0,11	0,12	0,14	0,16	0,19	0,23	0,27	0,32	0,39	0,53	0,70	0,85	0,50	0,00	0,00	0,00
recall	0,24	0,27	0,29	0,31	0,31	0,29	0,25	0,19	0,12	0,06	0,03	0,01	0,00	0,00	0,00	0,00	0,00	0,00
missrate	0,76	0,73	0,71	0,69	0,69	0,71	0,75	0,81	0,88	0,94	0,97	0,99	1,00	1,00	1,00	1,00	1,00	1,00
fppi	1,70	1,62	1,52	1,37	1,17	0,92	0,66	0,39	0,20	0,08	0,03	0,01	0,00	0,00	0,00	0,00	0,00	0,00
<b>fmi-inria_w96_EL_np_2.4 (amr: 0.91)</b>																		
fps	259,03	263,06	264,36	265,75	265,37	264,73	264,71	265,49	265,13	265,62	265,44	265,55	265,65	265,74	265,62	266,16	265,94	266,35
precision	0,22	0,33	0,48	0,57	0,61	0,59	0,51	0,52	0,63	0,80	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00
recall	0,43	0,40	0,35	0,30	0,22	0,15	0,09	0,05	0,03	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00
missrate	0,57	0,60	0,65	0,70	0,78	0,85	0,91	0,95	0,97	0,99	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00
fppi	1,16	0,60	0,29	0,17	0,11	0,08	0,07	0,03	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00

Variation des Trainings – Bildquellen für Negativbilder

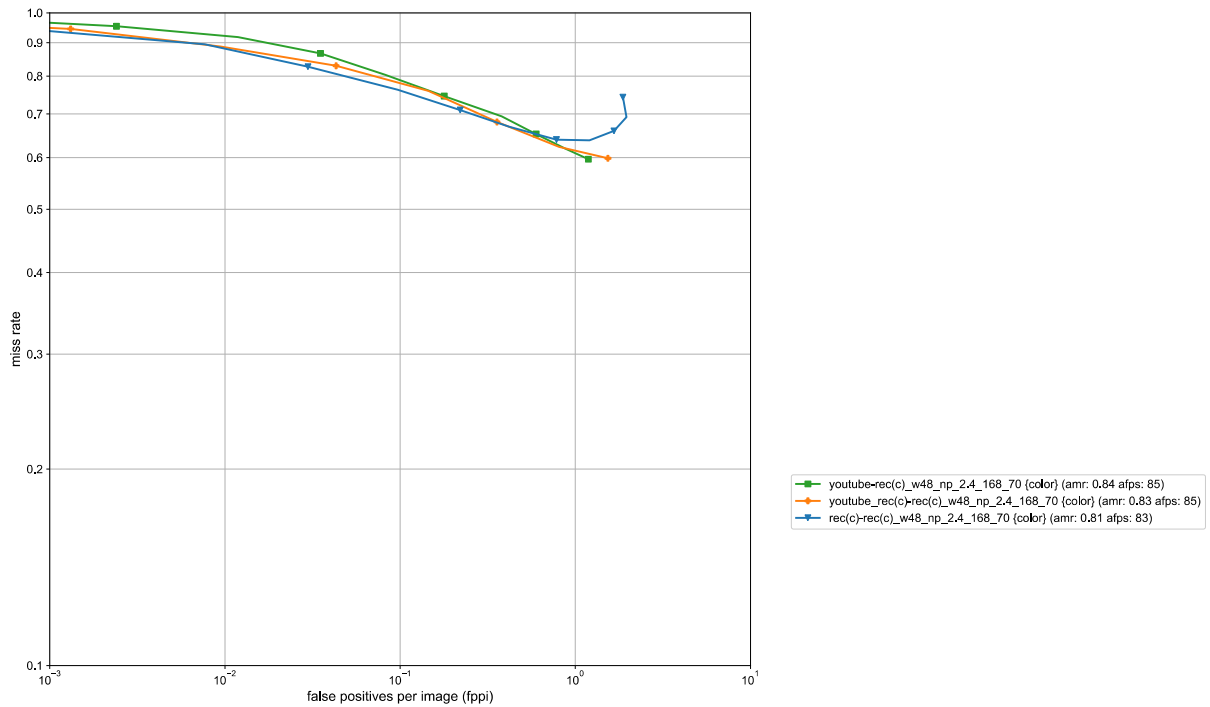


Abbildung A.1-3: Ergebnis der Variation der Bildquellen für die Positivtrainingsbilder.

Tabelle A.1-3: Rohdaten zur Variation der Bildquellen für die Positivtrainingsbilder.

v3_youtube-rec(c)_w48_EL_np_2.4_168_70_auto.yml {color} (amr: 0.84)																		
fps	84,85	85,26	85,21	84,80	84,88	85,21	85,31	85,31	85,52	85,44	85,51	85,28	85,02	85,35	85,56	85,46	85,45	85,12
precision	0,20	0,24	0,30	0,37	0,51	0,64	0,74	0,84	0,93	0,97	0,99	1,00	1,00	1,00	0,00	0,00	0,00	0,00
recall	0,40	0,38	0,35	0,31	0,25	0,20	0,13	0,08	0,05	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00
missrate	0,60	0,62	0,65	0,69	0,75	0,80	0,87	0,92	0,95	0,98	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00
fpfi	1,19	0,87	0,60	0,38	0,18	0,08	0,04	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
v3_youtube_rec(c)-rec(c)_w48_EL_np_2.4_168_70_auto.yml {color} (amr: 0.83)																		
fps	84,97	85,11	85,04	85,33	85,26	85,16	85,09	85,02	85,54	85,17	85,08	85,37	85,56	85,25	85,02	84,82	85,23	85,51
precision	0,16	0,26	0,40	0,55	0,75	0,90	0,97	0,99	1,00	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00
recall	0,40	0,38	0,32	0,24	0,17	0,11	0,06	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
missrate	0,60	0,62	0,68	0,76	0,83	0,89	0,94	0,98	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
fpfi	1,54	0,81	0,36	0,14	0,04	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
rec(c)-rec(c)_w48_EL_np_2.4_168_70 (color) (amr: 0.81)																		
fps	81,07	81,70	82,44	82,86	83,00	83,25	83,69	83,56	83,53	83,22	83,59	84,09	84,12	83,91	83,95	82,99	83,71	84,16
precision	0,09	0,10	0,13	0,18	0,26	0,37	0,49	0,65	0,81	0,91	0,98	1,00	1,00	1,00	1,00	1,00	1,00	0,00
recall	0,26	0,31	0,34	0,36	0,36	0,33	0,29	0,24	0,17	0,11	0,05	0,03	0,01	0,01	0,00	0,00	0,00	0,00
missrate	0,74	0,69	0,66	0,64	0,64	0,67	0,71	0,76	0,83	0,89	0,95	0,97	0,99	0,99	1,00	1,00	1,00	1,00
fpfi	1,87	1,96	1,66	1,21	0,78	0,42	0,22	0,10	0,03	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00

# A Ergebnisse der Evaluation einzelner Algorithmen

## Variation des Trainings – Anzahl der Trainingsbilder

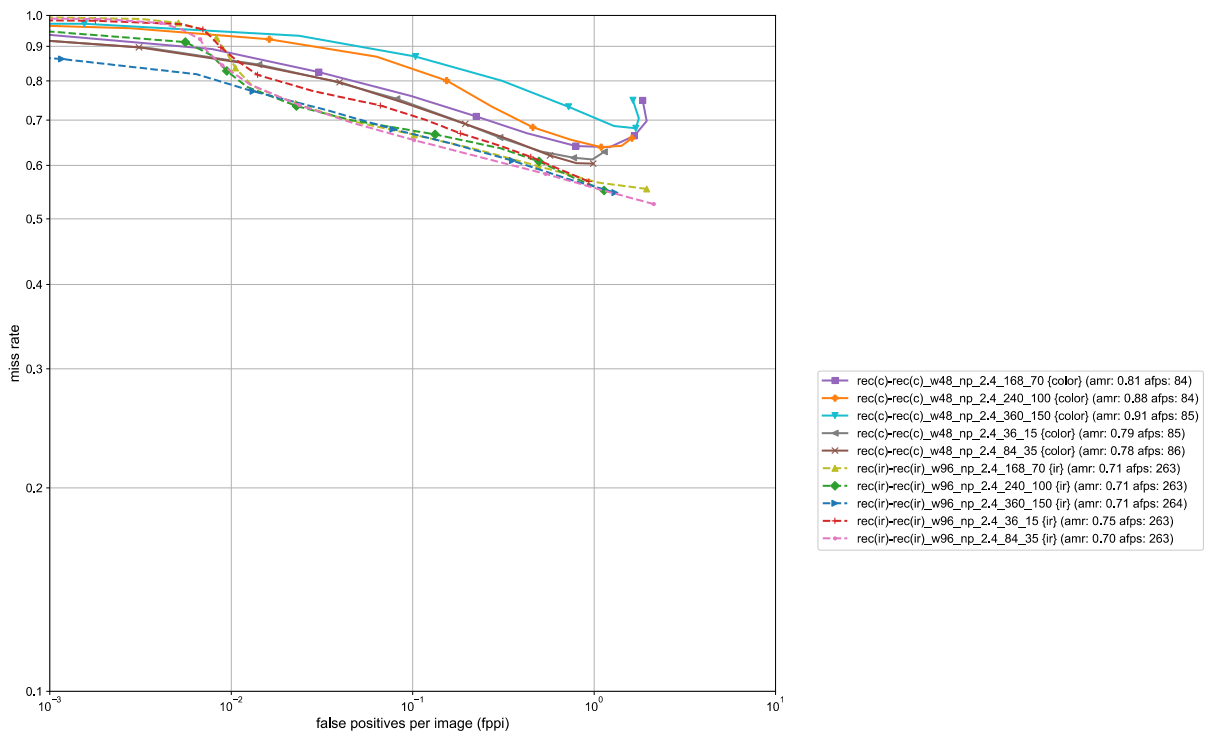


Abbildung A.1-4: Ergebnis der Variation der Anzahl an Positivtrainingsbildern.

Tabelle A.1-4: Rohdaten zur Variation der Anzahl an Positivtrainingsbildern.

<b>rec(c)-rec(c)_w48_EL_np_2.4_168_70_auto.yml {color} (amr: 0.81)</b>																			
fps	82,77	84,12	84,60	84,62	84,97	84,79	85,39	85,13	85,28	85,05	85,42	85,79	85,11	85,27	84,90	85,40	85,55	85,53	
precision	0,09	0,10	0,13	0,18	0,25	0,37	0,49	0,65	0,81	0,91	0,98	1,00	1,00	1,00	1,00	1,00	0,00	0,00	
recall	0,25	0,30	0,34	0,36	0,36	0,33	0,29	0,24	0,18	0,11	0,06	0,03	0,01	0,01	0,00	0,00	0,00	0,00	
missrate	0,75	0,70	0,66	0,64	0,64	0,67	0,71	0,76	0,82	0,89	0,94	0,97	0,99	0,99	1,00	1,00	1,00	1,00	
fppi	1,85	1,95	1,67	1,22	0,79	0,43	0,22	0,10	0,03	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
<b>rec(c)-rec(c)_w48_EL_np_2.4_240_100_auto.yml {color} (amr: 0.88)</b>																			
fps	83,46	84,25	84,76	84,81	85,28	84,66	84,93	85,62	85,17	85,35	85,07	85,03	85,32	85,54	85,10	84,72	85,40	85,04	
precision	0,14	0,16	0,20	0,26	0,34	0,42	0,49	0,61	0,78	0,92	0,99	1,00	1,00	1,00	0,00	0,00	0,00	0,00	
recall	0,34	0,36	0,36	0,35	0,32	0,27	0,20	0,13	0,08	0,04	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00	
missrate	0,66	0,64	0,64	0,65	0,68	0,73	0,80	0,87	0,92	0,96	0,98	0,99	1,00	1,00	1,00	1,00	1,00	1,00	
fppi	1,62	1,42	1,09	0,75	0,46	0,27	0,15	0,06	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
<b>rec(c)-rec(c)_w48_EL_np_2.4_360_150_auto.yml {color} (amr: 0.91)</b>																			
fps	82,83	84,49	85,45	85,31	85,81	85,44	85,30	85,21	85,31	85,63	85,85	85,32	85,65	85,55	85,57	85,88	85,77	85,70	
precision	0,10	0,11	0,12	0,15	0,22	0,32	0,48	0,68	0,93	1,00	1,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00	
recall	0,25	0,30	0,32	0,31	0,27	0,20	0,13	0,07	0,03	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
missrate	0,75	0,70	0,68	0,69	0,73	0,80	0,87	0,93	0,97	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	
fppi	1,64	1,77	1,70	1,29	0,72	0,31	0,10	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
<b>rec(c)-rec(c)_w48_EL_np_2.4_36_15_auto.yml {color} (amr: 0.79)</b>																			
fps	84,56	84,85	85,43	85,12	85,08	85,01	85,58	85,04	85,04	85,52	84,62	85,43	85,31	85,60	85,57	85,42	85,30	85,46	
precision	0,20	0,23	0,27	0,35	0,46	0,57	0,69	0,80	0,89	0,96	0,99	1,00	1,00	1,00	1,00	1,00	1,00	0,00	
recall	0,37	0,39	0,38	0,37	0,34	0,30	0,25	0,20	0,15	0,11	0,07	0,04	0,03	0,01	0,01	0,00	0,00	0,00	
missrate	0,63	0,61	0,62	0,63	0,66	0,70	0,75	0,80	0,85	0,89	0,93	0,96	0,97	0,99	0,99	1,00	1,00	1,00	
fppi	1,14	0,98	0,77	0,51	0,30	0,16	0,08	0,04	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
<b>rec(c)-rec(c)_w48_EL_np_2.4_84_35_auto.yml {color} (amr: 0.78)</b>																			
fps	86,28	87,10	86,40	86,65	87,19	87,27	86,78	86,88	86,99	87,07	86,88	86,96	87,22	87,06	87,11	87,18	87,03	86,78	
precision	0,23	0,27	0,33	0,42	0,54	0,68	0,79	0,90	0,96	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	
recall	0,40	0,40	0,38	0,35	0,31	0,26	0,20	0,15	0,10	0,06	0,03	0,02	0,01	0,00	0,00	0,00	0,00	0,00	
missrate	0,60	0,60	0,62	0,65	0,69	0,74	0,80	0,85	0,90	0,94	0,97	0,98	0,99	1,00	1,00	1,00	1,00	1,00	
fppi	0,99	0,80	0,57	0,35	0,19	0,09	0,04	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
<b>rec(ir)-rec(ir)_w96_EL_np_2.4_168_70_auto.yml {ir} (amr: 0.71)</b>																			
fps	255,1	259,9	262,3	263,8	263,8	265,2	265,3	265,3	266,4	264,5	264,9	263,6	264,9	264,7	264,9	263,6	265,0	263,2	
	3	3	6	4	4	8	0	5	2	4	8	3	4	6	7	1	4	5	
precision	0,15	0,23	0,38	0,55	0,71	0,84	0,89	0,92	0,92	0,90	0,87	0,83	0,79	0,74	0,84	1,00	0,00	0,00	
recall	0,45	0,43	0,40	0,37	0,33	0,30	0,26	0,21	0,16	0,11	0,07	0,05	0,03	0,01	0,00	0,00	0,00	0,00	
missrate	0,55	0,57	0,60	0,63	0,67	0,70	0,74	0,79	0,84	0,89	0,93	0,95	0,97	0,99	1,00	1,00	1,00	1,00	
fppi	1,95	1,05	0,49	0,23	0,10	0,04	0,02	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,00	0,00	0,00	0,00	

rec(ir)-rec(ir)_w96_EL_np_2.4_240_100_auto.yml {ir} (amr: 0.71)																		
fps	254,9	260,5	261,9	262,7	263,7	263,2	264,9	265,2	265,1	264,5	264,5	265,1	265,2	264,4	264,8	264,2	265,9	265,0
precision	0,23	0,29	0,37	0,47	0,65	0,83	0,90	0,93	0,93	0,92	0,92	0,98	0,99	1,00	1,00	1,00	1,00	0,00
recall	0,45	0,42	0,39	0,36	0,33	0,30	0,27	0,22	0,17	0,13	0,09	0,05	0,04	0,02	0,01	0,00	0,00	0,00
missrate	0,55	0,58	0,61	0,64	0,67	0,70	0,73	0,78	0,83	0,87	0,91	0,95	0,96	0,98	0,99	1,00	1,00	1,00
fppi	1,14	0,75	0,50	0,31	0,13	0,05	0,02	0,01	0,01	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00
rec(ir)-rec(ir)_w96_EL_np_2.4_360_150_auto.yml {ir} (amr: 0.71)																		
fps	255,4	259,6	263,2	264,2	265,1	264,3	264,0	266,3	265,8	265,4	264,7	266,3	265,3	263,7	263,2	265,6	265,7	265,2
precision	0,21	0,31	0,45	0,61	0,75	0,86	0,93	0,95	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
recall	0,45	0,43	0,39	0,35	0,32	0,27	0,23	0,18	0,14	0,10	0,05	0,03	0,01	0,01	0,00	0,00	0,00	0,00
missrate	0,55	0,57	0,61	0,65	0,68	0,73	0,77	0,82	0,86	0,90	0,95	0,97	0,99	0,99	1,00	1,00	1,00	1,00
fppi	1,30	0,72	0,36	0,17	0,08	0,03	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
rec(ir)-rec(ir)_w96_EL_np_2.4_36_15_auto.yml {ir} (amr: 0.75)																		
fps	257,4	260,5	261,7	262,6	263,3	264,0	263,3	264,9	264,1	266,0	265,4	264,1	265,2	263,6	264,4	265,0	263,9	264,6
precision	0,26	0,32	0,39	0,47	0,57	0,66	0,75	0,86	0,91	0,91	0,90	0,87	0,83	0,80	0,88	0,99	1,00	1,00
recall	0,43	0,41	0,38	0,36	0,33	0,30	0,26	0,23	0,18	0,14	0,10	0,07	0,05	0,03	0,02	0,01	0,01	0,01
missrate	0,57	0,59	0,62	0,64	0,67	0,70	0,74	0,77	0,82	0,86	0,90	0,93	0,95	0,97	0,98	0,99	0,99	0,99
fppi	0,94	0,64	0,45	0,30	0,18	0,12	0,07	0,03	0,01	0,01	0,01	0,01	0,01	0,01	0,00	0,00	0,00	0,00
rec(ir)-rec(ir)_w96_EL_np_2.4_84_35_auto.yml {ir} (amr: 0.70)																		
fps	255,4	259,0	262,7	264,7	265,3	263,9	265,2	264,6	265,3	264,3	264,6	264,0	263,2	264,6	265,2	264,9	265,6	265,5
precision	0,14	0,24	0,36	0,54	0,72	0,83	0,89	0,92	0,93	0,92	0,90	0,87	0,82	0,82	0,96	1,00	1,00	0,00
recall	0,47	0,45	0,42	0,38	0,35	0,31	0,26	0,21	0,16	0,12	0,08	0,05	0,03	0,01	0,01	0,00	0,00	0,00
missrate	0,53	0,55	0,58	0,62	0,65	0,69	0,74	0,79	0,84	0,88	0,92	0,95	0,97	0,99	0,99	1,00	1,00	1,00
fppi	2,13	1,07	0,54	0,24	0,10	0,05	0,02	0,01	0,01	0,01	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00

Variation des Trainings – Verhältnis negativer zu positiver Bilder

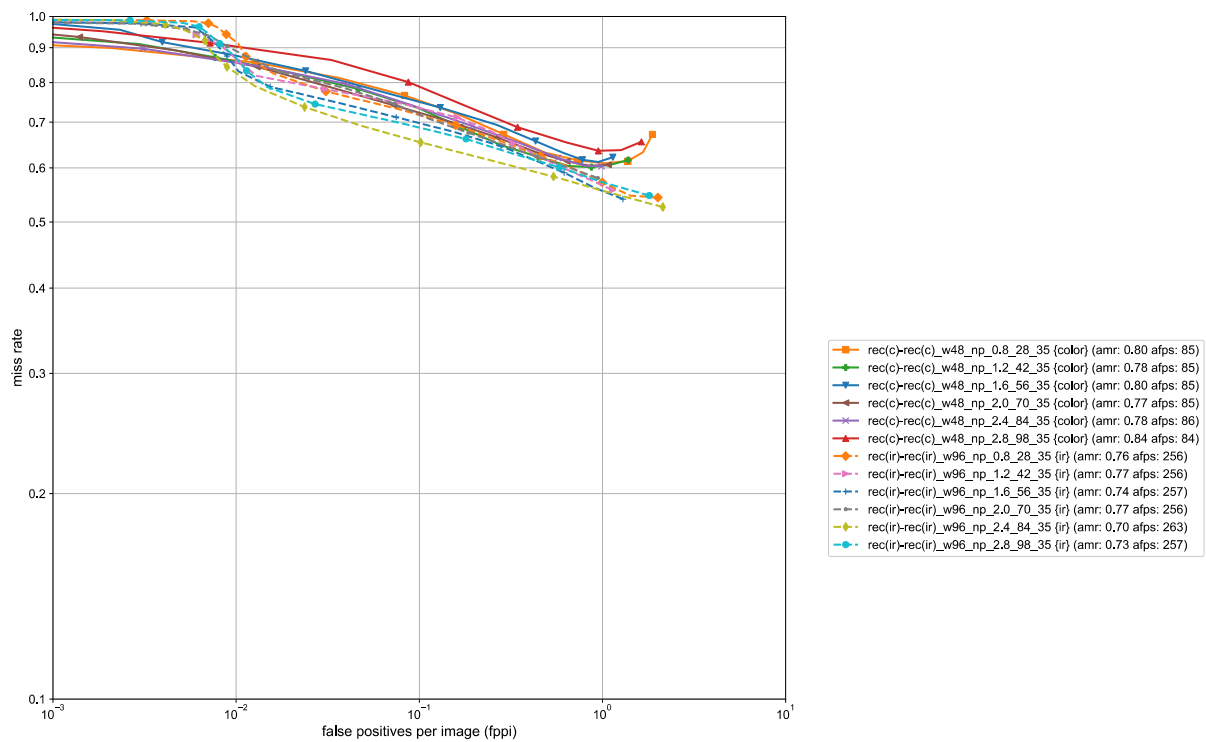


Abbildung A.1-5: Ergebnis der Variation des Verhältnisses trainierter Negativ- zu Positivbildern (np\_\*).

Tabelle A.1-5: Rohdaten zur Variation des Verhältnisses trainierter Negativ- zu Positivbildern (np\_\*).

rec(c)-rec(c)_w48_EL_np_0.8_28_35_auto.yml {color} (amr: 0.80)																		
fps	84,21	84,85	84,95	85,43	85,49	85,45	85,67	85,43	85,53	86,24	85,79	85,89	85,72	85,79	85,94	85,89	86,20	85,93
precision	0,12	0,14	0,17	0,22	0,28	0,36	0,46	0,56	0,68	0,79	0,89	0,97	1,00	1,00	1,00	1,00	1,00	1,00
recall	0,33	0,37	0,39	0,39	0,38	0,37	0,33	0,28	0,23	0,19	0,14	0,10	0,06	0,04	0,02	0,01	0,00	0,00
missrate	0,67	0,63	0,61	0,61	0,62	0,63	0,67	0,72	0,77	0,81	0,86	0,90	0,94	0,96	0,98	0,99	1,00	1,00
fppi	1,87	1,66	1,38	1,03	0,73	0,48	0,29	0,16	0,08	0,04	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00
rec(c)-rec(c)_w48_EL_np_1.2_42_35_auto.yml {color} (amr: 0.78)																		



# A Ergebnisse der Evaluation einzelner Algorithmen

fps	83,99	84,41	84,78	85,23	84,78	85,07	85,28	85,17	85,72	85,12	84,96	85,03	85,01	85,00	85,07	84,93	85,15	85,58
precision	0,17	0,21	0,25	0,31	0,39	0,48	0,58	0,67	0,78	0,87	0,93	0,96	0,99	0,99	0,99	1,00	1,00	1,00
recall	0,38	0,39	0,40	0,40	0,38	0,35	0,31	0,26	0,22	0,17	0,13	0,09	0,05	0,03	0,02	0,01	0,00	0,00
missrate	0,62	0,61	0,60	0,60	0,62	0,65	0,69	0,74	0,78	0,83	0,87	0,91	0,95	0,97	0,98	0,99	1,00	1,00
fppi	1,38	1,11	0,87	0,65	0,44	0,28	0,17	0,10	0,05	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00
rec(c)-rec(c)_w48_EL_np_1.6_56_35_auto.yml {color} (amr: 0.80)																		
fps	84,87	84,99	84,99	85,03	85,48	85,11	85,23	85,55	85,48	85,69	85,49	85,54	85,45	85,72	85,32	85,37	85,46	85,20
precision	0,20	0,23	0,27	0,31	0,37	0,46	0,60	0,74	0,84	0,90	0,94	0,93	0,95	0,98	1,00	1,00	1,00	1,00
recall	0,38	0,39	0,38	0,37	0,34	0,31	0,26	0,22	0,17	0,12	0,08	0,04	0,02	0,01	0,01	0,00	0,00	0,00
missrate	0,62	0,61	0,62	0,63	0,66	0,69	0,74	0,78	0,83	0,88	0,92	0,96	0,98	0,99	0,99	1,00	1,00	1,00
fppi	1,14	0,95	0,78	0,62	0,43	0,26	0,13	0,06	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
rec(c)-rec(c)_w48_EL_np_2.0_70_35_auto.yml {color} (amr: 0.77)																		
fps	84,64	85,22	84,83	85,33	85,53	85,58	85,76	85,71	85,16	85,46	85,28	85,29	85,26	85,51	85,79	85,54	85,54	85,74
precision	0,21	0,26	0,31	0,39	0,48	0,60	0,73	0,84	0,90	0,95	0,97	0,98	1,00	1,00	1,00	1,00	1,00	1,00
recall	0,39	0,40	0,39	0,37	0,33	0,30	0,26	0,21	0,16	0,11	0,07	0,04	0,02	0,01	0,00	0,00	0,00	0,00
missrate	0,61	0,60	0,61	0,63	0,67	0,70	0,74	0,79	0,84	0,89	0,93	0,96	0,98	0,99	1,00	1,00	1,00	1,00
fppi	1,08	0,85	0,64	0,43	0,27	0,15	0,07	0,03	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
rec(c)-rec(c)_w48_EL_np_2.4_84_35_auto.yml {color} (amr: 0.78)																		
fps	86,28	87,10	86,40	86,65	87,19	87,27	86,78	86,88	86,99	87,07	86,88	86,96	87,22	87,06	87,11	87,18	87,03	86,78
precision	0,23	0,27	0,33	0,42	0,54	0,68	0,79	0,90	0,96	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
recall	0,40	0,40	0,38	0,35	0,31	0,26	0,20	0,15	0,10	0,06	0,03	0,02	0,01	0,00	0,00	0,00	0,00	0,00
missrate	0,60	0,60	0,62	0,65	0,69	0,74	0,80	0,85	0,90	0,94	0,97	0,98	0,99	1,00	1,00	1,00	1,00	1,00
fppi	0,99	0,80	0,57	0,35	0,19	0,09	0,04	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
rec(c)-rec(c)_w48_EL_np_2.8_98_35_auto.yml {color} (amr: 0.84)																		
fps	83,22	83,52	84,19	83,91	84,33	84,28	84,19	84,10	84,13	84,20	84,23	84,37	84,51	84,03	84,12	84,27	84,16	84,43
precision	0,14	0,18	0,22	0,29	0,40	0,52	0,63	0,75	0,90	0,95	0,97	1,00	1,00	1,00	1,00	1,00	1,00	1,00
recall	0,34	0,36	0,36	0,35	0,31	0,26	0,20	0,14	0,09	0,05	0,03	0,01	0,01	0,00	0,00	0,00	0,00	0,00
missrate	0,66	0,64	0,64	0,65	0,69	0,74	0,80	0,86	0,91	0,95	0,97	0,99	0,99	1,00	1,00	1,00	1,00	1,00
fppi	1,63	1,26	0,95	0,64	0,34	0,18	0,09	0,03	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
rec(ir)-rec(ir)_w96_EL_np_0.8_28_35_auto.yml {ir} (amr: 0.76)																		
fps	240,31	247,97	253,47	255,30	256,64	258,47	257,51	259,31	258,41	259,76	259,76	258,96	260,11	259,43	259,28	259,57	259,69	259,16
precision	0,14	0,19	0,24	0,30	0,38	0,48	0,59	0,72	0,84	0,89	0,89	0,86	0,83	0,77	0,70	0,67	0,74	0,98
recall	0,46	0,45	0,43	0,40	0,38	0,34	0,31	0,27	0,22	0,17	0,13	0,09	0,06	0,04	0,02	0,02	0,01	0,01
missrate	0,54	0,55	0,57	0,60	0,62	0,66	0,69	0,73	0,78	0,83	0,87	0,91	0,94	0,96	0,98	0,98	0,99	0,99
fppi	2,01	1,42	1,00	0,69	0,46	0,28	0,16	0,08	0,03	0,02	0,01	0,01	0,01	0,01	0,01	0,01	0,00	0,00
rec(ir)-rec(ir)_w96_EL_np_1.2_42_35_auto.yml {ir} (amr: 0.77)																		
fps	245,94	251,27	255,41	253,16	258,14	257,15	256,13	256,22	253,91	255,49	256,12	259,06	259,35	263,71	259,84	261,00	261,82	253,58
precision	0,22	0,27	0,33	0,39	0,44	0,49	0,57	0,71	0,84	0,91	0,91	0,89	0,88	0,85	0,83	0,97	1,00	1,00
recall	0,44	0,42	0,40	0,37	0,35	0,32	0,29	0,25	0,22	0,18	0,14	0,09	0,06	0,04	0,02	0,02	0,01	0,01
missrate	0,56	0,58	0,60	0,63	0,65	0,68	0,71	0,75	0,78	0,82	0,86	0,91	0,94	0,96	0,98	0,98	0,99	0,99
fppi	1,14	0,83	0,60	0,44	0,33	0,24	0,16	0,08	0,03	0,01	0,01	0,01	0,01	0,00	0,00	0,00	0,00	0,00
rec(ir)-rec(ir)_w96_EL_np_1.6_56_35_auto.yml {ir} (amr: 0.74)																		
fps	244,28	248,24	250,67	251,83	253,39	257,77	259,51	260,08	258,50	260,91	260,61	260,38	261,55	260,25	259,38	260,13	259,49	259,83
precision	0,21	0,27	0,33	0,40	0,48	0,61	0,74	0,84	0,91	0,92	0,91	0,89	0,87	0,82	0,85	0,97	1,00	1,00
recall	0,46	0,44	0,41	0,38	0,35	0,32	0,29	0,25	0,21	0,17	0,13	0,09	0,06	0,04	0,02	0,02	0,01	0,01
missrate	0,54	0,56	0,59	0,62	0,65	0,68	0,71	0,75	0,79	0,83	0,87	0,91	0,94	0,96	0,98	0,98	0,99	0,99
fppi	1,30	0,90	0,62	0,42	0,28	0,15	0,07	0,04	0,02	0,01	0,01	0,01	0,01	0,01	0,00	0,00	0,00	0,00
rec(ir)-rec(ir)_w96_EL_np_2.0_70_35_auto.yml {ir} (amr: 0.77)																		
fps	252,01	254,46	255,87	256,30	256,96	258,11	257,94	258,11	257,58	257,00	255,17	255,25	256,14	256,07	256,62	257,94	261,73	262,49
precision	0,25	0,31	0,39	0,47	0,57	0,66	0,72	0,80	0,88	0,89	0,89	0,87	0,85	0,86	0,98	1,00	1,00	1,00
recall	0,42	0,40	0,38	0,35	0,32	0,29	0,25	0,21	0,17	0,13	0,10	0,06	0,04	0,02	0,01	0,01	0,01	0,01
missrate	0,58	0,60	0,62	0,65	0,68	0,71	0,75	0,79	0,83	0,87	0,90	0,94	0,96	0,98	0,99	0,99	0,99	0,99
fppi	0,94	0,65	0,44	0,30	0,19	0,11	0,07	0,04	0,02	0,01	0,01	0,01	0,01	0,00	0,00	0,00	0,00	0,00
rec(ir)-rec(ir)_w96_EL_np_2.4_84_35_auto.yml {ir} (amr: 0.70)																		
fps	255,41	259,00	262,78	264,71	265,31	263,96	265,23	264,64	265,38	264,33	264,62	264,08	263,20	264,60	265,20	264,98	265,60	265,59
precision	0,14	0,24	0,36	0,54	0,72	0,83	0,89	0,92	0,93	0,92	0,90	0,87	0,82	0,82	0,96	1,00	1,00	1,00
recall	0,47	0,45	0,42	0,38	0,35	0,31	0,26	0,21	0,16	0,12	0,08	0,05	0,03	0,01	0,01	0,00	0,00	0,00
missrate	0,53	0,55	0,58	0,62	0,65	0,69	0,74	0,79	0,84	0,88	0,92	0,95	0,97	0,99	0,99	1,00	1,00	1,00
fppi	2,13	1,07	0,54	0,24	0,10	0,05	0,02	0,01	0,01	0,01	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00
rec(ir)-rec(ir)_w96_EL_np_2.8_98_35_auto.yml {ir} (amr: 0.73)																		
fps	251,34	255,84	255,24	256,96	256,37	258,79	257,17	257,77	254,12	256,35	261,14	261,76	261,22	259,67	257,69	256,65	261,14	250,37
precision	0,16	0,24	0,34	0,45	0,58	0,75	0,88	0,91	0,92	0,90	0,89	0,85	0,80	0,76	0,79	0,99	1,00	1,00
recall	0,45	0,43	0,40	0,37	0,34	0,30	0,26	0,21	0,17	0,12	0,09	0,06	0,03	0,02	0,01	0,01	0,01	0,00
missrate	0,55	0,57	0,60	0,63	0,66	0,70	0,74	0,79	0,83	0,88	0,91	0,94	0,97	0,98	0,99	0,99	0,99	1,00
fppi	1,80	1,00	0,58	0,33	0,18	0,08	0,03	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,00	0,00	0,00	0,00

