

DIPLOMARBEIT

Advanced Parking Monitoring (APM)



Durchgeführt von

Philipp Kraft

Dennis Köb

Samuel Bleiner

Betreuer

Dipl.-Ing. Christoph Stüttler

Abgabevermerk

Original, 05.04.2021

Digital, 05.04.2021

Dipl.-Ing. Christoph Stüttler

AV Dipl.-Ing. Leopold Moosbrugger

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.

Rankweil, 30. März 2021

Philipp Kraft

Dennis Köb

Samuel Bleiner

Kurzfassung

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Vorwort

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Danksagung

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris. Ein lesenswertes Buch ist »Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]«¹

¹Albert Einstein. »Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]«. In: *Annalen der Physik* 322.10 (1905), S. 891–921. doi: <http://dx.doi.org/10.1002/andp.19053221004>.

Inhaltsverzeichnis

Eidesstattliche Erklärung	i
Kurzfassung	ii
Abstract	iii
Vorwort	iv
Danksagung	v
1 Projektteam	8
2 Projektbetreuer	9
3 Auftragnehmer	10
4 Projektplanung	11
5 Rechtliches	12
6 Einleitung	13
7 Projektantrag	14
8 Kennzeichenerkennung	15
8.1 Anforderungen	15
8.2 Vorstudie	16
8.3 Bildverarbeitung	17
8.3.1 Einleitung	17
8.3.2 Bilaterale Filterung	17
8.3.3 Thresholding	18
8.3.4 Erosion	19
8.3.5 Farbraum	20
8.3.5.1 RGB	20
8.3.5.2 Graustufen	20

8.3.5.3	BGR	21
8.3.6	Konturerkennung	21
8.4	Kennzeichenerkennungsprogramm	22
8.4.1	Einleitung	22
8.4.2	Programmiersprache	22
8.4.3	Konzept	22
8.4.3.1	Bildaufnahme	22
8.4.3.2	Kennzeichenerfassung	23
8.4.3.3	Kennzeichensegmentierung	23
8.4.3.4	Zeichenerkennung	23
8.4.3.5	Anbindung an Datenbank	23
8.4.4	Ablauf	24
8.4.5	Verwendete Libraries	25
8.4.5.1	Versionsübersicht der verwendeten Libraries	25
8.4.5.2	Jupyter Notebook	26
8.4.5.3	Numpy	27
8.4.5.4	OpenCV	27
8.4.5.5	Matplotlib	29
8.4.5.6	Keras	30
8.4.5.7	Tensorflow	31
8.4.5.8	localutils	33
8.4.5.9	Scikit-learn	33
8.4.5.10	Requests	34
8.4.5.11	Gpiozero	34
8.4.5.12	Picamera	35
8.4.6	Wichtige Programmteile	36
8.5	Raspberry Pi	36
8.5.1	Einleitung	36
8.5.2	Wahl des Raspberry Pi	37
8.5.3	Kamera	37

9	Fahrzeugerkennung	39
9.1	Anforderungen	39
9.2	Vorstudie	39
9.3	Erkennung von Metallen über Spulen	39
9.3.1	Messung ferromagnetischer Metalle	42
9.3.1.1	RL-Oszillator mit Timer Baustein	43
9.3.2	Messung paramagnetischer Metalle	48
9.3.2.1	LC Oszillatoren	49
9.3.2.2	Messung der Frequenz eines Colpitts-Oszillator	50
9.4	RS485 Bussystem	51
9.4.1	Überblick	51
9.4.2	Elektrische Spezifikation	51
9.4.3	Implementation eines eigenen Protokolls	51
9.5	Mikrokontroller Slave-Geräte	51
9.5.1	Überblick	51
9.5.2	Atmega328PB	51
9.5.3	Peripherie des Mikrokontrollers	51
9.5.3.1	Spannungswandler	51
9.5.3.2	RS485 Pegelwandler	51
9.5.3.3	Digitale Ein- und Ausgänge	51
9.5.4	Layout des Slave-Gerätes	51
9.5.5	Gehäuse	51
9.6	USB-Master	51
9.6.1	USB-Bussadapter Gerät	51
9.6.1.1	Überblick	51
9.6.1.2	FT232RL	51
9.6.1.3	Spannungsversorgung	51
9.6.1.4	USB-C Anschluss	51
9.6.1.5	Layout des Master-Geräts	51
9.6.1.6	Gehäuse	52
9.6.2	Master Programm	52

9.6.2.1	Benötigte Software	52
9.6.2.2	Adressvergabe	52
9.6.2.3	Frequenzauslesung	52
9.6.2.4	Auswertung	52
9.6.2.5	API-Post	52
9.6.3	RaspberryPi als Mastergerät	52
9.6.3.1	SSH Remote Zugriff	52
9.6.3.2	Code Deployment	52
9.6.3.3	Unittest	53
10	Webinterface	53
10.1	Einleitung	53
10.2	Anforderungen	53
10.3	Verwendete Technologien	53
10.3.1	HTML	53
10.3.1.1	Beispielhafte HTML Seite	53
10.3.2	CSS	55
10.3.3	JavaScript	57
10.3.4	PHP	58
10.3.5	TailwindCSS	58
10.3.6	Laravel	59
10.3.6.1	Routing	59
10.3.6.2	Blade Templates	61
10.3.6.3	Controllers	61
10.3.6.4	Artisan CLI	62
10.3.6.5	Migrations	63
10.3.6.6	Eloquent ORM	65
10.3.6.7	Laravel Sanctum	65
10.4	Lokale Entwicklungsumgebung mit Laragon	66
10.4.1	Benötigte Software	66
10.4.2	Konfiguration von PHP	66

10.4.3	Installation von phpMyAdmin	69
10.5	Lokale Entwicklungsumgebung mit WSL und Docker	70
10.5.1	Benötigte Software	70
10.5.2	Installation von WSL	70
10.5.2.1	2. Schritt: Virtual Machine Aktivieren	70
10.5.2.2	3. Schritt: Linux Kernel Update	71
10.5.2.3	4. Schritt: WSL 2	71
10.5.2.4	5. Schritt: Linux Distribution herunterladen	71
10.5.3	Installation von Docker	71
10.6	Production Server	72
10.6.1	Benötigte Software	73
10.6.2	Installation des LAMP Stacks	73
10.6.2.1	Apache	73
10.6.2.2	PHP	74
10.6.2.3	MariaDB	74
10.6.2.4	phpMyAdmin	75
10.6.2.5	Webinterface Virtual Host	78
10.6.2.6	Installation von Composer	79
10.6.3	Deployment mit Github Actions	79
10.6.3.1	Git Setup	79
10.6.3.2	Deploy Script	80
10.6.3.3	Server Deploy Script	81
10.6.3.4	Github Action	82
10.7	Grundlegender Aufbau Frontend	84
10.7.1	Components	84
10.7.1.1	Anonymous Components	84
10.7.1.2	Components erstellen	84
10.7.1.3	Components verwenden	85
10.7.1.4	Attribute übergeben	85
10.7.2	Layouts	86
10.7.3	Navbar	90

10.7.4	Sidebar	92
10.7.5	Footer	94
10.8	Funktionen	94
10.8.1	Dashboard	94
10.8.2	Login- und Registersystem	94
10.8.3	News	94
10.8.4	Benutzerverwaltung	94
10.8.5	Rechte- und Rollenverwaltung	94
10.8.6	Parkplatzverwaltung	94
10.8.7	Kennzeichenverwaltung	94
10.8.8	Erkennungsverlauf	94
10.8.9	Seiten Einstellungen	94
10.8.10	API Schlüssel	94
10.8.11	Displays	94
10.8.12	Notifications	94
10.8.13	Profil	94
10.8.14	Lokalisierung	94
10.9	Performance und Sicherheit	94
10.9.1	Form Validation	94
10.9.2	Authentifizierung	94
10.9.3	Authentisierung	94
10.9.4	SQL Injection	94
10.9.5	Cross-Site-Scripting	94
10.9.6	Lighthouse	94
11	Zusammenfassung und Ausblick	95
12	Anhang	96
	Abbildungsverzeichnis	97
	Codeverzeichnis	100

Abkürzungsverzeichnis	102
Literaturverzeichnis	103

1 Projektteam

2 Projektbetreuer

3 Auftragnehmer

4 Projektplanung

5 Rechtliches

6 Einleitung

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

7 Projektantrag

8 Kennzeichenerkennung

8.1 Anforderungen

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consetetur.

8.2 Vorstudie

Lorem ipsum dolor sit amet, consetetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consetetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque

tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

8.3 Bildverarbeitung

8.3.1 Einleitung

Die Bildverarbeitung ist ein zentrales Thema in dieser Applikation für Kennzeichenerkennung. Sie wird für die Zeichensegmentierung verwendet, sowie für die Vorbereitung von Bildern für andere Algorithmen. Im Folgenden werden die verwendeten Bildverarbeitungsfunktionen aufgelistet und deren Funktionsweise erläutert.

8.3.2 Bilaterale Filterung

Bilaterale Filterung ist eine Methode für eine kantenerhaltende Weichzeichnung eines Bildes.

Bei der Berechnung für den Farbwert des Ausgabepixels werden die benachbarten Pixel nicht nur mit ihrer Entfernung gewichtet, sondern auch mit ihrem eigenen Farbwert. Dadurch können einzelne farbliche Ausreißer herausgefiltert werden. Dies ist vor allem in der Bildverarbeitung wichtig,

da dadurch die wichtigen Eigenschaften eines Bildes, wie zum Beispiel Kanten, erhalten bleiben und verarbeitet werden können, aber einzelne abweichende Pixel herausgefiltert werden wodurch unnötige Informationen entfernt werden.



Abbildung 1: Vor Bilateraler Filterung



Abbildung 2: Nach Bilateraler Filterung

In Abbildung 1 kann man ein Bild von verschiedenen Lebensmitteln sehen. Wenn man genau hinsieht erkennt man vor allem bei den Blättern im Hintergrund und beim Brot viele detailreiche Texturen. Diese Texturen haben keine wichtige Texturen und sind deswegen unnötig. Um die Bildverarbeitung zu vereinfachen wendet man deswegen die bilaterale Filterung auf dieses Bild an, um diese detailreichen Texturen zu vereinfachen. In Abbildung 2 sieht man das Bild nach der bilateralen Filterung. Wenn man hier dann wieder genauer auf die Blätter und das Brot sieht, erkennt man, dass die detailreichen Texturen weichgezeichnet wurden, aber die Kanten sind genauso gut erkennbar wie vor der Filterung.

8.3.3 Thresholding

Das Thresholding oder auch Schwellenwertverfahren wird in der Bildverarbeitung verwendet, um Bilder zu segmentieren. Aus einem Graubild kann dadurch ein Binäres Bild erzeugt werden.

Bei diesem Verfahren wird ein bestimmter Schwellwert (En.: Threshold) definiert, welcher mit den Grauwerten der einzelnen Pixel des Bildes verglichen wird. Wenn der Grauwert den Schwellwert überschreitet, wird dieser durch einen weißen Pixel ersetzt und wenn der Grauwert kleiner als der Schwellwert ist, wird dieser durch einen schwarzen Pixel ersetzt. Dadurch erhält man ein Bild welches nur noch zwei Farben hat, Schwarz und Weiß. Dies wird deswegen eingesetzt, da dadurch

viele Bildverarbeitungsalgorithmen schneller arbeiten und die Effizienz gesteigert wird.



Abbildung 3: Graustufenbild



Abbildung 4: Binäres Bild nach Thresholding

In Abbildung 3 sieht man ein solches Graubild welches nur verschiedenen Graustufen aufweist. In Abbildung 4 sieht man das Bild nach dem Thresholding. Hier kann man nur noch das Boot mit den Menschen erkennen. Dies ist nicht nur für schnellere Bildverarbeitungsalgorithmen wichtig, sondern wird auch zur Objekterkennung in Bildern verwendet.

Um den Schwellwert zu bestimmen kann man diesen entweder variieren bis das gewünschte Ergebnis erscheint oder man verwendet Methoden, welche den Schwellwert automatisch bestimmen. Eine der bekanntesten Methoden zur Schwellwertbestimmung ist die Methode von Otsu², welche mit dem Schwellenwert die Pixel in Vordergrund und Hintergrund unterteilt.

8.3.4 Erosion

Erosion ist eine Funktion der Bildverarbeitung und ist in die morphologische Bildverarbeitung einzuordnen. Diese beschäftigt sich primär mit der Verarbeitung von binären Bildern, welche man nach Thresholding erhält.

