

# **Masterarbeit**

zur Erlangung des akademischen Grades  
Master of Science (M.Sc.)

## **Entwicklung einer Bildverarbeitung mit dem Schwerpunkt Personenerkennung für ein autonomes Logistik-Fahrzeug**

**Autor:** Giuliano Montorio  
giuliano.montorio@hs-bochum.de  
Matrikelnummer: 015202887

**Erstgutachter:** Prof. Dr.-Ing. Arno Bergmann  
**Zweitgutachter:** M.Sc. Bernd Möllenbeck

**Abgabedatum:** tt.mm.jjjj

# Eidesstattliche Erklärung

## Eidesstattliche Erklärung zur Abschlussarbeit:

«Entwicklung einer Bildverarbeitung mit dem Schwerpunkt Personenerkennung für ein autonomes Logistik-Fahrzeug»

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Bochum, 31. August 2020

---

Ort, Datum

---

Giuliano Montorio

# Danksagung

Ein großes Dankeschön gilt all jenen Personen, die mich im Rahmen dieser Masterarbeit begleitet und geholfen haben. Insbesondere möchte ich Herrn Prof. Dr.-Ing. Arno Bergmann, Herrn Bernd Möllenbeck, M.Sc. und Herrn Dr.-Ing. Christoph Krimpmann danken, die unsere Arbeit durch ihre fachliche und persönliche Unterstützung begleitet haben. Auch beim Fachbereich Elektrotechnik und Informatik der Hochschule Bochum, insbesondere Herrn Dipl.-Ing. Thorsten Bartsch möchten wir uns bedanken. Für die Bereitstellung von Informationen und Dokumente sind wir Herrn Dennis Hotze, M.Sc. und der Smart Mechatronics GmbH sehr dankbar, ohne deren Hilfe und finanzielle Unterstützung dieses Projekt nicht möglich gewesen wäre.

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>iv</b>
<b>Symbolverzeichnis</b>	<b>v</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Zielsetzung des Projekts . . . . .	2
<b>2 Grundlagen</b>	<b>3</b>
2.1 Neuronale Netze . . . . .	3
2.1.1 Eigenschaften von neuronalen Netzen . . . . .	3
2.1.2 Lernprozess . . . . .	4
2.1.3 Unterscheidung von neuronalen Netzen . . . . .	5
2.1.4 Evaluation neuronaler Netze . . . . .	5
2.2 Objekterkennung . . . . .	5
2.2.1 Bildklassifikation . . . . .	6
2.2.2 Objektlokalisierung . . . . .	6
2.2.3 Objekterkennung durch neuronale Netze . . . . .	6
2.3 Zustandsautomat . . . . .	8
2.4 Bestimmung von Positionskoordinaten . . . . .	9
2.5 Schnittstelle zwischen ROS und Python . . . . .	10
<b>3 Konzeptionierung</b>	<b>11</b>
3.1 Anforderungserhebung mit CONSENS . . . . .	11
3.2 Konzept und Aufbau der Personenerkennung . . . . .	11
3.2.1 Wirkstruktur der Personenerkennung . . . . .	11
3.2.2 Auswahl und Training der verwendeten neuronalen Netze . . . . .	11
3.2.3 Schnittstelle zwischen Python und ROS . . . . .	11
3.2.4 Erstellung von Objektinformationen . . . . .	11

3.3 Funktionsweise des Gesamtsystems . . . . .	11
<b>4 Verifikation</b>	<b>14</b>
<b>5 Zusammenfassung und Ausblick</b>	<b>15</b>
<b>Quellenverzeichnis</b>	<b>16</b>
<b>A Anhang</b>	<b>17</b>
A.1 Abbildungen . . . . .	17
A.2 Inhalt Datenträger . . . . .	18

# Abkürzungsverzeichnis

<b>ALF</b>	Autonomes Logistik Fahrzeug
<b>BLDC</b>	Brushless Direct Current
<b>CAN</b>	Controlled Area Network
<b>CONSENS</b>	Conceptual Design Specification Technique for the Engineering of Complex Systems
<b>EPOS</b>	Entwicklungsplattform Ortsfrequenzfilter-Sensor
<b>FMEA</b>	Failure Mode and Effects Analysis
<b>LIDAR</b>	Light Detection and Ranging
<b>LTI</b>	Linear, zeitinvariantes System
<b>MCM</b>	MotorController Module
<b>RALF</b>	Regelung eines Autonomen Logistik Fahrzeugs
<b>ROS</b>	Robot Operating System
<b>RVIZ</b>	ROS Visualization
<b>SLAM</b>	Simultaneous Localization and Mapping
<b>TEB</b>	Timed Elastic Band
<b>TFEST</b>	Transfer Function Estimation
<b>URDF</b>	Unified Robot Description Format
<b>USBFS</b>	Universal Serial Bus Filesystem