Variation des Eingangsbildes – Skalierungsfaktor

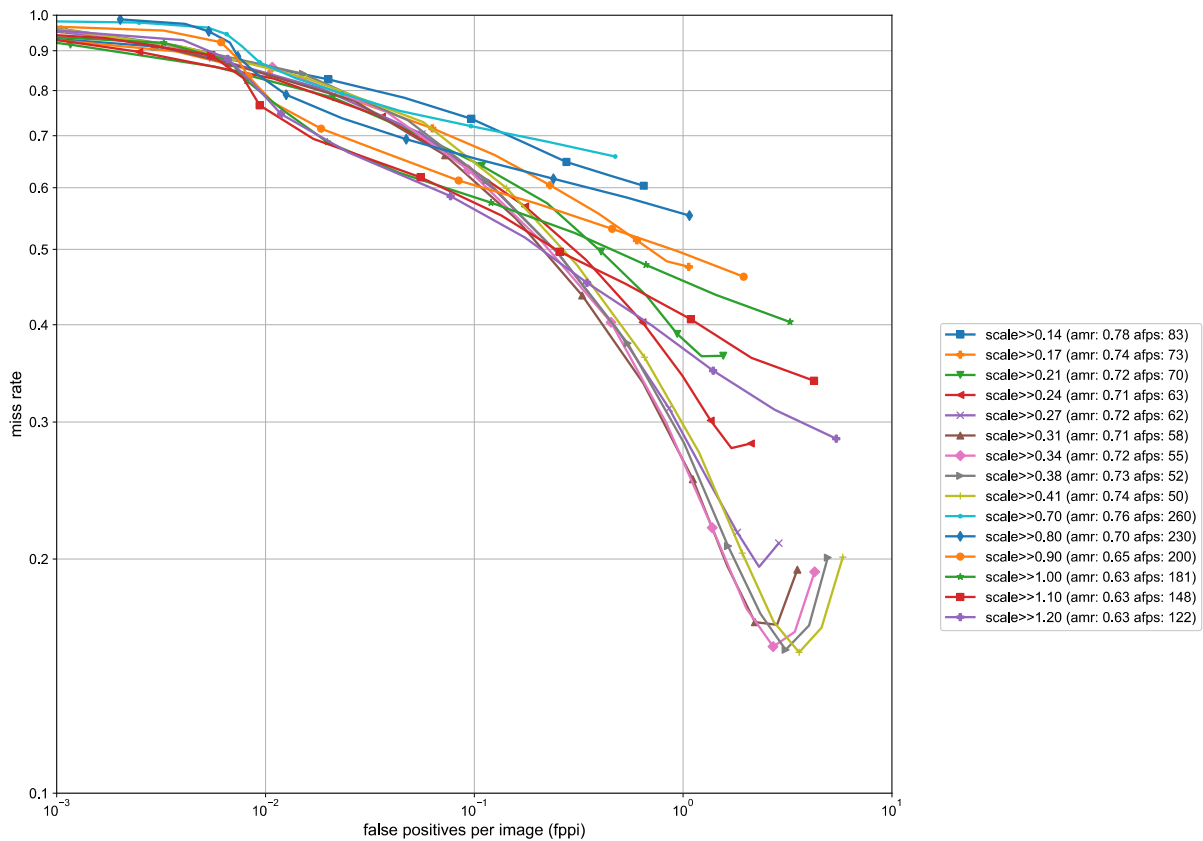


Abbildung A.1-6: Ergebnis der Variation der Skalierung des Eingangsbildes.

Tabelle A.1-6: Rohdaten zur Variation der Skalierung des Eingangsbildes.

Scale>>0.40 {color} (amr: 0.78)																
fps	82,58	83,25	81,95	82,52	84,19	83,60	82,60	82,92	83,50	83,31	83,03	83,58	83,00	82,78	84,08	83,32
precision	0,31	0,39	0,49	0,58	0,67	0,78	0,87	0,93	0,96	0,99	0,99	0,99	1,00	1,00	1,00	1,00
recall	0,40	0,38	0,35	0,31	0,26	0,22	0,17	0,13	0,09	0,05	0,03	0,02	0,01	0,00	0,00	0,00
missrate	0,60	0,62	0,65	0,69	0,74	0,78	0,83	0,87	0,91	0,95	0,97	0,98	0,99	1,00	1,00	1,00
fpfi	0,65	0,44	0,28	0,17	0,10	0,05	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Scale>>0.50 {color} (amr: 0.74)																
fps	73,01	73,18	75,37	74,15	70,91	72,62	73,94	73,33	73,20	75,39	72,99	74,02	75,11	73,87	73,32	72,99
precision	0,27	0,31	0,38	0,46	0,56	0,67	0,77	0,86	0,92	0,95	0,99	0,99	1,00	1,00	1,00	0,00
recall	0,53	0,52	0,49	0,44	0,40	0,34	0,28	0,22	0,16	0,10	0,06	0,03	0,01	0,00	0,00	0,00
missrate	0,47	0,48	0,51	0,56	0,60	0,66	0,72	0,78	0,84	0,90	0,94	0,97	0,99	1,00	1,00	1,00
fpfi	1,06	0,84	0,60	0,40	0,23	0,13	0,06	0,03	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00
Scale>>0.60 {color} (amr: 0.72)																
fps	69,91	70,48	69,83	68,06	69,19	70,80	69,33	70,42	69,89	70,60	71,07	70,31	70,57	71,03	69,97	69,85
precision	0,23	0,28	0,33	0,39	0,48	0,59	0,71	0,81	0,88	0,95	0,98	0,99	1,00	1,00	1,00	1,00
recall	0,64	0,64	0,61	0,56	0,50	0,43	0,36	0,29	0,22	0,14	0,08	0,05	0,02	0,01	0,00	0,00
missrate	0,36	0,36	0,39	0,44	0,50	0,57	0,64	0,71	0,78	0,86	0,92	0,95	0,98	0,99	1,00	1,00
fpfi	1,56	1,23	0,94	0,67	0,40	0,22	0,11	0,05	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00
Scale>>0.70 {color} (amr: 0.71)																
fps	63,51	64,21	63,21	63,81	63,63	63,39	64,58	64,42	63,96	62,56	62,89	64,81	64,33	63,13	66,40	64,02

## A Ergebnisse der Evaluation einzelner Algorithmen

---

<b>precision</b>	0,20	0,24	0,28	0,33	0,41	0,53	0,65	0,76	0,84	0,92	0,97	0,98	0,99	0,99	1,00	1,00
<b>recall</b>	0,72	0,72	0,70	0,66	0,60	0,52	0,43	0,35	0,26	0,17	0,10	0,06	0,03	0,01	0,00	0,00
<b>missrate</b>	0,28	0,28	0,30	0,34	0,40	0,48	0,57	0,65	0,74	0,83	0,90	0,94	0,97	0,99	1,00	1,00
<b>fppi</b>	2,11	1,70	1,35	0,99	0,64	0,34	0,18	0,08	0,04	0,01	0,00	0,00	0,00	0,00	0,00	0,00
<b>Scale&gt;&gt;0.80 {color} (amr: 0.72)</b>																
<b>fps</b>	59,55	61,50	61,76	62,10	62,65	62,63	60,99	62,13	61,95	62,42	63,22	62,09	63,19	62,54	62,50	63,67
<b>precision</b>	0,17	0,21	0,24	0,30	0,37	0,48	0,60	0,70	0,80	0,89	0,94	0,96	0,98	1,00	1,00	1,00
<b>recall</b>	0,79	0,80	0,78	0,75	0,69	0,60	0,50	0,40	0,30	0,19	0,12	0,07	0,03	0,01	0,01	0,00
<b>missrate</b>	0,21	0,20	0,22	0,25	0,31	0,40	0,50	0,60	0,70	0,81	0,88	0,93	0,97	0,99	0,99	1,00
<b>fppi</b>	2,87	2,31	1,82	1,31	0,86	0,48	0,25	0,13	0,06	0,02	0,01	0,00	0,00	0,00	0,00	0,00
<b>Scale&gt;&gt;0.90 {color} (amr: 0.71)</b>																
<b>fps</b>	56,33	57,38	57,87	58,59	58,68	58,99	59,53	58,63	58,90	58,71	58,10	58,78	59,45	59,59	58,60	59,65
<b>precision</b>	0,15	0,18	0,22	0,27	0,33	0,43	0,56	0,68	0,78	0,86	0,94	0,96	0,97	0,99	1,00	1,00
<b>recall</b>	0,81	0,84	0,83	0,80	0,75	0,66	0,56	0,45	0,34	0,22	0,13	0,07	0,04	0,01	0,01	0,00
<b>missrate</b>	0,19	0,16	0,17	0,20	0,25	0,34	0,44	0,55	0,66	0,78	0,87	0,93	0,96	0,99	0,99	1,00
<b>fppi</b>	3,53	2,82	2,21	1,63	1,11	0,65	0,33	0,16	0,07	0,03	0,01	0,00	0,00	0,00	0,00	0,00
<b>Scale&gt;&gt;1.00 {color} (amr: 0.72)</b>																
<b>fps</b>	52,51	53,98	54,78	55,57	55,32	55,98	55,36	56,18	55,65	56,14	56,01	54,83	55,49	55,25	54,65	54,60
<b>precision</b>	0,12	0,15	0,19	0,23	0,30	0,39	0,50	0,63	0,74	0,83	0,91	0,95	0,97	0,99	1,00	1,00
<b>recall</b>	0,81	0,84	0,85	0,83	0,78	0,70	0,60	0,49	0,37	0,25	0,14	0,08	0,04	0,02	0,01	0,00
<b>missrate</b>	0,19	0,16	0,15	0,17	0,22	0,30	0,40	0,51	0,63	0,75	0,86	0,92	0,96	0,98	0,99	1,00
<b>fppi</b>	4,26	3,43	2,70	2,01	1,38	0,83	0,45	0,21	0,09	0,04	0,01	0,00	0,00	0,00	0,00	0,00
<b>Scale&gt;&gt;1.10 {color} (amr: 0.73)</b>																
<b>fps</b>	49,55	51,04	52,09	52,78	52,59	53,02	52,59	52,87	53,10	52,66	53,23	52,68	52,68	53,08	53,24	52,86
<b>precision</b>	0,11	0,13	0,17	0,21	0,26	0,35	0,46	0,60	0,71	0,81	0,89	0,94	0,97	0,98	1,00	1,00
<b>recall</b>	0,80	0,84	0,85	0,83	0,79	0,72	0,62	0,51	0,39	0,27	0,16	0,09	0,04	0,02	0,01	0,00
<b>missrate</b>	0,20	0,16	0,15	0,17	0,21	0,28	0,38	0,49	0,61	0,73	0,84	0,91	0,96	0,98	0,99	1,00
<b>fppi</b>	4,94	4,01	3,10	2,34	1,64	1,01	0,54	0,25	0,11	0,05	0,02	0,00	0,00	0,00	0,00	0,00
<b>Scale&gt;&gt;1.20 {color} (amr: 0.74)</b>																
<b>fps</b>	46,01	48,07	49,24	49,94	50,15	50,40	50,40	50,35	50,68	50,65	50,54	50,78	50,67	50,71	50,89	50,81
<b>precision</b>	0,09	0,12	0,15	0,19	0,24	0,31	0,42	0,56	0,68	0,78	0,89	0,94	0,97	0,98	1,00	1,00
<b>recall</b>	0,80	0,84	0,85	0,83	0,80	0,73	0,64	0,52	0,40	0,27	0,17	0,09	0,04	0,02	0,01	0,00
<b>missrate</b>	0,20	0,16	0,15	0,17	0,20	0,27	0,36	0,48	0,60	0,73	0,83	0,91	0,96	0,98	0,99	1,00
<b>fppi</b>	5,82	4,60	3,59	2,71	1,92	1,19	0,65	0,31	0,14	0,06	0,02	0,00	0,00	0,00	0,00	0,00
<b>scale&gt;&gt;0.70 {ir} (amr: 0.76)</b>																
<b>fps</b>	269,89	264,19	263,23	251,20	260,50	255,24	256,68	256,92	269,78	249,95	267,85	257,41	262,19			
<b>precision</b>	0,35	0,51	0,68	0,81	0,87	0,90	0,91	0,89	0,86	0,83	0,87	0,98	1,00			
<b>recall</b>	0,34	0,31	0,28	0,25	0,21	0,17	0,13	0,09	0,05	0,04	0,02	0,01	0,01			
<b>missrate</b>	0,66	0,69	0,72	0,75	0,79	0,83	0,87	0,91	0,95	0,96	0,98	0,99	0,99			
<b>fppi</b>	0,47	0,22	0,10	0,04	0,02	0,01	0,01	0,01	0,01	0,01	0,01	0,00	0,00			
<b>scale&gt;&gt;0.80 {ir} (amr: 0.70)</b>																
<b>fps</b>	231,13	226,45	236,98	232,93	222,28	223,59	235,53	228,78	228,43	225,92	232,22	227,50	240,69			
<b>precision</b>	0,24	0,36	0,54	0,72	0,83	0,89	0,93	0,93	0,92	0,90	0,87	0,82	0,82			
<b>recall</b>	0,45	0,42	0,38	0,35	0,31	0,26	0,21	0,16	0,12	0,08	0,05	0,03	0,01			
<b>missrate</b>	0,55	0,58	0,62	0,65	0,69	0,74	0,79	0,84	0,88	0,92	0,95	0,97	0,99			
<b>fppi</b>	1,07	0,54	0,24	0,10	0,05	0,02	0,01	0,01	0,01	0,01	0,01	0,00	0,00			

scale>>0.90 {ir} (amr: 0.65)													
fps	195,07	198,14	199,79	200,93	201,43	201,37	201,90	201,75	201,31	202,28	201,87	202,67	202,82
precision	0,17	0,28	0,43	0,61	0,77	0,87	0,92	0,94	0,94	0,92	0,90	0,91	0,98
recall	0,54	0,50	0,47	0,43	0,39	0,33	0,29	0,23	0,17	0,12	0,08	0,04	0,02
missrate	0,46	0,50	0,53	0,57	0,61	0,67	0,71	0,77	0,83	0,88	0,92	0,96	0,98
fppi	1,95	0,94	0,46	0,20	0,08	0,04	0,02	0,01	0,01	0,01	0,01	0,00	0,00
scale>>1.00 {ir} (amr: 0.63)													
fps	171,59	179,25	181,75	182,05	181,98	182,84	182,74	182,85	183,67	182,85	182,98	183,00	183,20
precision	0,12	0,23	0,37	0,54	0,72	0,86	0,92	0,94	0,94	0,93	0,95	0,99	1,00
recall	0,60	0,56	0,52	0,48	0,43	0,37	0,31	0,25	0,18	0,13	0,08	0,04	0,02
missrate	0,40	0,44	0,48	0,52	0,57	0,63	0,69	0,75	0,82	0,87	0,92	0,96	0,98
fppi	3,25	1,43	0,66	0,30	0,12	0,04	0,02	0,01	0,01	0,01	0,00	0,00	0,00
scale>>1.10 {ir} (amr: 0.63)													
fps	140,66	146,02	148,29	149,44	149,70	149,69	149,72	149,44	149,37	148,92	147,92	147,66	147,70
precision	0,10	0,18	0,29	0,43	0,59	0,71	0,84	0,93	0,95	0,94	0,94	0,97	0,99
recall	0,66	0,64	0,59	0,55	0,50	0,45	0,38	0,31	0,23	0,17	0,11	0,07	0,04
missrate	0,34	0,36	0,41	0,45	0,50	0,55	0,62	0,69	0,77	0,83	0,89	0,93	0,96
fppi	4,23	2,12	1,09	0,53	0,26	0,13	0,06	0,02	0,01	0,01	0,01	0,00	0,00
scale>>1.20 {ir} (amr: 0.63)													
fps	114,30	120,59	123,35	123,58	123,76	124,28	124,20	124,23	124,66	123,91	123,49	123,71	123,96
precision	0,09	0,16	0,26	0,39	0,54	0,67	0,80	0,91	0,94	0,94	0,93	0,93	0,98
recall	0,71	0,69	0,65	0,60	0,55	0,48	0,41	0,33	0,25	0,18	0,12	0,07	0,04
missrate	0,29	0,31	0,35	0,40	0,45	0,52	0,59	0,67	0,75	0,82	0,88	0,93	0,96
fppi	5,41	2,74	1,39	0,71	0,35	0,17	0,08	0,02	0,01	0,01	0,01	0,00	0,00

Variation des Eingangsbildes – Weichzeichnen (Gauß-Filter & Bilateraler Filter)

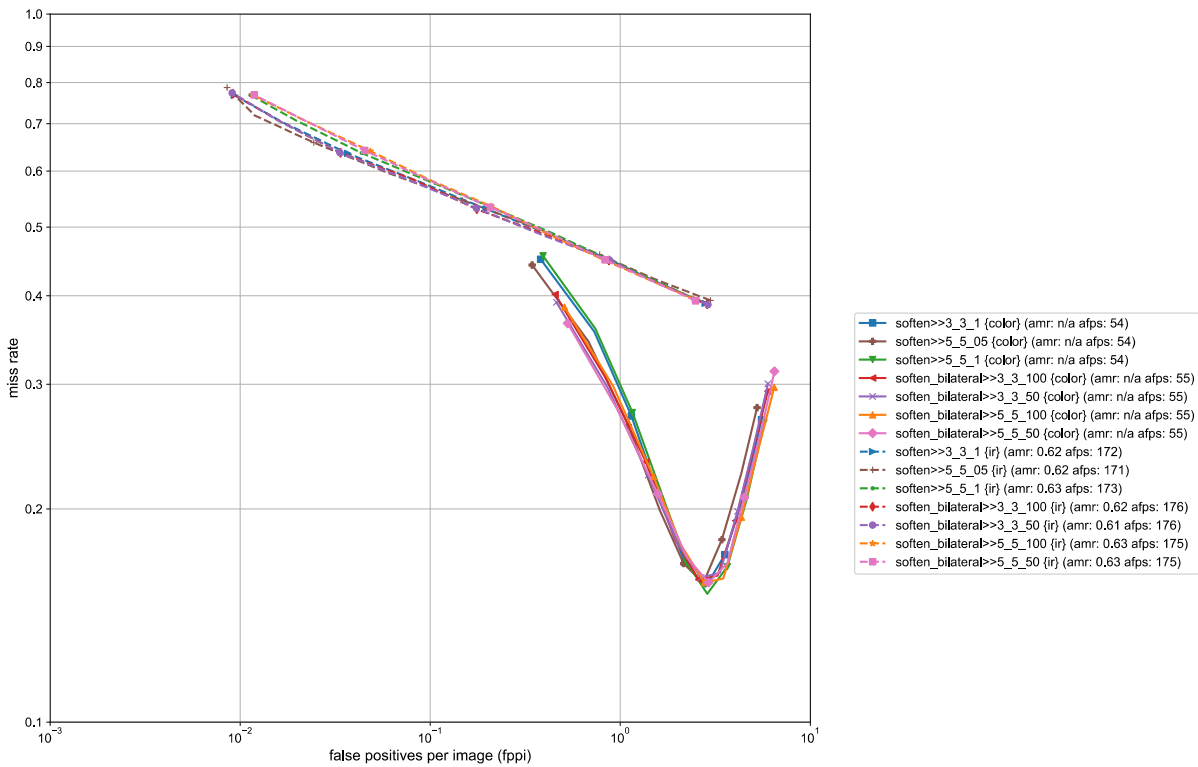


Abbildung A.1-7: Ergebnis der Variation der Weichzeichnung des Eingangsbildes.

Tabelle A.1-7: Rohdaten zur Variation der Weichzeichnung des Eingangsbildes.

soften>>3_3_1 (amr: n/a)									
fps	106,00	111,38	112,66	113,56	115,27	114,69	114,89	115,55	114,95
precision	0,11	0,14	0,19	0,24	0,29	0,34	0,41	0,48	0,60
recall	0,67	0,69	0,69	0,68	0,65	0,61	0,55	0,47	0,39
missrate	0,33	0,31	0,31	0,32	0,35	0,39	0,45	0,53	0,61
fpfi	4,17	3,07	2,22	1,62	1,19	0,86	0,59	0,37	0,20
soften>>5_5_05 (amr: n/a)									
fps	106,06	112,36	112,47	113,86	114,95	114,52	113,82	115,33	114,36
precision	0,11	0,15	0,19	0,24	0,29	0,35	0,42	0,50	0,62
recall	0,66	0,68	0,68	0,67	0,64	0,60	0,54	0,47	0,39
missrate	0,34	0,32	0,32	0,33	0,36	0,40	0,46	0,53	0,61
fpfi	4,12	2,92	2,10	1,55	1,15	0,84	0,56	0,35	0,18
soften>>5_5_1 (amr: n/a)									
fps	109,23	112,95	112,59	114,02	115,37	114,17	114,77	115,95	114,19
precision	0,11	0,14	0,19	0,23	0,28	0,34	0,40	0,48	0,59
recall	0,67	0,69	0,69	0,68	0,65	0,60	0,55	0,47	0,39
missrate	0,33	0,31	0,31	0,32	0,35	0,40	0,45	0,53	0,61
fpfi	4,14	3,10	2,25	1,65	1,22	0,87	0,60	0,38	0,20
soften_bilateral>>3_3_100 (amr: n/a)									
fps	110,08	113,99	115,91	116,96	117,54	117,92	118,18	118,18	117,90
precision	0,10	0,13	0,17	0,22	0,26	0,31	0,38	0,46	0,57
recall	0,66	0,67	0,68	0,68	0,66	0,62	0,57	0,49	0,41

<b>missrate</b>	0,34	0,33	0,32	0,32	0,34	0,38	0,43	0,51	0,59
<b>fppi</b>	4,42	3,30	2,46	1,83	1,38	1,01	0,70	0,43	0,23
<b>soften_bilateral&gt;&gt;3_3_50 (amr: n/a)</b>									
<b>fps</b>	109,61	113,48	116,13	117,03	117,83	118,14	117,93	118,49	118,21
<b>precision</b>	0,10	0,13	0,17	0,21	0,26	0,31	0,37	0,46	0,57
<b>recall</b>	0,66	0,67	0,68	0,68	0,65	0,62	0,57	0,50	0,42
<b>missrate</b>	0,34	0,33	0,32	0,32	0,35	0,38	0,43	0,50	0,58
<b>fppi</b>	4,44	3,32	2,51	1,87	1,42	1,04	0,72	0,44	0,24
<b>soften_bilateral&gt;&gt;5_5_100 (amr: n/a)</b>									
<b>fps</b>	109,99	112,85	115,05	116,26	116,46	117,30	116,95	117,17	116,82
<b>precision</b>	0,10	0,13	0,16	0,20	0,25	0,30	0,35	0,44	0,55
<b>recall</b>	0,65	0,67	0,68	0,68	0,65	0,62	0,57	0,50	0,42
<b>missrate</b>	0,35	0,33	0,32	0,32	0,35	0,38	0,43	0,50	0,58
<b>fppi</b>	4,45	3,43	2,58	1,96	1,49	1,10	0,77	0,47	0,26
<b>soften_bilateral&gt;&gt;5_5_50 (amr: n/a)</b>									
<b>fps</b>	109,66	112,97	115,24	116,22	117,12	117,49	117,43	117,28	117,29
<b>precision</b>	0,10	0,12	0,16	0,20	0,24	0,28	0,34	0,43	0,54
<b>recall</b>	0,65	0,66	0,67	0,67	0,65	0,62	0,57	0,51	0,43
<b>missrate</b>	0,35	0,34	0,33	0,33	0,35	0,38	0,43	0,49	0,57
<b>fppi</b>	4,49	3,49	2,67	2,04	1,56	1,17	0,81	0,49	0,27

### Variation des Eingangsbildes (Intensitätsbild) – IR-Konvertierung

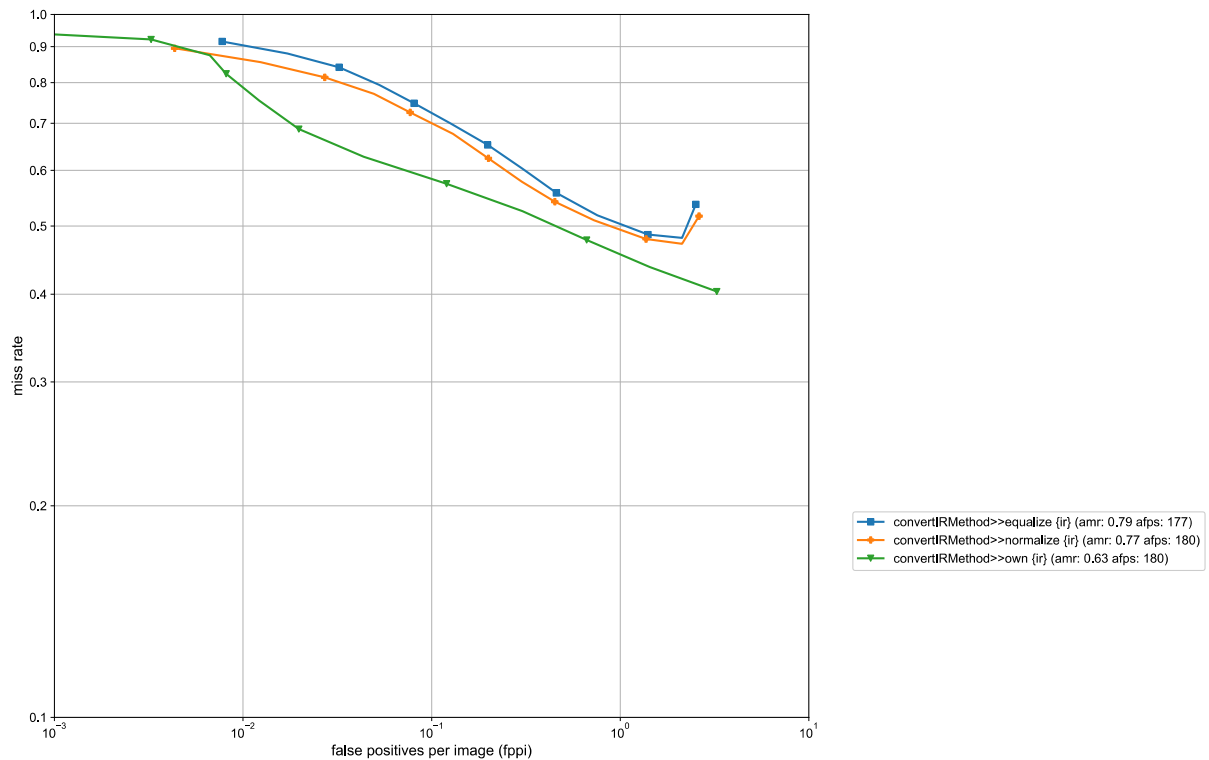


Abbildung A.1-8: Ergebnis der Variation der IR-Konvertierungsmethode.