Erosion benötigt zwei Eingaben, das binäre Bild und einen Kernel. Der Kernel ist dabei die Angabe, nach welcher die Erosion durchgeführt wird. Der Kernel ist auch eine binäre Struktur, welche über jeden einzelnen Pixel des binären Bildes geschoben wird. Wenn der Kernel komplett mit

²Benannt nach Nobuyuki Otsu

dem binären Bild übereinstimmt, behält dieser Pixel seinen Wert und ansonsten wird er invertiert. Dabei muss jedoch darauf geachtet werden, dass die Polarität des binären Bildes und des Kernels übereinstimmt, da sonst die Erosion nicht richtig funktioniert. Als Resultat erhält man danach ein deutlicheres Bild bei welchem einzelne Pixelfehler herausgefiltert wurden und die Konturen besser erkennbar sind.



Abbildung 5: Binäres Bild nach Thresholding



Abbildung 6: Nach Erosion

In Abbildung 5 und 6 sieht man die Anwendung der Erosion. Die Konturen der einzelnen Zeichen im Kennzeichen sind in Abbildung 6 nach der Erosion deutlicher erkennbar als davor.

8.3.5 Farbraum

Der Farbraum eines Bildes enthält alle möglichen Farben eines Farbmodells. Das Farbmodell beschreibt dabei die Parameter, aus welchen die einzelnen Farben gebildet werden. Dies ist in der Bildverarbeitung relevant, da verschiedene Funktionen der Bildverarbeitung, unterschiedliche Farbräume verwenden und dieser deswegen korrekt eingestellt werden muss.

In dieser spezifischen Applikation werden die folgenden Farbräume verwendet:

8.3.5.1 RGB

RGB ist einer der häufigsten und bekanntesten Farbräume. Er basiert auf den drei Grundfarben Rot, Grün und Blau und wird vor allem bei Bildschirmen und in der Fotografie genutzt. Die Farben setzen sich in diesem Modell aus dem jeweiligen Rot-, Grün- und Blauanteil der einzelnen Pixel zusammen.

8.3.5.2 Graustufen

Bei einem Graustufen-Bild, zu sehen in Abbildung 3, hat jeder Pixel einen Wert von 0 bis 255. Diese

Werte erstrecken sich also von Schwarz bis Weiß und dazwischen liegen verschiedene Grautöne. Dieser Farbraum wird in der Bildverarbeitung häufig verwendet, da Konturen einfacher erkennbar sind und es nur einen Parameter gibt, welcher verarbeitet werden muss, wodurch die Effizienz diverser Algorithmen gesteigert werden kann. Zudem wird dieser Farbraum auch oft in Verbindung mit Thresholding verwendet.

8.3.5.3 BGR

Der BGR ist ein relativ unbekannter und wenig verwendeter Farbraum, da er sehr ähnlich zum RGB-Farbraum ist. Der einzige Unterschied zwischen diesen beiden liegt in der Anordnung der Parameter. Bei BGR sind die Parameter spiegelverkehrt zu RGB, das heißt es kommt zuerst der Blauanteil, dann der Grünanteil und zum Schluss der Rotanteil. Insgesamt ergibt dies für die einzelnen Pixel zwar die gleichen Farben, aber die Funktionen der Bildverarbeitung müssen trotzdem das Bild im passenden Farbraum erhalten. So verwendet zum Beispiel die Funktion „imread“ von OpenCV den BGR-Farbraum und die Funktion „imshow“ von Matplotlib verwendet den RGB-Farbraum. Wenn man diese Funktionen also nacheinander anwendet, muss dazwischen der Farbraum umgewandelt werden.

8.3.6 Konturerkennung

Die Konturerkennung ist eine wichtige Funktion in der Bildverarbeitung mit welcher Objekte in einem Bild gefunden werden können. In dieser Applikation wird sie für die Zeichensegmentierung eingesetzt.

Die Konturerkennung wird hauptsächlich bei binären Bildern verwendet. Eine Kontur kann dabei wie im Folgenden definiert werden. Man überprüft jeden einzelnen Pixel und sieht nach, ob ein benachbarter Pixel einen anderen Farbwert aufweist. Falls dies zutrifft muss der zu prüfende Pixel zu einer Kontur gehören. Wenn dies auf mehrere zusammenhängende Pixel zutrifft, bedeutet das, dass diese zusammen eine Kontur bilden.

Die Funktion „findcontours“ von OpenCV, welche in dieser Applikation verwendet wird, ist eine Funktion für Konturerkennung und kann weiße Objekte auf einem schwarzen Hintergrund erkennen.

Sie basiert auf dem Algorithmus von Suzuki von 1985³ und liefert eine Liste mit allen Konturen. Die Konturen werden in der Liste als ein Array von Koordinaten abgespeichert.

8.4 Kennzeichenerkennungsprogramm

8.4.1 Einleitung

Die Software ist der wichtigste und größte Teil der Kennzeichenerkennung. Sie erhält ein Bild, in welchem ein Auto mit einem Kennzeichen enthalten ist und liefert am Ende dieses Kennzeichen und sendet dieses dann automatisch an die Datenbank. Die Software kann entweder über Bildverarbeitung oder mit Machine Learning Modellen realisiert werden. Der erste Ansatz bei dieser Anwendung war mit klassischer Bildverarbeitung, welche aber nicht die gewünschte Genauigkeit erreicht hat, weswegen dann auf Machine Learning gewechselt wurde.

8.4.2 Programmiersprache

Die verwendete Programmiersprache für die Kennzeichenerkennung ist Python. Python ist eine höhere Programmiersprache, welche übersichtlich und leicht lesbar ist. Sie ist vor allem für Bildverarbeitung und Machine Learning Anwendungen gut geeignet, da es dafür hoch optimierte und effiziente Bibliotheken gibt wie zum Beispiel OpenCV, Numpy und Tensorflow. Dadurch ist Python für diese Anwendung besser geeignet als zum Beispiel C++. Dieses wäre zwar normalerweise effizienter, bietet aber weniger optimierte Bibliotheken in diesem Bereich, wodurch es hier weniger gut geeignet ist.

8.4.3 Konzept

Das Programm für die Kennzeichenerkennung basiert auf fünf Stufen. Die erste Stufe ist die Bildaufnahme, die zweite ist die Kennzeichenerfassung mittels Machine Learning, die dritte ist die Kennzeichensegmentierung mithilfe von Bildverarbeitung, die vierte ist die Zeichenerkennung mittels Machine Learning und die fünfte ist die Anbindung an die Datenbank.

8.4.3.1 Bildaufnahme

Um ein Bild verarbeiten zu können und aus diesem ein Kennzeichen auslesen zu können, muss

³Topological structural analysis of digitized binary images by border following

zuerst ein Bild vorliegen. Dieses wird über den RaspberryPi mit der RaspberryPi-Kamera aufgenommen. Um das Bild aufzunehmen, muss einfach ein Auslöser aktiviert werden und dann wird das Bild aufgenommen und im richtigen Ordner abgespeichert. Zuvor wird noch überprüft ob sich in diesem Ordner bereits ein Bild befindet und falls eines vorhanden ist wird es gelöscht. Dadurch werden mögliche Fehler durch mehrere Bilder verhindert.

8.4.3.2 Kennzeichenerfassung

Die Kennzeichenerfassung hat die Aufgabe, das Kennzeichen im Eingabebild zu lokalisieren. Dies geschieht mittels Machine Learning mit dem Modul WPOD-NET von Sérgio Montazolli Silva und Cláudio Rosita Jung⁴. Dieses verwendet zuerst das Modul YOLOv2 welches zur Echtzeitobjekterkennung verwendet werden kann und in dieser Anwendung zur Erkennung von Fahrzeugen verwendet wird. Danach werden die Koordinaten des Kennzeichens ermittelt und dieses aus dem Bild ausgeschnitten und abgespeichert.

8.4.3.3 Kennzeichensegmentierung

Die Kennzeichensegmentierung hat das Ziel die einzelnen Zeichen im Kennzeichen zu separieren und so zu vorbereiten, dass die darauffolgende Zeichenerkennung damit arbeiten kann. Dazu wird das Bild mit dem Kennzeichen zuerst in Graustufen konvertiert, dann mit einem bilateralen Filter gefiltert, mit Thresholding in ein binäres Bild umgewandelt und anschließend mittels Erosion besser erkennbar gemacht. Danach werden im verarbeiteten Bild die Konturen gesucht, sortiert und anhand dieser die einzelnen Zeichen herausgefiltert.

8.4.3.4 Zeichenerkennung

Die letzte Stufe der Kennzeichenerkennung ist die Zeichenerkennung. In dieser werden die einzelnen Zeichen erkannt und als Text abgespeichert. Dies funktioniert über eine eigenes Neuronales Netz basierend auf MobileNetV2⁵, welches mit einem Datensatz von über 35 000 Bildern auf die Erkennung von Zeichen aus Bildern trainiert wurde.

8.4.3.5 Anbindung an Datenbank

Nachdem das Kennzeichen als Text abgespeichert wurde, muss diese Information in die Daten-

⁴License Plate Detection and Recognition in Unconstrained Scenarios

⁵Neuronales Netz für Computer Vision

bank übergeben werden. Dazu wird die eigene API angewandt, welcher man diese Informationen übergeben muss und als Rückgabe die Information bekommt, ob sich das Fahrzeug nun innerhalb oder außerhalb des Parkplatzes befindet.

8.4.4 Ablauf

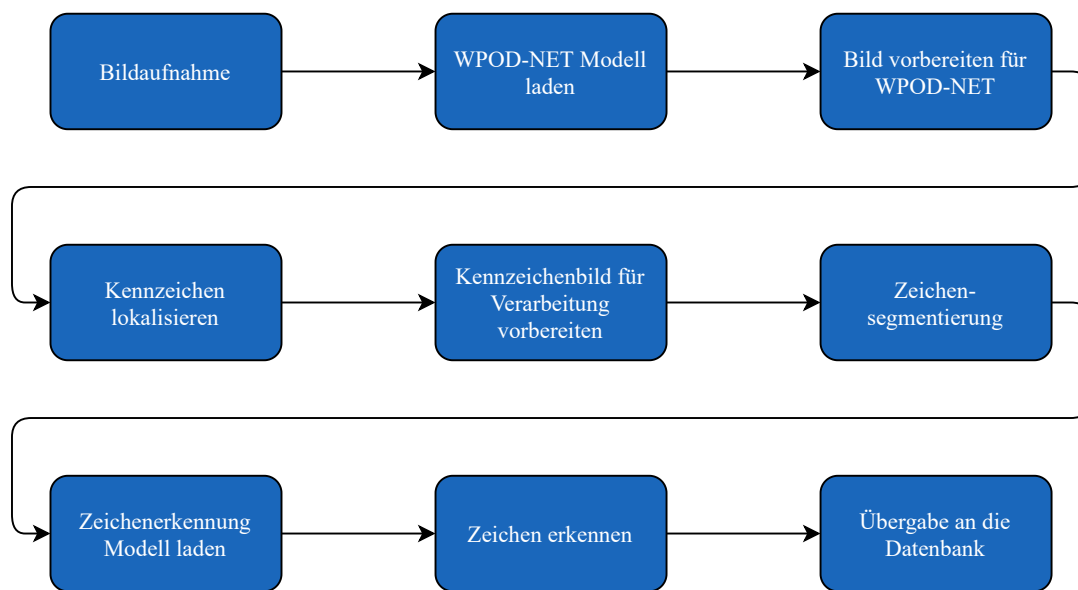


Abbildung 7: Ablaufdiagramm der Kennzeichenerkennung

Im oberen Diagramm ist der Ablauf des Programms angegeben. Zuerst wird mit einem Button der Auslöser betätigt und damit das Foto aufgenommen. Dann wird das erste Machine Learning Modell für die Kennzeichenerkennung „WPOD-NET“⁶ geladen. Bevor das Bild diesem Modell übergeben werden kann, muss es noch angepasst werden damit das Modell damit arbeiten kann. Danach kann damit das Kennzeichen im Bild lokalisiert werden. Im Anschluss wird dieses Kennzeichenbild mit mehreren Bildverarbeitungsalgorithmen verarbeitet, um dann die einzelnen Zeichen zu segmentieren. Danach kann dann das Modell für die Zeichenerkennung geladen werden und dieses dann auch angewendet werden, um das Ergebnis zu erhalten. Dieses Ergebnis wird dann noch mit einer API an die Datenbank übergeben.