# Symbolverzeichnis

Symbol	Bedeutung
$D$	Dämpfung der Übertragungsfunktion
$G$	Übertragungsfunktion
$\underline{H}$	Hilfsmatrix
$K_P$	Verstärkungsfaktor
$K_s$	Streckenverstärkung
$\mathcal{L}$	Laplace-Transformation
$M$	Momentanpol
$\mathcal{O}_{\mathcal{T}}$	Menge aller Odometriedaten
$P_a$	Skalierungsfaktor für manuellen Betrieb
$P_m$	Skalierungsfaktor für automatischen Betrieb
$R_c$	Circumscribed Radius
$R_i$	Inscribed Radius
$T$	Abklingzeitkonstante
$T_g$	Anstiegszeit
$T_n$	Nachstellzeit
$T_u$	Verzugszeit
$T_v$	Vorhaltezeit
$U$	Laplacetransformierte Eingangsgröße

Symbol	Bedeutung
$\mathcal{X}_{\mathcal{T}}$	Menge aller Positionsvektoren
$\mathcal{Z}_{\mathcal{T}}$	Menge aller Umgebungsmessungen
$Y$	Laplacetransformierte Ausgangsgröße
$\vec{a}$	Umrechnungsvektor
$a_i$	Koeffizienten der Differentialgleichung der Ausgangsgröße
$b_j$	Koeffizienten der Differentialgleichung der Eingangsgröße
$\vec{b}$	Allgemeines Bewegungsziel
$c$	Rotatorischer Bewegungsbefehl
$d_i$	Reelle Zahl
$f$	Cost Scaling Factor
$g$	Impulsantwort
$h$	Übergangsfunktion
$\vec{h}$	Hilfsvektor
$i$	imaginäre Einheit $i = \sqrt{-1}$
$j$	Komplexe Zahl $j = \sqrt{-1}$
$k$	Komplexe Zahl $k = \sqrt{-1}$
$m$	Karte der Umgebung
$m_i$	Landmarken
$o$	Odometrie
$p$	Wahrscheinlichkeitsfunktion
$\vec{p}$	Orientierungsvektor
$q$	Quaternion



Symbol	Bedeutung
$r$	Distanz
$\vec{r}_a$	Rotatorisches Bewegungsziel aus automatischen Betrieb
$\vec{r}_m$	Rotatorisches Bewegungsziel aus manuellen Betrieb
$s$	Komplexe Frequenz
$t$	Zeit
$\vec{t}_{ma}$	Translatorisches Bewegungsziel aus manuellen oder automatischen Betrieb
$u$	Eingangsgröße
$\hat{u}$	Sprunghöhe
$\vec{v}$	Geschwindigkeitsvektor
$w$	Führungsgröße
$\vec{x}_t$	Positionsvektor
$\vec{x}_0$	Startpositionsvektor
$y$	Ausgangsgröße eines Systems
$z$	Messwert der Umgebung
$\alpha$	Fahrtwinkel
$\beta$	Posenwinkel
$\delta$	Impulsfunktion
$\sigma$	Sprungfunktion

# 1 Einleitung

## 1.1 Motivation

Das Thema der künstlichen Intelligenz (KI) ist heutzutage allgegenwärtig. Smart Home Geräte wie Amazons Alexa, Siri der Firma Apple oder der Google Assistant gehören mittlerweile in jeden ... deutschen Haushalt und enthalten KI zur Spracherkennung. Derartige Technologien begleiten den Menschen jedoch nicht nur Zuhause sondern auch in der Transport- und Logistikbranche. Eine Potenzialanalyse zur künstlichen Intelligenz der Firma Sopra Steria zeigt, dass bereits im Jahr 2017 20% aller befragten Unternehmen solche Systeme einsetzten. 37% planten den zukünftigen Einsatz. Die Implementierung solcher Systeme hat Einfluss auf verschiedenste Eigenschaften der Wertschöpfungskette. Die Qualität des Fachprozesses wird mit ebenfalls steigender Geschwindigkeit erhöht. Die zur Logistikbranche gehörenden Transportfahrzeuge sind ebenfalls mit KI ausgestattet und sorgen so für weniger Arbeitsunfälle und eine schnellere, präzisere Abarbeitung der Logistikaufgaben.