# A Ergebnisse der Evaluation einzelner Algorithmen

Tabelle A.1-8: Rohdaten zur Variation der IR-Konvertierungsmethode.

convertIRMethod>>equalize {ir} (amr: 0.79)														
fps	164,45	172,93	175,88	175,37	176,22	178,21	179,06	179,30	179,76	181,24	180,34	180,26	180,81	
precision	0,12	0,15	0,21	0,32	0,42	0,49	0,57	0,64	0,70	0,74	0,78	0,84	0,89	
recall	0,46	0,52	0,51	0,48	0,44	0,40	0,35	0,30	0,25	0,21	0,16	0,12	0,08	
missrate	0,54	0,48	0,49	0,52	0,56	0,60	0,65	0,70	0,75	0,79	0,84	0,88	0,92	
fppi	2,52	2,13	1,40	0,76	0,46	0,30	0,20	0,13	0,08	0,05	0,03	0,02	0,01	
convertIRMethod>>normalize {ir} (amr: 0.77)														
fps	167,83	175,59	178,92	180,51	181,41	181,77	182,80	183,72	183,62	182,00	182,75	182,50	182,31	
precision	0,12	0,16	0,22	0,33	0,43	0,51	0,58	0,65	0,73	0,77	0,84	0,90	0,95	
recall	0,48	0,53	0,52	0,49	0,46	0,42	0,38	0,32	0,27	0,23	0,19	0,14	0,10	
missrate	0,52	0,47	0,48	0,51	0,54	0,58	0,62	0,68	0,73	0,77	0,81	0,86	0,90	
fppi	2,62	2,13	1,37	0,73	0,45	0,30	0,20	0,13	0,08	0,05	0,03	0,01	0,00	
convertIRMethod>>own {ir} (amr: 0.63)														
fps	172,73	177,50	179,63	182,02	182,83	182,83	183,26	182,52	182,85	181,49	182,09	180,83	181,90	
precision	0,12	0,23	0,37	0,54	0,72	0,86	0,92	0,94	0,94	0,93	0,95	0,99	1,00	
recall	0,60	0,56	0,52	0,48	0,43	0,37	0,31	0,25	0,18	0,13	0,08	0,04	0,02	
missrate	0,40	0,44	0,48	0,52	0,57	0,63	0,69	0,75	0,82	0,87	0,92	0,96	0,98	
fppi	3,25	1,43	0,66	0,30	0,12	0,04	0,02	0,01	0,01	0,01	0,00	0,00	0,00	

## Variation des Eingangsbildes (Intensitätsbild) – Normalisierung und Egalisierung

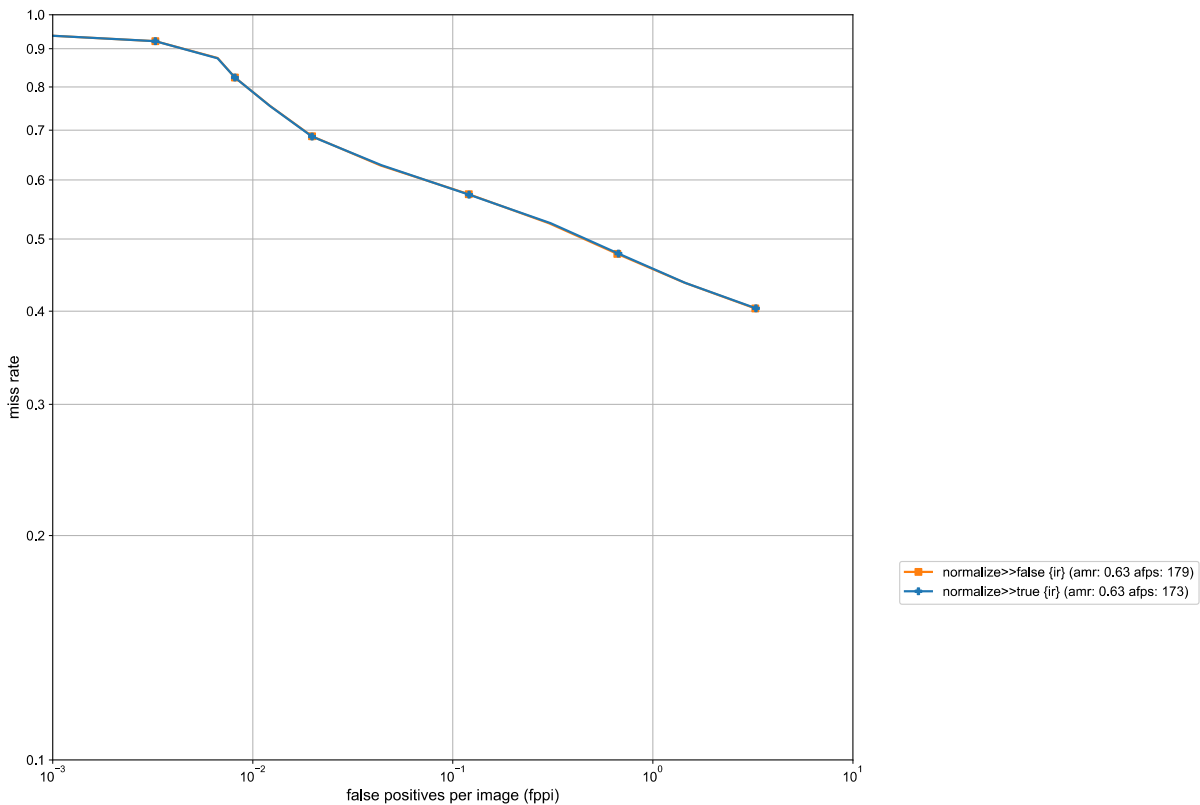


Abbildung A.1-9: Ergebnis der Erkennungsleistung bei Vorverarbeitung des Intensitätsbildes durch Normalisierung.

Tabelle A.1-9: Rohdaten zur Erkennungsleistung bei Vorverarbeitung des Intensitätsbildes durch Normalisierung.

normalize>>>false {ir} (amr: 0.63)														
fps	172,33	176,37	179,23	180,54	180,72	181,47	180,15	180,85	180,84	180,06	180,17	182,92	181,24	
precision	0,12	0,23	0,37	0,54	0,72	0,86	0,92	0,94	0,94	0,93	0,95	0,99	1,00	
recall	0,60	0,56	0,52	0,48	0,43	0,37	0,31	0,25	0,18	0,13	0,08	0,04	0,02	
missrate	0,40	0,44	0,48	0,52	0,57	0,63	0,69	0,75	0,82	0,87	0,92	0,96	0,98	
fppi	3,25	1,43	0,66	0,30	0,12	0,04	0,02	0,01	0,01	0,01	0,00	0,00	0,00	
normalize>>>true {ir} (amr: 0.63)														
fps	170,55	175,10	177,33	178,56	179,44	178,21	173,82	170,35	170,22	177,23	170,95	170,65	168,71	
precision	0,12	0,22	0,37	0,53	0,72	0,86	0,92	0,94	0,94	0,93	0,95	0,99	1,00	
recall	0,60	0,56	0,52	0,48	0,43	0,37	0,31	0,25	0,18	0,13	0,08	0,04	0,02	
missrate	0,40	0,44	0,48	0,52	0,57	0,63	0,69	0,75	0,82	0,87	0,92	0,96	0,98	
fppi	3,28	1,45	0,67	0,31	0,12	0,04	0,02	0,01	0,01	0,01	0,00	0,00	0,00	

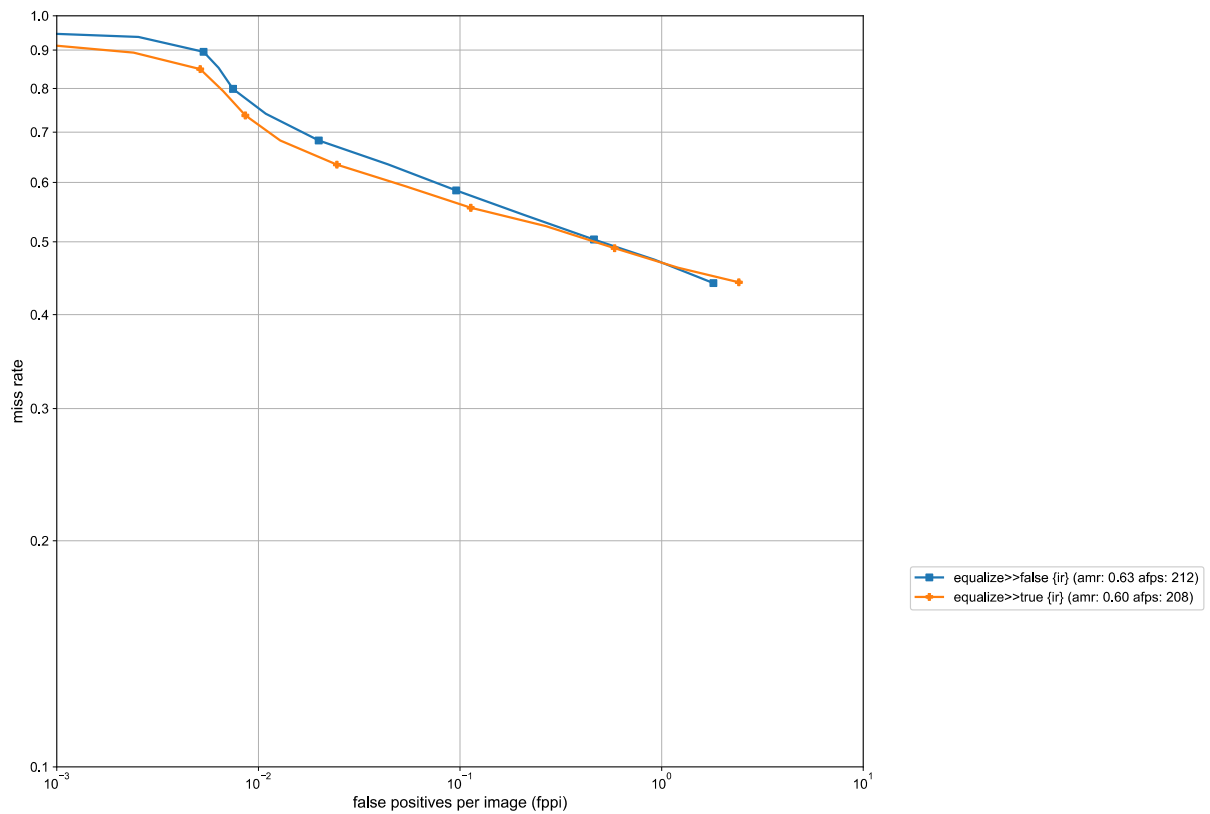


Abbildung A.1-10: Ergebnis der Erkennungsleistung bei Vorverarbeitung des Intensitätsbildes durch Egalisierung.



# A Ergebnisse der Evaluation einzelner Algorithmen

Tabelle A.1-10: Rohdaten zur Erkennungsleistung bei Vorverarbeitung des Intensitätsbildes durch Egalisierung.

equalize>>>false {ir} (amr: 0.63)													
fps	209,87	209,75	214,79	215,04	209,69	214,10	212,22	210,72	210,48	213,44	210,06	213,12	214,35
precision	0,18	0,29	0,44	0,62	0,76	0,86	0,92	0,95	0,95	0,94	0,93	0,95	0,99
recall	0,56	0,53	0,50	0,46	0,41	0,37	0,32	0,26	0,20	0,15	0,10	0,06	0,04
missrate	0,44	0,47	0,50	0,54	0,59	0,63	0,68	0,74	0,80	0,85	0,90	0,94	0,96
fppi	1,80	0,92	0,46	0,21	0,10	0,04	0,02	0,01	0,01	0,01	0,01	0,00	0,00
equalize>>>true {ir} (amr: 0.60)													
fps	202,17	206,86	209,99	208,45	207,53	211,50	209,13	209,57	209,14	207,23	207,50	210,09	211,25
precision	0,14	0,25	0,39	0,57	0,74	0,85	0,92	0,95	0,96	0,96	0,96	0,97	0,99
recall	0,56	0,54	0,51	0,48	0,44	0,41	0,37	0,32	0,26	0,21	0,15	0,11	0,07
missrate	0,44	0,46	0,49	0,52	0,56	0,59	0,63	0,68	0,74	0,79	0,85	0,89	0,93
fppi	2,41	1,21	0,58	0,27	0,11	0,05	0,02	0,01	0,01	0,01	0,01	0,00	0,00

## Variation der SVM-Parameter – $h_{scale}$

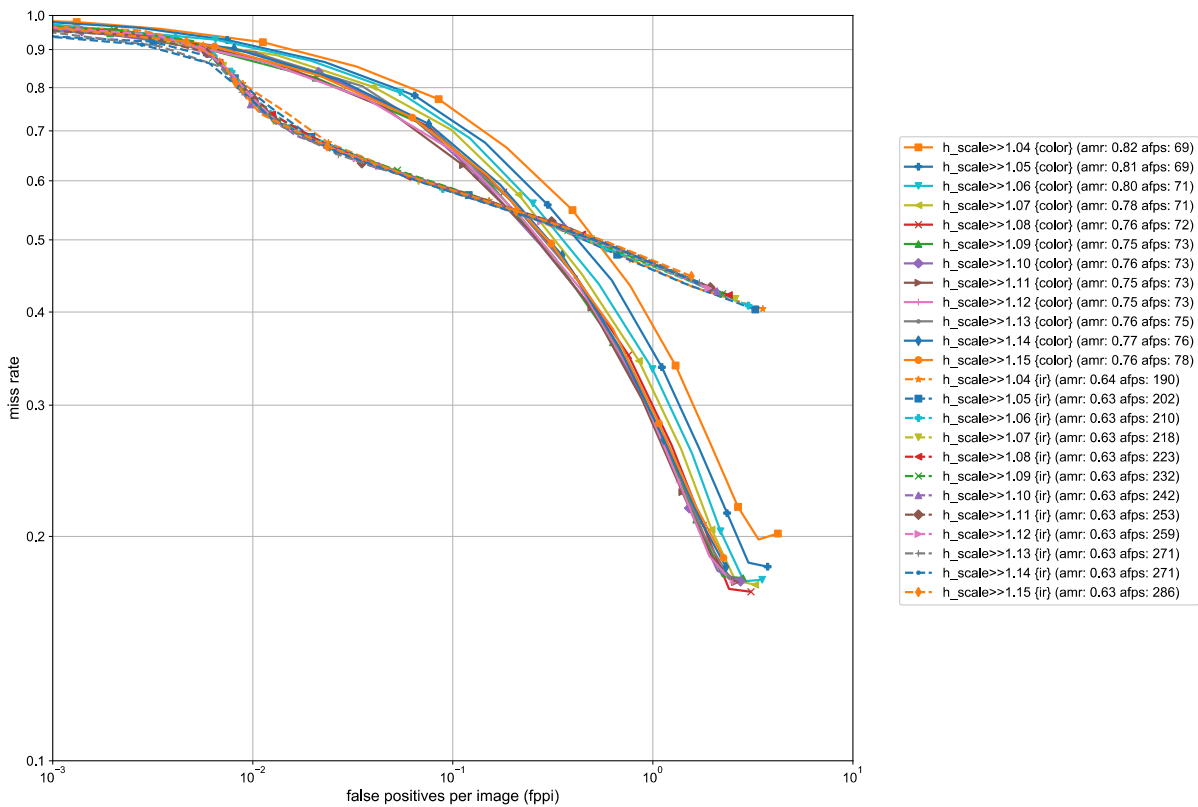


Abbildung A.1-11: Ergebnis der Variation des SVM-Parameters  $h_{scale}$ .

Tabelle A.1-11: Rohdaten zur Variation des SVM-Parameters  $h_{scale}$ .

h_scale>>1.04 {color} (amr: 0.82)															
fps	67,28	69,05	68,09	68,79	70,15	69,39	70,08	69,76	69,59	70,46	70,10	69,73	70,42	69,27	70,59
precision	0,12	0,15	0,18	0,22	0,27	0,35	0,46	0,57	0,67	0,77	0,84	0,90	0,92	0,94	0,98
recall	0,80	0,80	0,78	0,73	0,66	0,57	0,45	0,34	0,23	0,15	0,08	0,04	0,02	0,01	0,01

missrate	0,20	0,20	0,22	0,27	0,34	0,43	0,55	0,66	0,77	0,85	0,92	0,96	0,98	0,99	0,99
fppi	4,21	3,38	2,66	1,93	1,30	0,77	0,40	0,19	0,09	0,03	0,01	0,00	0,00	0,00	0,00
<b>h_scale&gt;&gt;1.04 (ir) (amr: 0.64)</b>															
fps	181,13	187,06	189,37	190,95	192,26	190,95	188,89	193,30	193,05	192,32	192,80	193,06	192,26		
precision	0,11	0,21	0,33	0,51	0,68	0,84	0,91	0,93	0,94	0,94	0,92	0,92	0,98		
recall	0,60	0,57	0,53	0,49	0,44	0,38	0,33	0,26	0,19	0,14	0,09	0,05	0,02		
missrate	0,40	0,43	0,47	0,51	0,56	0,62	0,67	0,74	0,81	0,86	0,91	0,95	0,98		
fppi	3,54	1,61	0,78	0,35	0,15	0,05	0,02	0,01	0,01	0,01	0,01	0,00	0,00		
<b>h_scale&gt;&gt;1.05 (color) (amr: 0.81)</b>															
fps	68,78	70,14	69,45	70,39	68,85	69,97	70,82	70,19	69,81	70,49	69,87	69,28	68,90	69,56	69,81
precision	0,14	0,17	0,20	0,24	0,31	0,40	0,52	0,63	0,72	0,81	0,88	0,91	0,94	0,98	0,97
recall	0,82	0,82	0,79	0,74	0,66	0,56	0,44	0,32	0,22	0,13	0,07	0,04	0,02	0,01	0,00
missrate	0,18	0,18	0,21	0,26	0,34	0,44	0,56	0,68	0,78	0,87	0,93	0,96	0,98	0,99	1,00
fppi	3,75	3,01	2,35	1,70	1,11	0,62	0,30	0,14	0,06	0,02	0,01	0,00	0,00	0,00	0,00
<b>h_scale&gt;&gt;1.05 (ir) (amr: 0.63)</b>															
fps	193,47	199,09	202,23	202,96	204,03	203,66	204,91	204,13	204,15	203,97	202,72	203,70	203,24		
precision	0,12	0,23	0,37	0,54	0,72	0,86	0,92	0,94	0,94	0,93	0,95	0,99	1,00		
recall	0,60	0,56	0,52	0,47	0,43	0,37	0,31	0,24	0,18	0,12	0,08	0,04	0,02		
missrate	0,40	0,44	0,48	0,53	0,57	0,63	0,69	0,76	0,82	0,88	0,92	0,96	0,98		
fppi	3,25	1,43	0,66	0,30	0,12	0,04	0,02	0,01	0,01	0,01	0,00	0,00	0,00		
<b>h_scale&gt;&gt;1.06 (color) (amr: 0.80)</b>															
fps	68,91	70,26	70,57	71,50	71,58	71,36	70,39	72,00	70,62	71,34	70,55	71,96	71,19	71,80	71,27
precision	0,15	0,18	0,21	0,26	0,33	0,44	0,57	0,66	0,74	0,83	0,89	0,95	0,95	0,99	1,00
recall	0,83	0,83	0,80	0,74	0,66	0,56	0,44	0,32	0,21	0,13	0,07	0,04	0,02	0,01	0,00
missrate	0,17	0,17	0,20	0,26	0,34	0,44	0,56	0,68	0,79	0,87	0,93	0,96	0,98	0,99	1,00
fppi	3,52	2,85	2,18	1,57	1,00	0,54	0,25	0,12	0,05	0,02	0,01	0,00	0,00	0,00	0,00
<b>h_scale&gt;&gt;1.06 (ir) (amr: 0.63)</b>															
fps	202,78	206,26	209,72	211,25	210,47	212,74	210,74	211,28	210,47	212,36	212,32	211,87	211,80		
precision	0,13	0,24	0,39	0,59	0,78	0,89	0,93	0,94	0,94	0,93	0,92	0,98	0,98		
recall	0,59	0,55	0,51	0,46	0,41	0,36	0,30	0,22	0,16	0,11	0,07	0,03	0,01		
missrate	0,41	0,45	0,49	0,54	0,59	0,64	0,70	0,78	0,84	0,89	0,93	0,97	0,99		
fppi	3,01	1,29	0,59	0,24	0,09	0,03	0,02	0,01	0,01	0,01	0,00	0,00	0,00		
<b>h_scale&gt;&gt;1.07 (color) (amr: 0.78)</b>															
fps	71,11	71,19	71,84	72,15	72,77	72,16	72,74	72,17	72,30	72,53	70,48	71,29	72,96	72,23	71,17
precision	0,16	0,19	0,23	0,28	0,36	0,48	0,60	0,69	0,79	0,87	0,94	0,97	0,97	0,98	1,00
recall	0,83	0,83	0,80	0,74	0,66	0,55	0,42	0,30	0,20	0,12	0,06	0,03	0,01	0,00	0,00
missrate	0,17	0,17	0,20	0,26	0,34	0,45	0,58	0,70	0,80	0,88	0,94	0,97	0,99	1,00	1,00
fppi	3,25	2,58	1,96	1,39	0,85	0,44	0,21	0,10	0,04	0,01	0,00	0,00	0,00	0,00	0,00
<b>h_scale&gt;&gt;1.07 (ir) (amr: 0.63)</b>															
fps	218,47	220,85	219,26	219,77	219,31	216,46	218,27	220,54	217,72	217,01	216,73	215,84	215,24		
precision	0,14	0,27	0,43	0,64	0,82	0,90	0,94	0,94	0,94	0,93	0,95	0,98	1,00		
recall	0,58	0,54	0,50	0,45	0,40	0,34	0,28	0,21	0,14	0,10	0,05	0,03	0,01		
missrate	0,42	0,46	0,50	0,55	0,60	0,66	0,72	0,79	0,86	0,90	0,95	0,97	0,99		
fppi	2,59	1,10	0,49	0,19	0,07	0,03	0,01	0,01	0,01	0,01	0,00	0,00	0,00		
<b>h_scale&gt;&gt;1.08 (color) (amr: 0.76)</b>															
fps	70,60	71,85	71,98	71,30	72,10	72,34	72,78	72,47	72,71	73,09	72,90	73,24	72,84	72,99	73,37

## A Ergebnisse der Evaluation einzelner Algorithmen

<b>precision</b>	0,17	0,20	0,25	0,31	0,39	0,51	0,64	0,74	0,83	0,89	0,94	0,96	0,97	0,97	1,00
<b>recall</b>	0,83	0,83	0,79	0,73	0,65	0,54	0,42	0,29	0,19	0,11	0,06	0,03	0,01	0,00	0,00
<b>missrate</b>	0,17	0,17	0,21	0,27	0,35	0,46	0,58	0,71	0,81	0,89	0,94	0,97	0,99	1,00	1,00
<b>fppi</b>	3,09	2,41	1,79	1,24	0,76	0,39	0,18	0,08	0,03	0,01	0,00	0,00	0,00	0,00	0,00
<b>h_scale&gt;&gt;1.08 {ir} (amr: 0.63)</b>															
<b>fps</b>	224,55	223,71	218,99	219,18	222,24	223,04	227,25	226,79	220,82	222,62	224,80	221,49	223,79		
<b>precision</b>	0,15	0,28	0,45	0,66	0,83	0,91	0,94	0,94	0,94	0,92	0,94	0,99	1,00		
<b>recall</b>	0,58	0,54	0,49	0,44	0,39	0,33	0,26	0,19	0,14	0,09	0,05	0,02	0,01		
<b>missrate</b>	0,42	0,46	0,51	0,56	0,61	0,67	0,74	0,81	0,86	0,91	0,95	0,98	0,99		
<b>fppi</b>	2,40	1,02	0,44	0,17	0,06	0,02	0,01	0,01	0,01	0,01	0,00	0,00	0,00		
<b>h_scale&gt;&gt;1.09 {color} (amr: 0.75)</b>															
<b>fps</b>	71,40	72,15	73,86	73,70	73,72	73,55	73,55	72,37	73,49	73,45	73,08	74,38	73,89	73,42	70,70
<b>precision</b>	0,18	0,22	0,26	0,33	0,43	0,55	0,66	0,76	0,86	0,92	0,95	0,96	1,00	1,00	1,00
<b>recall</b>	0,82	0,82	0,79	0,73	0,64	0,53	0,41	0,28	0,17	0,10	0,05	0,02	0,01	0,00	0,00
<b>missrate</b>	0,18	0,18	0,21	0,27	0,36	0,47	0,59	0,72	0,83	0,90	0,95	0,98	0,99	1,00	1,00
<b>fppi</b>	2,83	2,23	1,65	1,11	0,63	0,32	0,15	0,07	0,02	0,01	0,00	0,00	0,00	0,00	0,00
<b>h_scale&gt;&gt;1.09 {ir} (amr: 0.63)</b>															
<b>fps</b>	228,19	234,72	231,70	229,70	230,20	232,34	232,99	234,71	231,68	235,12	233,96	230,69	233,35		
<b>precision</b>	0,16	0,30	0,49	0,69	0,84	0,93	0,94	0,94	0,94	0,92	0,94	0,99	1,00		
<b>recall</b>	0,58	0,53	0,49	0,43	0,38	0,32	0,25	0,18	0,12	0,08	0,04	0,02	0,01		
<b>missrate</b>	0,42	0,47	0,51	0,57	0,62	0,68	0,75	0,82	0,88	0,92	0,96	0,98	0,99		
<b>fppi</b>	2,23	0,92	0,38	0,15	0,05	0,02	0,01	0,01	0,01	0,01	0,00	0,00	0,00		
<b>h_scale&gt;&gt;1.10 {color} (amr: 0.76)</b>															
<b>fps</b>	73,10	72,76	72,54	72,82	73,43	71,93	73,06	71,77	72,49	71,91	72,48	74,00	74,11	75,09	74,11
<b>precision</b>	0,18	0,22	0,28	0,34	0,45	0,58	0,69	0,76	0,85	0,93	0,96	0,99	0,99	1,00	1,00
<b>recall</b>	0,83	0,82	0,78	0,71	0,61	0,51	0,39	0,27	0,16	0,09	0,05	0,02	0,01	0,00	0,00
<b>missrate</b>	0,17	0,18	0,22	0,29	0,39	0,49	0,61	0,73	0,84	0,91	0,95	0,98	0,99	1,00	1,00
<b>fppi</b>	2,74	2,11	1,52	1,01	0,56	0,27	0,13	0,06	0,02	0,01	0,00	0,00	0,00	0,00	0,00
<b>h_scale&gt;&gt;1.10 {ir} (amr: 0.63)</b>															
<b>fps</b>	235,30	245,35	240,72	240,14	245,99	244,03	237,95	239,80	246,29	250,80	242,58	242,00	238,69		
<b>precision</b>	0,17	0,31	0,50	0,71	0,87	0,93	0,95	0,94	0,93	0,93	0,97	0,99	1,00		
<b>recall</b>	0,57	0,53	0,48	0,43	0,37	0,31	0,24	0,17	0,11	0,07	0,04	0,01	0,01		
<b>missrate</b>	0,43	0,47	0,52	0,57	0,63	0,69	0,76	0,83	0,89	0,93	0,96	0,99	0,99		
<b>fppi</b>	2,09	0,85	0,36	0,13	0,04	0,02	0,01	0,01	0,01	0,00	0,00	0,00	0,00		
<b>h_scale&gt;&gt;1.11 {color} (amr: 0.75)</b>															
<b>fps</b>	72,99	73,82	73,86	73,32	72,65	73,48	73,76	73,17	72,39	73,80	74,22	75,03	75,27	74,47	73,05
<b>precision</b>	0,19	0,23	0,29	0,37	0,47	0,60	0,71	0,80	0,88	0,93	0,97	0,98	1,00	1,00	1,00
<b>recall</b>	0,83	0,81	0,77	0,69	0,59	0,48	0,37	0,25	0,15	0,08	0,04	0,02	0,01	0,00	0,00
<b>missrate</b>	0,17	0,19	0,23	0,31	0,41	0,52	0,63	0,75	0,85	0,92	0,96	0,98	0,99	1,00	1,00
<b>fppi</b>	2,58	1,98	1,40	0,88	0,49	0,24	0,11	0,05	0,01	0,00	0,00	0,00	0,00	0,00	0,00
<b>h_scale&gt;&gt;1.11 {ir} (amr: 0.63)</b>															
<b>fps</b>	251,25	257,25	255,26	248,56	245,33	247,27	254,08	258,41	259,34	250,72	257,94	255,42	250,98		
<b>precision</b>	0,18	0,32	0,53	0,74	0,89	0,94	0,94	0,94	0,93	0,94	0,99	1,00	1,00		
<b>recall</b>	0,57	0,52	0,47	0,42	0,37	0,30	0,23	0,16	0,11	0,06	0,03	0,01	0,00		
<b>missrate</b>	0,43	0,48	0,53	0,58	0,63	0,70	0,77	0,84	0,89	0,94	0,97	0,99	1,00		
<b>fppi</b>	1,94	0,80	0,31	0,11	0,04	0,02	0,01	0,01	0,01	0,00	0,00	0,00	0,00		