⁶Machine Learning Modell für Kennzeichenerfassung

8.4.5 Verwendete Libraries

8.4.5.1 Versionsübersicht der verwendeten Libraries

Im Folgenden werden die verwendeten Versionen der benötigten Libraries aufgelistet, um das Programm zu starten. Dies wird noch einmal unterteilt in die Versionen, welche auf Windows 10 benötigt werden und jene auf Raspbian⁷, da auf Raspbian manche Versionen noch nicht verfügbar sind, neuere und verbesserte Versionen verfügbar sind oder manche Versionen nur auf komplizierten Umwegen installierbar sind.

Windows:

- h5py = 2.10.0
- imutils = 0.5.3
- Keras = 2.4.3
- matplotlib = 3.3.2
- notebook = 6.1.5
- numpy = 1.18.5
- opencv-python = 4.4.0.44
- scikit-learn = 0.23.2
- tensorflow = 2.3.1
- requests = 2.24.0

Raspbian:

- h5py = 2.10.0
- imutils = 0.5.3
- Keras = 2.4.3
- matplotlib = 3.3.3

⁷Raspbian ist ein Unix-basiertes Betriebssystem für den Raspberry Pi

- notebook = 6.1.5
- numpy = 1.19.4
- opencv-python = 4.4.0.46
- scikit-learn = 0.24.0
- tensorflow = 2.4.0
- requests = 2.21.0

8.4.5.2 Jupyter Notebook

Jupyter Notebook ist eine nützliche Erweiterung für Python, wenn es um den Bereich der Daten-Visualisierung und Machine Learning geht. Sie ist eine Unteranwendung des Open-Source Projektes „Project Jupyter“, welches von großen Partnern wie Microsoft und Google unterstützt und verwendet wird. Mit Jupyter Notebook ist es möglich in einer Web-Anwendung ein Live-Script abzuarbeiten und in Kombination mit Matplotlib die Daten visuell darzustellen, abzuspeichern und zu überwachen. Im Gegensatz zur direkten Darstellung in der Entwicklungsumgebung, wird der letzte Durchgang des Scripts, inklusive aller Bilder und Grafiken gespeichert. Zudem ist es auch hervorragend für das Training und die dazugehörige Überwachung von Machine Learning Modulen geeignet. Außerdem bietet Jupyter Notebook noch die Möglichkeit mehrere Scripts parallel abzuarbeiten und diese individuell zu überwachen. In dieser Applikation wird es für ein solches Training und für die anschauliche Visualisierung von Daten genutzt.

Die Installation von Jupyter Notebook ist simpel, da man es einfach über pip⁸ mit dem folgenden Befehl installieren kann:

```
1 pip install notebook
```

Code 1: PIP Installation von Jupyter Notebook

Um das Jupyter Notebook aufzurufen muss im Terminal „jupyter notebook“ eingegeben werden und dann öffnet sich automatisch die Web-Applikation. Danach hat man Zugriff auf komplette

⁸Paketverwaltungsprogramm für Python

Dateistruktur des eigenen Projektes und kann die einzelnen Python-Scripts starten. Dabei ist zu beachten, dass man keine gewöhnliche .py-Datei benötigt, sondern eine .ipynb-Datei benötigt, wozu man einfach eine Kopie des normalen Scripts erstellt und die Dateiendung ändert.

8.4.5.3 Numpy

Numpy ist eine weit verbreitete Library für Python, welche sich mit der Berechnung und Verarbeitung von mehrdimensionalen Arrays beschäftigt. Damit können komplexe Berechnungen und Algorithmen oft effizienter verarbeitet werden, wobei aber beachtet werden muss, dass der Code an die mehrdimensionalen Arrays angepasst werden muss. Numpy selbst ist auch eine Voraussetzung für OpenCV, welches Numpy-Arrays verwendet, um Bilder zu verarbeiten. Dazu werden 3-dimensionale Arrays verwendet, wodurch die Verarbeitung dieser Bilder und die Bildverarbeitungsalgorithmen schneller sind. In dieser Applikation wird Numpy in Zusammenarbeit mit OpenCV verwendet und auch für die weitere Verarbeitung der Daten von OpenCV, um zum Beispiel die Koordinaten des Kennzeichens zu speichern.

Um Numpy zu installieren muss folgender Befehl in der Konsole eingegeben werden:

```
1 pip install numpy
```

Code 2: PIP Installation von Numpy

8.4.5.4 OpenCV

OpenCV ist die wichtigste und mächtigste Library die in dieser Applikation verwendet wird. Sie wurde ursprünglich in C++ geschrieben, aber es gibt auch eine Version in Python, welche in dieser Applikation verwendet wird. OpenCV ist eine freie Library für Bildverarbeitung, Computer Vision⁹ und Machine Learning, welche von Intel gestartet wurde und mittlerweile die wichtigste und am weitesten verbreitete Library in diesem Bereich ist. Sie verwendet Numpy-Arrays, um für eine effiziente Verarbeitung von Bildern zu sorgen. Einige der wichtigsten Funktionen stellen die klassischen Bildverarbeitungsalgorithmen wie zum Beispiel Filterung und Farbraumanpassungen, die Verarbeitung und Auswertung von Kamerabildern und auch die Kompatibilität mit Deep-Learning¹⁰ dar. In

⁹Die Computergestützte Auswertung und Verarbeitung von Kamerabildern

¹⁰Machine Learning Methode welche Neuronale Netze nutzt

dieser Applikation wird OpenCV für diverse Bildverarbeitungsalgorithmen, die Zusammenarbeit mit Machine Learning Modulen, die Verarbeitung eines Kamerabildes und das allgemeine Arbeiten mit Bildern verwendet.

Die Installation von OpenCV ist dabei etwas komplizierter als bei anderen Libraries.

Windows:

Unter Windows ist die Installation vergleichbar mit anderen Libraries, es muss einfach der folgende Befehl in der Konsole eingegeben werden:

```
1 pip install opencv-python
```

Code 3: PIP Installation von OpenCV

Raspbian:

Auf dem Raspberry Pi gestaltet sich die Installation von OpenCV um einiges schwieriger, da die Installation nicht über pip gemacht werden kann, sondern es manuell kompiliert werden muss.

Als erstes werden alle Tools und Bibliotheken installiert, welche für OpenCV benötigt werden. Dazu verwendet man folgenden Befehl im Terminal:

```
1 $ sudo apt-get install build-essential git cmake pkg-config libjpeg8-dev  
  ↳ libtiff4-dev libjasper-dev libpng12-dev libavcodec-dev  
  ↳ libavformat-dev libswscale-dev libv4l-dev libgtk2.0-dev  
  ↳ libatlas-base-dev gfortran
```

Code 4: Installation benötigter Tools und Bibliotheken für OpenCV

Danach kann man mit dem nächsten Befehl OpenCV von einem GitHub-Repository klonen:

```
1 $ sudo apt-get install build-essential git cmake pkg-config libjpeg8-dev  
  ↳ libtiff4-dev libjasper-dev libpng12-dev libavcodec-dev  
  ↳ libavformat-dev libswscale-dev libv4l-dev libgtk2.0-dev  
  ↳ libatlas-base-dev gfortran
```

Code 5: Klonen von OpenCV von GitHub

Im nächsten Schritt wird OpenCV mit den folgenden Befehlen kompiliert:

```
1 cd ~/opencv && mkdir build && cd build  
2  
3 cmake -D CMAKE_BUILD_TYPE=RELEASE \  
4 -D CMAKE_INSTALL_PREFIX=/usr/local \  
5 -D INSTALL_PYTHON_EXAMPLES=ON \  
6 -D INSTALL_C_EXAMPLES=ON \  
7 -D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules \  
8 -D BUILD_EXAMPLES=ON ..  
9 make -j4
```

Code 6: Kompilieren von OpenCV

Wenn das Kompilieren erfolgreich beendet wurde, kann OpenCV abschließend installiert werden.

```
1 $ sudo make install && sudo ldconfig
```

Code 7: Abschließende Installation von OpenCV

Danach sollte OpenCV fertig installiert und eingerichtet sein und es kann in einem Python Projekt verwendet werden.

8.4.5.5 Matplotlib

Matplotlib ist eine Python-Library mit welcher Daten und Berechnungen visuell dargestellt werden

können. Damit können statische, animierte und interaktive Diagramme erstellt werden, was die Datenauswertung um einiges erleichtert. Die Syntax ähnelt sehr stark jener von MATLAB, wodurch die Bedienung sehr einfach ist, wenn man schon Erfahrung mit MATLAB hat. Wie auch in MATLAB kann man die Diagramme beliebig anordnen und dadurch das Layout der Darstellung selbst festlegen. Es funktioniert auch hervorragend in Zusammenarbeit mit Jupyter Notebook, wodurch man die Daten in einer Web-Applikation visualisieren kann. Die visuelle Darstellung von Daten ist vor allem in der Entwicklung und beim Arbeiten mit visuellen Ergebnissen wie Bildern von Vorteil. In dieser Applikation wird Matplotlib für die Darstellung der Ergebnisse von Bildverarbeitungsalgorithmen, die Ausgabe von Daten und Resultaten und auch für Test- und Entwicklungszwecke verwendet. Im unteren Bild kann man ein Beispiel sehen bei welchem Matplotlib verwendet wird, um einige Bilder mit einem Titel anzuzeigen.

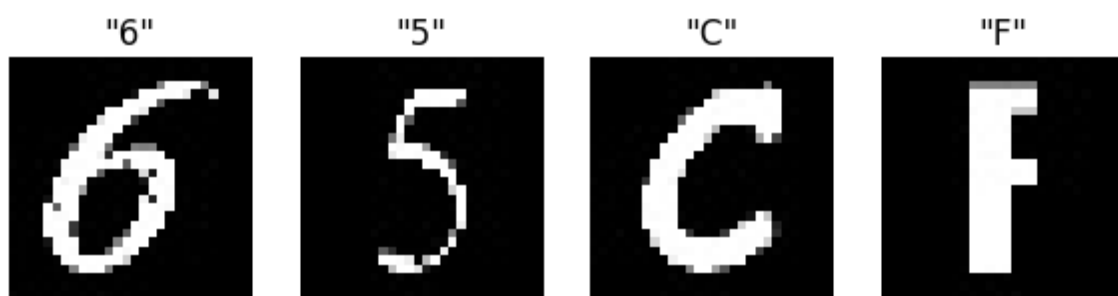


Abbildung 8: Beispiel einer Anwendung von Matplotlib

Um Matplotlib zu installieren muss das Folgende in der Konsole eingegeben werden:

```
1 pip install matplotlib
```

Code 8: PIP Installation von Matplotlib

8.4.5.6 Keras

Keras ist eine Deep-Learning Schnittstelle für diverse Machine Learning Frameworks wie zum Beispiel Tensorflow oder Theanos. Damit wird die Bedienung und Anwendung dieser Frameworks vereinfacht und es bietet auch diverse Funktionen, um Inputs mit den Machine Learning Modellen kompatibel zu machen. Keras ist ein Teil von Tensorflow, wird aber eigenständig weiterentwickelt, um die Kompatibilität mit anderen Machine Learning Frameworks aufrechtzuerhalten. In dieser

Applikation wird es in Zusammenarbeit mit Tensorflow verwendet, um mit den verwendeten Machine Learning Modellen zu arbeiten.

Um Keras zu installieren führt man folgenden Befehl in der Konsole aus:

```
1 pip install keras
```

Code 9: PIP Installation von Keras

8.4.5.7 Tensorflow

Tensorflow ist eines der weltweit beliebtesten Frameworks für Machine Learning und wird von weltweit erfolgreichen Firmen wie zum Beispiel Google, AMD oder auch Intel verwendet. Es bietet eine umfassende Plattformen für jegliche Machine Learning Anwendungen und ist in Zusammenarbeit mit APIs wie zum Beispiel Keras leicht zu verwenden. In dieser Applikation wird damit ein neuronales Netz trainiert und mehrere Machine Learning Modelle verwaltet und verwendet.

Bei der Installation von Tensorflow muss beachtet werden, dass sich diese von Windows zu Raspbian stark unterscheidet. Raspbian unterstützt offiziell die benötigte Version von Tensorflow noch nicht und deswegen muss dieses über ein paar Umwege installiert werden.

Windows:

Die Installation von Tensorflow unter Windows ist einfach, da man es wie andere Libraries einfach über pip installieren kann.

```
1 pip install tensorflow
```

Code 10: PIP Installation von Tensorflow

Raspbian:

Bei Raspbian muss Tensorflow manuell kompiliert werden, da die aktuelle Version noch nicht offiziell unterstützt wird. Diese funktioniert aber trotz dieses Umweges einwandfrei und wird mit den folgenden Befehlen in der Konsole installiert.