Das Projekt dieser Masterarbeit wird praktisch am autonomen Logistikfahrzeug angewendet, das aus dem Labor für Antriebstechnik der Hochschule Bochum stammt. Die Idee des ALFs ist es ein Fahrzeug zu entwickeln, das nach seiner Fertigstellung Logistikaufgaben am Standort der Hochschule Bochum lösen soll. Der Entwicklungsprozess stellt sich aus diversen Bachelor- und Masterarbeiten zusammen, die sowohl Hardware, als auch Softwareimplementierungen vorsehen. Bisher wurden zwei Abschlussarbeiten inklusive der praktischen Anwendung am ALF geschrieben. M.Sc. Dennis Hotze und M.Sc. Dominik Eickmann entwickelten in ihrem Masterprojekt das Fahrzeug und konnten Fahraufgaben ferngesteuert und manuell erledigen. Während der darauffolgenden Bachelorarbeit wurde eine Schlupfkompensation entwickelt, die den Drift am Fahrzeug durch Eingabe von Umgebungsinformationen verhindert. Weiterhin wurden Funktionen entwickelt, um grundlegende und autonome Fahraufgaben zu lösen. Das autonome Logistikfahrzeug aus der vorangegangenen Bachelorarbeit dient auch in dieser Masterarbeit als

Versuchsplattform.

## 1.2 Zielsetzung des Projekts

Die Grundidee und Herausforderung dieser Masterarbeit ist die Interaktion zwischen Menschen und Roboter und der dadurch resultierenden Bedienung des Systems ohne Eingabegerät. Bisher wurden anzufahrende Posen per Mausklick eingegeben oder Fahrmodi manuell gewechselt. Der Informationsfluss wird hierbei rein visuell und akustisch passieren. Letzteres wird in der Masterarbeit von Herrn Dittmann behandelt und durch ausgewählte Schnittstellen mit diesem Projekt verknüpft, um ein Gesamtsystem zu bilden. Die visuelle Komponente kann im weiteren Entwicklungsprozess des Roboters für verschiedene Anwendungsbereiche genutzt werden. Ziel dieser Masterarbeit ist die Erkennung und Unterscheidung von Personen. Das System wird zwischen bekannten und unbekannten Personen unterscheiden können und Informationen aus den gegebenen Daten generieren, die in der weiteren Entwicklung nützlich sind.

## 2 Grundlagen

Für ein besseres Verständnis, der in Kapitel Konzept... angewandten Methoden, werden anbei die Grundlagen behandelt. Informationen zu der verwendeten Hard- und Software wurden bereits in der vorangegangenen Bachelorarbeit vermittelt. Aufgrund Anforderung A... ist während der praktischen Anwendung keine Änderung der Hardware vorgesehen.

### 2.1 Neuronale Netze

Das Neuronennetz des menschlichen Gehirns dient als Vorbild für künstliche, neuronale Netze (KNN). Diese werden heutzutage als Lösung diverser Anwendungsprobleme angewendet, in denen komplexe Strukturen und Muster aus großen Datenmengen erkannt werden sollen. Das in diesem Projekt zugrundeliegende Bildverarbeitungsproblem besitzt die beschriebenen Eigenschaften und eignet sich somit für den Einsatz zur Erkennung von Personen. Anders als bei den meisten programmierten Applikationen ist die Ausgabe von KNN's lediglich probabilistisch. Beim vorliegenden, autonomen Logistikfahrzeug werden zur Personenerkennung derartige neuronale Netze verwendet.

#### 2.1.1 Eigenschaften von neuronalen Netzen

Das Grundlage für die Eingabe in ein neuronales Netz ist die Skalierung der vorliegenden Daten auf eine definierte Größe. Diese wäre beispielsweise bei einem Anwendungsfall mit einer Audiospur die Frequenzspektren oder bei einem Bildverarbeitungsproblem die Pixel eines Bildes. Die skalierten Daten werden in einem Tensor gegeben der die Dimensionen der Eingabe hat. Somit unterteilt sich ein Bild in die drei Dimensionen, die Höhe, die Weite und die Farbwerte der Primärfarben pro Pixel.