<b>h_scale&gt;&gt;1.12 (color) (amr: 0.75)</b>															
fps	73,12	73,57	74,95	75,94	74,35	73,09	74,80	74,17	70,62	73,43	75,18	72,04	71,96	74,23	74,34
precision	0,20	0,24	0,30	0,38	0,49	0,61	0,72	0,82	0,90	0,94	0,97	0,97	1,00	1,00	1,00
recall	0,83	0,81	0,76	0,68	0,57	0,46	0,34	0,23	0,13	0,07	0,03	0,01	0,00	0,00	0,00
missrate	0,17	0,19	0,24	0,32	0,43	0,54	0,66	0,77	0,87	0,93	0,97	0,99	1,00	1,00	1,00
fppi	2,51	1,92	1,35	0,83	0,45	0,21	0,10	0,04	0,01	0,00	0,00	0,00	0,00	0,00	0,00
<b>h_scale&gt;&gt;1.12 (ir) (amr: 0.63)</b>															
fps	267,61	260,74	262,66	253,61	264,80	256,18	260,03	259,75	256,32	251,86	257,74	264,53	262,71		
precision	0,18	0,34	0,56	0,77	0,89	0,94	0,94	0,94	0,93	0,93	0,97	0,99	1,00		
recall	0,57	0,52	0,47	0,42	0,36	0,29	0,22	0,15	0,10	0,06	0,03	0,01	0,00		
missrate	0,43	0,48	0,53	0,58	0,64	0,71	0,78	0,85	0,90	0,94	0,97	0,99	1,00		
fppi	1,89	0,75	0,28	0,09	0,03	0,01	0,01	0,01	0,01	0,00	0,00	0,00	0,00		
<b>h_scale&gt;&gt;1.13 (color) (amr: 0.76)</b>															
fps	74,28	74,47	72,76	77,11	76,72	74,50	75,65	75,18	76,22	74,16	76,38	76,33	75,25	74,77	74,33
precision	0,20	0,25	0,31	0,39	0,50	0,63	0,73	0,81	0,89	0,95	0,98	0,98	1,00	1,00	0,00
recall	0,82	0,80	0,74	0,66	0,55	0,43	0,31	0,20	0,11	0,06	0,03	0,01	0,00	0,00	0,00
missrate	0,18	0,20	0,26	0,34	0,45	0,57	0,69	0,80	0,89	0,94	0,97	0,99	1,00	1,00	1,00
fppi	2,41	1,81	1,25	0,76	0,41	0,19	0,09	0,04	0,01	0,00	0,00	0,00	0,00	0,00	0,00
<b>h_scale&gt;&gt;1.13 (ir) (amr: 0.63)</b>															
fps	274,90	266,57	255,82	274,27	263,81	265,62	271,25	272,10	279,80	280,30	287,52	273,00	269,78		
precision	0,20	0,36	0,60	0,79	0,91	0,94	0,95	0,94	0,95	0,98	0,99	1,00	1,00		
recall	0,56	0,51	0,46	0,41	0,35	0,28	0,21	0,15	0,09	0,05	0,02	0,01	0,00		
missrate	0,44	0,49	0,54	0,59	0,65	0,72	0,79	0,85	0,91	0,95	0,98	0,99	1,00		
fppi	1,71	0,67	0,23	0,08	0,03	0,01	0,01	0,01	0,00	0,00	0,00	0,00	0,00		
<b>h_scale&gt;&gt;1.14 (color) (amr: 0.77)</b>															
fps	75,41	76,40	76,32	75,32	76,27	77,10	77,33	77,90	77,32	74,91	75,50	75,35	77,00	76,54	76,16
precision	0,21	0,25	0,32	0,41	0,53	0,64	0,74	0,83	0,90	0,95	0,98	0,97	1,00	1,00	1,00
recall	0,82	0,79	0,73	0,64	0,52	0,41	0,28	0,18	0,10	0,05	0,02	0,01	0,00	0,00	0,00
missrate	0,18	0,21	0,27	0,36	0,48	0,59	0,72	0,82	0,90	0,95	0,98	0,99	1,00	1,00	1,00
fppi	2,32	1,72	1,14	0,68	0,35	0,17	0,08	0,03	0,01	0,00	0,00	0,00	0,00	0,00	0,00
<b>h_scale&gt;&gt;1.14 (ir) (amr: 0.63)</b>															
fps	269,50	262,41	277,51	259,21	277,45	268,95	272,97	259,29	271,88	275,12	275,06	279,52	277,48		
precision	0,20	0,38	0,61	0,81	0,91	0,94	0,95	0,94	0,96	0,99	1,00	1,00	1,00		
recall	0,56	0,51	0,46	0,41	0,34	0,27	0,20	0,14	0,09	0,04	0,02	0,01	0,00		
missrate	0,44	0,49	0,54	0,59	0,66	0,73	0,80	0,86	0,91	0,96	0,98	0,99	1,00		
fppi	1,65	0,61	0,22	0,07	0,03	0,01	0,01	0,01	0,00	0,00	0,00	0,00	0,00		
<b>h_scale&gt;&gt;1.15 (color) (amr: 0.76)</b>															
fps	77,50	78,46	78,73	78,88	79,27	77,99	79,13	79,32	78,29	76,58	77,29	78,60	78,73	78,49	77,36
precision	0,21	0,26	0,33	0,42	0,55	0,66	0,76	0,85	0,91	0,94	0,98	1,00	1,00	1,00	0,00
recall	0,81	0,78	0,72	0,62	0,51	0,39	0,27	0,17	0,09	0,05	0,02	0,01	0,00	0,00	0,00
missrate	0,19	0,22	0,28	0,38	0,49	0,61	0,73	0,83	0,91	0,95	0,98	0,99	1,00	1,00	1,00
fppi	2,25	1,66	1,07	0,63	0,31	0,15	0,06	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00
<b>h_scale&gt;&gt;1.15 (ir) (amr: 0.63)</b>															
fps	288,72	290,86	274,43	297,57	290,59	300,60	280,76	284,37	288,67	278,58	288,82	280,74	279,74		
precision	0,21	0,39	0,62	0,82	0,91	0,95	0,94	0,93	0,93	0,96	0,99	1,00	1,00		
recall	0,55	0,50	0,45	0,40	0,33	0,26	0,19	0,13	0,08	0,04	0,02	0,01	0,00		

# A Ergebnisse der Evaluation einzelner Algorithmen

<b>missrate</b>	0,45	0,50	0,55	0,60	0,67	0,74	0,81	0,87	0,92	0,96	0,98	0,99	1,00
<b>fppi</b>	1,56	0,59	0,21	0,07	0,02	0,01	0,01	0,01	0,00	0,00	0,00	0,00	0,00

## Variation der SVM-Parameter – $n_{levels}$

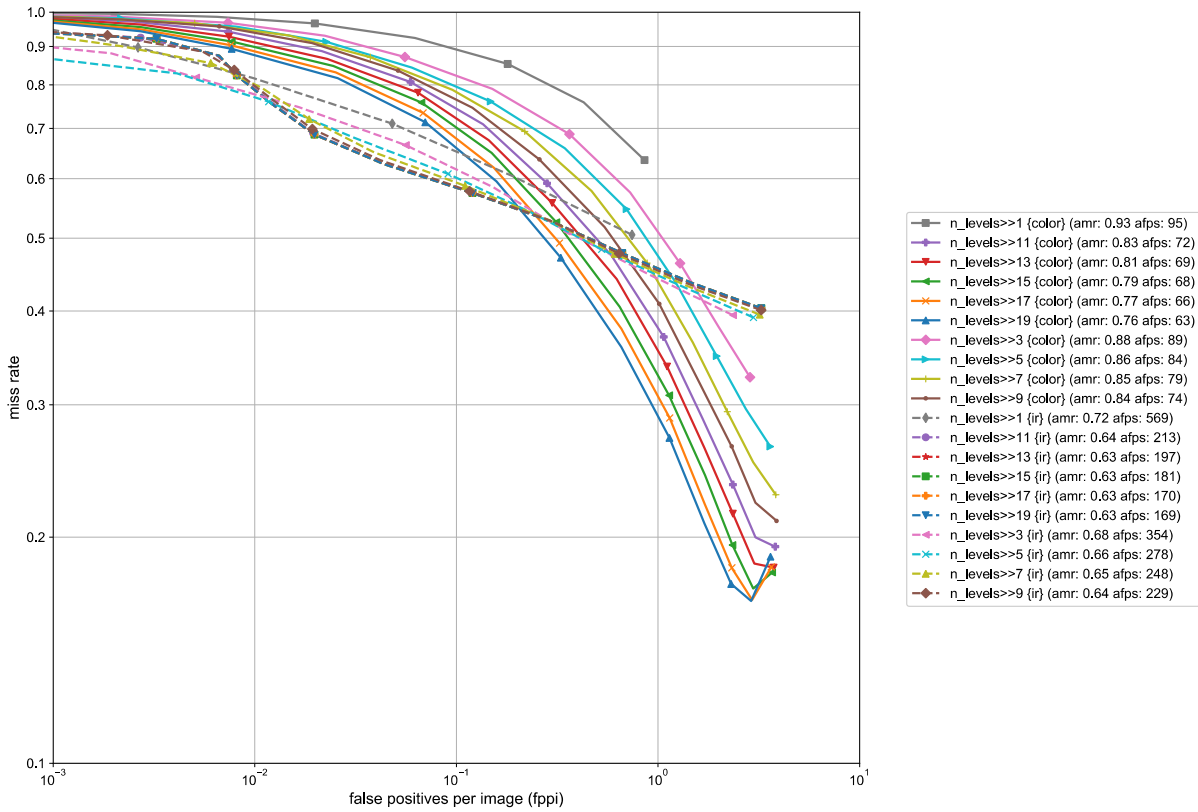


Abbildung A.1-12: Ergebnis der Variation des SVM-Parameters  $n_{levels}$ .

Tabelle A.1-12: Rohdaten zur Variation des SVM-Parameters  $n_{levels}$ .

n_levels>>1 {color} (amr: 0.93)															
<b>fps</b>	95,55	96,07	95,59	95,86	95,84	95,95	95,80	96,23	96,03	96,79	95,84	95,75	95,86	96,03	96,61
<b>precision</b>	0,24	0,30	0,38	0,48	0,56	0,63	0,65	0,72	0,75	0,00	0,00	0,00	0,00	0,00	0,00
<b>recall</b>	0,36	0,24	0,15	0,08	0,03	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
<b>missrate</b>	0,64	0,76	0,85	0,92	0,97	0,99	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
<b>fppi</b>	0,86	0,43	0,18	0,06	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
n_levels>>1 {ir} (amr: 0.72)															
<b>fps</b>	574,38	563,51	568,83	569,81	571,52	563,94	571,93	568,51	577,61	562,78	574,51	571,32	568,34		
<b>precision</b>	0,33	0,60	0,82	0,93	0,97	0,98	0,97	0,94	0,90	0,93	1,00	0,00	0,00		
<b>recall</b>	0,49	0,40	0,29	0,18	0,10	0,05	0,02	0,01	0,00	0,00	0,00	0,00	0,00		
<b>missrate</b>	0,51	0,60	0,71	0,82	0,90	0,95	0,98	0,99	1,00	1,00	1,00	1,00	1,00		
<b>fppi</b>	0,74	0,19	0,05	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00		
n_levels>>11 {color} (amr: 0.83)															
<b>fps</b>	70,09	71,66	72,01	71,68	72,65	71,77	70,25	73,83	72,85	74,03	72,11	72,70	72,93	73,71	71,95
<b>precision</b>	0,14	0,16	0,19	0,24	0,31	0,40	0,52	0,61	0,71	0,80	0,85	0,89	0,93	0,99	1,00
<b>recall</b>	0,81	0,80	0,77	0,71	0,63	0,53	0,41	0,29	0,19	0,11	0,06	0,03	0,01	0,01	0,00

<b>missrate</b>	0,19	0,20	0,23	0,29	0,37	0,47	0,59	0,71	0,81	0,89	0,94	0,97	0,99	0,99	1,00
<b>fppi</b>	3,82	3,04	2,35	1,67	1,07	0,59	0,28	0,14	0,06	0,02	0,01	0,00	0,00	0,00	0,00
<b>n_levels&gt;&gt;11 {ir} (amr: 0.64)</b>															
<b>fps</b>	203,39	209,29	214,92	217,09	218,81	212,65	213,15	213,15	215,25	214,43	216,15	214,93	214,45		
<b>precision</b>	0,12	0,23	0,37	0,54	0,73	0,87	0,92	0,94	0,94	0,93	0,95	0,99	1,00		
<b>recall</b>	0,60	0,56	0,52	0,47	0,42	0,37	0,31	0,24	0,17	0,12	0,08	0,03	0,02		
<b>missrate</b>	0,40	0,44	0,48	0,53	0,58	0,63	0,69	0,76	0,83	0,88	0,92	0,97	0,98		
<b>fppi</b>	3,25	1,43	0,66	0,30	0,12	0,04	0,02	0,01	0,01	0,01	0,00	0,00	0,00		
<b>n_levels&gt;&gt;13 {color} (amr: 0.81)</b>															
<b>fps</b>	67,97	68,35	69,94	70,07	69,62	69,94	68,88	68,95	69,91	69,91	68,68	71,21	68,78	70,48	70,75
<b>precision</b>	0,14	0,17	0,20	0,24	0,31	0,40	0,52	0,63	0,72	0,81	0,88	0,91	0,94	0,98	0,97
<b>recall</b>	0,82	0,82	0,79	0,74	0,66	0,56	0,44	0,32	0,22	0,13	0,07	0,04	0,02	0,01	0,00
<b>missrate</b>	0,18	0,18	0,21	0,26	0,34	0,44	0,56	0,68	0,78	0,87	0,93	0,96	0,98	0,99	1,00
<b>fppi</b>	3,75	3,01	2,35	1,70	1,11	0,62	0,30	0,14	0,06	0,02	0,01	0,00	0,00	0,00	0,00
<b>n_levels&gt;&gt;13 {ir} (amr: 0.63)</b>															
<b>fps</b>	190,38	191,57	199,56	196,49	197,05	201,91	199,93	197,64	196,96	197,71	196,29	197,59	197,96		
<b>precision</b>	0,12	0,23	0,37	0,54	0,72	0,86	0,92	0,94	0,94	0,93	0,95	0,99	1,00		
<b>recall</b>	0,60	0,56	0,52	0,47	0,43	0,37	0,31	0,24	0,18	0,12	0,08	0,04	0,02		
<b>missrate</b>	0,40	0,44	0,48	0,53	0,57	0,63	0,69	0,76	0,82	0,88	0,92	0,96	0,98		
<b>fppi</b>	3,25	1,43	0,66	0,30	0,12	0,04	0,02	0,01	0,01	0,01	0,00	0,00	0,00		
<b>n_levels&gt;&gt;15 {color} (amr: 0.79)</b>															
<b>fps</b>	64,49	66,28	67,51	68,63	67,90	67,55	69,25	67,92	68,84	68,85	68,33	69,08	67,86	69,08	68,51
<b>precision</b>	0,14	0,17	0,20	0,25	0,31	0,41	0,53	0,64	0,73	0,82	0,89	0,92	0,95	0,98	0,98
<b>recall</b>	0,82	0,83	0,80	0,76	0,69	0,59	0,47	0,35	0,24	0,15	0,09	0,04	0,02	0,01	0,00
<b>missrate</b>	0,18	0,17	0,20	0,24	0,31	0,41	0,53	0,65	0,76	0,85	0,91	0,96	0,98	0,99	1,00
<b>fppi</b>	3,68	2,97	2,34	1,71	1,13	0,65	0,31	0,15	0,07	0,02	0,01	0,00	0,00	0,00	0,00
<b>n_levels&gt;&gt;15 {ir} (amr: 0.63)</b>															
<b>fps</b>	173,69	176,85	180,85	181,78	184,50	182,66	184,74	182,61	183,54	181,69	181,94	181,25	181,68		
<b>precision</b>	0,12	0,23	0,37	0,54	0,72	0,86	0,92	0,94	0,94	0,93	0,95	0,99	1,00		
<b>recall</b>	0,60	0,56	0,52	0,48	0,43	0,37	0,31	0,25	0,18	0,13	0,08	0,04	0,02		
<b>missrate</b>	0,40	0,44	0,48	0,52	0,57	0,63	0,69	0,75	0,82	0,87	0,92	0,96	0,98		
<b>fppi</b>	3,25	1,43	0,66	0,30	0,12	0,04	0,02	0,01	0,01	0,01	0,00	0,00	0,00		
<b>n_levels&gt;&gt;17 {color} (amr: 0.77)</b>															
<b>fps</b>	63,96	63,88	65,72	66,13	66,39	66,09	65,46	67,14	67,47	66,95	66,67	66,30	67,12	67,22	66,99
<b>precision</b>	0,14	0,17	0,21	0,25	0,32	0,41	0,54	0,65	0,74	0,83	0,90	0,93	0,96	0,98	0,98
<b>recall</b>	0,82	0,83	0,82	0,78	0,71	0,62	0,51	0,38	0,27	0,17	0,10	0,05	0,03	0,01	0,00
<b>missrate</b>	0,18	0,17	0,18	0,22	0,29	0,38	0,49	0,62	0,73	0,83	0,90	0,95	0,97	0,99	1,00
<b>fppi</b>	3,65	2,93	2,32	1,70	1,14	0,66	0,32	0,15	0,07	0,03	0,01	0,00	0,00	0,00	0,00
<b>n_levels&gt;&gt;17 {ir} (amr: 0.63)</b>															
<b>fps</b>	162,75	165,69	166,96	168,66	168,72	168,51	167,90	169,33	174,67	173,30	176,04	175,67	172,25		
<b>precision</b>	0,12	0,23	0,37	0,54	0,72	0,86	0,92	0,94	0,94	0,93	0,95	0,99	1,00		
<b>recall</b>	0,60	0,56	0,52	0,48	0,43	0,37	0,31	0,25	0,18	0,13	0,08	0,04	0,02		
<b>missrate</b>	0,40	0,44	0,48	0,52	0,57	0,63	0,69	0,75	0,82	0,87	0,92	0,96	0,98		
<b>fppi</b>	3,25	1,43	0,66	0,30	0,12	0,04	0,02	0,01	0,01	0,01	0,00	0,00	0,00		
<b>n_levels&gt;&gt;19 {color} (amr: 0.76)</b>															
<b>fps</b>	62,27	63,19	63,68	65,27	64,04	64,15	64,27	64,59	64,64	63,21	64,58	64,30	62,00	63,65	64,69

## A Ergebnisse der Evaluation einzelner Algorithmen

<b>precision</b>	0,14	0,18	0,21	0,26	0,32	0,42	0,54	0,66	0,75	0,84	0,91	0,94	0,96	0,98	0,98
<b>recall</b>	0,81	0,84	0,83	0,79	0,73	0,64	0,53	0,40	0,29	0,18	0,11	0,06	0,03	0,01	0,01
<b>missrate</b>	0,19	0,16	0,17	0,21	0,27	0,36	0,47	0,60	0,71	0,82	0,89	0,94	0,97	0,99	0,99
<b>fppi</b>	3,61	2,89	2,31	1,70	1,14	0,66	0,33	0,16	0,07	0,03	0,01	0,00	0,00	0,00	0,00
<b>n_levels&gt;&gt;19 {ir} (amr: 0.63)</b>															
<b>fps</b>	162,06	165,92	167,74	169,10	170,03	169,44	166,71	168,56	168,63	168,89	170,68	175,46	176,03		
<b>precision</b>	0,12	0,23	0,37	0,54	0,72	0,86	0,92	0,94	0,94	0,93	0,95	0,99	1,00		
<b>recall</b>	0,60	0,56	0,52	0,48	0,43	0,37	0,31	0,25	0,18	0,13	0,08	0,04	0,02		
<b>missrate</b>	0,40	0,44	0,48	0,52	0,57	0,63	0,69	0,75	0,82	0,87	0,92	0,96	0,98		
<b>fppi</b>	3,25	1,43	0,66	0,30	0,12	0,04	0,02	0,01	0,01	0,01	0,00	0,00	0,00		
<b>n_levels&gt;&gt;3 {color} (amr: 0.88)</b>															
<b>fps</b>	88,57	88,88	89,12	89,18	89,28	89,49	89,29	89,46	89,40	89,38	89,67	89,45	89,36	89,54	89,57
<b>precision</b>	0,15	0,19	0,24	0,30	0,39	0,51	0,63	0,70	0,76	0,83	0,94	1,00	1,00	0,00	0,00
<b>recall</b>	0,67	0,62	0,54	0,42	0,31	0,21	0,13	0,07	0,03	0,01	0,00	0,00	0,00	0,00	0,00
<b>missrate</b>	0,33	0,38	0,46	0,58	0,69	0,79	0,87	0,93	0,97	0,99	1,00	1,00	1,00	1,00	1,00
<b>fppi</b>	2,86	2,00	1,29	0,73	0,36	0,15	0,06	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00
<b>n_levels&gt;&gt;3 {ir} (amr: 0.68)</b>															
<b>fps</b>	345,96	353,09	343,47	348,23	334,45	343,79	357,69	369,93	353,46	375,92	355,91	372,66	348,60		
<b>precision</b>	0,16	0,32	0,51	0,67	0,82	0,92	0,96	0,98	0,99	1,00	1,00	0,98	0,97		
<b>recall</b>	0,61	0,55	0,49	0,42	0,33	0,25	0,18	0,12	0,07	0,04	0,02	0,01	0,00		
<b>missrate</b>	0,39	0,45	0,51	0,58	0,67	0,75	0,82	0,88	0,93	0,96	0,98	0,99	1,00		
<b>fppi</b>	2,36	0,89	0,35	0,15	0,06	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00		
<b>n_levels&gt;&gt;5 {color} (amr: 0.86)</b>															
<b>fps</b>	83,45	83,22	83,77	84,09	84,19	84,54	84,17	84,02	83,99	84,34	84,11	84,37	84,70	84,13	84,39
<b>precision</b>	0,13	0,16	0,20	0,26	0,33	0,42	0,55	0,66	0,74	0,80	0,87	0,94	1,00	1,00	0,00
<b>recall</b>	0,74	0,70	0,65	0,56	0,45	0,34	0,24	0,16	0,09	0,04	0,02	0,01	0,00	0,00	0,00
<b>missrate</b>	0,26	0,30	0,35	0,44	0,55	0,66	0,76	0,84	0,91	0,96	0,98	0,99	1,00	1,00	1,00
<b>fppi</b>	3,61	2,73	1,96	1,22	0,70	0,35	0,15	0,06	0,02	0,01	0,00	0,00	0,00	0,00	0,00
<b>n_levels&gt;&gt;5 {ir} (amr: 0.66)</b>															
<b>fps</b>	264,79	291,89	284,74	284,38	286,90	273,96	265,87	277,04	275,16	269,69	279,86	291,40	277,31		
<b>precision</b>	0,13	0,25	0,42	0,60	0,76	0,89	0,94	0,97	0,99	1,00	1,00	1,00	1,00		
<b>recall</b>	0,61	0,57	0,52	0,46	0,39	0,32	0,24	0,17	0,12	0,08	0,05	0,02	0,01		
<b>missrate</b>	0,39	0,43	0,48	0,54	0,61	0,68	0,76	0,83	0,88	0,92	0,95	0,98	0,99		
<b>fppi</b>	2,97	1,23	0,52	0,23	0,09	0,03	0,01	0,00	0,00	0,00	0,00	0,00	0,00		
<b>n_levels&gt;&gt;7 {color} (amr: 0.85)</b>															
<b>fps</b>	76,63	78,36	79,04	79,06	78,87	79,94	79,60	79,53	79,54	79,27	79,70	79,50	79,68	79,34	79,61
<b>precision</b>	0,13	0,16	0,19	0,24	0,31	0,40	0,51	0,62	0,72	0,78	0,83	0,90	0,95	1,00	1,00
<b>recall</b>	0,77	0,75	0,71	0,64	0,54	0,42	0,31	0,21	0,13	0,07	0,03	0,02	0,01	0,00	0,00
<b>missrate</b>	0,23	0,25	0,29	0,36	0,46	0,58	0,69	0,79	0,87	0,93	0,97	0,98	0,99	1,00	1,00
<b>fppi</b>	3,84	2,97	2,21	1,49	0,88	0,47	0,22	0,10	0,04	0,01	0,01	0,00	0,00	0,00	0,00
<b>n_levels&gt;&gt;7 {ir} (amr: 0.65)</b>															
<b>fps</b>	235,38	238,85	251,31	253,95	255,03	257,60	255,15	252,38	254,23	247,86	242,76	243,36	248,95		
<b>precision</b>	0,12	0,23	0,39	0,55	0,74	0,87	0,92	0,93	0,95	0,97	0,98	1,00	1,00		
<b>recall</b>	0,60	0,57	0,52	0,47	0,41	0,35	0,28	0,21	0,14	0,10	0,06	0,03	0,01		
<b>missrate</b>	0,40	0,43	0,48	0,53	0,59	0,65	0,72	0,79	0,86	0,90	0,94	0,97	0,99		
<b>fppi</b>	3,20	1,37	0,61	0,28	0,11	0,04	0,02	0,01	0,01	0,00	0,00	0,00	0,00		

n_levels>>9 {color} (amr: 0.84)															
fps	72,83	73,89	74,73	74,35	75,55	74,07	77,02	73,94	75,86	75,57	75,00	74,57	74,30	75,90	75,64
precision	0,13	0,16	0,19	0,24	0,30	0,40	0,51	0,61	0,70	0,78	0,83	0,88	0,93	0,98	1,00
recall	0,79	0,78	0,74	0,68	0,59	0,48	0,36	0,25	0,16	0,09	0,04	0,02	0,01	0,01	0,00
missrate	0,21	0,22	0,26	0,32	0,41	0,52	0,64	0,75	0,84	0,91	0,96	0,98	0,99	0,99	1,00
fppi	3,87	3,04	2,32	1,61	1,01	0,54	0,26	0,12	0,05	0,02	0,01	0,00	0,00	0,00	0,00

n_levels>>9 {ir} (amr: 0.64)													
fps	220,73	229,38	228,32	232,04	228,73	232,63	229,60	230,98	232,44	232,38	226,47	229,24	232,04
precision	0,12	0,23	0,38	0,54	0,73	0,87	0,92	0,94	0,94	0,94	0,96	0,99	1,00
recall	0,60	0,57	0,52	0,47	0,42	0,36	0,30	0,23	0,16	0,11	0,07	0,03	0,02
missrate	0,40	0,43	0,48	0,53	0,58	0,64	0,70	0,77	0,84	0,89	0,93	0,97	0,98
fppi	3,25	1,42	0,64	0,30	0,12	0,04	0,02	0,01	0,01	0,01	0,00	0,00	0,00

### Variation der SVM-Parameter – Group Threshold

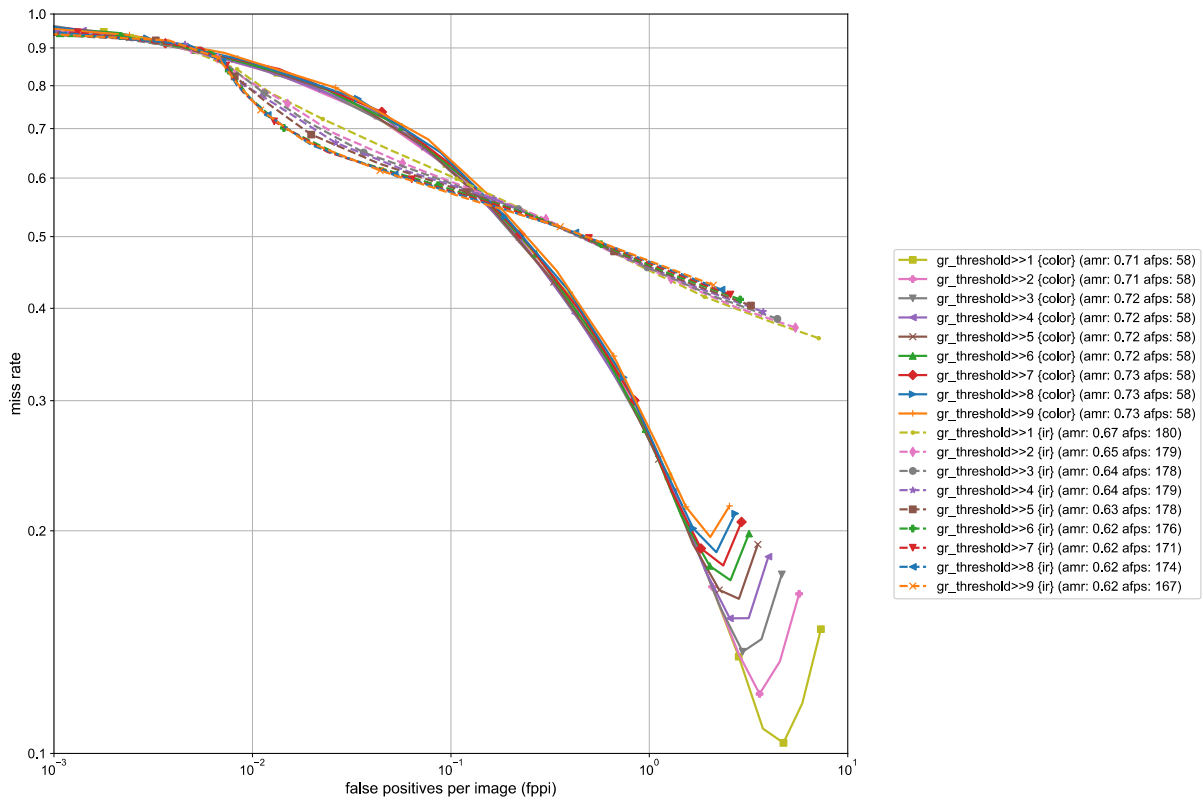


Abbildung A.1-13: Ergebnis der Variation des SVM-Parameters „group threshold“.