Mit dem ersten Befehl werden die benötigten Tools installiert:

```
1 $ sudo apt-get install cmake curl
```

Code 11: Benötigte Tools für Tensorflow

Danach kann man die neuste Tensorflow Version von GitHub herunterladen:

```
1 $ wget -O tensorflow.zip  
↪ https://github.com/tensorflow/tensorflow/archive/v2.4.0.zip
```

Code 12: Tensorflow von GitHub herunterladen

Anschließend muss Tensorflow entpackt werden:

```
1 $ unzip tensorflow.zip  
2 $ mv tensorflow-2.4.0 tensorflow  
3 $ cd tensorflow
```

Code 13: Entpacken von Tensorflow

Im nächsten Schritt müssen noch zusätzlich erforderliche Libraries installiert werden:

```
1 $ ./tensorflow/lite/tools/make/download_dependencies.sh
```

Code 14: Zusätzlich erforderliche Librarys

Danach kann die Installation kompiliert werden:

```
1 $ ./tensorflow/lite/tools/make/build_aarch32_lib.sh
```

Code 15: Kompilieren von Tensorflow

Danach muss man die Installation mit den folgenden Befehlen abschließen:

```
1 $ cd ~/tensorflow/tensorflow/lite/tools/make/downloads/flatbuffers
2 $ mkdir build
3 $ cd build
4 $ cmake ..
5 $ make -j4
6 $ sudo make install
7 $ sudo ldconfig
```

Code 16: Abschließen der Installation von Tensorflow

Nach diesem Schritt sollte Tensorflow Installation funktionieren und man kann es wie jede andere Library in Python einbinden.

8.4.5.8 localutils

localutils ist ein einfaches Python-Script, welches die Arbeit mit WPOD-NET im Bereich der Kennzeichenerkennung vereinfacht und in dieser Applikation für die einfachere Nutzung von WPOD-NET verwendet wird. Es beinhaltet die Funktion „detectlp“ mit welcher ein Bild des Kennzeichens und die Koordinaten des Kennzeichens ermittelt werden können. Dieses Script kann über den folgenden Link von GitHub heruntergeladen werden: https://github.com/quangnhat185/Plate_detect_and_recognize/blob/master/local_utils.py

8.4.5.9 Scikit-learn

Scikit-learn ist eine freie Library für Machine Learning und basiert auf Numpy und Matplotlib. Es wird vor allem verwendet, um mit großen visuellen Datensätzen neuronale Netze zu trainieren. Es bietet Funktionen, um Modelle anzupassen, Daten vorzubereiten, Modelle zu trainieren und

ähnliches. In dieser Applikation wird es für die Normalisierung von Labels verwendet, um diese für Machine Learning Modelle anzupassen und für das zufällige Aufteilen von Arrays in Test und Trainingsteile, welche für das Training eines Neuronalen Netzes benötigt werden.

Für die Installation von Scikit-learn muss die folgende Zeile in der Konsole ausgeführt werden:

```
1 pip install scikit-learn
```

Code 17: PIP Installation von Scikit-learn

8.4.5.10 Requests

Requests ist eine Library mit welcher HTTP-Anfragen in Python vereinfacht werden. Sie wird häufig verwendet, um mit APIs zu kommunizieren und ist eine der beliebtesten Python Librarys, da sie für API-Anwendungen so gut wie notwendig ist. In dieser Applikation wird sie für die Kommunikation zu einer eigenen API für den Datenbankzugriff verwendet. Dadurch ist es ohne größere Schwierigkeiten möglich, das Bild des Kennzeichens, die ID für den jeweiligen Parkplatz und das Resultat des Kennzeichens an die eigene Datenbank zu senden und auch eine Antwort zu erhalten, ob der Zugriff erfolgreich war. Um die Daten mit dieser Library zu übertragen müssen diese einfach nur in einem Dictionary eingetragen werden und dann mit dem korrekten Schlüsselwort über die API gesendet werden.

Die Installation von Requests erfolgt mit folgendem Befehl in der Konsole:

```
1 pip install requests
```

Code 18: PIP Installation von Requests

8.4.5.11 Gpiozero

Gpiozero ist eine Python-Library für den Raspberry Pi. Mit ihr ist es möglich auf die GPIO-Pins des Raspberry zuzugreifen, Signale von diesen auszulesen und Signale an diesen auszugeben. Dies wird oft verwendet, um Sensoren auszuwerten oder um Aktoren zu steuern, da diese Pins frei

programmierbar sind, wodurch man jede beliebige Anwendung realisieren kann. Der Raspberry Pi kann zwar nur Aktoren in seiner Leistungskategorie ansteuern, dies kann aber zum Beispiel mit einem Relais umgangen werden, wodurch der Raspberry Pi zu einem mächtigen Werkzeug für die Sensorik und Aktorik wird. Ein Beispiel für die möglichen Anwendungen ist ein Button oder eine Lampe. In dieser Applikation wird mit dieser Library ein gedrückter Button detektiert, welcher als Auslöser für die Kamera fungiert.

In der folgenden Abbildung sieht man ein Bild der GPIO-Pins des Raspberry Pi, welche mit dieser Library angesteuert werden können:

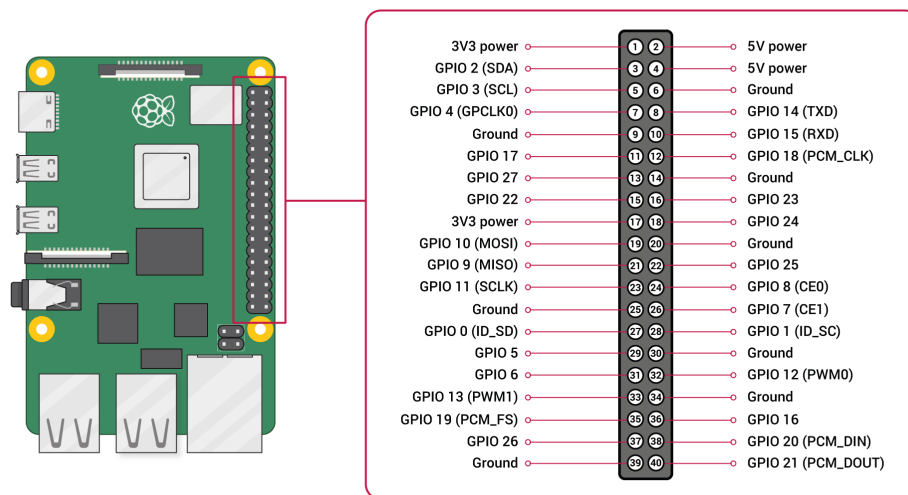


Abbildung 9: GPIO-Pins des Raspberry Pi

Diese Library kann nur auf dem Raspberry Pi installiert werden. Um die Library Gpiozero zu installieren kann pip verwendet werden:

```
1 pip install gpiozero
```

Code 19: PIP Installation von Gpiozero

8.4.5.12 Picamera

Die Library picamera wird verwendet, um auf das Kameramodul des Raspberry Pi zuzugreifen.

Die Library bietet etliche Funktionen an, mit welchen die Kamera gesteuert werden kann, wie zum Beispiel Fotos aufzunehmen, Videoaufnahmen zu starten und zu beenden, Fotoeinstellungen zu ändern und ähnliches. Die Library ist notwendig, wenn man über Python das Kameramodul bedienen will. In dieser Applikation wird sie verwendet, um ein Vorschaubild des aufgenommenen Bildes auf dem Bildschirm darzustellen, falls ein Bildschirm angeschlossen wird und um im Anschluss das Bild aufzunehmen und korrekt zu skalieren.

Die Installation von picamera ist nur auf dem Raspberry Pi möglich, da nur dieser damit arbeiten kann. Für die Installation von picamera muss folgender Befehl in der Konsole eingegeben werden:

```
1 pip install picamera
```

Code 20: PIP Installation von Picamera

8.4.6 Wichtige Programmteile

8.5 Raspberry Pi

8.5.1 Einleitung

Damit die Software der Kennzeichenerkennung arbeiten kann, benötigt es eine geeignete Hardware. Diese muss dabei die folgenden Eigenschaften aufweisen, um für die Kennzeichenerkennung geeignet zu sein:

- Möglichst klein
- Nicht zu teuer
- Möglichkeit eine Kamera anzuschließen
- Schnell
- Internetanbindung

8.5.2 Wahl des Raspberry Pi

In dieser Applikation wird ein RaspberryPi 4B 2GB verwendet, da er alle zuvor genannten Eigenschaften am besten erfüllt.

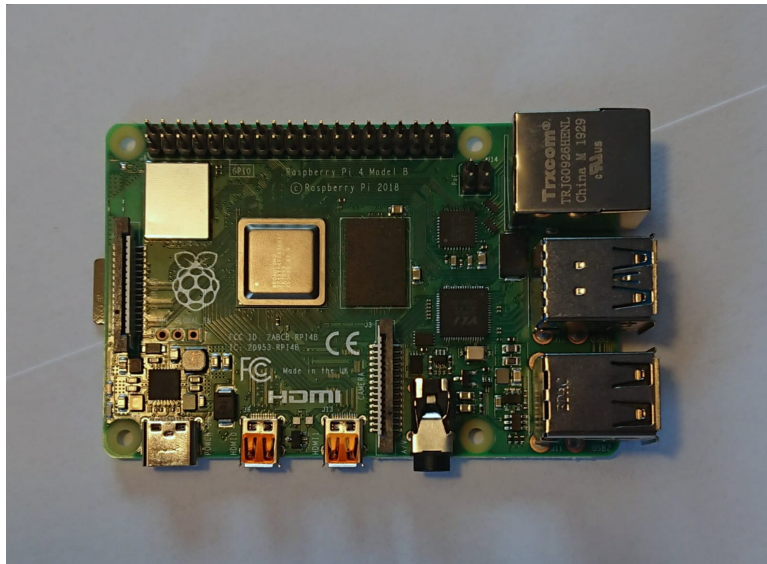


Abbildung 10: Raspberry Pi

Er hat die Größe einer Scheckkarte, wodurch es möglich ist ein kompaktes Gehäuse zu bauen, welches man einfach montieren kann. Er ist mit 35€ pro Stück nicht zu teuer. Er hat einen integrierten Kameraanschluss und es gibt unzählige Kameras, welche damit kompatibel sind. Er hat für seine Größe eine sehr gute Rechenkraft und ist damit in der Lage das komplette Programm für die Kennzeichenerkennung in einer akzeptablen Zeit abzuarbeiten. Er besitzt außerdem einen Ethernet-Anschluss und ein WLAN-Modul, wodurch er sehr einfach mit dem Internet verbunden werden kann.

8.5.3 Kamera

Für den Raspberry Pi muss zudem noch die passende Kamera ausgewählt werden, um die Bilder von den Kennzeichen aufzunehmen. Dafür gibt es Unmengen an kompatiblen Kameras, aber hier wird das Originalzubehör von RaspberryPi verwendet. Dies hat den Grund, dass diese Kamera leicht zu verwenden, sehr klein, mit Schrauben einfach zu befestigen und günstig ist. Zudem liefert sie ein qualitativ hochwertiges Bild, welches für die Software gut verarbeitbar ist.



Abbildung 11: Raspberry Pi mit Kamera

9 Fahrzeugerkennung

9.1 Anforderungen

Das Ziel der Fahrzeugerkennung ist es Fahrzeuge auf mehrere Parklücken eines Parkplatzes zu erkennen. Die daraus gewonnen Zustände sollen an das Webinterface übermittelt und an den jeweiligen Parklücken über LEDs ausgegeben werden.

9.2 Vorstudie

9.3 Erkennung von Metallen über Spulen

Grundsätzlich werden in der Realität häufig Spulen verwendet, welche unter dem Asphalt verbaut sind, um darüberliegende Fahrzeuge zu detektieren. Als Messprinzip wird die Änderung des magnetischen Widerstands R_m bei konstanter magnetischer Spannung U_m und daraus resultierende magnetischen Fluss Φ Es gelten für diese Größen der folgende Zusammenhänge:

$$\Phi = \frac{U_m}{R_m} \quad (1)$$

Wobei

Φ = magnetischer Fluss

R_m = magnetischer Widerstand

U_m = magnetische Spannung

Der magnetische Widerstand lässt sich wiederum durch die Eigenschaften der Spule bestimmen. Die Formel hierfür lautet:

$$R_m = \frac{N \cdot (2a + 2b)}{\mu_0 \cdot \mu_r \cdot A} \quad (2)$$

Wobei

$$A = a \cdot b \quad (3)$$

N = Anzahl der Windungen der Spule

a = Breite der Spule in m

b = Länge der Spule in m

$\mu_0 = 1,2566 \cdot 10^{-6} \text{ N/A}^2$ = magnetische Feldkonstante

μ_r = relative Permeabilität

A = Fläche der Spule

Bis auf die relative Permeabilität sind alle anderen Variablen konstant. Daraus lässt sich schlussfolgern, dass der magnetische Fluss Φ anhand der Gleichung 1 und 2 proportional zur relativen Permeabilität ist.

$$\Phi \propto \mu_r \quad (4)$$

Der Fluss Φ hängt somit auch von den Materialien ab durch die er fließt. In der nächsten Abbildung kann man erkennen, wie ein Fahrzeug über eine im Boden installierte Spule den magnetischen Fluss Φ und somit die magnetische Flussdichte B beeinflussen kann.