/Bild Neuron

Der grundlegende Aufbau eines neuronalen Netzes besteht aus miteinander verbundenen Schichten, die häufig aus Neuronen bestehen. Jedes Neuron verarbeitet im wesentlichen einge-

hende Zahlenwerte und gibt diese aus. Hierbei wird eine gewichtete Summe gebildet, die dann auf eine Aktivierungsfunktion angewendet wird.

$$s = \sum_{j=1}^n w_{ij} x_j \quad (2.1)$$

Gleichung 2.1 zeigt das mathematische Modell der gewichteten Summe  $s$ . Das jeweilige Gewicht  $w$  wird mit dem Index  $j$  inkrementiert und mit dem dazugehörigen Eingang des Neurons  $i$  multipliziert. Alle Produkte werden aufsummiert und ergeben die gewichtete Summe. Es gibt verschiedene Varianten der Aktivierungsfunktion, die je nach Netzart zur Anwendung kommen können. In welchen Eigenschaften sich neuronale Netze unterscheiden können, wird in Kapitel... beschrieben. Häufig werden für Aktivierungsfunktion Schwellwertfunktionen angewendet.

/Bild neuronales netz

In Abbildung... ist Grundstruktur eines neuronalen Netzes veranschaulicht. Neuronen sind hier als Kreise dargestellt und bilden in den beispielhaft vertikal veranschaulichten Formationen einzelne Schichten. Hierbei wird zwischen Eingabe-, Zwischen- und Ausgabeschichten unterschieden. Die Eingabeschicht nimmt Informationen in Form von Daten auf und gibt diese an die erste Zwischenschicht weiter. Die Anzahl der Zwischenschichten, oder auch verdeckte Schichten, ist in der Anwendung der neuronalen Netzen variabel. Am rechten Bildrand ist die Ausgabeschicht gezeigt, die die entsprechende Ausgabe des Netzes generiert.

### 2.1.2 Lernprozess

Der Lernprozess von neuronalen Netzen zielt darauf hinaus, einer Netzstruktur ein gewünschtes Verhalten beizubringen. Genauer sollen die in Kapitel... beschriebenen Gewichte modifiziert werden. Dafür ist vor dem Lernprozess eine Trainingsmenge nötig. Im Falle einer Personenerkennung wäre beispielsweise ein Datensatz aus Bildern von Personen ein geeignete Trainingsmenge.

Zunächst wird zwischen drei Lernverfahren unterschieden, dem unüberwachten Lernen, dem bestärkenden Lernen und dem überwachten Lernen. Beim unüberwachten Lernen erkennt

das Netz selbst Muster und Klassen aus der eingegebenen Menge. Anders als beim unüberwachten Lernen, lernt das Netz beim bestärkten Lernen mit einer Rückmeldung. Diese enthält Informationen darüber, ob ein errechnetes Ergebnis einer Trainingseinheit richtig oder falsch ist. Das überwachte Lernen setzt eine Trainingsmenge voraus, die neben der Eingabedaten auch das dazugehörige korrekte Ergebnis enthält. So wird in der Vorwärtspropagation durch eine Eingabe eine entsprechende Ausgabe erzeugt und diese mit dem korrekten Ergebnis verglichen. Das KNN wird dann mithilfe des aus dem vorangegangenen Vergleich entstandenen Fehler korrigiert.[1]

Die meist genutzte Form des überwachten Lernens ist die Rückwärtspropagierung (engl. Backpropagation) oder Fehlerrückführung genannt [2]. Die mathematische Grundlage für dieses Lernverfahren sind Gradientenabstiegsverfahren. Durch die Fehlerrückführung werden Gewichte durch die Ausgabeschicht, die dann als Eingabeschicht interpretiert wird, mithilfe des Fehlervektors optimiert.[1]

### 2.1.3 Unterscheidung von neuronalen Netzen

/handbuch ki

### 2.1.4 Evaluation neuronaler Netze

Die Ausgabe von neuronalen Netzen ist probabilistisch und nicht vorhersehbar. Folglich bestehen diverse Metriken für Evaluationen, die derartige Systeme messbar machen.

## 2.2 Objekterkennung

Bei der visuellen Objekterkennung wird ein Objekt, das auf einem Bild gezeigt ist, mit einer gewissen Wahrscheinlichkeit inklusive der Position in der Abbildung erkannt. Die drei Abstraktionsebenen einer solchen Erkennung unterteilen sich in Bildklassifikation, Objektlokalisierung und semantische Segmentierung ...2014Bild. Letzteres kommt in dieser Arbeit nicht zur Anwendung und wird aus diesem Grund im Folgenden nicht behandelt. Weiterhin wird ebenfalls die Objekterkennung durch neuronale Netze betrachtet und in Kapitel... verglichen.