Tabelle A.1-13: Rohdaten zur Variation des SVM-Parameters „group threshold“.

gr_threshold>>1 {color} (amr: 0.71)															
fps	56,27	57,47	57,16	57,29	59,01	58,81	58,14	58,60	58,86	59,46	59,36	59,06	59,21	58,93	58,45
precision	0,08	0,10	0,12	0,15	0,19	0,24	0,31	0,41	0,52	0,64	0,74	0,84	0,90	0,94	0,96
recall	0,85	0,88	0,90	0,89	0,86	0,83	0,76	0,68	0,58	0,48	0,37	0,26	0,17	0,11	0,05
missrate	0,15	0,12	0,10	0,11	0,14	0,17	0,24	0,32	0,42	0,52	0,63	0,74	0,83	0,89	0,95
fppi	7,33	5,90	4,74	3,73	2,82	1,95	1,25	0,73	0,40	0,20	0,09	0,04	0,01	0,01	0,00



## A Ergebnisse der Evaluation einzelner Algorithmen

gr_threshold>>1 {ir} (amr: 0.67)															
fps	172,73	178,42	179,87	180,96	180,64	181,20	181,60	182,03	182,02	181,89	180,82	181,49	180,19		
precision	0,06	0,10	0,19	0,30	0,45	0,60	0,74	0,84	0,90	0,93	0,93	0,94	0,95		
recall	0,64	0,62	0,59	0,55	0,50	0,45	0,40	0,34	0,28	0,22	0,16	0,11	0,07		
missrate	0,36	0,38	0,41	0,45	0,50	0,55	0,60	0,66	0,72	0,78	0,84	0,89	0,93		
fppi	7,15	4,00	1,90	0,93	0,45	0,22	0,11	0,05	0,02	0,01	0,01	0,00	0,00		
gr_threshold>>2 {color} (amr: 0.71)															
fps	55,37	56,49	58,16	57,96	58,26	58,61	58,95	58,48	59,08	58,71	58,37	58,55	59,01	59,35	58,20
precision	0,10	0,12	0,15	0,18	0,23	0,30	0,39	0,50	0,63	0,73	0,83	0,90	0,94	0,95	0,97
recall	0,84	0,87	0,88	0,86	0,83	0,78	0,70	0,61	0,50	0,39	0,27	0,17	0,11	0,06	0,03
missrate	0,16	0,13	0,12	0,14	0,17	0,22	0,30	0,39	0,50	0,61	0,73	0,83	0,89	0,94	0,97
fppi	5,69	4,55	3,60	2,83	2,07	1,37	0,82	0,44	0,22	0,11	0,04	0,01	0,00	0,00	0,00
gr_threshold>>2 {ir} (amr: 0.65)															
fps	172,52	176,80	179,33	180,27	180,56	181,18	180,95	181,01	180,61	181,60	181,87	179,10	179,66		
precision	0,08	0,14	0,24	0,38	0,54	0,69	0,83	0,90	0,92	0,94	0,93	0,95	0,98		
recall	0,62	0,60	0,56	0,52	0,47	0,43	0,37	0,31	0,24	0,18	0,13	0,08	0,05		
missrate	0,38	0,40	0,44	0,48	0,53	0,57	0,63	0,69	0,76	0,82	0,87	0,92	0,95		
fppi	5,45	2,73	1,29	0,62	0,30	0,14	0,06	0,03	0,02	0,01	0,01	0,00	0,00		
gr_threshold>>3 {color} (amr: 0.72)															
fps	56,01	58,02	58,06	59,27	58,52	59,11	59,29	58,82	57,73	58,65	58,58	58,04	58,67	58,20	58,53
precision	0,12	0,15	0,18	0,22	0,27	0,35	0,46	0,58	0,69	0,79	0,88	0,93	0,96	0,96	0,97
recall	0,83	0,86	0,86	0,84	0,80	0,74	0,65	0,55	0,44	0,32	0,21	0,12	0,07	0,04	0,02
missrate	0,17	0,14	0,14	0,16	0,20	0,26	0,35	0,45	0,56	0,68	0,79	0,88	0,93	0,96	0,98
fppi	4,65	3,68	2,95	2,28	1,61	1,03	0,57	0,29	0,14	0,06	0,02	0,01	0,00	0,00	0,00
gr_threshold>>3 {ir} (amr: 0.64)															
fps	168,31	174,29	176,97	177,03	175,59	178,20	179,62	181,17	181,40	181,51	180,70	181,36	181,66		
precision	0,09	0,17	0,29	0,44	0,61	0,77	0,88	0,92	0,93	0,94	0,94	0,97	1,00		
recall	0,61	0,58	0,55	0,50	0,46	0,40	0,35	0,28	0,22	0,16	0,11	0,06	0,03		
missrate	0,39	0,42	0,45	0,50	0,54	0,60	0,65	0,72	0,78	0,84	0,89	0,94	0,97		
fppi	4,42	2,08	0,98	0,47	0,22	0,09	0,04	0,02	0,01	0,01	0,01	0,00	0,00		
gr_threshold>>4 {color} (amr: 0.72)															
fps	56,12	57,30	58,34	58,17	59,13	58,70	58,89	58,78	58,31	58,86	58,30	58,58	58,83	58,12	58,58
precision	0,13	0,17	0,20	0,24	0,30	0,39	0,52	0,63	0,74	0,82	0,91	0,94	0,96	0,97	0,97
recall	0,82	0,85	0,85	0,82	0,77	0,70	0,61	0,50	0,38	0,26	0,17	0,09	0,05	0,02	0,01
missrate	0,18	0,15	0,15	0,18	0,23	0,30	0,39	0,50	0,62	0,74	0,83	0,91	0,95	0,98	0,99
fppi	3,98	3,17	2,55	1,93	1,32	0,80	0,42	0,21	0,10	0,04	0,01	0,00	0,00	0,00	0,00
gr_threshold>>4 {ir} (amr: 0.64)															
fps	171,65	176,99	179,04	178,10	178,09	180,18	179,85	179,18	179,73	180,66	181,84	181,26	181,71		
precision	0,11	0,20	0,33	0,49	0,66	0,83	0,90	0,93	0,94	0,94	0,94	0,98	1,00		
recall	0,60	0,57	0,53	0,49	0,44	0,39	0,33	0,27	0,20	0,14	0,09	0,05	0,03		
missrate	0,40	0,43	0,47	0,51	0,56	0,61	0,67	0,73	0,80	0,86	0,91	0,95	0,97		
fppi	3,75	1,69	0,80	0,37	0,16	0,06	0,03	0,02	0,01	0,01	0,00	0,00	0,00		
gr_threshold>>5 {color} (amr: 0.72)															
fps	55,45	56,59	57,94	58,69	58,25	59,20	59,08	59,48	58,72	59,00	58,81	58,75	58,32	58,88	58,22
precision	0,15	0,18	0,22	0,27	0,33	0,43	0,56	0,68	0,77	0,86	0,92	0,95	0,97	0,99	0,98
recall	0,81	0,84	0,83	0,81	0,75	0,67	0,57	0,45	0,34	0,22	0,13	0,07	0,04	0,02	0,01

## A.1 Personenerkennung

<b>missrate</b>	0,19	0,16	0,17	0,19	0,25	0,33	0,43	0,55	0,66	0,78	0,87	0,93	0,96	0,98	0,99
<b>fppi</b>	3,52	2,83	2,25	1,66	1,11	0,65	0,33	0,16	0,07	0,03	0,01	0,00	0,00	0,00	0,00
<b>gr_threshold&gt;&gt;5 {ir} (amr: 0.63)</b>															
<b>fps</b>	173,44	175,42	176,51	178,57	178,47	178,94	179,52	180,70	180,57	180,26	179,91	180,01	180,44		
<b>precision</b>	0,12	0,23	0,37	0,54	0,72	0,86	0,92	0,94	0,94	0,93	0,95	0,99	1,00		
<b>recall</b>	0,60	0,56	0,52	0,48	0,43	0,37	0,31	0,25	0,18	0,13	0,08	0,04	0,02		
<b>missrate</b>	0,40	0,44	0,48	0,52	0,57	0,63	0,69	0,75	0,82	0,87	0,92	0,96	0,98		
<b>fppi</b>	3,25	1,43	0,66	0,30	0,12	0,04	0,02	0,01	0,01	0,01	0,00	0,00	0,00		
<b>gr_threshold&gt;&gt;6 {color} (amr: 0.72)</b>															
<b>fps</b>	55,87	56,98	58,05	57,77	57,83	59,26	59,17	58,87	58,83	58,81	58,41	59,11	58,85	59,12	59,15
<b>precision</b>	0,16	0,19	0,23	0,29	0,36	0,47	0,59	0,70	0,80	0,88	0,94	0,95	0,98	0,99	1,00
<b>recall</b>	0,80	0,83	0,82	0,79	0,73	0,64	0,53	0,41	0,30	0,18	0,11	0,06	0,03	0,01	0,00
<b>missrate</b>	0,20	0,17	0,18	0,21	0,27	0,36	0,47	0,59	0,70	0,82	0,89	0,94	0,97	0,99	1,00
<b>fppi</b>	3,18	2,57	2,02	1,46	0,96	0,54	0,27	0,13	0,06	0,02	0,01	0,00	0,00	0,00	0,00
<b>gr_threshold&gt;&gt;6 {ir} (amr: 0.62)</b>															
<b>fps</b>	170,82	176,59	177,91	176,75	177,16	178,89	179,01	178,45	179,61	179,21	175,15	171,67	178,21		
<b>precision</b>	0,13	0,25	0,40	0,59	0,78	0,90	0,94	0,95	0,94	0,93	0,96	1,00	1,00		
<b>recall</b>	0,59	0,55	0,51	0,46	0,41	0,36	0,30	0,23	0,16	0,11	0,07	0,03	0,01		
<b>missrate</b>	0,41	0,45	0,49	0,54	0,59	0,64	0,70	0,77	0,84	0,89	0,93	0,97	0,99		
<b>fppi</b>	2,87	1,25	0,57	0,24	0,09	0,03	0,01	0,01	0,01	0,01	0,00	0,00	0,00		
<b>gr_threshold&gt;&gt;7 {color} (amr: 0.73)</b>															
<b>fps</b>	55,98	57,48	58,09	58,81	59,05	59,13	59,11	59,24	59,16	58,82	58,93	58,65	59,01	58,89	59,01
<b>precision</b>	0,17	0,21	0,25	0,31	0,38	0,50	0,62	0,73	0,81	0,90	0,95	0,96	0,98	0,98	1,00
<b>recall</b>	0,79	0,82	0,81	0,77	0,70	0,60	0,50	0,38	0,26	0,16	0,09	0,05	0,02	0,01	0,00
<b>missrate</b>	0,21	0,18	0,19	0,23	0,30	0,40	0,50	0,62	0,74	0,84	0,91	0,95	0,98	0,99	1,00
<b>fppi</b>	2,92	2,36	1,82	1,30	0,84	0,45	0,22	0,10	0,04	0,01	0,00	0,00	0,00	0,00	0,00
<b>gr_threshold&gt;&gt;7 {ir} (amr: 0.62)</b>															
<b>fps</b>	169,66	171,86	164,99	170,31	174,44	170,65	177,77	167,91	170,32	173,85	172,67	173,17	177,49		
<b>precision</b>	0,14	0,27	0,43	0,64	0,82	0,92	0,94	0,95	0,94	0,93	0,97	1,00	1,00		
<b>recall</b>	0,58	0,54	0,50	0,45	0,40	0,35	0,28	0,21	0,15	0,10	0,06	0,03	0,01		
<b>missrate</b>	0,42	0,46	0,50	0,55	0,60	0,65	0,72	0,79	0,85	0,90	0,94	0,97	0,99		
<b>fppi</b>	2,56	1,10	0,50	0,19	0,06	0,02	0,01	0,01	0,01	0,01	0,00	0,00	0,00		
<b>gr_threshold&gt;&gt;8 {color} (amr: 0.73)</b>															
<b>fps</b>	56,06	56,97	58,43	58,16	59,02	59,24	58,98	59,14	59,02	58,73	58,63	58,82	58,02	58,69	58,91
<b>precision</b>	0,18	0,22	0,26	0,32	0,40	0,52	0,65	0,75	0,84	0,91	0,95	0,97	0,99	1,00	1,00
<b>recall</b>	0,79	0,81	0,80	0,75	0,68	0,58	0,47	0,35	0,23	0,14	0,07	0,04	0,01	0,00	0,00
<b>missrate</b>	0,21	0,19	0,20	0,25	0,32	0,42	0,53	0,65	0,77	0,86	0,93	0,96	0,99	1,00	1,00
<b>fppi</b>	2,71	2,18	1,66	1,17	0,74	0,39	0,19	0,09	0,03	0,01	0,00	0,00	0,00	0,00	0,00
<b>gr_threshold&gt;&gt;8 {ir} (amr: 0.62)</b>															
<b>fps</b>	168,93	173,23	175,32	171,94	177,63	177,83	178,91	173,32	178,44	172,99	178,15	170,54	173,90		
<b>precision</b>	0,16	0,29	0,46	0,69	0,85	0,93	0,94	0,95	0,94	0,93	0,98	1,00	1,00		
<b>recall</b>	0,58	0,54	0,49	0,45	0,39	0,34	0,27	0,20	0,14	0,09	0,05	0,02	0,01		
<b>missrate</b>	0,42	0,46	0,51	0,55	0,61	0,66	0,73	0,80	0,86	0,91	0,95	0,98	0,99		
<b>fppi</b>	2,31	0,99	0,42	0,15	0,05	0,02	0,01	0,01	0,01	0,00	0,00	0,00	0,00		
<b>gr_threshold&gt;&gt;9 {color} (amr: 0.73)</b>															
<b>fps</b>	55,68	56,14	58,02	58,27	57,94	58,70	58,46	58,94	58,79	58,38	58,35	58,72	57,62	58,39	58,58

# A Ergebnisse der Evaluation einzelner Algorithmen

<b>precision</b>	0,19	0,23	0,28	0,34	0,42	0,54	0,67	0,76	0,85	0,92	0,95	0,98	0,99	1,00	1,00
<b>recall</b>	0,78	0,80	0,78	0,74	0,66	0,55	0,44	0,32	0,21	0,11	0,06	0,03	0,01	0,00	0,00
<b>missrate</b>	0,22	0,20	0,22	0,26	0,34	0,45	0,56	0,68	0,79	0,89	0,94	0,97	0,99	1,00	1,00
<b>fppi</b>	2,53	2,03	1,53	1,08	0,66	0,34	0,16	0,08	0,03	0,01	0,00	0,00	0,00	0,00	0,00
<b>gr_threshold&gt;&gt;9 {ir} (amr: 0.62)</b>															
<b>fps</b>	164,27	163,89	166,64	170,13	170,13	164,80	168,50	167,98	166,80	169,21	166,23	168,73	172,52		
<b>precision</b>	0,17	0,31	0,50	0,73	0,87	0,93	0,95	0,94	0,93	0,94	0,99	1,00	1,00		
<b>recall</b>	0,57	0,53	0,48	0,44	0,39	0,32	0,26	0,19	0,13	0,08	0,04	0,02	0,01		
<b>missrate</b>	0,43	0,47	0,52	0,56	0,61	0,68	0,74	0,81	0,87	0,92	0,96	0,98	0,99		
<b>fppi</b>	2,10	0,89	0,36	0,12	0,04	0,02	0,01	0,01	0,01	0,00	0,00	0,00	0,00		

## Variation der SVM-Parameter – Anwendung der besten Parameter

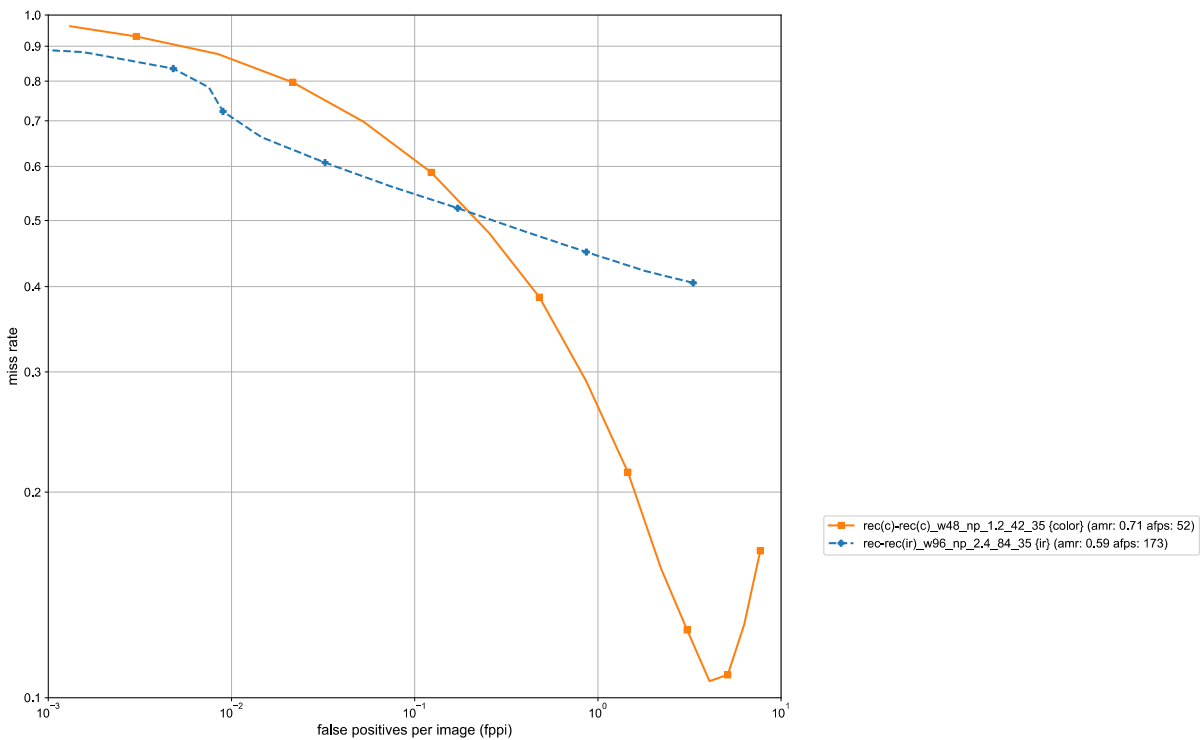


Abbildung A.1-14: Ergebnis der Anwendung der optimierten SVM-Konfiguration.

Tabelle A.1-14: Rohdaten zur Anwendung der optimierten SVM-Konfiguration.

<b>rec(c)-rec(c)_w48_EL_np_1.2_42_35_auto.yml {color} (amr: 0.71)</b>																
<b>fps</b>	50,61	52,03	52,51	52,75	53,21	52,71	52,84	53,07	52,78	53,24	52,37	53,55	52,65	53,21	53,10	52,97
<b>precision</b>	0,07	0,09	0,11	0,14	0,17	0,22	0,29	0,38	0,49	0,60	0,71	0,81	0,87	0,92	0,94	0,95
<b>recall</b>	0,84	0,87	0,89	0,89	0,87	0,85	0,79	0,71	0,61	0,52	0,41	0,30	0,20	0,12	0,07	0,04
<b>missrate</b>	0,16	0,13	0,11	0,11	0,13	0,15	0,21	0,29	0,39	0,48	0,59	0,70	0,80	0,88	0,93	0,96
<b>fppi</b>	7,71	6,30	5,11	4,07	3,07	2,21	1,46	0,86	0,48	0,25	0,12	0,05	0,02	0,01	0,00	0,00
<b>rec-rec(ir)_w96_EL_np_2.4_84_35_auto.yml {ir} (amr: 0.59)</b>																
<b>fps</b>	161,47	170,27	173,47	174,45	172,50	172,40	176,14	175,63	176,09	176,01	173,48	173,95	175,41			
<b>precision</b>	0,12	0,19	0,32	0,50	0,67	0,81	0,90	0,95	0,96	0,96	0,96	0,98	1,00			
<b>recall</b>	0,59	0,58	0,55	0,52	0,48	0,44	0,39	0,34	0,28	0,22	0,17	0,12	0,08			

missrate	0,41	0,42	0,45	0,48	0,52	0,56	0,61	0,66	0,72	0,78	0,83	0,88	0,92
fppi	3,31	1,80	0,87	0,39	0,17	0,07	0,03	0,01	0,01	0,01	0,00	0,00	0,00

Variation der detektierten Personen – Verdeckungsgrad (Intensitätsbild)

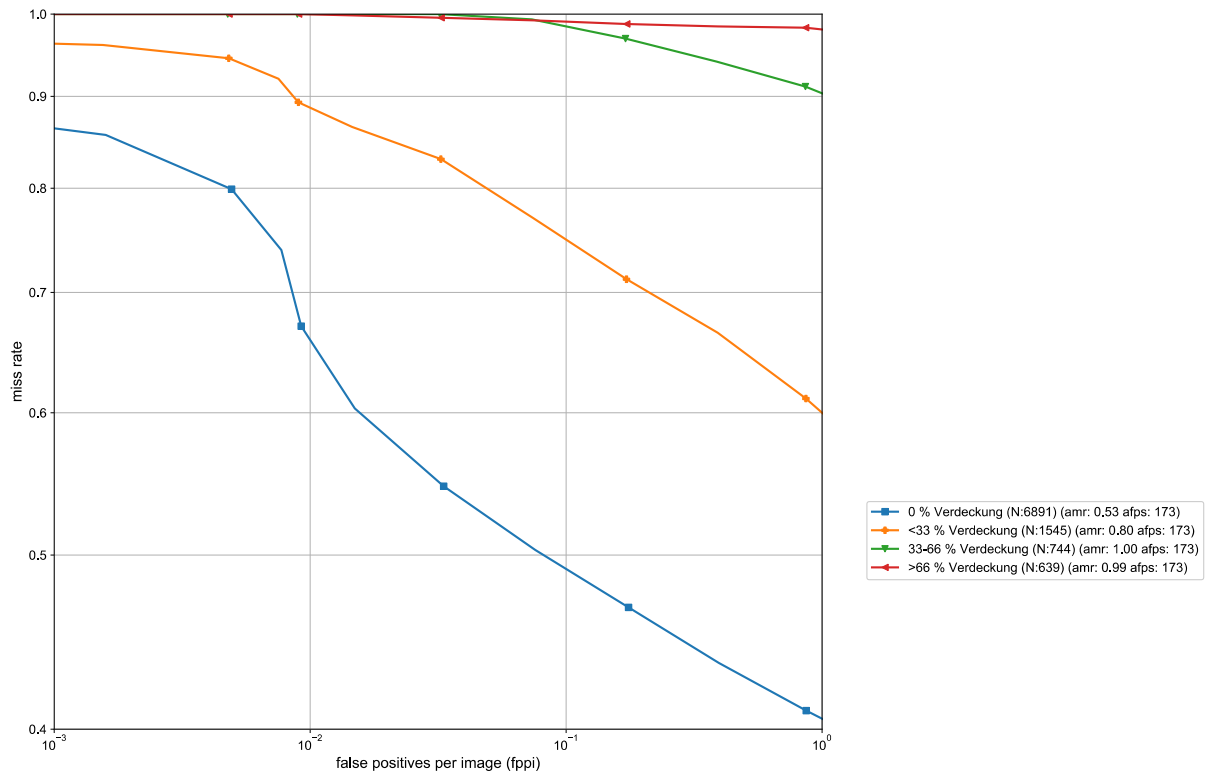


Abbildung A.1-15: Erkennungsleistung der SVM in Abhängigkeit der Verdeckung von Personen am Beispiel des Intensitätsbilds.

Tabelle A.1-15: Rohdaten zur Erkennungsleistung der SVM in Abhängigkeit der Verdeckung von Personen am Beispiel des Intensitätsbilds.

0 % Verdeckung (N:6891) (amr: 0.53)													
fps	161,87	170,46	173,99	174,53	172,64	172,49	176,32	175,80	176,25	176,11	173,48	174,10	175,69
precision	0,09	0,16	0,27	0,44	0,62	0,78	0,88	0,94	0,95	0,95	0,96	0,98	1,00
recall	0,63	0,61	0,59	0,56	0,53	0,50	0,45	0,40	0,33	0,26	0,20	0,14	0,10
missrate	0,37	0,39	0,41	0,44	0,47	0,50	0,55	0,60	0,67	0,74	0,80	0,86	0,90
fppi	3,30	1,79	0,87	0,40	0,18	0,08	0,03	0,01	0,01	0,01	0,00	0,00	0,00
<33 % Verdeckung (N:1545) (amr: 0.80)													
fps	161,47	170,27	173,47	174,45	172,50	172,40	176,14	175,63	176,09	176,01	173,48	173,95	175,41
precision	0,02	0,03	0,05	0,09	0,17	0,27	0,39	0,53	0,59	0,56	0,58	0,75	0,98
recall	0,49	0,44	0,39	0,34	0,29	0,23	0,17	0,13	0,11	0,08	0,06	0,04	0,03
missrate	0,51	0,56	0,61	0,66	0,71	0,77	0,83	0,87	0,89	0,92	0,94	0,96	0,97
fppi	3,31	1,80	0,87	0,39	0,17	0,07	0,03	0,01	0,01	0,01	0,00	0,00	0,00
33-66 % Verdeckung (N:744) (amr: 1.00)													
fps	161,47	170,24	173,44	174,47	172,53	172,34	176,19	175,68	176,10	176,00	173,55	174,02	175,44
precision	0,00	0,00	0,01	0,01	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00
recall	0,17	0,13	0,09	0,06	0,03	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00
missrate	0,83	0,87	0,91	0,94	0,97	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00
fppi	3,31	1,80	0,86	0,39	0,17	0,07	0,03	0,01	0,01	0,01	0,00	0,00	0,00

# A Ergebnisse der Evaluation einzelner Algorithmen

>66 % Verdeckung (N:639) (amr: 0.99)														
fps	161,85	170,50	173,90	174,57	172,56	172,38	176,29	175,73	176,08	175,93	173,51	173,97	175,48	
precision	0,00	0,00	0,00	0,00	0,00	0,01	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00
recall	0,05	0,03	0,02	0,02	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
missrate	0,95	0,97	0,98	0,98	0,99	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
fppi	3,31	1,78	0,86	0,39	0,17	0,07	0,03	0,01	0,01	0,01	0,01	0,00	0,00	0,00

## Variation der detektierten Personen – Posen (Intensitätsbild)

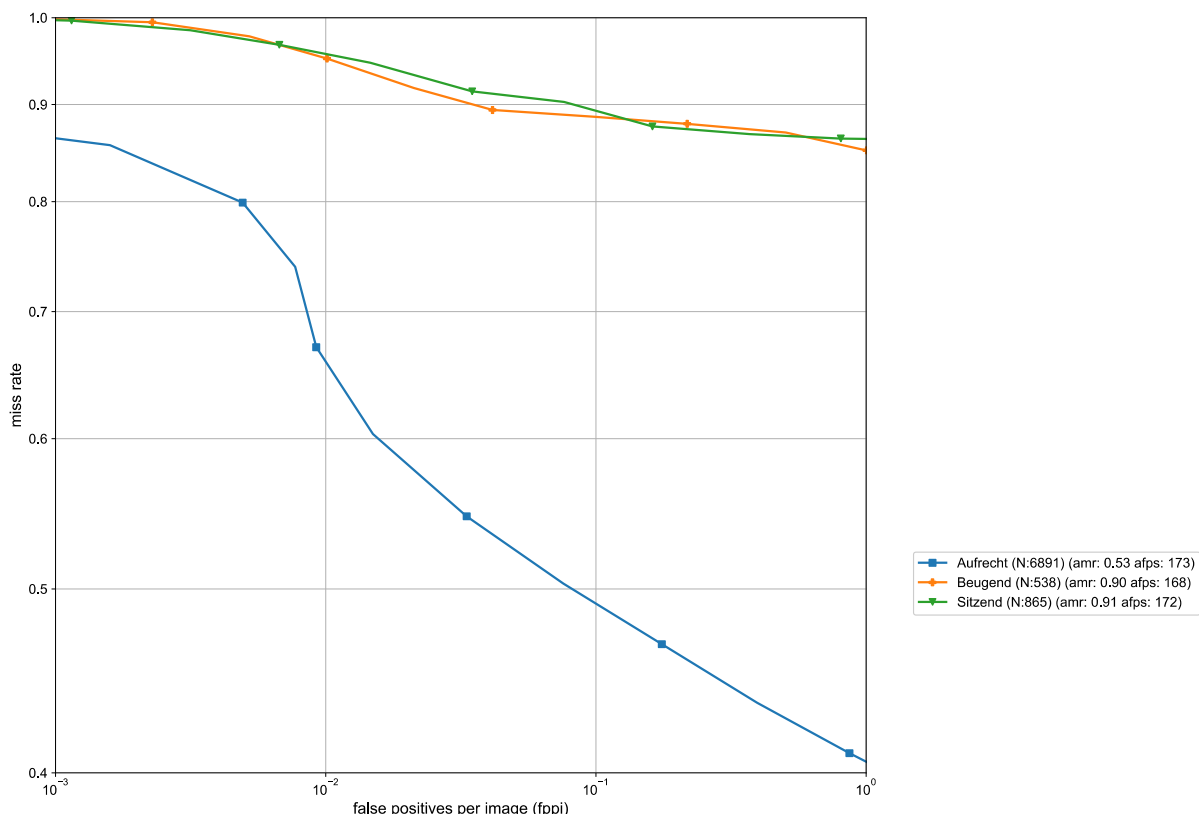


Abbildung A.1-16: Erkennungsleistung der SVM in Abhängigkeit der Posen von Personen am Beispiel des Intensitätsbilds.

Tabelle A.1-16: Rohdaten zur Erkennungsleistung der SVM in Abhängigkeit der Posen von Personen am Beispiel des Intensitätsbilds.