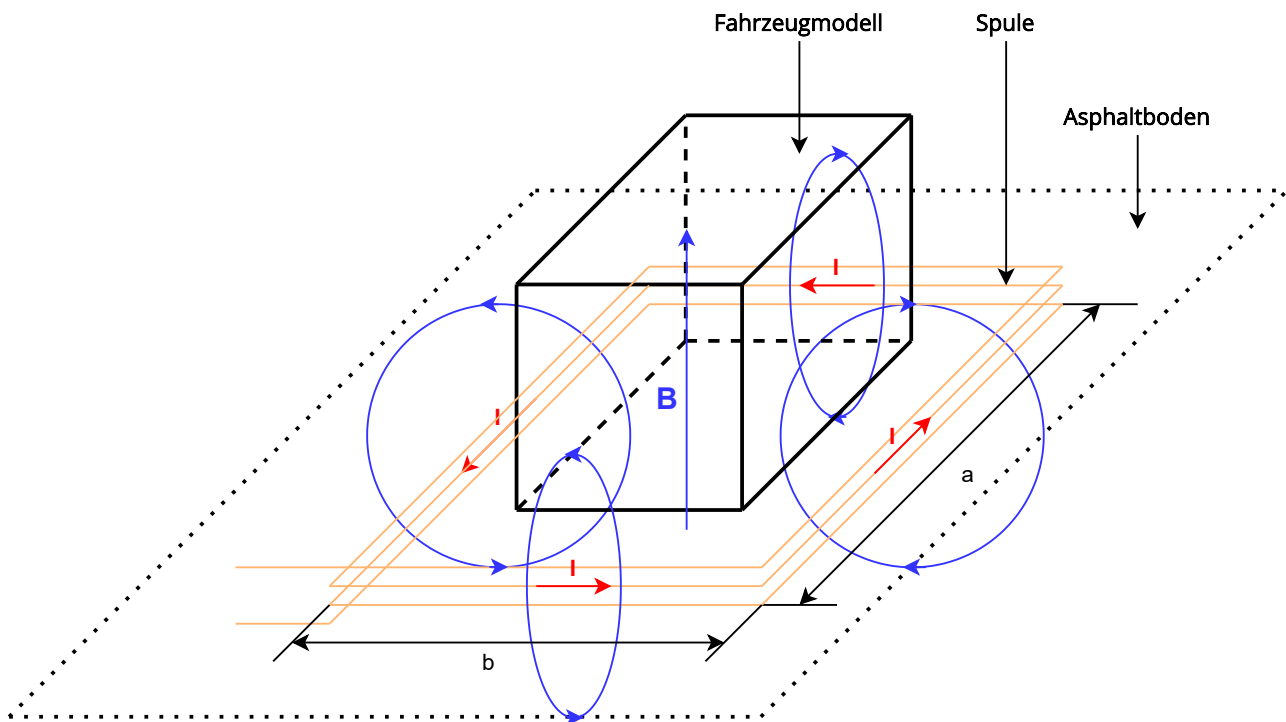


Abbildung 12: Installation der Spule

Um diese Größen auslesbar zu machen, müssen diese magnetischen bei einer direkten Messung in elektrische Größen umgewandelt werden. Hierfür gelten folgende Zusammenhänge:

$$L = \frac{N \cdot \Phi}{I} = \frac{\Psi}{I} \quad (5)$$

Wobei

L = Induktivität der Spule

I = Strom der durch die Spule fließt

N = Anzahl der Windungen der Spule

Ψ = Verkettete Fluss

$$u(t) = L \cdot \frac{di(t)}{dt} \quad (6)$$

Wobei

L = Induktivität der Spule

$i(t)$ = Strom der durch die Spule fließt zum Zeitpunkt t

$u(t)$ = Spannung die an der Spule anliegt zum Zeitpunkt t

t = Zeit in s

So lässt sich bei Bekanntheit von Strom und Spannung auf die Induktivität und mit der Gleichung 5 und mit den Zusammenhang 4 auf die magnetischen Eigenschaften des Materials rückschließen. Diese Art der Detektion bietet viele praktische Vorteile.

- **Größerer Messbereich**

Im Vergleich zu anderen Detektionsmethoden wie einer Lichtschranke kann man einen größeren Bereich durch die Wirkfläche der Spule abdecken. So lassen sich auch kleinere Kraftfahrzeuge wie Motorräder oder Mopeds besser erkennen.

- **Schutz vor Umweltfaktoren**

Durch den Verbau im Boden ist die Messeinrichtung vor Umwelteinflüssen wie Regen, Frost, hohen beziehungsweise niedrigen Temperaturen und Korrosion besser geschützt. Dies verringert auch den Einfluss dieser Störfaktoren auf die Eigenschaften Spule und somit auf die daraus resultierenden Messergebnisse.

- **Ausschließung von Materialien**

Alle nicht metallische Stoffe werden von diesem Messprinzip nicht wahrgenommen. So können Verschmutzungen wie Blätter und Staub, welche visuelle Sensoren stören können, die Detektion nicht behindern.

9.3.1 Messung ferromagnetischer Metalle

Um diese Detektionsverfahren zu verstehen muss zuerst der Zusammenhang zwischen Induktivität und relativer Permeabilität verstanden werden. Aus den Gleichungen 2, 1 und 5 kann die folgende Proportionalität ermittelt werden:

$$L \propto \mu_r \quad (7)$$

Bei ferromagnetischen Stoffen wie Eisen ist die relative Permeabilität sehr viel größer als 1. Wenn nun ein Auto oder eine anderes Vehikel sehr viel Eisen beinhaltet wirkt sich dies steigernd auf die Induktivität der Spule aus. Die folgenden Verfahren nutzen dieses Prinzip um die Belegung einer Parklücke zu bestimmen.

9.3.1.1 RL-Oszillator mit Timer Baustein

Bei diesem Messverfahren wird die Änderung der Induktivität L über die Änderung einer Stromladekurve über einen Widerstand R ermittelt. Eine Simulation in LTSPice mit folgendem Schaltbild kann die Auswirkung auf eine Ladekurve bei gleicher Spannung und gleichem Widerstand gut darstellen.

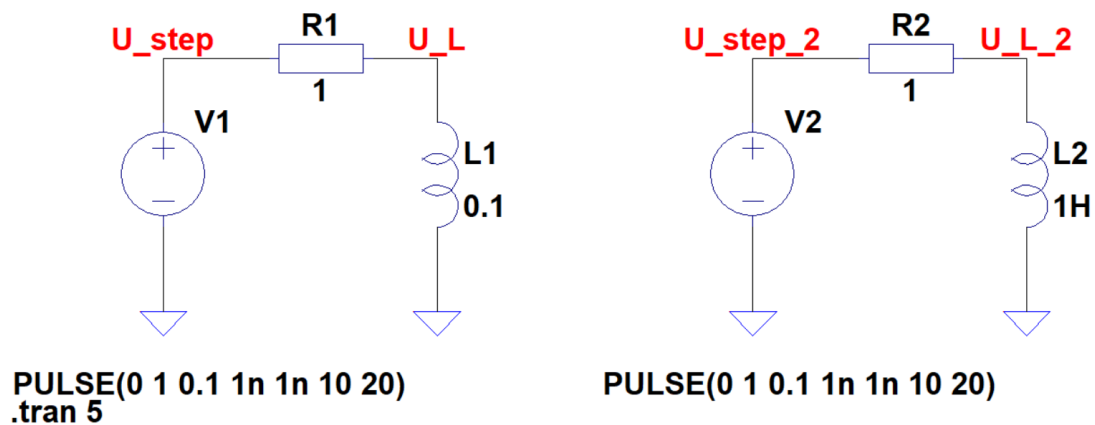


Abbildung 13: LTSPice Blockbild zweier RL-Glieder

Die daraus Resultierende Simulation im Zeitbereich ergibt folgendes Ergebnis: hmm

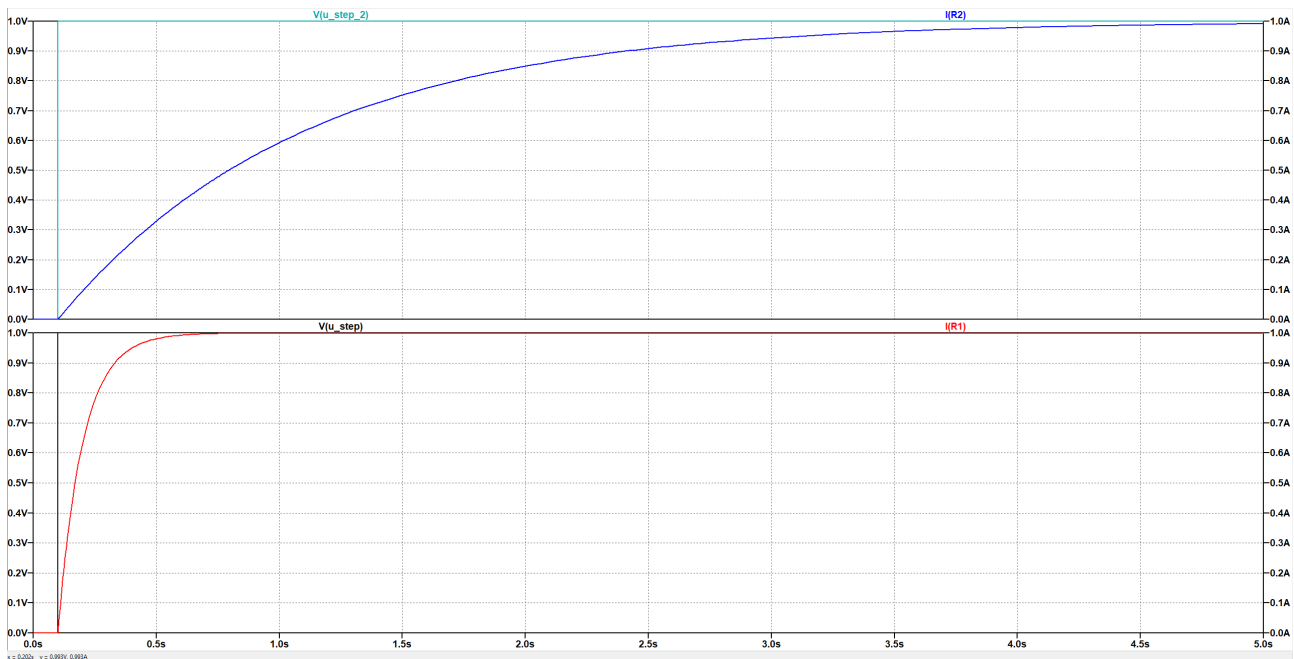


Abbildung 14: Stromkurven zweier RL-Glieder

Es ist aus der letzten Abbildung erkennbar, dass sich eine größere Induktivität verlangsamt auf die Zunahme des Stromes auswirkt. Für Einschaltvorgänge kann diese Stromkurve für RL-Glieder folgendermaßen beschrieben werden.

$$i(t) = \frac{U_0}{R} \cdot (1 - e^{-\frac{t}{\tau}}) \quad (8)$$

$$\tau = \frac{L}{R} \quad (9)$$

$i(t)$ = Strom der durch die Spule fließt zum Zeitpunkt t

L = Induktivität der Spule in H

R = Widerstand in Ω

U_0 = Spannung die nach dem einschalten anliegt in V

t = Zeit in s

τ = Zeitkonstante in $\frac{1}{s}$

Timer Bausteine wie der NE555 nutzen diese solche Verlaufskurven wenn sie als astabile Kippstufe konfiguriert sind. Es gibt einen Eingang dieses Baustein meist CV für Control Voltage der die

Spannung überwacht. Überschreitet diese Spannung zwei Drittel der Betriebsspannung schaltet er einen weiteren Pin, meist DIS für Discharge, auf 0V beziehungsweise auf Masse. Unterschreitet jedoch die Spannung am CV Pin ein Drittel der Betriebsspannung so wird der DIS Pin auf Betriebsspannung geschaltet. Diese beiden Pins sind über RC- oder RL- Glieder verbunden. Es entsteht somit ein Oszillator, welcher die Spannung am DIS Pin anhebt und absenkt. Der NE555 kann dieses Signal über einen internen Komparator in ein Rechtecksignal umwandeln, welches besser von Mikrokontrollern ausgelesen werden kann. Der Mikrokontroller kann so die Anzahl der einkommenden Takte über einen gewissen Zeitraum zählen und daraus eine Oszillatorfrequenz errechnen. Die Zeitsignale lassen sich in LTSPice mit dem NE555 als Timer-Baustein simulieren.