/Bild      Bildaufnahme      und      Digitalisierung→Segmentierung→Bildverarbeitung→  
>merkmalsextraktion→Klassifikation

### 2.2.1 Bildklassifikation

Die Bildklassifikation beschreibt eine Zuweisung von Objektkategorien zu einem gegebenen Bild. Mithilfe einer Merkmalsextraktion werden Merkmalsvektoren extrahiert und können so in einem Klassifikator berechnet werden. Ein gängiges Verfahren zur Merkmalsgewinnung ist das sogenannte Histogram of oriented gradients (HOG). Bei diesem Verfahren werden in einem Bild auftretende Intensitäten geprüft und so Kanten und Ecken als Histogramm gespeichert. Die Support Vektor Maschine ist ein typischer Funktionsapproximator für eine Objektklassifikation. Es handelt sich hierbei um ein Verfahren, das Klassen durch eine sogenannte Hyperebene voneinander trennt. Diese Methode hat sich vor allem aufgrund ihrer kurzen Rechenzeit durchgesetzt.

### 2.2.2 Objektlokalisierung

Wie in Kapitel... beschrieben, soll die Ausgabe einer Objekterkennung auch den Ort eines Objektes enthalten. Für jedes erkannte Objekt wird ein Rechteck in Form von Pixelkoordinaten erzeugt, das den Interessensbereich beschreibt.

### 2.2.3 Objekterkennung durch neuronale Netze

Die bisher besten Ergebnisse in der Bildverarbeitung im Zusammenspiel mit neuronalen Netzen wurden durch *Convolutional Neural Network* (CNN) ermöglicht [3]. Anders als bei den bereits erwähnten Methoden geschieht die Merkmalsextraktion hierbei innerhalb des Netzes. Derartige Netzwerke nutzen Faltung zur Verarbeitung der Eingangsdaten statt der üblichen Matrizenmultiplikation [3]. Grundsätzlich besteht der Aufbau eines CNNs aus der Klassifikation durch eine vorangegangene Merkmalsextraktion.

/bild merkmalsextraktion

Die Merkmalsextraktion setzen sich im grundlegenden Fall aus drei Stufen zusammen. Dabei kann sich die Zusammensetzung beliebig oft hintereinander wiederholen. Die erste, sogenannte Konvolutionsschicht führt Faltungsprozesse mit den Eingangsdaten durch [3]. Während der Konvolution, oder auch Faltung genannt, werden eingehende Daten in Filter, sogenannte Kernels, eingegeben. Diese extrahieren bestimmte Merkmale, dabei können sich die Filter je Schicht ändern. So können verschiedene Schichten diverse Merkmale extrahieren. Im zweiten Schritt wird eine nichtlineare Aktivierungsfunktion wie der *Rectified Linear Unit* (ReLU) Funktion

auf die Ausgangsdaten der Kovolutionsschicht angewendet. Beim Pooling werden eingehende Matrizen minimiert [3]. Gleichzeitig gehen die aussagekräftigsten Merkmale jedoch nicht verloren. In einigen Fällen wird die Zusammensetzung der drei Stufen als Kovolutionsschicht bezeichnet [3].

### /bild Klassifikation

Für die Klassifikation der Daten setzen sich die letzten Schichten meistens aus einer vollständig verbundenen Schicht und einer Softmax Schicht. Diese gibt einen  $K$ -dimensionalen Vektor aus, wobei  $K$  für die Anzahl der ausgegebenen Klassen steht. Durch die folgende Softmax-Schicht wird der Vektor in einem Zahlenbereich von Null bis Eins transformiert. Die Summe aller Elemente des Vektors ergeben 1. An dieser Stelle sind alle Daten vollständig bearbeitet und werden als Vektor aus dem CNN ausgegeben. Jedes Element wird als Konfidenz der jeweiligen Klasse interpretiert.