Aufrecht (N:6891) (amr: 0.53)														
fps	161,87	170,46	173,99	174,53	172,64	172,49	176,32	175,80	176,25	176,11	173,48	174,10	175,69	
precision	0,09	0,16	0,27	0,44	0,62	0,78	0,88	0,94	0,95	0,95	0,96	0,98	1,00	
recall	0,63	0,61	0,59	0,56	0,53	0,50	0,45	0,40	0,33	0,26	0,20	0,14	0,10	
missrate	0,37	0,39	0,41	0,44	0,47	0,50	0,55	0,60	0,67	0,74	0,80	0,86	0,90	
fppi	3,30	1,79	0,87	0,40	0,18	0,08	0,03	0,01	0,01	0,01	0,00	0,00	0,00	
Beugend (N:538) (amr: 0.90)														
fps	159,84	168,85	172,35	171,94	171,00	162,24	176,08	167,93	168,65	169,89	166,36	165,97	172,71	
precision	0,01	0,01	0,03	0,04	0,09	0,17	0,31	0,40	0,46	0,43	0,30	0,00	0,00	
recall	0,16	0,15	0,15	0,13	0,12	0,11	0,11	0,08	0,05	0,02	0,01	0,00	0,00	
missrate	0,84	0,85	0,85	0,87	0,88	0,89	0,89	0,92	0,95	0,98	0,99	1,00	1,00	

fppi	3,30	1,87	1,00	0,50	0,22	0,10	0,04	0,02	0,01	0,01	0,00	0,00	0,00
<b>Sitzend (N:865) (amr: 0.91)</b>													
fps	165,99	172,09	174,97	175,07	173,16	170,64	176,00	173,17	173,64	174,60	170,37	171,28	174,38
precision	0,01	0,01	0,02	0,04	0,09	0,14	0,23	0,31	0,37	0,37	0,27	0,00	0,00
recall	0,13	0,14	0,14	0,13	0,12	0,10	0,09	0,05	0,03	0,02	0,00	0,00	0,00
missrate	0,87	0,86	0,86	0,87	0,88	0,90	0,91	0,95	0,97	0,98	1,00	1,00	1,00
fppi	3,18	1,69	0,81	0,37	0,16	0,08	0,03	0,01	0,01	0,00	0,00	0,00	0,00

**Variation der detektierten Personen – Sicherheitsweste (Farb- und Intensitätsbild)**

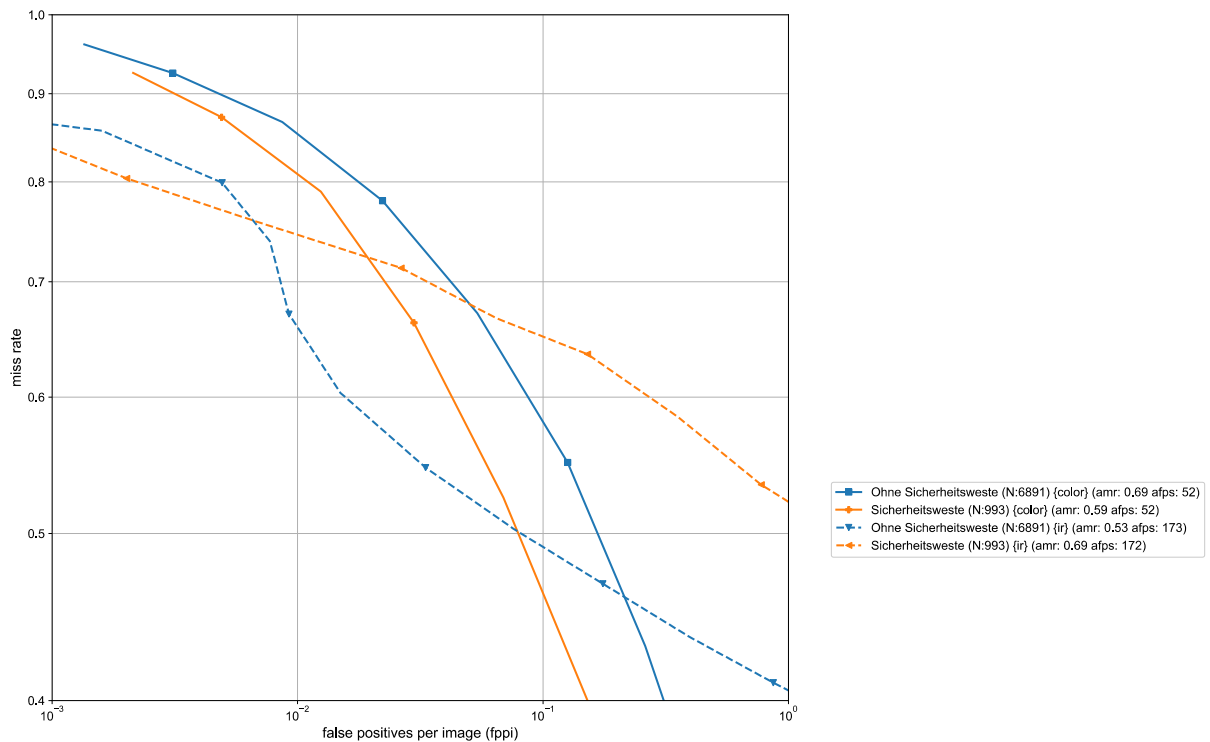


Abbildung A.1-17: Erkennungsleistung der SVM in Abhängigkeit des Tags einer Sicherheitsweste.

Tabelle A.1-17: Rohdaten zur Erkennungsleistung der SVM in Abhängigkeit des Tags einer Sicherheitsweste.

<b>Ohne Sicherheitsweste (N:6891) {color} (amr: 0.69)</b>																
fps	50,54	51,98	52,45	52,71	53,21	52,70	52,79	53,07	52,80	53,25	52,35	53,55	52,77	53,16	53,07	52,92
precision	0,06	0,07	0,09	0,11	0,14	0,18	0,24	0,32	0,43	0,54	0,66	0,77	0,84	0,89	0,93	0,94
recall	0,88	0,93	0,96	0,96	0,94	0,91	0,85	0,77	0,67	0,57	0,45	0,33	0,22	0,13	0,07	0,04
missrate	0,12	0,07	0,04	0,04	0,06	0,09	0,15	0,23	0,33	0,43	0,55	0,67	0,78	0,87	0,93	0,96
fppi	7,76	6,36	5,17	4,13	3,12	2,25	1,49	0,88	0,49	0,26	0,13	0,05	0,02	0,01	0,00	0,00
<b>Sicherheitsweste (N:993) {color} (amr: 0.59)</b>																
fps	50,32	51,77	52,68	52,65	53,42	52,53	52,98	52,68	53,06	52,82	52,26	53,45	52,83	53,34	53,54	53,35
precision	0,01	0,02	0,02	0,03	0,03	0,05	0,07	0,10	0,14	0,22	0,32	0,46	0,59	0,68	0,77	0,81
recall	0,79	0,81	0,83	0,87	0,92	0,93	0,91	0,86	0,79	0,70	0,61	0,48	0,34	0,21	0,13	0,07
missrate	0,21	0,19	0,17	0,13	0,08	0,07	0,09	0,14	0,21	0,30	0,39	0,52	0,66	0,79	0,87	0,93

# A Ergebnisse der Evaluation einzelner Algorithmen

fppi	7,88	6,56	5,32	4,21	3,22	2,34	1,59	1,00	0,58	0,32	0,16	0,07	0,03	0,01	0,00	0,00	
<b>Ohne Sicherheitsweste (N:6891) {ir} (amr: 0.53)</b>																	
fps	161,87	170,46	173,99	174,53	172,64	172,49	176,32	175,80	176,25	176,11	173,48	174,10	175,69				
precision	0,09	0,16	0,27	0,44	0,62	0,78	0,88	0,94	0,95	0,95	0,96	0,98	1,00				
recall	0,63	0,61	0,59	0,56	0,53	0,50	0,45	0,40	0,33	0,26	0,20	0,14	0,10				
missrate	0,37	0,39	0,41	0,44	0,47	0,50	0,55	0,60	0,67	0,74	0,80	0,86	0,90				
fppi	3,30	1,79	0,87	0,40	0,18	0,08	0,03	0,01	0,01	0,01	0,00	0,00	0,00				
<b>Sicherheitsweste (N:993) {ir} (amr: 0.69)</b>																	
fps	162,10	169,78	173,88	173,80	172,15	170,65	176,08	174,15	174,56	174,83	171,02	171,91	174,54				
precision	0,02	0,04	0,07	0,13	0,23	0,39	0,58	0,82	0,92	0,96	0,99	0,99	1,00				
recall	0,49	0,50	0,47	0,42	0,36	0,33	0,29	0,24	0,20	0,15	0,11	0,08	0,05				
missrate	0,51	0,50	0,53	0,58	0,64	0,67	0,71	0,76	0,80	0,85	0,89	0,92	0,95				
fppi	3,11	1,63	0,77	0,36	0,15	0,06	0,03	0,01	0,00	0,00	0,00	0,00	0,00				

## Variation der detektierten Personen – Aktivitäten (Intensitätsbild)

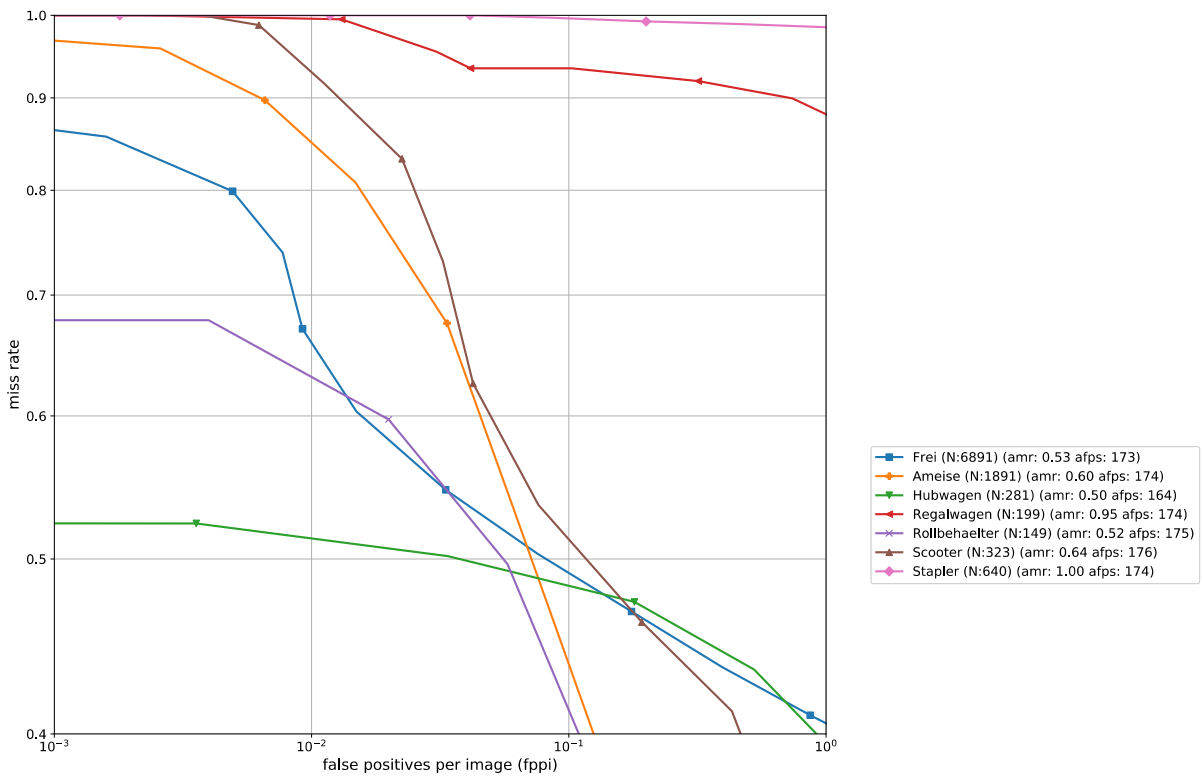


Abbildung A.1-18: Erkennungsleistung der SVM in Abhängigkeit der Aktivitäten von Personen am Beispiel des Intensitätsbilds.

Tabelle A.1-18: Rohdaten zur Erkennungsleistung der SVM in Abhängigkeit der Aktivitäten von Personen am Beispiel des Intensitätsbilds.

<b>Frei (N:6891) (amr: 0.53)</b>																	
fps	161,87	170,46	173,99	174,53	172,64	172,49	176,32	175,80	176,25	176,11	173,48	174,10	175,69				

## A.1 Personenerkennung

<b>precision</b>	0,09	0,16	0,27	0,44	0,62	0,78	0,88	0,94	0,95	0,95	0,96	0,98	1,00
<b>recall</b>	0,63	0,61	0,59	0,56	0,53	0,50	0,45	0,40	0,33	0,26	0,20	0,14	0,10
<b>missrate</b>	0,37	0,39	0,41	0,44	0,47	0,50	0,55	0,60	0,67	0,74	0,80	0,86	0,90
<b>fppi</b>	3,30	1,79	0,87	0,40	0,18	0,08	0,03	0,01	0,01	0,01	0,00	0,00	0,00
<b>Amelse (N:1891) (amr: 0.60)</b>													
<b>fps</b>	150,30	167,21	173,35	175,26	177,53	177,80	178,50	178,46	178,24	177,06	176,43	177,70	177,07
<b>precision</b>	0,09	0,12	0,21	0,33	0,46	0,58	0,68	0,77	0,81	0,85	0,87	0,88	0,97
<b>recall</b>	0,86	0,91	0,90	0,86	0,80	0,71	0,60	0,47	0,32	0,19	0,10	0,04	0,02
<b>missrate</b>	0,14	0,09	0,10	0,14	0,20	0,29	0,40	0,53	0,68	0,81	0,90	0,96	0,98
<b>fppi</b>	3,79	2,96	1,55	0,78	0,42	0,23	0,13	0,06	0,03	0,01	0,01	0,00	0,00
<b>Hubwagen (N:281) (amr: 0.50)</b>													
<b>fps</b>	147,36	153,44	160,30	172,44	157,99	159,18	159,66	172,84	161,73	161,30	175,70	177,79	173,37
<b>precision</b>	0,08	0,11	0,19	0,35	0,59	0,88	0,99	1,00	1,00	1,00	1,00	1,00	1,00
<b>recall</b>	0,76	0,70	0,62	0,57	0,53	0,50	0,48	0,41	0,30	0,20	0,14	0,07	0,02
<b>missrate</b>	0,24	0,30	0,38	0,43	0,47	0,50	0,52	0,59	0,70	0,80	0,86	0,93	0,98
<b>fppi</b>	4,27	2,76	1,28	0,53	0,18	0,03	0,00	0,00	0,00	0,00	0,00	0,00	0,00
<b>Regalwagen (N:199) (amr: 0.95)</b>													
<b>fps</b>	164,79	174,59	173,74	178,73	180,52	177,63	178,54	173,57	172,12	175,72	176,08	174,11	171,93
<b>precision</b>	0,05	0,05	0,04	0,06	0,10	0,22	0,41	0,39	0,14	0,00	0,00	0,00	0,00
<b>recall</b>	0,43	0,32	0,14	0,10	0,08	0,07	0,07	0,05	0,01	0,00	0,00	0,00	0,00
<b>missrate</b>	0,57	0,68	0,86	0,90	0,92	0,93	0,93	0,95	0,99	1,00	1,00	1,00	1,00
<b>fppi</b>	3,64	2,45	1,46	0,74	0,32	0,10	0,04	0,03	0,01	0,00	0,00	0,00	0,00
<b>Rollbehälter (N:149) (amr: 0.52)</b>													
<b>fps</b>	166,07	172,69	173,92	173,39	173,62	177,08	179,23	175,00	178,54	177,69	180,08	176,92	175,54
<b>precision</b>	0,05	0,10	0,21	0,36	0,54	0,72	0,86	0,96	1,00	1,00	1,00	1,00	1,00
<b>recall</b>	0,74	0,81	0,81	0,79	0,65	0,50	0,40	0,32	0,20	0,16	0,09	0,06	0,05
<b>missrate</b>	0,26	0,19	0,19	0,21	0,35	0,50	0,60	0,68	0,80	0,84	0,91	0,94	0,95
<b>fppi</b>	4,14	2,12	0,93	0,42	0,16	0,06	0,02	0,00	0,00	0,00	0,00	0,00	0,00
<b>Scooter (N:323) (amr: 0.64)</b>													
<b>fps</b>	167,65	172,61	176,68	175,34	176,93	178,72	178,65	179,97	176,22	175,40	179,79	177,66	177,86
<b>precision</b>	0,08	0,13	0,21	0,36	0,53	0,71	0,78	0,77	0,75	0,75	0,44	0,00	0,00
<b>recall</b>	0,77	0,77	0,71	0,59	0,54	0,46	0,37	0,27	0,17	0,08	0,01	0,00	0,00
<b>missrate</b>	0,23	0,23	0,29	0,41	0,46	0,54	0,63	0,73	0,83	0,92	0,99	1,00	1,00
<b>fppi</b>	3,38	2,17	1,06	0,43	0,19	0,08	0,04	0,03	0,02	0,01	0,01	0,00	0,00
<b>Stapler (N:640) (amr: 1.00)</b>													
<b>fps</b>	159,96	170,46	174,10	175,51	172,59	174,74	177,42	178,73	179,09	176,98	177,15	177,74	177,07
<b>precision</b>	0,00	0,00	0,00	0,00	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00
<b>recall</b>	0,03	0,02	0,02	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
<b>missrate</b>	0,97	0,98	0,98	0,99	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
<b>fppi</b>	3,43	2,30	1,12	0,45	0,20	0,09	0,04	0,02	0,01	0,01	0,00	0,00	0,00





## A.1.2 Neuronales Netz

### Variation der trainierten Bildarten

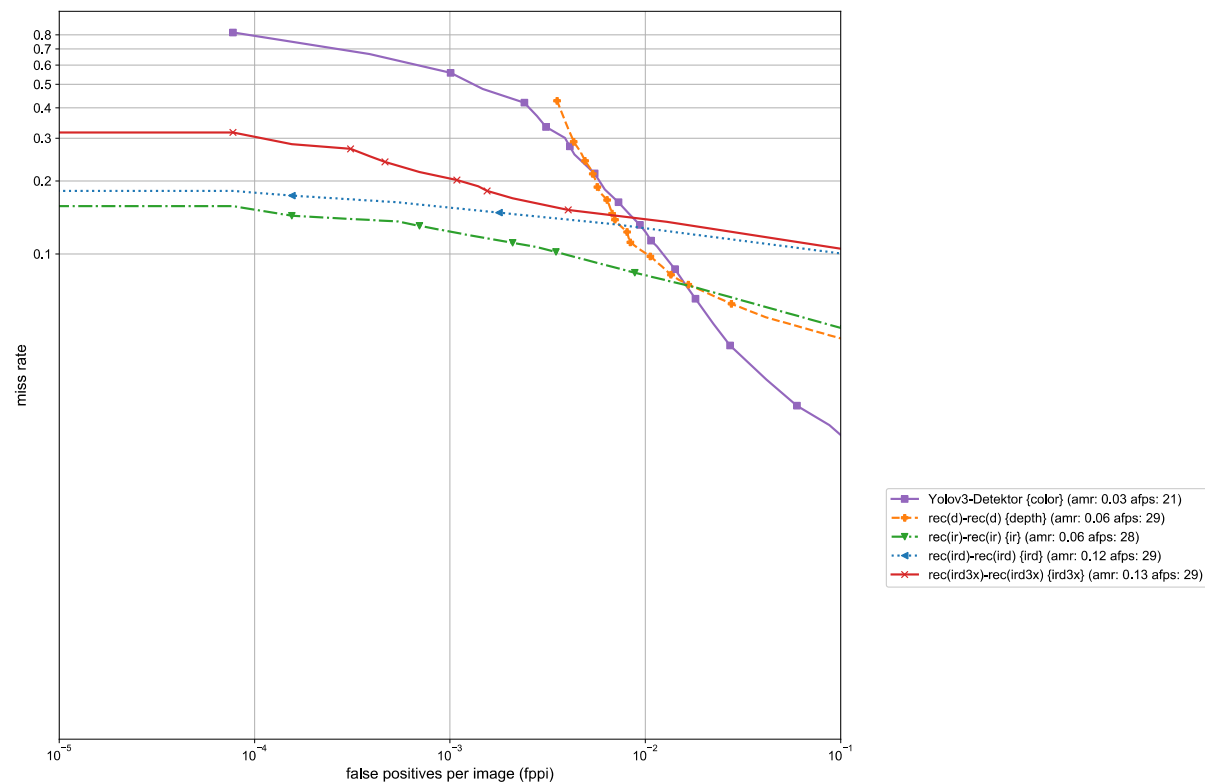


Abbildung A.1-19: Ergebnisse der Variation der Bildarten mit einem DL-Detektor für Personen, die maximal zu 33 % verdeckt sind.

Tabelle A.1-19: Rohdaten zur Variation der Bildarten mit einem DL-Detektor für Personen, die maximal zu 33 % verdeckt sind.

Yolov3-Detektor {color} (amr: 0.03)

fps 21,32

## A Ergebnisse der Evaluation einzelner Algorithmen

<b>precision</b>	0,02	0,49	0,67	0,80	0,87	0,89	0,92	0,95	0,96	0,97	0,97	0,98	0,98	0,98	0,98	0,98	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	1,00	1,00	1,00	1,00
<b>recall</b>	1,00	0,99	0,99	0,99	0,98	0,98	0,98	0,97	0,96	0,95	0,93	0,93	0,91	0,89	0,89	0,88	0,87	0,86	0,84	0,82	0,79	0,74	0,72	0,70	0,67	0,63	0,58	0,52	0,44	0,33	0,18		
<b>missrate</b>	0,00	0,01	0,01	0,01	0,02	0,02	0,02	0,03	0,04	0,05	0,07	0,07	0,09	0,11	0,11	0,12	0,13	0,14	0,16	0,18	0,21	0,26	0,28	0,30	0,33	0,37	0,42	0,48	0,56	0,67	0,82		
<b>fppi</b>	40,03	0,78	0,37	0,18	0,11	0,09	0,06	0,04	0,03	0,02	0,02	0,02	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
<b>rec(d)-rec(d) {depth} (amr: 0.06)</b>																																	
<b>fps</b>	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00
<b>precision</b>	0,79	0,93	0,95	0,97	0,97	0,97	0,98	0,98	0,98	0,98	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99
<b>recall</b>	0,96	0,95	0,94	0,93	0,93	0,92	0,92	0,91	0,90	0,90	0,89	0,88	0,88	0,87	0,86	0,86	0,85	0,85	0,84	0,83	0,82	0,81	0,80	0,80	0,79	0,78	0,76	0,74	0,71	0,67	0,57		
<b>missrate</b>	0,04	0,05	0,06	0,07	0,07	0,08	0,08	0,09	0,10	0,10	0,11	0,12	0,12	0,13	0,14	0,14	0,15	0,15	0,16	0,17	0,18	0,19	0,20	0,20	0,21	0,22	0,24	0,26	0,29	0,33	0,43		
<b>fppi</b>	0,16	0,04	0,03	0,02	0,02	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	
<b>rec(ir)-rec(ir) {ir} (amr: 0.06)</b>																																	
<b>fps</b>	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	28,23	
<b>precision</b>	0,87	0,97	0,99	0,99	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	
<b>recall</b>	0,95	0,93	0,92	0,91	0,90	0,89	0,89	0,88	0,87	0,86	0,86	0,85	0,84	0,83	0,83	0,83	0,82	0,82	0,82	0,81	0,80	0,79	0,79	0,78	0,77	0,77	0,76	0,74	0,72	0,69	0,62		
<b>missrate</b>	0,05	0,07	0,08	0,09	0,10	0,11	0,11	0,12	0,13	0,14	0,14	0,15	0,16	0,17	0,17	0,17	0,18	0,18	0,18	0,19	0,20	0,21	0,21	0,22	0,23	0,23	0,24	0,26	0,28	0,31	0,38		
<b>fppi</b>	0,10	0,02	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
<b>rec(ird)-rec(ird) {ird} (amr: 0.12)</b>																																	
<b>fps</b>	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	
<b>precision</b>	0,80	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	
<b>recall</b>	0,90	0,87	0,85	0,84	0,83	0,82	0,81	0,80	0,78	0,77	0,76	0,75	0,74	0,72	0,72	0,71	0,70	0,69	0,68	0,66	0,64	0,63	0,62	0,60	0,59	0,56	0,54	0,50	0,44	0,33			
<b>missrate</b>	0,10	0,13	0,15	0,16	0,17	0,18	0,19	0,20	0,22	0,23	0,24	0,25	0,26	0,28	0,28	0,28	0,29	0,30	0,31	0,32	0,34	0,36	0,37	0,38	0,40	0,41	0,44	0,46	0,50	0,56	0,67		
<b>fppi</b>	0,17	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
<b>rec(ird3x)-rec(ird3x) {ird3x} (amr: 0.13)</b>																																	
<b>fps</b>	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	29,00	
<b>precision</b>	0,73	0,98	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	
<b>recall</b>	0,91	0,86	0,85	0,83	0,82	0,81	0,80	0,78	0,76	0,75	0,73	0,72	0,70	0,68	0,68	0,67	0,66	0,65	0,64	0,63	0,61	0,60	0,59	0,57	0,56	0,55	0,53	0,50	0,46	0,41	0,31		
<b>missrate</b>	0,09	0,14	0,15	0,17	0,18	0,19	0,20	0,22	0,24	0,25	0,27	0,28	0,30	0,32	0,32	0,33	0,34	0,35	0,36	0,37	0,39	0,40	0,41	0,43	0,44	0,45	0,47	0,50	0,54	0,59	0,69		
<b>fppi</b>	0,25	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	

Variation der detektierten Personen – Verdeckung (Intensitätsbild)

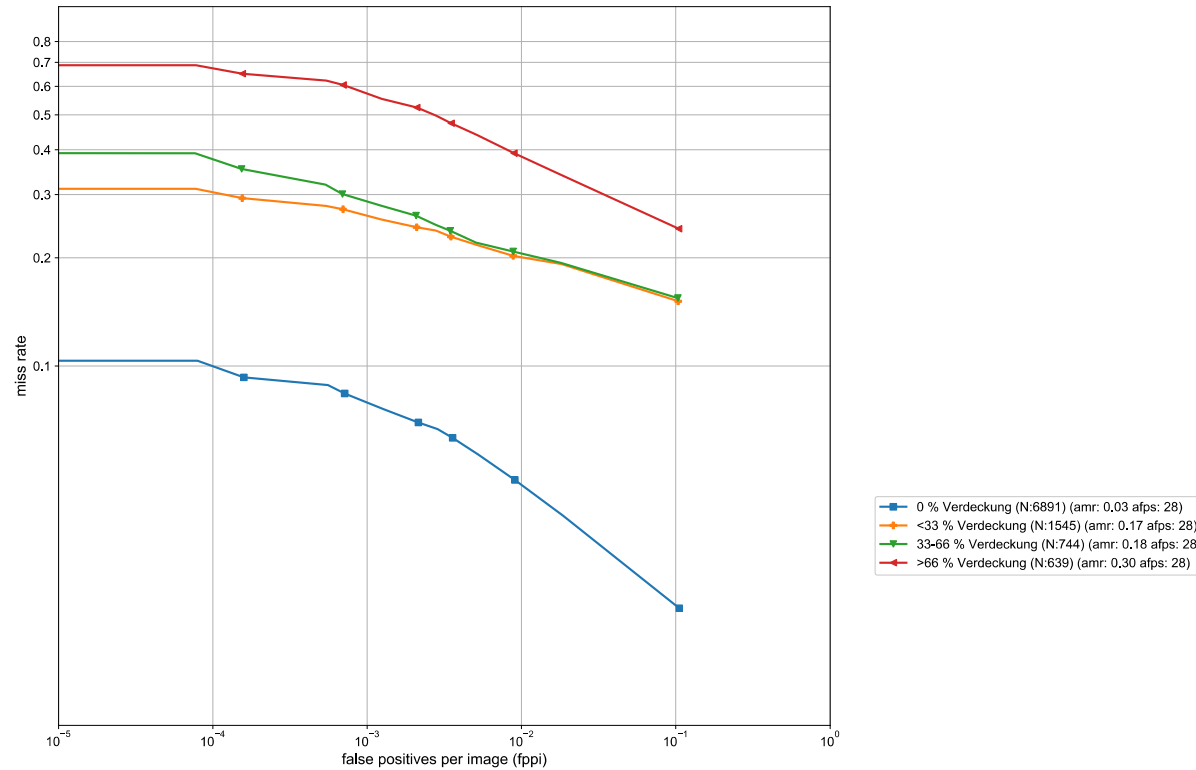


Abbildung A.1-20: Unterteilung der DL-Detektionsergebnisse in Abhängigkeit des Verdeckungsgrads markierter Personen.

Tabelle A.1-20: Rohdaten zur Unterteilung der DL-Detektionsergebnisse in Abhängigkeit des Verdeckungsgrads markierter Personen.