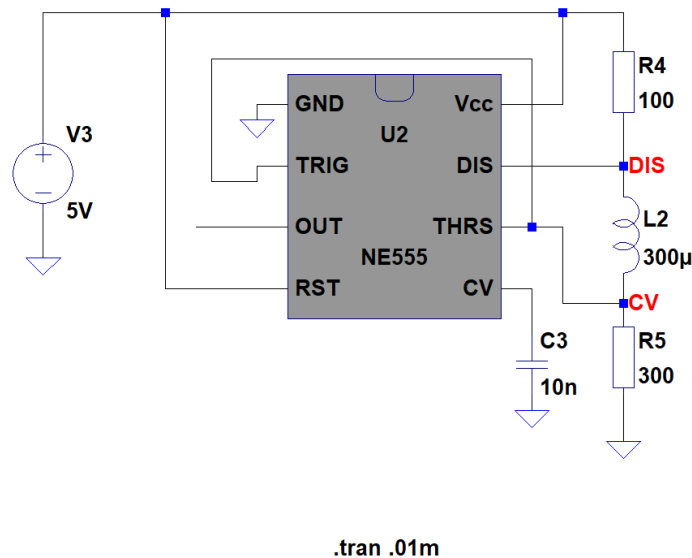


Abbildung 15: RL-Oszillator mit NE555

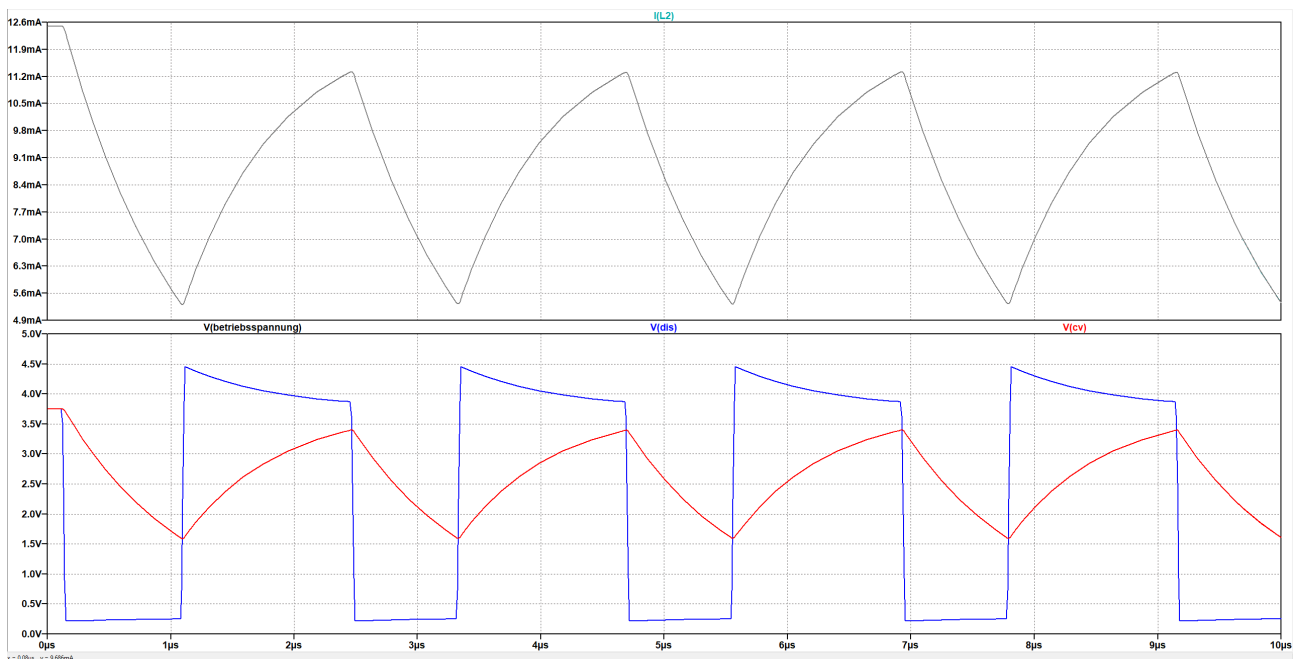


Abbildung 16: RL-Oszillator mit NE555 Zeitsignale

Im oberen Diagramm ist der Verlauf des Stromes durch die Spule zu erkennen der wie erwartet zu- und abnimmt. Im unteren Diagramm ist in schwarz die Betriebsspannung als Referenz in rot der CV Pin und in blau der DIS Pin des NE555. Das rote Signal wechselt zwischen zwei Drittel der Betriebsspannung und einem Drittel der Betriebsspannung mit einer Frequenz von $f = 438\text{kHz}$ bei einer Induktivität von $L = 300\mu\text{H}$, einem Widerstand $R4 = 100\Omega$ und einem Widerstand $R5 = 300\Omega$. Die Simulation entspricht hier der Theorie, jedoch bricht das blaue Signal am PIN DIS in der Einschaltphase ein. Der Grund dafür ist der interne Widerstand des NE555, der ab einem gewissen Stromverbrauch für einen Spannungsabfall sorgt. Da es meist besser ist mit niedrigen Frequenzen zu arbeiten ist es nach den Gleichungen 8 und 9 entweder nötig die Induktivität zu erhöhen oder die Widerstände zu verkleinern. Die Induktivität lässt sich aber entweder durch erhöhen der Windungen oder durch Vergrößerung der Spule erreichen, was in vielen Fällen nicht möglich ist oder zunehmend kostspielig ist. Die Reduktion der Widerstände führt zu einer Zunahme des Stromes und der Verlustleistung, was auch unerwünscht ist. Aus diesen Gründen ist der RL-Oszillator nur für hohe Frequenzen geeignet. In der nächsten Abbildung sieht man die Schaltung bei verschiedenen Induktivität nämlich bei $L2 = 100\mu\text{H}$ und bei $L = 1\text{mH}$

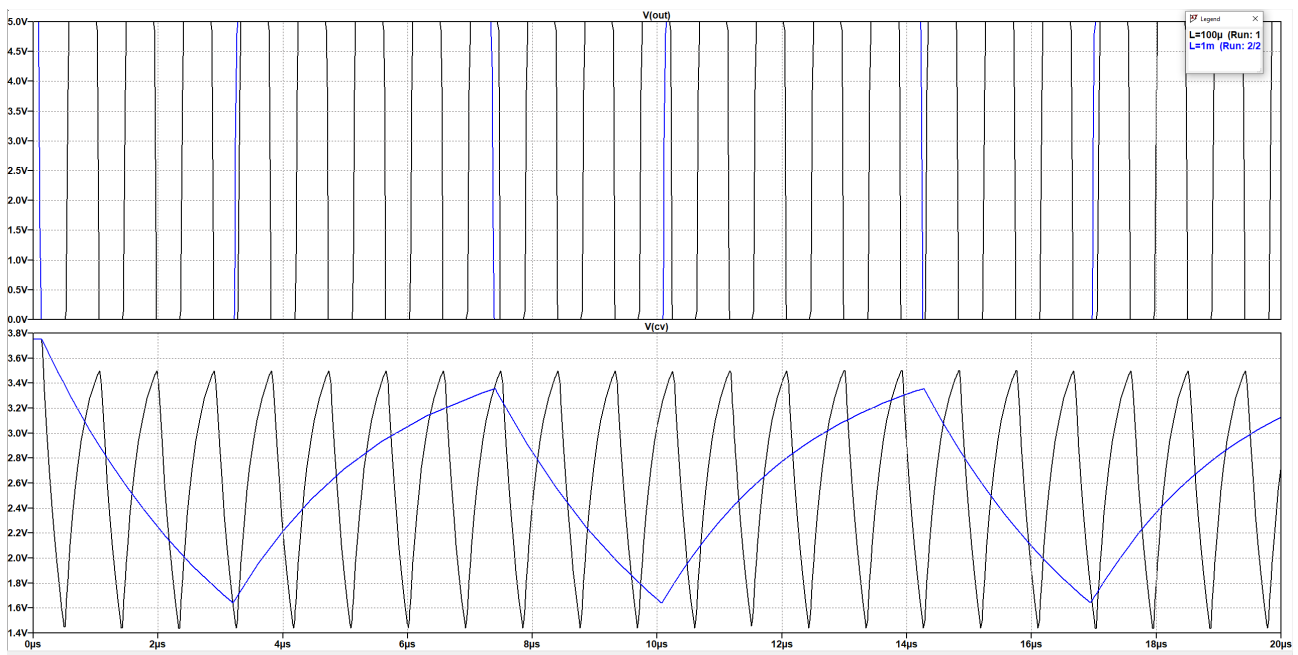


Abbildung 17: Vergleich des NE555-Oszillators bei unterschiedlichen L

Eine Verzehnfachung der Induktivität führt zu einer Reduktion der Oszillatorfrequenz um den Faktor 10.

$$f \propto \frac{1}{L} \quad (10)$$

9.3.2 Messung paramagnetischer Metalle

Da nicht alle Metalle ferromagnetische Eigenschaften haben werden viele Stoffe wie Aluminium, Kupfer oder diverse Legierungen von einer niederfrequenten Messung der Induktivität nicht wahrgenommen. Ein Effekt, welcher bei elektrischen Leitern abhilfe schaffen kann, sind die Eddy Currents. Wenn in einem Leiter ein magnetisches Feld einwirkt erzeugt dieses im Objekt Ströme, die wiederum ihr eigenes magnetische Feld erzeugen. Die fließenden Ströme erzeugen im Objekt Ohmsche Verluste, welche normalerweise unerwünscht sind. Sie wirken auch einer Änderung der magnetischen Feldes entgegen, da sie selbst Energie brauchen um ihre Richtung zu ändern. Je größer die Frequenz der eingespeisten Felddichte B desto stärker wirkt die Flussdichte B_{eddy} senkend auf die Gesamtflussdichte und so auf den magnetischen Fluss Φ . Nach der Gleichung 5 reduziert sich danach die Induktivität mit zunehmender Frequenz.

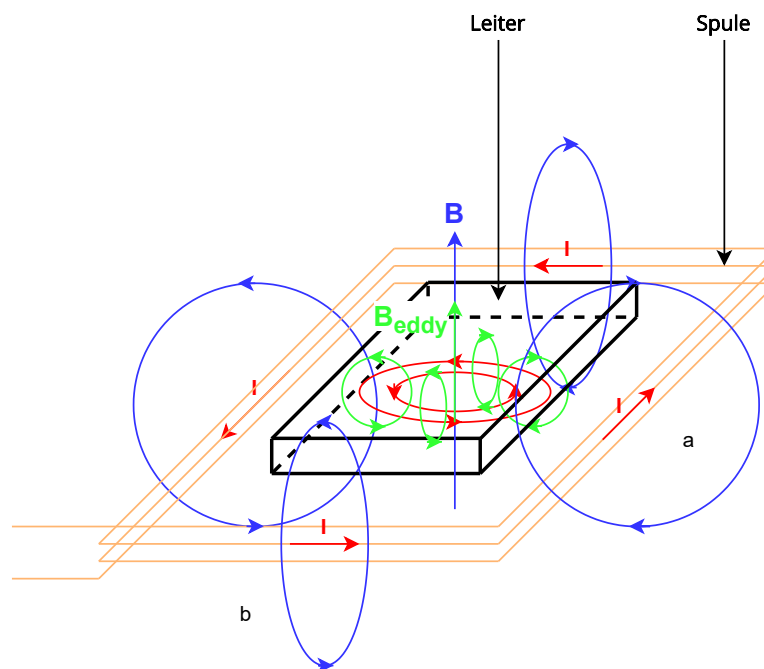


Abbildung 18: Eddy Currents in einem Leiter

Bei sehr hohen Frequenzen ist jedoch zu beachten, dass der Skin-Effekt die Querschnittsfläche, durch die der Wechselstrom fließen kann, reduziert. Somit nimmt der Effekt der Eddy-Currents ab. Diese Phänomene werden jedoch nicht in dieser Arbeit behandelt, da sie außerhalb des Rahmens der Fahrzeugerkennung liegen.