Die Artenvielfalt der CNNs ist sehr breit gefächert. Jedes Netzwerk unterscheidet sich in der jeweiligen Architektur der Schichten. Durch die Änderung verschiedener Parameter, beispielsweise bei der Faltung oder beim Pooling, können CNNs im Einsatz jeweils anders reagieren. Hierbei entscheidet man bei der Auswahl des Modells häufig unter den Gesichtspunkten Bearbeitungszeit, Genauigkeit und je nach Anwendungsfall spielt der Speicherplatz ebenfalls eine große Rolle. Die Anforderungen ... und ... beschreiben die Geschwindigkeit und die Genauigkeit der Personenerkennung. Außerdem soll laut Anforderung ... das neuronale Netz für eine mögliche Auslagerung auf einem eingebettetem System entwickelt werden. Hinsichtlich der genannten Aspekte wird im Zuge dieser Arbeit ein KNN mit der MobileNet Architektur und dem *Single Shot Detector* (SSD) Framework verwendet. Die Grundlage der Auswahl wird mithilfe der Paper *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications* [4] und *The Implementation of CNN-based Object Detector on ARM Embedded Platforms* [5] begründet.

Die *MobileNet* Architektur ermöglicht im Vergleich zu anderen Lösung zur Merkmalsextraktion den gesuchten Spagat zwischen Schnelligkeit, Genauigkeit und Größe des Netzes [4]. Das Hauptmerkmal von MobileNet ist der Aufbau der Konvolutionsschichten, der zu den genannten Eigenschaften in der Praxis führen [4] [5]. Vergleichsmessungen zu anderen Architekturen werden in den erwähnten Paper durchgeführt. Der *Single Shot Detector* dient als Klassifikator und zeichnet sich besonders durch seine Schnelligkeit aus [6]. Er erreicht eine höhere Genauigkeit



als andere state-of-the-art Lösungen wie zum Beispiel *Faster R-CNN* [7] und ist zudem dreimal schneller[6].

### 2.3 Zustandsautomat

Die Idee der Nutzung eines Zustandsautomaten oder auch endlicher Automat (EA) ergab sich im Laufe der Entwicklungsphase. In der in Kapitel ... erwähnten Bachelorarbeit werden diverse Modi beschrieben, die den Aufruf von unterschiedlichen ROS-Knoten voraussetzen. Aufgrund der Analogie zwischen den beschriebenen Modi und der Zustände eines Zustandsautomaten wird die Nutzung eines solchen Automaten begründet. Im Folgenden wird auf die Eigenschaften eines endlichen Automats eingegangen.

Im Allgemeinen geht es bei einem Zustandsautomaten um die Beschreibung der Zustände (engl. States) eines Objekts. Dabei stellt das Objekt meist das Gesamtsystem dar, etwa ein Getränkeautomat oder wie dieser Arbeit autonomes Fahrzeug. States sind durch Bedingungen verknüpft und lösen während sogenannter Ereignisse eine Transition aus, die den Wechsel des Zustands ausübt. Weiterhin bilden die Zustände in ihrer Gesamtheit den Lebenszyklus des Objekts. Ein Getränkeautomat befindet sich bekanntermaßen beim Eintreffen eines Kunden in einer Art Bereitschaft. Übertragen auf die Theorie eines Zustandsautomaten wäre dies ein Bereitschaftszustand. Die Auswahl des Getränks und die Eingabe des entsprechenden Geldbetrags können beispielhaft als Ereignisse interpretiert werden. Somit wird ein Transition durchgeführt und der Zustand der Getränkeausgabe wird losgetreten. Wurde das Getränk ausgegeben und entnommen, geschieht der Wechsel in den Bereitschaftszustand und der beschriebene Zyklus ist komplettiert.

Seit dem Bestehen der endlichen Automaten haben sich in der Praxis zwei Typen durchgesetzt. Mealy und Moore Automaten unterscheiden sich grundlegend in ihrem Verhalten und können durch folgende Gleichungen beschrieben werden.

$$\alpha_{t+1} = \phi(\zeta_t, \alpha_t) \text{ mit } t \in \mathbb{N} \quad (2.2)$$

$$\gamma_t = \psi(\zeta_t, \alpha_t) \text{ mit } t \in \mathbb{N} \quad (2.3)$$

In den Gleichung ... beschreibt  $\varphi$  die Transitionsfunktion und  $\psi$  die Ausgabefunktion des Mooreautomats. Die Transition steht in Abhängigkeit von  $\zeta_t$ , die aktuelle Eingabe, und  $\alpha_t$ , der aktuelle Zustand selbst. Mithilfe der Transitionsfunktion lässt sich der Zustand bestimmen, der im folgenden Zeitschritt  $t$  angestrebt werden soll. Der Ausgang des Moore Automaten wird durch die Ausgangsfunktion *psiup* berechnet. Diese hängt genau wie die Transitionsfunktion von der Eingabe und dem Zustand zum Zeitpunkt  $t$  ab.