0 % Verdeckung (N:6891) (amr: 0.03)																																	
fps	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22





## A Ergebnisse der Evaluation einzelner Algorithmen

<b>recall</b>	0,98	0,96	0,95	0,94	0,94	0,93	0,93	0,92	0,92	0,91	0,91	0,90	0,90	0,89	0,89	0,88	0,88	0,88	0,87	0,87	0,86	0,86	0,85	0,85	0,84	0,83	0,82	0,81	0,79	0,76	0,69	
<b>missrate</b>	0,02	0,04	0,05	0,06	0,06	0,07	0,07	0,08	0,08	0,09	0,09	0,10	0,10	0,11	0,11	0,12	0,12	0,12	0,13	0,13	0,14	0,14	0,15	0,15	0,16	0,17	0,18	0,19	0,21	0,24	0,31	
<b>fppi</b>	0,11	0,02	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00		
<b>Beugend (N:538) (amr: 0.31)</b>																																
<b>fps</b>	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	28,09	
<b>precision</b>	0,46	0,80	0,86	0,92	0,94	0,95	0,96	0,96	0,97	0,97	0,97	0,98	0,98	0,98	0,98	0,98	0,98	0,98	0,98	0,98	0,98	0,99	0,99	0,99	1,00	1,00	1,00	1,00	1,00	1,00		
<b>recall</b>	0,72	0,68	0,67	0,65	0,65	0,65	0,65	0,64	0,64	0,64	0,63	0,63	0,62	0,62	0,62	0,62	0,61	0,61	0,61	0,60	0,60	0,59	0,59	0,58	0,58	0,57	0,55	0,54	0,52	0,49	0,43	
<b>missrate</b>	0,28	0,32	0,33	0,35	0,35	0,35	0,35	0,36	0,36	0,36	0,37	0,37	0,38	0,38	0,38	0,38	0,39	0,39	0,39	0,40	0,40	0,41	0,41	0,42	0,42	0,43	0,45	0,46	0,48	0,51	0,57	
<b>fppi</b>	0,15	0,03	0,02	0,01	0,01	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00		
<b>Sitzend (N:865) (amr: 0.76)</b>																																
<b>fps</b>	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36	28,36		
<b>precision</b>	0,17	0,53	0,68	0,81	0,87	0,90	0,92	0,94	0,96	0,97	0,99	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00		
<b>recall</b>	0,25	0,24	0,23	0,22	0,22	0,22	0,22	0,22	0,22	0,22	0,21	0,21	0,21	0,20	0,20	0,20	0,20	0,20	0,20	0,20	0,19	0,19	0,18	0,18	0,17	0,16	0,15	0,14	0,12	0,11	0,08	0,05
<b>missrate</b>	0,75	0,76	0,77	0,78	0,78	0,78	0,78	0,78	0,78	0,78	0,79	0,79	0,80	0,80	0,80	0,80	0,80	0,80	0,80	0,81	0,81	0,82	0,82	0,83	0,84	0,85	0,86	0,88	0,89	0,92	0,95	
<b>fppi</b>	0,15	0,03	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	

Variation der detektierten Personen – Sicherheitsweste (Farb- und Intensitätsbild)

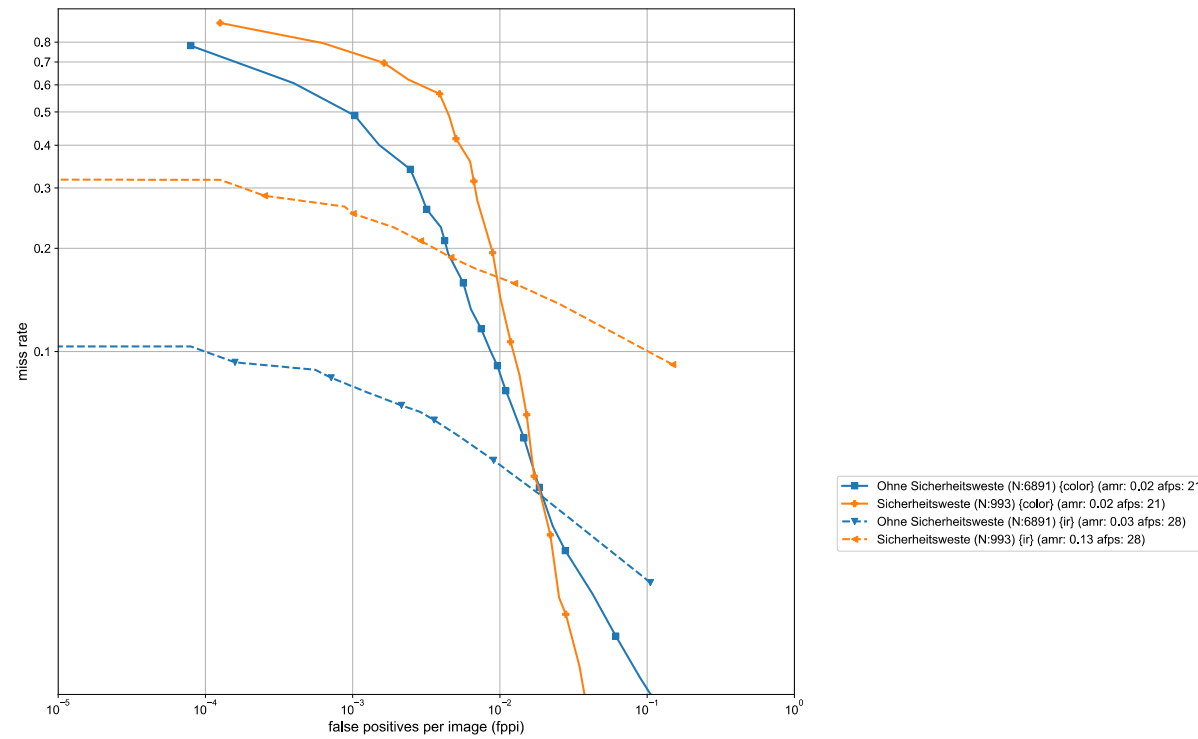


Abbildung A.1-22: Einfluss der Sicherheitsweste bei der DL-Detektion im Farb- und Intensitätsbild.

Tabelle A.1-22: Rohdaten zum Einfluss der Sicherheitsweste bei der DL-Detektion im Farb- und Intensitätsbild.

Ohne Sicherheitsweste (N:6891) {color} (amr: 0.02)																																	
fps	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33	21,33
precision	0,01	0,41	0,59	0,75	0,82	0,86	0,90	0,93	0,95	0,96	0,97	0,97	0,97	0,98	0,98	0,98	0,98	0,98	0,98	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	0,99	1,00	1,00	1,00	1,00
recall	1,00	1,00	0,99	0,99	0,99	0,99	0,99	0,98	0,97	0,97	0,96	0,95	0,94	0,93	0,92	0,92	0,91	0,90	0,88	0,87	0,84	0,81	0,79	0,77	0,74	0,71	0,66	0,60	0,51	0,39	0,22		
missrate	0,00	0,00	0,01	0,01	0,01	0,01	0,01	0,02	0,03	0,03	0,04	0,05	0,06	0,07	0,08	0,08	0,09	0,10	0,12	0,13	0,16	0,19	0,21	0,23	0,26	0,29	0,34	0,40	0,49	0,61	0,78		





Variation der detektierten Personen – Aktivitäten (Intensitätsbild)

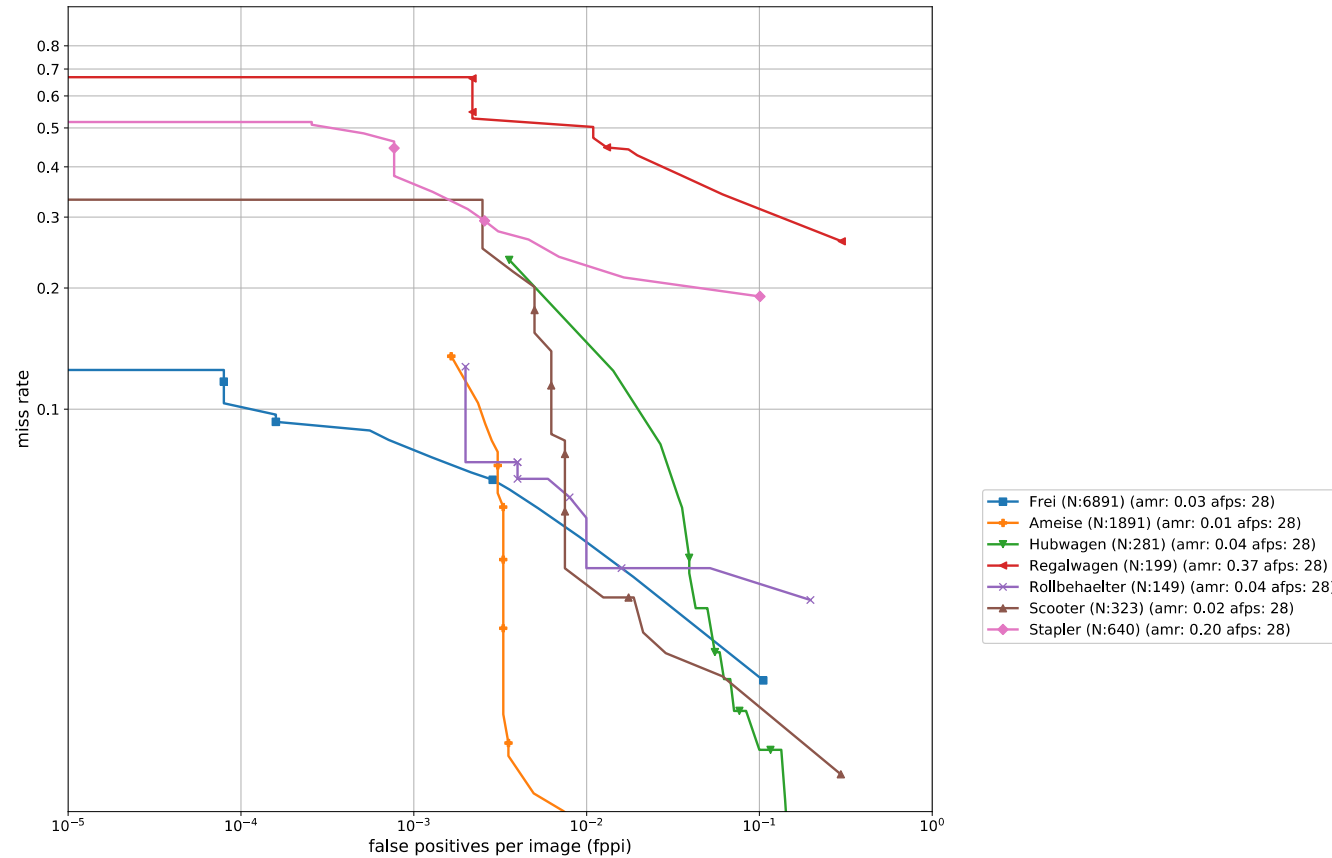


Abbildung A.1-23: Unterteilung der DL-Detektionsergebnisse in Abhängigkeit der Aktivität markierter Personen.

Tabelle A.1-23: Rohdaten zur Unterteilung der DL-Detektionsergebnisse in Abhängigkeit der Aktivität markierter Personen.

Frei (N:6891) (amr: 0.03)																																
fps	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22	28,22



Scooter (N:323) (amr: 0.02)																																		
<b>fps</b>	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00	28,00
<b>precision</b>	0,57	0,87	0,93	0,95	0,95	0,96	0,96	0,97	0,98	0,98	0,98	0,98	0,98	0,98	0,98	0,98	0,98	0,98	0,98	0,98	0,98	0,98	0,98	0,98	0,99	0,99	0,98	0,99	0,99	0,99	0,99	0,99	1,00	
<b>recall</b>	0,99	0,98	0,98	0,97	0,97	0,97	0,97	0,97	0,96	0,96	0,94	0,94	0,94	0,93	0,93	0,92	0,92	0,92	0,91	0,90	0,89	0,87	0,86	0,86	0,85	0,82	0,80	0,78	0,75	0,67	0,46			
<b>misrate</b>	0,01	0,02	0,02	0,03	0,03	0,03	0,03	0,03	0,04	0,04	0,06	0,06	0,06	0,07	0,07	0,08	0,08	0,08	0,09	0,10	0,11	0,13	0,14	0,14	0,15	0,18	0,20	0,22	0,25	0,33	0,54			
<b>fppi</b>	0,30	0,06	0,03	0,02	0,02	0,02	0,02	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00		
Stapler (N:640) (amr: 0.20)																																		
<b>fps</b>	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06	28,06
<b>precision</b>	0,57	0,89	0,95	0,96	0,97	0,98	0,98	0,99	0,99	0,99	0,99	0,99	0,99	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
<b>recall</b>	0,81	0,79	0,76	0,74	0,72	0,71	0,69	0,65	0,62	0,59	0,55	0,54	0,52	0,49	0,48	0,48	0,47	0,46	0,44	0,43	0,40	0,38	0,36	0,34	0,31	0,30	0,29	0,26	0,23	0,19	0,13			
<b>misrate</b>	0,19	0,21	0,24	0,26	0,28	0,29	0,31	0,35	0,38	0,41	0,45	0,46	0,48	0,51	0,52	0,52	0,53	0,54	0,56	0,57	0,60	0,62	0,64	0,66	0,69	0,70	0,71	0,74	0,77	0,81	0,87			
<b>fppi</b>	0,10	0,02	0,01	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	



## A.2 Kollisionsgefahrerkennung

Tabelle A.2-1: Ergebnisse der Kollisionstests ohne Personen, Vergleich der OF-Algorithmen im Intensitätsbild (RANASC-Intervall: 0, Entfernung des Bodens & Hintergrunds: An, Mindestanzahl an Kollisionen: 2).

Testfall	Testszenario	BR	D-TVL	FB	LK	PD
<b>Warnungsrate</b>						
1	1a	1	0,89	1	1	0,89
2	1a	1	1	1	1	1
3	1a	1	1	1	1	1
4	1a	0	1	1	1	1
5	1a	0	0,9	1	1	1
6	1a	1	1	1	1	1
7	1a	0,9	1	1	1	1
8	1a	0,9	1	1	1	0,9
9	1a	1	1	0,9	1	1
10	1a	0,9	1	1	1	0,9
11	1b	0,4	1	0,9	1	0,6
12	1b	0	0,3	0	1	0,9
13	1b	0	0	0	1	1
14	1b	0	0	0	1	1
15	1c	1	1	1	1	1
16	1c	0,6	1	1	1	0,9
17	1c	0,4	1	0,9	1	1
18	1c	0,4	1	1	1	1
19	2a	1	0,3	0,8	0,9	1
20	2a	1	0	1	1	1
21	2a	0,7	0	0,9	1	1
22	2a	0	0	0,6	1	0,6
23	2b	0,7	0	0,9	1	1
24	2b	0,8	0	1	1	1
25	2b	1	0	1	1	1
26	2b	0,4	0	0,9	1	1
27	2c	1	1	1	1	1
28	2c	1	1	0,9	1	1
29	2c	0,9	1	1	1	0,8
30	2c	0,5	0,9	1	1	1
31	2d	0,8	1	0,9	1	1
32	2d	1	1	1	1	1
33	2d	0,9	1	1	1	0,9
34	2d	1	1	0,8	1	1
<b>Rechtzeitig</b>						
1	1a	0,22	1	1	1	1
2	1a	0	1	1	1	1
3	1a	0	1	1	0,9	1

## A Ergebnisse der Evaluation einzelner Algorithmen

---

4 1a	-	0,9	0	0,9	0
5 1a	-	1	1	1	0,6
6 1a	0	1	1	1	1
7 1a	0	1	1	1	1
8 1a	0	1	0,9	1	0,78
9 1a	0	1	1	1	1
10 1a	0	1	1	1	1
11 1b	-	-	-	-	-
12 1b	-	-	-	-	-
13 1b	-	-	-	-	-
14 1b	-	-	-	-	-
15 1c	0	1	1	1	0,9
16 1c	0	1	1	1	0,67
17 1c	0	1	1	1	1
18 1c	0	1	1	1	1

### Permanenz

1 1a	0,16	0,82	0,84	0,79	0,68
2 1a	0,31	0,77	0,82	0,96	0,78
3 1a	0,5	0,76	0,8	0,78	0,9
4 1a	-	0,82	0,86	0,99	0,56
5 1a	-	0,79	0,93	0,84	0,65
6 1a	0,47	0,71	1	0,94	0,82
7 1a	0,72	0,86	0,92	0,92	0,93
8 1a	0,25	0,81	0,7	0,89	0,64
9 1a	0,42	0,81	0,75	0,85	0,7
10 1a	0,4	0,86	0,82	0,76	0,65
11 1b	0,73	0,8	0,95	0,76	0,28
12 1b	-	0,83	-	0,81	0,33
13 1b	-	-	-	0,91	0,36
14 1b	-	-	-	1	0,28
15 1c	0,2	0,98	0,77	0,95	0,65
16 1c	0,3	0,85	0,95	0,84	0,57
17 1c	0,13	0,96	0,75	0,8	0,64
18 1c	0,11	0,73	0,62	1	0,79
19 2a	0,27	0,65	0,59	0,84	0,75
20 2a	0,26	-	0,88	0,95	0,73
21 2a	0,14	-	0,71	0,81	0,53
22 2a	-	-	0,17	0,85	0,06
23 2b	0,2	-	0,51	0,84	0,65
24 2b	0,17	-	0,45	0,98	0,51
25 2b	0,17	-	0,48	1	0,27
26 2b	0,1	-	0,18	0,95	0,13
27 2c	0,23	0,78	0,25	0,69	0,28
28 2c	0,17	0,64	0,31	0,69	0,26
29 2c	0,12	0,42	0,32	0,79	0,21
30 2c	0,09	0,43	0,28	0,74	0,13
31 2d	0,45	0,59	0,48	0,78	0,3
32 2d	0,46	0,69	0,54	0,95	0,28

<b>33</b> 2d	0,55	0,93	0,61	0,85	0,19
<b>34</b> 2d	0,47	0,91	0,48	0,77	0,24

### A.2.1 Variation der Bildart

Tabelle A.2-2: Ergebnisse der Kollisionstests ohne Personen, Vergleich der OF-Algorithmen im Farbbild. (RANASC-Intervall: 0, Entfernung des Bodens & Hintergrunds: An, Mindestanzahl an Kollisionen: 2).

Testfall	Testszenario	BR	D-TVL	FB	LK	PD
<b>Warnungsrate</b>						
<b>1</b>	1a	1	0,89	0,89	0,89	1
<b>2</b>	1a	0,9	1	1	1	1
<b>3</b>	1a	1	1	1	1	1
<b>4</b>	1a	0	1	1	1	1
<b>5</b>	1a	0,1	1	1	1	1
<b>6</b>	1a	0,8	1	1	1	1
<b>7</b>	1a	1	1	1	1	1
<b>8</b>	1a	1	1	0,9	1	1
<b>9</b>	1a	0,8	1	1	1	1
<b>10</b>	1a	0,8	1	1	1	1
<b>11</b>	1b	0,1	0,9	0	1	1
<b>12</b>	1b	0	1	0	1	1
<b>13</b>	1b	0	1	0	1	0,9
<b>14</b>	1b	0	1	0	1	0,7
<b>15</b>	1c	0,8	1	1	1	1
<b>16</b>	1c	0,8	1	1	1	0,9
<b>17</b>	1c	1	1	1	1	1
<b>18</b>	1c	0,8	1	1	1	1
<b>19</b>	2a	0,9	1	1	1	0,9
<b>20</b>	2a	1	1	0,9	1	1
<b>21</b>	2a	0,7	1	1	1	1
<b>22</b>	2a	0,2	1	1	1	1
<b>23</b>	2b	0,7	1	0,9	1	1
<b>24</b>	2b	0,4	1	1	1	1
<b>25</b>	2b	0,7	1	1	1	1
<b>26</b>	2b	0,4	1	1	1	1
<b>27</b>	2c	0,4	0,9	0,8	1	0,9
<b>28</b>	2c	0,4	1	0,8	1	1
<b>29</b>	2c	0,2	1	0,8	1	0,8
<b>30</b>	2c	0,1	1	1	1	0,9
<b>31</b>	2d	0,9	0,9	1	1	0,9
<b>32</b>	2d	0,9	1	1	1	0,8
<b>33</b>	2d	1	1	1	1	0,8
<b>34</b>	2d	0,9	1	1	1	0,8
<b>Rechtzeitig</b>						
<b>1</b>	1a	0,33	1	1	1	1



## A Ergebnisse der Evaluation einzelner Algorithmen

---

2 1a	0	1	1	1	1
3 1a	0	1	1	1	1
4 1a	-	0,9	0,1	0,9	0
5 1a	0	1	1	1	0,7
6 1a	0	1	1	1	1
7 1a	0	1	1	1	1
8 1a	0	1	0,33	1	1
9 1a	0	1	1	1	1
10 1a	0	1	1	1	1
11 1b	-	-	-	-	-
12 1b	-	-	-	-	-
13 1b	-	-	-	-	-
14 1b	-	-	-	-	-
15 1c	0,12	1	1	1	1
16 1c	0	1	1	1	0,78
17 1c	0	1	1	1	0,9
18 1c	0	1	1	1	1

### Permanenz

1 1a	0,31	0,8	0,94	0,88	0,88
2 1a	0,24	0,9	0,92	0,91	0,84
3 1a	0,34	0,98	1	0,94	0,94
4 1a	-	0,89	0,87	0,73	0,74
5 1a	1	0,94	0,97	0,84	0,75
6 1a	0,67	0,93	0,85	1	0,89
7 1a	0,64	0,95	0,85	0,95	0,87
8 1a	0,24	0,95	0,77	0,9	0,75
9 1a	0,4	0,98	0,79	1	0,79
10 1a	0,28	0,9	0,88	1	0,78
11 1b	0,5	0,78	-	0,82	0,37
12 1b	-	0,89	-	0,93	0,25
13 1b	-	0,95	-	0,8	0,11
14 1b	-	0,9	-	0,95	0,05
15 1c	0,33	0,95	1	0,95	0,76
16 1c	0,27	0,95	0,84	1	0,72
17 1c	0,22	0,95	0,9	1	0,54
18 1c	0,28	0,95	0,99	1	0,92
19 2a	0,26	0,95	0,78	1	0,66
20 2a	0,31	0,85	0,78	1	0,82
21 2a	0,11	0,87	0,7	0,9	0,62
22 2a	0,08	0,95	0,25	1	0,18
23 2b	0,16	0,95	0,48	1	0,46
24 2b	0,11	1	0,57	0,95	0,47
25 2b	0,15	0,91	0,45	0,95	0,28
26 2b	0,1	0,99	0,4	0,96	0,32
27 2c	0,07	0,6	0,23	0,91	0,1
28 2c	0,08	0,75	0,14	0,84	0,24
29 2c	0,04	0,42	0,28	0,85	0,19
30 2c	0,03	0,51	0,4	0,81	0,15

<b>31</b>	2d	0,22	0,84	0,49	0,9	0,12
<b>32</b>	2d	0,13	0,85	0,59	1	0,08
<b>33</b>	2d	0,14	0,76	0,63	1	0,08
<b>34</b>	2d	0,13	0,56	0,56	1	0,07

## A.2.2 Variation der Vorverarbeitung – Entfernung des Bodens und Hintergrunds

Tabelle A.2-3: Ergebnisse der Kollisionstests ohne Personen, ohne Entfernung des Bodens und des Hintergrunds.

Testfall	Testszenario	Warnungsrate			Rechtzeitig			Permanenz		
		FB	LK	PD	FB	LK	PD	FB	LK	PD
1	1a	1	1	0,89	0,89	1	1	0,67	0,86	0,8
2	1a	0,8	1	0,9	1	1	1	0,71	0,9	0,67
3	1a	1	0,9	0,9	0,9	1	0,89	0,59	0,82	0,56
4	1a	0,9	1	0,8	0,67	0,9	0,5	0,76	0,76	0,39
5	1a	1	1	0,8	0,9	1	0,5	0,82	0,73	0,52
6	1a	1	1	1	1	1	1	0,89	0,89	0,69
7	1a	1	1	1	1	1	1	0,9	0,95	0,73
8	1a	1	1	1	0,9	1	1	0,65	1	0,67
9	1a	1	1	1	1	1	1	0,77	0,76	0,64
10	1a	0,9	1	1	1	0,9	1	0,64	0,74	0,68
11	1b	0,8	1	0,7	-	-	-	0,9	0,56	0,22
12	1b	0	1	0,9	-	-	-	-	0,81	0,31
13	1b	0	1	1	-	-	-	-	0,83	0,28
14	1b	0	1	0,9	-	-	-	-	0,93	0,25
15	1c	0,9	1	0,9	1	1	0,78	0,9	0,9	0,7
16	1c	0,9	1	0,8	1	1	0,62	0,84	0,95	0,54
17	1c	1	1	1	1	1	0,8	0,75	0,85	0,55
18	1c	0,8	1	1	1	1	1	0,6	0,93	0,75
19	2a	1	1	1				0,51	0,87	0,69
20	2a	0,9	1	1				0,87	0,86	0,79
21	2a	1	1	1				0,74	0,68	0,47
22	2a	0,7	1	0,8				0,17	0,9	0,15
23	2b	1	1	1				0,48	0,94	0,76
24	2b	0,8	1	1				0,41	0,99	0,56
25	2b	1	1	1				0,31	0,99	0,44
26	2b	1	1	1				0,11	0,89	0,24
27	2c	0,9	1	1				0,25	0,68	0,24
28	2c	1	1	1				0,24	0,75	0,27
29	2c	1	1	0,9				0,23	0,76	0,14
30	2c	1	1	0,8				0,32	0,83	0,14
31	2d	0,8	0,9	0,9				0,4	0,78	0,21
32	2d	1	1	1				0,7	0,94	0,22
33	2d	0,8	1	0,9				0,64	0,9	0,17

## A Ergebnisse der Evaluation einzelner Algorithmen

34 2d                    0,9   1   1                    0,6 0,89 0,21

### A.2.3 Variation der Nachbearbeitung – RANSAC-Filterung

Tabelle A.2-4: Ergebnisse der Kollisionstests ohne Personen, RANSAC-Intervall = 3.

Testfall	Testszenario	Warnungsrate				Rechtzeitig				Permanenz			
		BR	FB	LK	PD	BR	FB	LK	PD	BR	FB	LK	PD
1	1a	0,22	0,89	1	0,78	0	1	1	1	0,09	0,75	0,89	0,7
2	1a	0,3	1	0,9	0,9	0	1	1	1	0,09	0,7	0,91	0,86
3	1a	0,6	1	0,9	1	0	1	1	0,9	0,19	0,71	0,74	0,74
4	1a	0	1	1	0,5	-	0	0,9	0	-	0,82	0,8	0,32
5	1a	0	1	1	1	-	1	1	0,2	-	0,94	0,99	0,56
6	1a	0,9	1	1	0,8	0,11	1	1	1	0,45	0,85	1	0,7
7	1a	1	1	1	1	0	1	1	1	0,8	0,9	0,93	0,85
8	1a	0,4	0,9	1	1	0	0	1	0,8	0,07	0,75	0,95	0,53
9	1a	0,6	1	1	1	0	0,6	1	0,9	0,33	0,65	0,89	0,69
10	1a	0,9	0,9	1	1	0	1	1	1	0,25	0,68	0,81	0,84
11	1b	0	0,9	1	0,3	-	-	-	-	-	0,88	0,8	0,35
12	1b	0	0	1	0,5	-	-	-	-	-	-	0,8	0,42
13	1b	0	0	1	1	-	-	-	-	-	-	0,86	0,21
14	1b	0	0	1	0,8	-	-	-	-	-	-	0,8	0,33
15	1c	0,3	1	1	1	0	1	1	0,1	0,19	0,78	0,84	0,67
16	1c	0,2	1	1	1	0	0,9	1	0	0,13	0,71	0,89	0,63
17	1c	0	0,8	1	1	-	1	1	0	-	0,78	0,83	0,65
18	1c	0,1	0,9	1	1	0	1	1	0,5	0,27	0,81	0,94	0,81
19	2a	0	0,9	1	1					-	0,7	0,75	0,5
20	2a	0,1	1	1	1					0,03	0,82	0,95	0,59
21	2a	0	1	1	0,9					-	0,79	0,9	0,33
22	2a	0	0,2	1	0,3					-	0,06	0,89	0,07
23	2b	0	1	1	1					-	0,49	0,85	0,66
24	2b	0	0,9	1	1					-	0,65	1	0,45
25	2b	0	1	1	1					-	0,64	1	0,15
26	2b	0	0,9	1	0,8					-	0,3	0,94	0,03
27	2c	0,1	0,7	0,9	1					0,19	0,16	0,51	0,22
28	2c	0	0,8	1	0,9					-	0,06	0,45	0,15
29	2c	0	0,5	1	0,9					-	0,08	0,38	0,08
30	2c	0	0,7	0,9	0,5					-	0,08	0,42	0,06
31	2d	0,9	1	1	0,9					0,18	0,28	0,68	0,25
32	2d	1	1	1	0,9					0,21	0,43	0,88	0,19
33	2d	1	1	1	0,9					0,12	0,44	0,89	0,15
34	2d	1	1	1	1					0,16	0,49	0,82	0,2

Tabelle A.2-5: Ergebnisse der Kollisionstests ohne Personen, RANSAC-Intervall = 4.

Testszenario	Testfall	Warnungsrate				Rechtzeitig				Permanenz			
		BR	FB	LK	PD	BR	FB	LK	PD	BR	FB	LK	PD
1	1a	0	1	1	1	-	1	1	1	-	0,78	0,89	0,72
2	1a	0	1	1	1	-	1	1	1	-	0,9	0,93	0,66
3	1a	0	0,9	1	1	-	1	1	0,4	-	0,82	0,78	0,56
4	1a	0	1	1	0	-	0	0,9	-	-	0,75	0,79	-
5	1a	0	0,9	1	1	-	0,78	1	0,3	-	0,57	0,8	0,77
6	1a	0,3	1	1	1	0	1	1	0,8	0,43	0,78	0,95	0,85
7	1a	1	1	1	1	0	1	1	0,7	0,66	0,93	0,89	0,77
8	1a	0	0,9	1	1	-	0	1	0,3	-	0,84	0,67	0,54
9	1a	0	1	1	1	-	0,4	1	0,4	-	0,84	0,8	0,74
10	1a	0	1	0,9	1	-	0,6	1	0,6	-	0,7	0,84	0,65
11	1b	0	0,3	0,9	0	-	-	-	-	-	0,89	0,95	-
12	1b	0	0	1	0,8	-	-	-	-	-	-	1	0,39
13	1b	0	0	0,8	0,4	-	-	-	-	-	-	0,71	0,32
14	1b	0	0	1	0,7	-	-	-	-	-	-	0,75	0,33
15	1c	0	0,9	1	1	-	1	1	0	-	0,89	0,95	0,72
16	1c	0	1	0,9	1	-	1	1	0	-	0,82	0,94	0,65
17	1c	0	0,9	1	0,9	-	1	1	0	-	0,71	0,68	0,7
18	1c	0	0,8	1	1	-	1	1	0,5	-	0,79	0,88	0,56
19	2a	0	0,9	1	1					-	0,48	0,91	0,42
20	2a	0,1	1	1	1					0,03	0,8	0,85	0,61
21	2a	0	1	1	1					-	0,5	0,85	0,28
22	2a	0	0,1	1	0,2					-	0,02	0,98	0,06
23	2b	0	0,9	1	1					-	0,65	0,95	0,51
24	2b	0	0,9	1	1					-	0,53	0,98	0,3
25	2b	0	0,9	1	1					-	0,49	0,98	0,08
26	2b	0	0,6	1	0,3					-	0,19	0,88	0,03
27	2c	0	0,4	1	0,8					-	0,12	0,56	0,23
28	2c	0	0,7	0,9	0,9					-	0,04	0,34	0,11
29	2c	0	0,5	0,8	0,5					-	0,07	0,35	0,1
30	2c	0	0,4	1	0,1					-	0,09	0,56	0,02
31	2d	0	0,9	0,9	0,8					-	0,28	0,78	0,17
32	2d	0,3	1	1	0,8					0,05	0,39	0,81	0,13
33	2d	0,3	0,9	1	0,7					0,03	0,42	0,88	0,1
34	2d	0,1	0,8	1	0,9					0,02	0,44	0,99	0,16

Tabelle A.2-6: Ergebnisse der Kollisionstests ohne Personen, RANSAC-Intervall = 5.