9.3.2.1 LC Oszillatoren

Im Vergleich zum RL-Oszillator, der über eine astabile Kippstufe läuft, kann ein LC Oszillator seine Resonanzfrequenz nur durch das Verstellen der Induktivität und der Kapazität eines Kondensators verändern. Zudem gibt es in der Theorie keine Wirkverluste, da die Energie abwechselnd zwischen magnetischer und elektrischer Energie umgewandelt wird. In der Realität gibt es jedoch ohmsche Verluste, die zu einem Abklingen der Schwingung führen. Es braucht daher ein aktives Glied wie ein Transistor oder ein Operationsverstärker, der den Oszillator mit Energie versorgt.

Der zeitliche Verlauf einer ungedämpften Schwingung kann in LTSpice simuliert werden indem wir eine Sprungfunktion auf ein LC Glied geben und einen Widerstand in Serie zum Kondensator oder zur Spule geben.

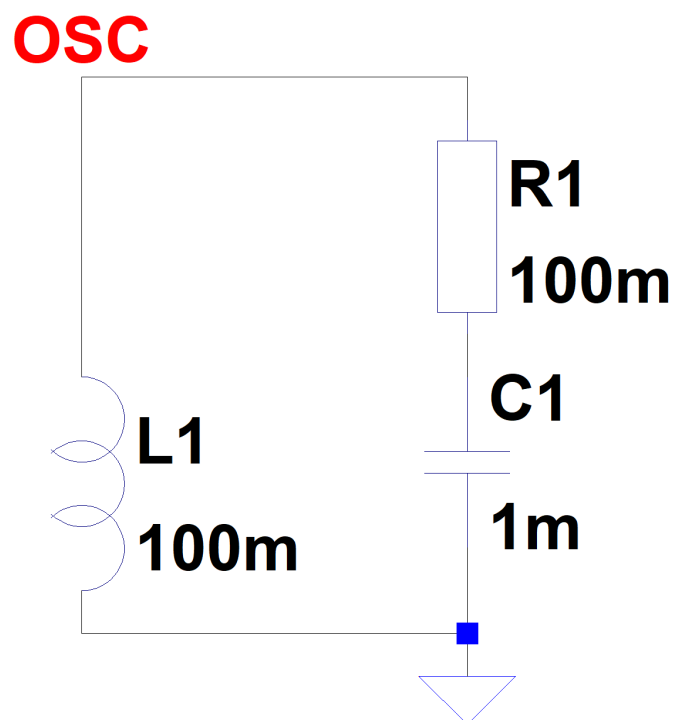


Abbildung 19: Parallelschwingkreis in LTSpice

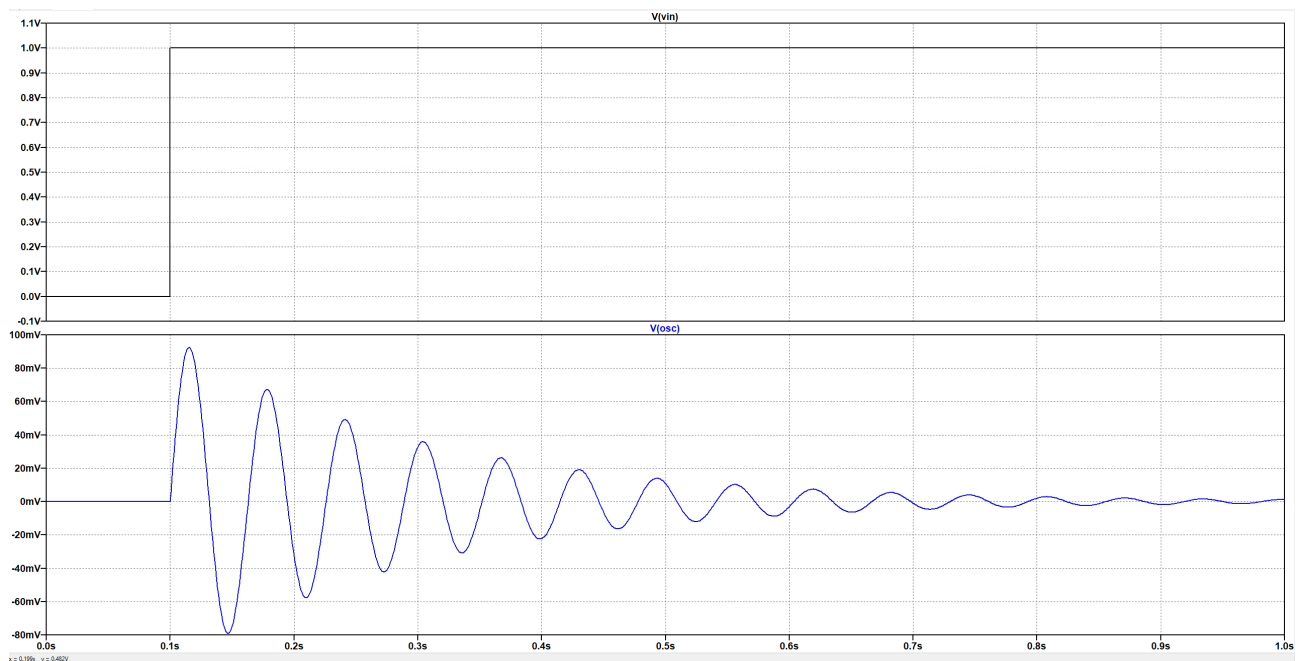


Abbildung 20: Gedämpfte Schwingung

Aus der Simulation ergibt sich über eine Cursormessung für die angegebenen Werte $L = 100mH$ und $C = 1mF$ eine Resonanzfrequenz einen Wert von $f = 15.85Hz$. Die Resonanzfrequenz für einfache LC-Schwingkreise lässt sich mit der Thomsonschen Schwingungsgleichung errechnen:

$$f = \frac{1}{2 \cdot \pi \sqrt{L \cdot C}} \quad (11)$$

Wenn man die Werte $L = 100mH$ und $C = 1mF$ in die Gleichung 11 einsetzt erhält man eine Resonanzfrequenz von $f = 15.915Hz$. Allgemein tritt in LC-Glieder bei einer Phasenlage von $\varphi_{LC} = 180^\circ$ und bei der Frequenz nach der Thomsonschen Gleichung 11 Resonanz auf.

9.3.2.2 Messung der Frequenz eines Colpitts-Oszillator

9.4 RS485 Bussystem

9.4.1 Überblick

9.4.2 Elektrische Spezifikation

9.4.3 Implementation eines eigenen Protokolls

9.5 Mikrokontroller Slave-Geräte

9.5.1 Überblick

9.5.2 Atmega328PB

9.5.3 Peripherie des Mikrokontrollers

9.5.3.1 Spannungswandler

9.5.3.2 RS485 Pegelwandler

9.5.3.3 Digitale Ein- und Ausgänge

9.5.4 Layout des Slave-Gerätes

9.5.5 Gehäuse

9.6 USB-Master

9.6.1 USB-Bussadapter Gerät

9.6.1.1 Überblick

9.6.1.2 FT232RL

9.6.1.3 Spannungsversorgung

9.6.1.4 USB-C Anschluss

9.6.1.5 Layout des Master-Geräts

9.6.1.6 Gehäuse

9.6.2 Master Programm

9.6.2.1 Benötigte Software

9.6.2.2 Adressvergabe

9.6.2.3 Frequenzauslesung

9.6.2.4 Auswertung

9.6.2.5 API-Post

9.6.3 RaspberryPi als Mastergerät

9.6.3.1 SSH Remote Zugriff

9.6.3.2 Code Deployment

9.6.3.3 Unittest

10 Webinterface

10.1 Einleitung

10.2 Anforderungen

Das Webinterface hat auf der einen Seite die Aufgabe die Kommunikation mit der Kennzeichenerkennung und der Fahrzeugerkennung sicherzustellen und auf der anderen Seite die Verwaltung und Darstellung der gewonnen Daten.

10.3 Verwendete Technologien

10.3.1 HTML

HTML steht hierbei für Hypertext Markup Language und ist eine Auszeichnungssprache welche vom World Wide Web Consortium (W3C)¹¹ entwickelt wird. Hypertext Markup Language (HTML) ist De-Facto-Standard um Inhalte in Browsern darzustellen. HTML ist dabei aber nicht für die visuelle Darstellung verantwortlich sondern nur für die semantische Struktur. Der Sinn dahinter ist, dass der Inhalt und die Vorgaben an die Darstellung möglichst gut getrennt ist. Für die Formatierung kommt die Stylesheet-Sprache Cascading Style Sheets (CSS) zum Einsatz, welche ebenfalls vom World Wide Web Consortium entwickelt wird. Die aktuellste Version der HTML Spezifikation ist HTML5¹² und wurde am 28. Oktober 2014 vom W3C vorgelegt.

10.3.1.1 Beispielhafte HTML Seite

Eine HTML Seite setzt sich aus einer Vielzahl von sogenannten Elementen zusammen. Ein Element besteht aus einem Start Tag und aus einem End Tag, der Inhalt wird zwischen diese Tags geschrieben. Nun folgt eine einfache HTML Seite, welche die Grundlegenden Funktionen von HTML und CSS darlegen soll.

¹¹<https://www.w3.org>

¹²<https://www.w3.org/2014/10/html5-rec.html.en>

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  <title>Titel</title>
5  </head>
6  <body>
7
8  <h1>Überschrift</h1>
9  <p>Paragraph</p>
10
11 </body>
12 </html>
```

Code 21: index.html

Das Element `<!DOCTYPE html>` deklariert, dass die folgende Seite den HTML5 Standard verwendet. Danach folgt mit `<html>` das Wurzelement, das alle anderen Elemente beinhaltet. Das `<head>` Element beinhaltet verschiedene Metadaten d.h. Daten die nicht angezeigt werden. Im oben gezeigten Beispiel Code 21 wird nur der Title des Dokuments gesetzt, dieser wird im Browser Tab angezeigt. Es können aber auch noch andere Daten gesetzt bzw. eingebunden werden:

- Character Set
- Styles
- Scripts
- Viewport
- Sonstige Metainformationen (Author, Keywords)

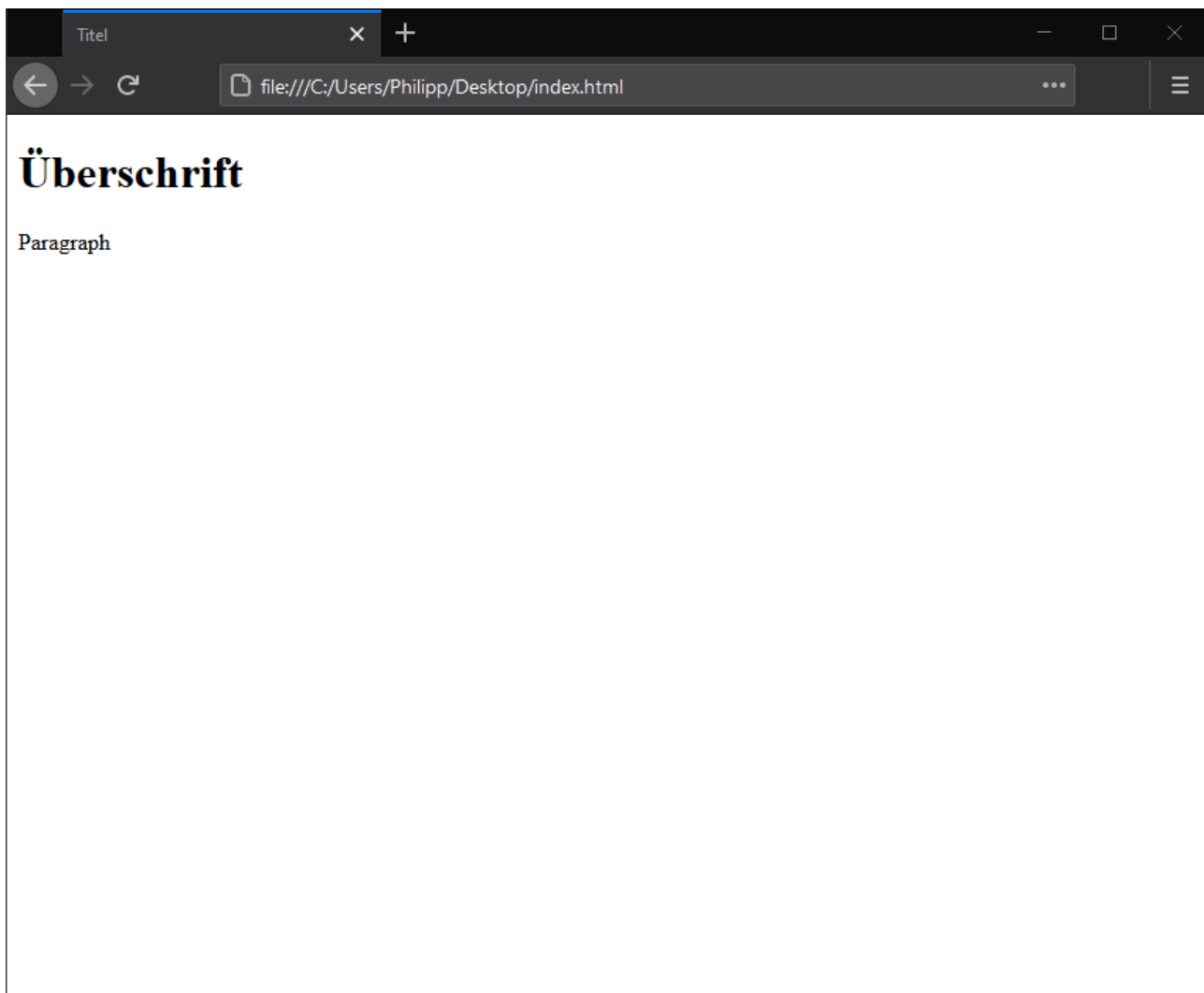


Abbildung 21: Einfache HTML Seite

10.3.2 CSS

Wie bereits angesprochen ist Cascading Style Sheets (CSS) für die Formatierung bzw. die visuelle Darstellung der einzelnen HTML-Elemente verantwortlich. Der Standard wird wie bei HTML vom W3C spezifiziert und die aktuellste Version ist CSS3 was so viel bedeutet wie CSS Level 3, wobei nur einzelne Teile als Empfehlung durch das W3C vorgelegt wurden, beispielweise das CSS Color Module Level 3¹³. Um die Funktion darzustellen wird die vorherige HTML Seite nun mit CSS ergänzt.

¹³<https://www.w3.org/TR/css-color-3>