$$\alpha_{t+1} = \phi(\zeta_t, \alpha_t) \text{ mit } t \in \mathbb{N} \quad (2.4)$$

$$\gamma_t = \psi(\alpha_t) \text{ mit } t \in \mathbb{N} \quad (2.5)$$

Beim Vergleich der beiden Gleichungen ... und ... stellt sich heraus, dass die Ausgangsfunktion *psi* in der Beschreibung des Verhaltens eines endlichen Automaten durch Moore lediglich vom Ausgang zum Zeitpunkt  $t$  abhängig ist.

Eine Unterkategorie der Finiten Automaten ist der Hierarchische Zustandsautomat. Die Besonderheit hierbei ist die Zusammensetzung aller vorangegangenen Zustände eines aktiven Zustands. Diese sind bei der hier beschriebenen hierarchisch aufgebauten Maschine nämlich ebenfalls aktiv. So besteht die Möglichkeit eines aufeinander aufbauenden Endzustands.

## 2.4 Bestimmung von Positionskoordinaten

Während der Durchführung autonomer Fahr- bzw. Logistikaufgaben können diverse Probleme auftreten, die eine erfolgreiche Bearbeitung verhindern können. Beispielsweise können Türen geschlossen sein oder Gegenstände die geplante Route blockieren. Da das ALF nicht über die technischen Möglichkeiten besitzt derartige Problemstellungen zu lösen, müssen Menschen Abhilfe schaffen. Für diese Zwecke ist die Kenntnis über die letzte Position der erfassten Personen relativ zur statischen Karte notwendig. Anstehende Fahraufgaben werden, bedingt durch das Vorgängerprojekt, mithilfe des Robot Operating Systems gelöst. Personen können folglich als Position in das ROS Netzwerk veröffentlicht. Dies ermöglicht dem Roboter die veröffentlichten

Positionen anzufahren. Die Eintragung der Position in die statische Karte setzt die Beschreibung der Position als Koordinaten voraus. Für die Bestimmung der Positionskordinaten wird ein zweidimensionales Bild und die dazugehörigen Tiefeninformationen genutzt. Die Koordinate  $x_{loc}$  beschreibt hier die longitudinale Entfernung von der Kamera zur Person. Eingehende laterale Distanzen werden durch die Koordinate  $y_{loc}$  dargestellt.

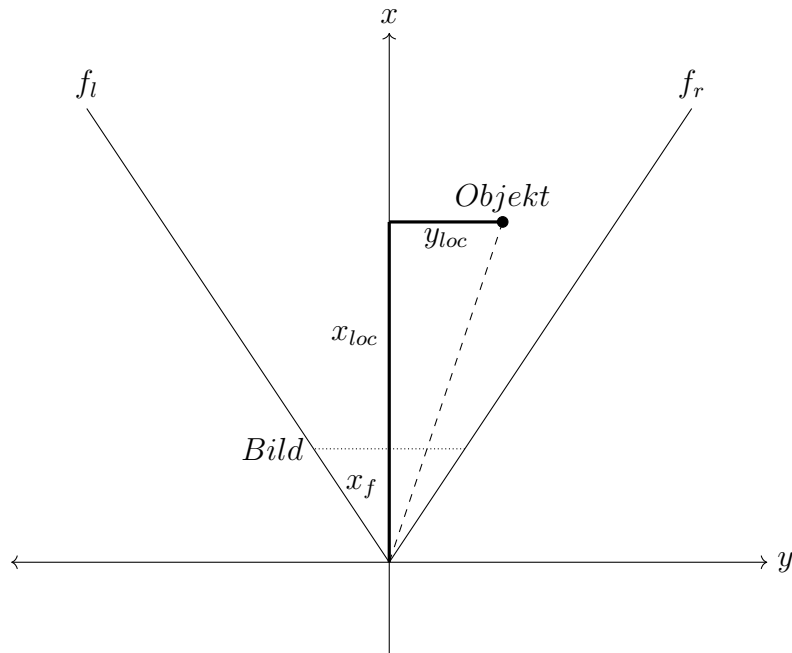


Abbildung 2.1: (a) Die Abbildung zeigt die Übergangsfunktion  $h(t)$  eines Verzögerungsglieds erster Ordnung als Antwort auf die sprunghörmige Eingangsgröße  $u(t)$  mit  $\hat{u} = 1$ . (b) Die Impulsantwort auf das Eingangssignal  $\delta(t)$ . Zwischen den Funktionen  $h(t)$  und  $g(t)$  besteht der Zusammenhang  $g(t) = \frac{d}{dt} h(t)$ .