Testfall	Testszenario	Warnungsrate				Rechtzeitig				Permanenz			
		BR	FB	LK	PD	BR	FB	LK	PD	BR	FB	LK	PD
1	1a	0	1	1	1	-	1	1	1	-	0,83	0,77	0,71
2	1a	0	0,9	1	0,9	-	1	0,8	1	-	0,79	0,76	0,86
3	1a	0	0,9	1	0,9	-	1	1	0,44	-	0,72	0,98	0,72
4	1a	0	1	0,9	0	-	0	0,89	-	-	0,91	0,74	-

## A Ergebnisse der Evaluation einzelner Algorithmen

5	1a	0	1	0,8	0,9	-	0,8	0,88	0,11	-	0,87	0,77	0,51
6	1a	0	1	1	0,8	-	1	1	0,5	-	0,86	0,85	0,64
7	1a	1	1	1	1	0	1	1	0,2	0,67	0,87	0,94	0,84
8	1a	0	1	1	1	-	0	1	0	-	0,71	0,95	0,38
9	1a	0	1	1	1	-	0,3	1	0	-	0,75	0,94	0,77
10	1a	0	1	0,9	0,9	-	0,3	0,89	0,44	-	0,69	0,43	0,76
11	1b	0	0,8	1	0	-	-	-	-	-	0,76	0,91	-
12	1b	0	0	1	0,5	-	-	-	-	-	-	0,85	0,57
13	1b	0	0	1	0,2	-	-	-	-	-	-	0,9	0,35
14	1b	0	0	1	0,2	-	-	-	-	-	-	0,9	0,38
15	1c	0	1	1	1	-	1	1	0	-	0,8	0,85	0,57
16	1c	0	1	0,9	1	-	1	1	0	-	0,77	0,89	0,72
17	1c	0	1	1	1	-	1	1	0	-	0,74	1	0,63
18	1c	0	0,9	1	1	-	1	1	0,3	-	0,76	0,9	0,55
19	2a	0	0,8	1	1					-	0,44	0,82	0,4
20	2a	0,1	1	1	1					0,07	0,63	0,85	0,62
21	2a	0	0,8	1	0,9					-	0,48	0,81	0,3
22	2a	0	0	1	0					-	-	0,78	-
23	2b	0	1	1	1					-	0,53	0,83	0,46
24	2b	0	0,7	1	1					-	0,35	0,9	0,31
25	2b	0	0,8	1	0,6					-	0,27	0,9	0,06
26	2b	0	0,1	1	0,4					-	0,08	0,91	0,02
27	2c	0	0,3	1	0,7					-	0,04	0,42	0,17
28	2c	0	0,4	1	0,8					-	0,02	0,41	0,11
29	2c	0	0,3	1	0,4					-	0,06	0,41	0,08
30	2c	0	0,3	1	0,1					-	0,04	0,56	0,03
31	2d	0	1	1	0,8					-	0,28	0,65	0,17
32	2d	0	1	1	0,9					-	0,42	0,87	0,11
33	2d	0,1	1	0,9	0,5					0,05	0,32	0,81	0,1
34	2d	0	0,9	1	0,9					-	0,35	0,94	0,1

### A.2.4 Variation der Auswertung – Mindestanzahl an Erkennungen pro Kollision

Tabelle A.2-7: Ergebnisse der Kollisionstests ohne Personenerkennung, Mindestanzahl an Kollisionen = 1.

Testfall	Testszenario	Warnungsrate			Rechtzeitig			Permanenz		
		FB	LK	PD	FB	LK	PD	FB	LK	PD
1	1a	1	0,78	1	1	1	1	1	0,89	0,99
2	1a	1	0,9	1	1	1	1	1	0,94	0,95
3	1a	1	1	1	1	1	1	0,95	0,93	0,77
4	1a	1	1	1	0,2	1	0,1	0,79	0,89	0,55
5	1a	1	1	0,9	1	1	0,89	0,88	0,94	0,64

6 1a	1	1	1	1	1	1	0,9	0,89	0,85
7 1a	1	1	1	1	1	1	0,77	0,95	0,86
8 1a	1	1	1	1	1	1	0,71	0,87	0,65
9 1a	1	1	0,9	1	1	0,89	0,81	0,85	0,73
10 1a	1	1	0,8	1	1	1	0,85	1	0,72
11 1b	1	1	0,8	-	-	-	0,84	0,72	0,2
12 1b	1	1	0,9	-	-	-	0,84	0,77	0,23
13 1b	0,9	1	1	-	-	-	0,56	0,99	0,39
14 1b	0,6	1	1	-	-	-	0,68	0,86	0,31
15 1c	1	1	1	1	1	1	0,95	0,9	0,73
16 1c	1	1	0,9	1	1	1	0,9	1	0,48
17 1c	1	1	1	1	1	1	0,84	1	0,69
18 1c	1	1	0,9	1	1	1	1	0,8	0,8
19 2a	1	1	1				0,67	0,79	0,65
20 2a	1	1	1				0,81	0,87	0,77
21 2a	1	1	0,9				0,76	0,84	0,45
22 2a	0,7	1	0,8				0,1	1	0,07
23 2b	1	1	1				0,43	0,94	0,73
24 2b	0,9	1	1				0,45	0,97	0,63
25 2b	1	1	1				0,36	1	0,31
26 2b	1	1	1				0,11	0,95	0,16
27 2c	1	1	1				0,31	0,71	0,31
28 2c	1	1	0,9				0,32	0,91	0,28
29 2c	0,9	1	0,9				0,3	0,89	0,21
30 2c	1	1	0,9				0,28	0,78	0,12
31 2d	1	1	1				0,5	0,91	0,35
32 2d	1	1	1				0,66	0,92	0,28
33 2d	1	1	0,9				0,64	0,85	0,18
34 2d	1	1	1				0,72	0,92	0,24

Tabelle A.2-8: Ergebnisse der Kollisionstests ohne Personenerkennung, Mindestanzahl an Kollisionen = 3.

Testfall	TestszENARIO	Warnungsrate			Rechtzeitig			Permanenz		
		FB	LK	PD	FB	LK	PD	FB	LK	PD
1 1a		0,67	1	1	1	1	1	0,84	0,89	0,93
2 1a		1	1	0,9	0,9	1	1	0,81	0,9	0,66
3 1a		1	1	1	1	1	1	0,96	0,88	0,87
4 1a		1	1	0,9	0	0,8	0	0,92	0,68	0,49
5 1a		1	0,8	1	0,8	0,88	0,3	0,82	0,73	0,56
6 1a		1	1	1	1	1	1	0,9	0,75	0,91
7 1a		1	1	1	1	1	1	0,95	0,95	0,81
8 1a		1	1	1	0,6	1	1	0,77	0,87	0,75
9 1a		1	1	1	1	1	1	0,8	1	0,86
10 1a		1	1	1	1	1	1	0,76	0,93	0,77
11 1b		1	1	0,5	-	-	-	0,92	0,85	0,23
12 1b		0,5	1	1	-	-	-	0,8	0,83	0,28

## A Ergebnisse der Evaluation einzelner Algorithmen

13	1b	0	0,9	0,7	-	-	-	-	0,85	0,28
14	1b	0	0,9	0,8	-	-	-	-	0,73	0,28
15	1c	1	1	0,9	1	1	0,89	0,9	0,86	0,62
16	1c	1	1	1	1	1	0,8	0,9	1	0,57
17	1c	1	1	1	1	1	1	0,85	1	0,56
18	1c	1	1	1	1	1	0,9	0,7	0,74	0,72
19	2a	1	1	1				0,79	0,74	0,62
20	2a	0,8	1	1				0,89	0,92	0,8
21	2a	0,9	1	1				0,6	0,88	0,56
22	2a	0	1	0,6				-	0,95	0,07
23	2b	0,9	1	1				0,46	0,62	0,75
24	2b	0,9	1	1				0,48	0,8	0,48
25	2b	1	1	1				0,48	1	0,26
26	2b	0,7	1	0,9				0,13	1	0,14
27	2c	1	1	1				0,27	0,69	0,24
28	2c	1	1	1				0,3	0,76	0,26
29	2c	1	1	0,9				0,26	0,88	0,18
30	2c	0,8	1	0,9				0,28	0,87	0,09
31	2d	0,9	1	1				0,51	0,9	0,3
32	2d	1	1	1				0,72	0,97	0,24
33	2d	0,9	1	0,9				0,61	0,93	0,18
34	2d	1	1	1				0,52	0,92	0,23

Tabelle A.2-9: Ergebnisse der Kollisionstests ohne Personenerkennung, Mindestanzahl an Kollisionen = 4.

Testfall	Testszenario	Warnungsrate			Rechtzeitig			Permanenz		
		FB	LK	PD	FB	LK	PD	FB	LK	PD
1	1a	0,89	1	1	1	1	1	0,75	0,72	0,71
2	1a	1	0,9	1	1	1	1	1	1	0,95
3	1a	1	1	1	1	1	0,9	1	0,78	0,93
4	1a	1	1	0,8	0	0,8	0	0,78	0,93	0,52
5	1a	1	1	1	0,7	0,8	0,3	0,84	0,75	0,65
6	1a	1	0,9	1	0,9	0,89	0,9	0,95	0,78	0,83
7	1a	1	1	1	1	1	1	0,88	0,95	0,66
8	1a	1	1	1	0,8	1	1	0,71	0,68	0,73
9	1a	1	0,9	0,9	1	1	1	0,75	0,63	0,74
10	1a	1	1	0,9	1	1	1	0,76	0,95	0,59
11	1b	0,1	1	0,4	-	-	-	0,8	0,97	0,34
12	1b	0	1	0,9	-	-	-	-	0,8	0,4
13	1b	0	1	0,9	-	-	-	-	0,68	0,39
14	1b	0	0,9	0,9	-	-	-	-	0,74	0,33
15	1c	1	0,9	1	1	1	0,7	0,85	0,88	0,62
16	1c	1	1	1	1	1	0,7	0,9	0,97	0,6
17	1c	1	1	1	1	1	0,8	0,81	0,94	0,67
18	1c	1	1	0,9	1	1	0,78	1	0,84	0,75
19	2a	1	1	1				0,91	0,88	0,57

20	2a	1	1	1	0,94	0,95	0,75
21	2a	0,9	1	1	0,85	1	0,59
22	2a	0,2	1	0,6	0,14	1	0,06
23	2b	1	1	1	0,56	0,96	0,71
24	2b	0,8	1	0,9	0,64	0,98	0,56
25	2b	1	1	1	0,61	0,85	0,28
26	2b	1	1	0,9	0,14	0,8	0,13
27	2c	1	1	1	0,3	0,74	0,2
28	2c	1	1	1	0,27	0,83	0,16
29	2c	0,8	1	0,9	0,21	0,83	0,15
30	2c	0,9	1	0,8	0,24	0,77	0,08
31	2d	0,9	1	1	0,55	0,79	0,27
32	2d	1	1	0,9	0,74	0,8	0,19
33	2d	1	1	0,9	0,76	0,91	0,17
34	2d	1	1	1	0,72	0,98	0,22

Tabelle A.2-10: Ergebnisse der Kollisionstests ohne Personenerkennung, Mindestanzahl an Kollisionen = 5.

Testfall	Testszenario	Warnungsrate			Rechtzeitig			Permanenz		
		FB	LK	PD	FB	LK	PD	FB	LK	PD
1	1a	0,89	1	0,89	1	1	1	0,69	0,89	0,73
2	1a	1	1	1	1	1	1	0,95	1	0,79
3	1a	0,9	1	0,9	0,89	1	0,89	0,96	0,79	0,56
4	1a	1	1	0,9	0	0,7	0	0,85	0,98	0,57
5	1a	1	1	1	0,5	1	0,3	0,99	0,96	0,64
6	1a	1	1	0,9	0,9	0,9	0,78	1	0,9	0,77
7	1a	1	1	1	1	1	1	0,81	0,9	0,76
8	1a	1	1	1	0,4	1	1	0,78	0,76	0,73
9	1a	1	1	1	1	1	1	0,67	0,96	0,73
10	1a	0,9	1	0,9	1	1	1	0,86	1	0,85
11	1b	0,8	1	0,4	-	-	-	0,7	0,81	0,48
12	1b	1	1	0,6	-	-	-	0,9	0,86	0,36
13	1b	1	1	0,9	-	-	-	0,82	0,91	0,27
14	1b	0,4	1	0,8	-	-	-	0,63	0,9	0,3
15	1c	1	0,9	1	1	1	0,4	0,85	0,77	0,63
16	1c	1	1	0,9	1	1	0,22	0,86	0,95	0,53
17	1c	0,7	1	0,8	1	1	0,62	0,93	0,95	0,52
18	1c	1	1	0,9	1	1	0,78	0,95	1	0,81
19	2a	1	1	1				0,9	0,92	0,65
20	2a	1	1	1				0,83	0,85	0,47
21	2a	1	1	1				0,98	0,81	0,45
22	2a	0,1	1	0,5				0,2	0,75	0,06
23	2b	1	1	1				0,56	0,95	0,67
24	2b	1	1	1				0,64	0,89	0,5
25	2b	1	1	1				0,56	0,83	0,21
26	2b	0,6	1	0,9				0,13	1	0,1



## A Ergebnisse der Evaluation einzelner Algorithmen

<b>27</b>	2c	1	1	1	0,24	0,81	0,21
<b>28</b>	2c	1	1	0,8	0,23	0,68	0,2
<b>29</b>	2c	0,9	1	0,9	0,22	0,72	0,13
<b>30</b>	2c	0,8	1	0,5	0,24	0,78	0,09
<b>31</b>	2d	1	1	0,9	0,55	0,85	0,31
<b>32</b>	2d	0,8	1	0,9	0,67	0,87	0,24
<b>33</b>	2d	0,9	1	0,8	0,72	0,91	0,15
<b>34</b>	2d	1	1	0,9	0,62	0,96	0,16

### A.2.5 Anwendung der optimalen Konfiguration für die Kollisionsgefahrerkennung

Tabelle A.2-11: Ergebnisse der Kollisionstests ohne Personenerkennung, optimierte Konfiguration.

Testfall	Testszenario	Warnungsrate			Rechtzeitig			Permanenz		
		FB	LK	PD	FB	LK	PD	FB	LK	PD
<b>1</b>	1a	1	1	1	1	1	1	0,72	0,84	0,88
<b>2</b>	1a	1	1	1	1	1	1	0,86	0,88	0,75
<b>3</b>	1a	1	1	1	1	1	1	0,9	0,95	0,95
<b>4</b>	1a	1	1	1	0,6	1	0,1	0,67	0,86	0,77
<b>5</b>	1a	1	1	1	1	1	1	0,9	0,94	0,85
<b>6</b>	1a	1	1	1	1	1	1	0,95	0,82	0,94
<b>7</b>	1a	1	1	1	1	1	1	0,9	0,95	0,91
<b>8</b>	1a	1	1	1	1	0,9	1	0,72	0,88	0,75
<b>9</b>	1a	1	1	1	1	1	1	0,87	0,87	0,84
<b>10</b>	1a	1	1	1	1	1	1	0,87	0,85	0,69
<b>11</b>	1b	0,1	1	1	-	-	-	0,09	0,76	0,34
<b>12</b>	1b	0	1	0,9	-	-	-	-	0,85	0,21
<b>13</b>	1b	0	0,9	1	-	-	-	-	0,81	0,14
<b>14</b>	1b	0	1	1	-	-	-	-	0,9	0,07
<b>15</b>	1c	1	1	1	1	1	1	0,72	0,92	0,61
<b>16</b>	1c	0,7	1	0,9	1	1	0,89	0,53	0,87	0,55
<b>17</b>	1c	1	1	1	1	1	1	0,86	0,8	0,69
<b>18</b>	1c	1	1	1	1	1	1	0,9	0,85	0,92
<b>19</b>	2a	1	1	1				0,69	0,91	0,64
<b>20</b>	2a	1	1	1				0,93	0,9	0,72
<b>21</b>	2a	1	1	1				0,74	0,85	0,6
<b>22</b>	2a	1	1	1				0,4	0,89	0,21
<b>23</b>	2b	1	1	1				0,82	0,92	0,64
<b>24</b>	2b	1	1	1				0,56	0,79	0,53
<b>25</b>	2b	1	1	1				0,47	0,91	0,44
<b>26</b>	2b	1	1	1				0,35	0,92	0,36
<b>27</b>	2c	1	1	1				0,32	0,57	0,14
<b>28</b>	2c	1	1	1				0,36	0,84	0,24
<b>29</b>	2c	1	1	0,8				0,31	0,74	0,28

<b>30</b>	2c	1	1	0,9	0,36	0,77	0,21
<b>31</b>	2d	1	1	1	0,6	0,89	0,11
<b>32</b>	2d	1	1	0,8	0,72	0,77	0,07
<b>33</b>	2d	1	1	0,9	0,74	0,72	0,08
<b>34</b>	2d	1	1	0,9	0,74	0,94	0,08

### A.2.6 Vergleich mit statischer Kollisionsgefahrerkennung

Tabelle A.2-12: Ergebnisse der Kollisionstests ohne Personenerkennung mit der statischen Kollisionsgefahrerkennung in Abhängigkeit der minimalen Distanz  $r$  [m].

Testfall	Testszenario	Warnungsrate					
		$r=0,5$	$r=1,0$	$r=1,5$	$r=2,0$	$r=2,5$	$r=3,0$
<b>1</b>	1a	1	1	1	1	1	1
<b>2</b>	1a	1	1	1	1	1	1
<b>3</b>	1a	1	1	1	1	1	1
<b>4</b>	1a	0	1	1	1	1	1
<b>5</b>	1a	0	1	1	1	1	1
<b>6</b>	1a	1	1	1	1	1	1
<b>7</b>	1a	1	1	1	1	1	1
<b>8</b>	1a	0,4	1	1	1	1	1
<b>9</b>	1a	0,9	1	1	1	1	1
<b>10</b>	1a	1	1	1	1	1	1
<b>11</b>	1b	1	1	1	1	1	1
<b>12</b>	1b	1	1	1	1	1	1
<b>13</b>	1b	1	1	1	1	1	1
<b>14</b>	1b	1	1	1	1	1	1
<b>15</b>	1c	1	1	1	1	1	1
<b>16</b>	1c	1	1	1	1	1	1
<b>17</b>	1c	1	1	1	1	1	1
<b>18</b>	1c	1	1	1	1	1	1
<b>19</b>	2a	0,6	1	1	1	1	1
<b>20</b>	2a	0,4	1	1	1	1	1
<b>21</b>	2a	0	1	1	1	1	1
<b>22</b>	2a	0	0,3	1	1	1	1
<b>23</b>	2b	0	1	1	1	1	1
<b>24</b>	2b	0	1	1	1	1	1
<b>25</b>	2b	0	1	1	1	1	1
<b>26</b>	2b	0	1	1	1	1	1
<b>27</b>	2c	1	1	1	1	1	1
<b>28</b>	2c	1	1	1	1	1	1
<b>29</b>	2c	1	1	1	1	1	1
<b>30</b>	2c	0,9	1	1	1	1	1
<b>31</b>	2d	1	1	1	1	1	1
<b>32</b>	2d	1	1	1	1	1	1
<b>33</b>	2d	1	1	1	1	1	1
<b>34</b>	2d	1	1	1	1	1	1

## A Ergebnisse der Evaluation einzelner Algorithmen

Rechtzeitig							
1 1a	0	0	1	1	1	1	1
2 1a	0	0	0	0,3	1	1	1
3 1a	0	0	0,1	0,3	0,5	0,9	0,9
4 1a	-	0	0	0	0,2	0,3	0,3
5 1a	-	0	0	0,1	0,3	0,4	0,4
6 1a	0	0	0	0,1	0,3	0,6	0,6
7 1a	0	0	0,3	0,5	0,9	1	1
8 1a	0	0	0,8	1	1	1	1
9 1a	0	0	0,6	1	1	1	1
10 1a	0	0	0,3	0,9	1	1	1
11 1b	-	-	-	-	-	-	-
12 1b	-	-	-	-	-	-	-
13 1b	-	-	-	-	-	-	-
14 1b	-	-	-	-	-	-	-
15 1c	0	0,3	1	1	1	1	1
16 1c	0	0,3	1	1	1	1	1
17 1c	0	0,2	1	1	1	1	1
18 1c	0	0,1	1	1	1	1	1
Permanenz							
1 1a	1	1	1	1	1	1	1
2 1a	0,99	1	1	0,96	1	1	1
3 1a	0,99	1	0,94	0,84	0,8	0,78	0,78
4 1a	-	0,51	0,73	0,79	0,77	0,8	0,8
5 1a	-	0,8	0,88	0,87	0,9	0,92	0,92
6 1a	1	1	1	0,96	0,89	0,87	0,87
7 1a	0,92	0,97	0,8	0,86	0,76	0,81	0,81
8 1a	0,14	0,99	0,53	0,87	0,9	0,91	0,91
9 1a	0,28	1	0,56	0,83	0,85	0,88	0,88
10 1a	0,35	1	0,74	0,79	0,81	0,85	0,85
11 1b	1	1	1	1	1	1	1
12 1b	1	1	1	1	1	1	1
13 1b	1	1	1	1	1	1	1
14 1b	1	1	1	1	1	1	1
15 1c	1	1	1	1	1	1	1
16 1c	1	1	1	1	1	1	1
17 1c	1	1	1	1	1	1	1
18 1c	1	1	1	1	1	1	1
19 2a	0,28	0,86	0,64	0,71	0,79	0,87	0,87
20 2a	0,28	0,94	0,93	0,96	1	1	1
21 2a	-	0,84	0,81	0,83	0,9	0,94	0,94
22 2a	-	0,49	0,56	0,93	0,97	0,99	0,99
23 2b	-	0,99	0,78	0,94	0,99	1	1
24 2b	-	0,96	0,96	0,8	0,96	1	1
25 2b	-	0,98	0,57	0,99	1	1	1
26 2b	-	0,88	0,78	0,9	0,99	0,99	0,99
27 2c	1	1	1	1	1	1	1
28 2c	0,98	1	1	1	1	1	1

---

<b>29</b>	2c	0,86	0,98	0,99	0,99	0,99	0,99
<b>30</b>	2c	0,37	0,98	0,98	0,99	0,99	1
<b>31</b>	2d	0,98	1	1	1	1	1
<b>32</b>	2d	0,95	1	1	1	1	1
<b>33</b>	2d	0,91	0,99	1	1	1	1
<b>34</b>	2d	0,81	1	1	1	1	1



# Anhang B Ergebnisse der Evaluation des Gesamtsystems

## B.1 Support-Vector-Machine

Tabelle B.1-1: Rohdaten zu den Kollisionstests mit Personen, Vergleich der OF-Algorithmen in Verbindung mit einem SVM bei  $fppi=0,01$ .

Testfall	TestszENARIO	Warnungsrate			Rechtzeitig			Permanenz		
		FB	LK	PD	FB	LK	PD	FB	LK	PD
35	3a	1	1	1	1	1	1	0,75	0,71	0,94
36	3a	1	1	1	1	1	1	0,96	0,88	0,93
37	3a	1	1	1	1	1	1	0,93	0,59	0,83
38	3b	1	1	1	1	1	1	0,84	0,69	0,87
39	3b	0,88	1	1	0,86	1	1	0,61	0,86	0,93
40	3b	1	1	1	1	0,9	0,9	0,9	0,87	0,64
41	3b	1	1	1	1	1	1	0,95	0,71	0,87
42	3c	1	1	1	-	-	-	0,41	0,98	0,67
43	3c	1	1	0,9	-	-	-	0,42	0,86	0,65
44	3c	1	1	1	-	-	-	0,44	0,71	0,42
45	3c	1	1	1	-	-	-	0,47	0,89	0,49

Tabelle B.1-2: Rohdaten zu den Kollisionstests mit Personen, Vergleich der OF-Algorithmen in Verbindung mit einem SVM bei  $fppi=0,1$ .

Testfall	TestszENARIO	Warnungsrate			Rechtzeitig			Permanenz		
		FB	LK	PD	FB	LK	PD	FB	LK	PD
35	3a	0,9	1	1	0,89	1	1	0,58	0,78	0,82
36	3a	1	1	1	0,9	1	1	0,65	1	0,82
37	3a	1	1	1	0,9	1	1	0,77	0,9	0,95
38	3b	1	1	1	0,9	1	1	0,84	0,99	0,96
39	3b	1	1	1	1	1	1	0,96	0,96	0,92
40	3b	1	1	1	1	1	1	0,99	1	0,89
41	3b	1	1	0,9	1	1	1	0,95	0,84	0,9
42	3c	1	1	1	-	-	-	0,44	0,9	0,65
43	3c	1	1	1	-	-	-	0,44	0,75	0,63
44	3c	1	1	1	-	-	-	0,34	0,95	0,61
45	3c	1	1	1	-	-	-	0,42	0,95	0,56

## B.2 Neuronales Netz

Tabelle B.2-1: Rohdaten zu den Kollisionstests mit Personen, Vergleich der OF-Algorithmen in Verbindung mit einem DL-Detektor bei  $fppi=0$ .

Testfall	Testszenario	Warnungsrate			Rechtzeitig			Permanenz		
		FB	LK	PD	FB	LK	PD	FB	LK	PD
35	3a	1	1	1	1	1	1	1	1	0,99
36	3a	1	1	1	1	1	1	1	1	0,99
37	3a	1	1	1	1	1	1	0,99	1	0,99
38	3b	1	1	1	1	1	1	0,98	1	0,98
39	3b	1	1	1	1	1	1	0,97	1	0,92
40	3b	1	1	1	1	1	1	1	1	0,91
41	3b	1	1	1	1	1	1	1	1	0,94
42	3c	1	1	1	-	-	-	0,48	1	0,61
43	3c	1	1	1	-	-	-	0,47	1	0,64
44	3c	1	1	1	-	-	-	0,48	1	0,65
45	3c	1	1	1	-	-	-	0,5	1	0,62

Tabelle B.2-2: Rohdaten zu den Kollisionstests mit Personen, Vergleich der OF-Algorithmen in Verbindung mit einem DL-Detektor bei  $fppi=0,01$ .

Testfall	Testszenario	Warnungsrate			Rechtzeitig			Permanenz		
		FB	LK	PD	FB	LK	PD	FB	LK	PD
35	3a	1	1	1	1	1	1	1	1	0,99
36	3a	1	1	1	1	1	1	1	1	0,99
37	3a	1	1	1	1	1	1	0,99	1	0,99
38	3b	1	1	1	1	1	1	0,98	1	0,97
39	3b	1	1	1	1	1	1	0,97	1	0,92
40	3b	1	1	1	1	1	1	1	1	0,91
41	3b	1	1	1	1	1	1	1	1	0,94
42	3c	1	1	1	-	-	-	0,48	1	0,66
43	3c	1	1	1	-	-	-	0,47	1	0,64
44	3c	1	1	1	-	-	-	0,48	1	0,56
45	3c	1	1	1	-	-	-	0,5	1	0,64

Tabelle B.2-3: Rohdaten zu den Kollisionstests mit Personen, Vergleich der OF-Algorithmen in Verbindung mit einem DL-Detektor bei  $fppi=0,1$ .

Testfall	Testszenario	Warnungsrate			Rechtzeitig			Permanenz		
		FB	LK	PD	FB	LK	PD	FB	LK	PD
35	3a	1	1	1	1	1	1	1	1	0,99
36	3a	1	1	1	1	1	1	1	1	0,99
37	3a	1	1	1	1	1	1	0,99	1	1
38	3b	1	1	1	1	1	1	0,98	1	0,98
39	3b	1	1	1	1	1	1	0,97	1	0,93

B.3 Keine Personenerkennung - Statische Kollisionsgefahrerkennung

40	3b	1	1	1	1	1	1	1	1	0,91
41	3b	1	1	1	1	1	1	1	1	0,93
42	3c	1	1	1	-	-	-	0,48	1	0,66
43	3c	1	1	1	-	-	-	0,47	1	0,64
44	3c	1	1	1	-	-	-	0,48	1	0,56
45	3c	1	1	1	-	-	-	0,5	1	0,67

### B.3 Keine Personenerkennung - Statische Kollisionsgefahrerkennung

Tabelle B.3-1: Rohdaten der Auswertung der statischen Kollisionsgefahrerkennung bei Kollisionen mit Personen in Abhängigkeit der minimalen Distanz  $r$  [m].

		r=0,5	r=1,0	r=1,5	r=2,0	r=2,5	r=3,0
<b>Testfall</b>	<b>TestszENARIO</b>	<b>Warnungsrate</b>					
35	3a	1	1	1	1	1	1
36	3a	1	1	1	1	1	1
37	3a	1	1	1	1	1	1
38	3b	0,6	1	1	1	1	1
39	3b	0	1	1	1	1	1
40	3b	0	1	1	1	1	1
41	3b	1	1	1	1	1	1
42	3c	1	1	1	1	1	1
43	3c	1	1	1	1	1	1
44	3c	1	1	1	1	1	1
45	3c	1	1	1	1	1	1
		0,7818182	1	1	1	1	1
		<b>Permanenz</b>					
35	3a	1	1	1	1	1	1
36	3a	0,95	1	0,87	1	1	1
37	3a	0,99	1	0,71	0,94	0,96	0,97
38	3b	0,92	0,99	0,44	0,96	0,98	0,99
39	3b	-	0,99	0,6	0,91	0,95	0,96
40	3b	-	1	0,72	0,94	0,96	0,97
41	3b	1	1	0,48	0,98	0,99	0,99
42	3c	1	1	0,76	0,94	0,94	0,95
43	3c	1	1	0,73	0,96	0,98	0,99
44	3c	1	1	0,75	0,95	0,97	0,98
45	3c	0,99	1	0,76	0,98	0,98	0,99
		0,9833333	0,9981818	0,7109091	0,96	0,9736364	0,9809091
		<b>Rechtzeitig</b>					
35	3a	0	0	1	1	1	1
36	3a	0	0	0,6	1	1	1
37	3a	0	0	0,6	1	1	1
38	3b	0	0	1	1	1	1
39	3b	-	0	0,62	1	1	1



## B Ergebnisse der Evaluation des Gesamtsystems

---

<b>40</b>	3b	-		0	0,6	1	1	1
<b>41</b>	3b		0	0	0,8	1	1	1
<b>42</b>	3c	-	-	-	-	-	-	-
<b>43</b>	3c	-	-	-	-	-	-	-
<b>44</b>	3c	-	-	-	-	-	-	-
<b>45</b>	3c	-	-	-	-	-	-	-
			0	0	0,7457143	1	1	1