## 2.5 Schnittstelle zwischen ROS und Python

# **3 Konzeptionierung**

## **3.1 Anforderungserhebung mit CONSENS**

## **3.2 Konzept und Aufbau der Personenerkennung**

### **3.2.1 Wirkstruktur der Personenerkennung**

### **3.2.2 Auswahl und Training der verwendeten neuronalen Netze**

### **3.2.3 Schnittstelle zwischen Python und ROS**

### **3.2.4 Erstellung von Objektinformationen**

## **3.3 Funktionsweise des Gesamtsystems**

Der Kern dieser Arbeit ist die Personenerkennung im praktischen Kontext des im Kapitel ... beschriebenen autonomen Logistikfahrzeugs. Aufgrund dessen sind einzelne Programme nicht als abgeschlossenes System zu betrachten. In Kapitel .../consens wurden bereits alle Schnittstellen zu verbauten Hardware- und Softwarekomponenten präsentiert. Das vollständige System der Personenerkennung und der Aufbau des entwickelten, endlichen Automats wird im folgenden Kapitel erklärt.

Die Personenerkennung am ALF wird mithilfe der Bildinformationen von zwei *Kinect*-Kameras betrieben. In Kapitel.../rospython wurden bereits zwei Lösungsansätze in der Softwareentwicklung in Zusammenspiel mit ROS und Python präsentiert. Aufgrund einer starken Belastung der im Roboter verbauten Recheneinheit während der parallelen Bildverarbeitung beider eingehender Bilder, wurde das Programm auf eine serielle Verarbeitung umgestellt. Der Befehl der parallelen Verarbeitung erzwingt Berechnungsprozesse mit derselben Frequenz, die durch den Eingang

der Bilder beider Kameras vorgegeben wird. Mithilfe der seriellen Abarbeitung war es ebenfalls möglich die Häufigkeit der Berechnungen zu steuern und damit verbundene Programmoptimierungen vorzunehmen. Auslastungen des verbauten Computers können so eingespart werden und für weitere, parallel laufende Prozesse genutzt werden. Dies wird zum Beispiel durch eine gezielte Verzögerung der Personenerkennung erreicht. Hierbei werden Pausen mit der gewünschten Dauer zwischen Bildverarbeitungsprozessen eingelegt bis ein relevantes Bild erkannt wird. Erst dann arbeitet die Personenerkennung mit der maximalen Geschwindigkeit. Als relevant werden Bilder eingestuft, die eine Person enthalten.



## **4 Verifikation**

## **5 Zusammenfassung und Ausblick**



# Quellenverzeichnis

- [1] Kriesel, D. *Ein kleiner Überblick über Neuronale Netze*. 2005.
- [2] Wolfgang Ertel. *Grundkurs Künstliche Intelligenz*. 3. Auflage. Springer Vieweg, 2013.
- [3] Goodfellow, I. , Bengio, Y. und Courville, A. *Deep Learning*. [http : / / www . deeplearningbook . org](http://www.deeplearningbook.org). MIT Press, 2016.
- [4] Howard, A. G. , Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. und Adam, H. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. Apr. 2017.
- [5] Zhang, Y., Bi, S., Dong, M. und Liu, Y. *The Implementation of CNN-based Object Detector of ARM Embedded Platforms. 2018 IEEE 16th Int. Conf. on Dependable, Autonomic & Secure Comp., 16th Int. Conf. on Pervasive Intelligence & Comp., 4th Int. Conf. on Big Data Intelligence & Comp., and 3rd Cyber Sci. and Tech. Cong.* 2018.
- [6] Liu, W. , Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y. und Berg, A. C. *SSD: Single Shot MultiBox Detector. European conference on computer vision*. pp. 21-37. 2016.
- [7] Ren, S. , He, K., Girshick, R. und Sun, J. *Faster R-CNN: Towards real-time object detection with region proposal networks. NIPS*. 2016.

# **A Anhang**

## **A.1 Abbildungen**

## **A.2 Inhalt Datenträger**

- 1** Datenblätter
- 2** Programm
- 3** Lastenheft