```
1  body {  
2      background-color: deepskyblue;  
3  }  
4  
5  h1 {  
6      color: white;  
7      text-align: center;  
8      font-family: verdana;  
9  }  
10  
11 p {  
12     color: wheat;  
13     font-family: verdana;  
14     font-size: 20px;  
15 }
```

Code 22: style.css

Nun muss dieses Stylesheet nur noch im <head> Tag mit

`<link rel="stylesheet" href="mystyle.css">` eingebunden werden.

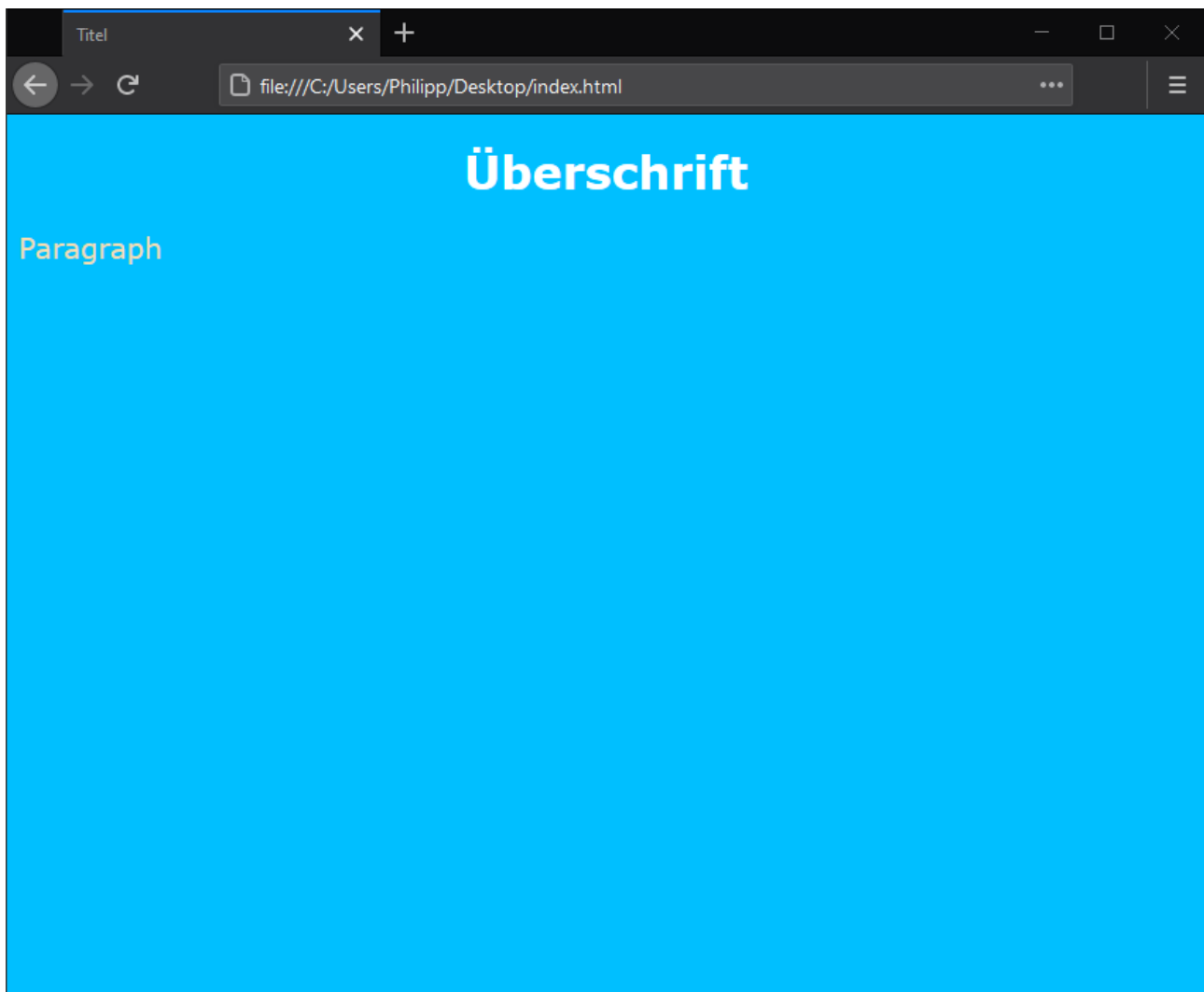


Abbildung 22: Einfache HTML Seite mit CSS

Es lässt sich nun deutlich erkennen, dass sich die Webseite stark verändert hat von den Farben bis zu der Schriftart. HTML und CSS bieten noch viel mehr Funktionen, eine Vielzahl der Funktionen sind auf der Website [w3schools](https://www.w3schools.com/html)¹⁴ zu finden.

10.3.3 JavaScript

Es folgt nun eine weitere sehr wichtige Technologie und die meist verwendete Programmiersprache überhaupt laut der Stack Overflow Developer Survey 2020¹⁵. JavaScript (JS) ermöglicht es dynamische Webseiten zu erstellen, dabei wird der Code direkt lokal im Browser ausgeführt. Jedoch

¹⁴<https://www.w3schools.com/html> und <https://www.w3schools.com/css>

¹⁵<https://insights.stackoverflow.com/survey/2020>

ist JavaScript nicht mehr nur auf das Frontend¹⁶ mehr beschränkt, es möglich mit Frameworks wie Node.js auch Backend¹⁷ Applikationen zu schreiben und somit ist es möglich Full-Stack¹⁸-Anwendungen vollständig mit JavaScript zu entwickeln. Eine der wichtigsten Anwendungsgebiete ist die Manipulation von Elementen über das Document Object Model (DOM). Der Standards wird unter dem Namen ECMA Script von der Organisation Ecma International¹⁹ veröffentlicht und die aktuellste Version ist **ECMA-262**.

10.3.4 PHP

Hypertext Preprocessor (PHP) ist eine Skriptsprache um dynamische Webseiten zu realisieren, jedoch wird nicht wie bei JavaScript der Code auf dem Client ausgeführt sondern auf dem Server, dort wird die HTML-Ausgabe generiert und dem Client zugesendet. Somit ist es nicht möglich den Code als Benutzer zu betrachten. Grundsätzlich ist PHP als synchrone Sprache geplant worden, es ist jedoch auch möglich asynchron zu Programmieren um die Performance zu steigern. Die aktuellste Version ist PHP 8²⁰.

10.3.5 TailwindCSS

Es gibt eine Vielzahl von CSS-Frameworks, das Ziel ist ein einfacheres und schnelleres erstellen von Webseiten, dazu gehören neben TailwindCSS folgende relevanten Frameworks.

- Bootstrap (<https://getbootstrap.com>)
- Foundation (<https://get.foundation>)
- Materialize (<https://materializecss.com>)

Viele von diesen CSS Frameworks verwenden vorgefertigte Components welche direkt verwendet werden können, dies führt dazu, dass die Entwicklung sehr rasch ist. Dort unterscheidet sich TailwindCSS von den anderen CSS Frameworks, dort existieren sogenannte Utility-Classes, diese können direkt im HTML auf die einzelnen Elemente angewendet werden.

¹⁶Grafische Benutzeroberfläche mit der der Benutzer interagiert

¹⁷Verarbeitung von Daten auf beispielsweise einem Server

¹⁸Front- und Backend

¹⁹<https://www.ecma-international.org>

²⁰<https://www.php.net/docs.php>

10.3.6 Laravel

Laravel²¹ ist ein Open-Source PHP-Framework, es erleichtert die Entwicklung und erhöht die Sicherheit und auch durch die zahlreichen First-Party-Packages bietet Laravel ein sehr hochwertiges Ecosystem. Da Laravel ein sehr wichtiger Bestandteil des Webinterfaces ist wird später noch genauer auf die einzelnen Funktionen des Frameworks eingegangen. Laravel wird seit Juni 2011 entwickelt und erhält jährlich eine neue Version, aktuell ist Laravel 8 die neuste Version.

Das Framework folgt dem sogenannten Model-View-Controller (MVC) Muster das bedeutet, dass die Programmierlogik in drei verschiedene Teile unterteilt wird. Der Sinn dahinter ist, dass die Anwendung dadurch sehr flexibel ist und später leichter erweitert werden kann oder einzelne Teile wiederverwendet werden können.

- **Model**

Hier befindet sich die Datenstruktur der Anwendung. In Laravel kann dies beispielsweise das Model User sein.

- **View**

In der View befindet sich die Präsentationsebene, im Fall von Laravel sind das Components und Layouts.

- **Controller**

In den Controllern der Anwendung befindet sich die Logik um beispielsweise durch das Absenden eines Formulars einen Eintrag in der Datenbank zu erstellen.

Laravel besitzt eine sehr ausführliche Dokumentation welche unter (<https://laravel.com/docs/8.x>) zu erreichen ist.

10.3.6.1 Routing

In Laravel erfolgt durch ein Routefile unter `routes/web.php`. API Requests haben ein eigenes Routefile unter `routes/api.php` und erhalten einen `/api` Prefix. Diese Dateien werden dann automatisch durch einen Service Provider von Laravel geladen.

²¹<https://laravel.com/>

Der Laravel Router erlaubt folgende HTTP-Anfragemethoden:

- **GET** (Fordert Ressource an)

```
Route::get($uri, $callback);
```

- **POST** (Neue Ressource erstellen)

```
Route::post($uri, $callback);
```

- **PUT** (Ressource ersetzen oder erstellen)

```
Route::put($uri, $callback);
```

- **PATCH** (Ressource ändern)

```
Route::patch($uri, $callback);
```

- **DELETE** (Löscht Ressource)

```
Route::delete($uri, $callback);
```

- **OPTIONS** (Liste von unterstützten Methoden des Servers)

```
Route::options($uri, $callback);
```

Im folgenden Ausschnitt Code 23 werden ein Teil der Routen von den Benutzern dargelegt. Dabei lässt sich erkennen, dass die Routen sich in einer Gruppe befinden, welche den Prefix `/admin` und die Middleware `verified` hat das bedeutet, dass der Benutzer eingeloggt sein muss um diese Routen aufzurufen. Dabei verweisen die Routen auf einzelne Methoden im UserController. Schlussendlich werden mit der `name`-Methode die Routen benannt, damit sie später im Code einfach referenziert werden können.

```

1  <?php
2  Route::group(['middleware' => 'verified', 'prefix' => 'admin'],
3  function () {
4      Route::get('users', [UserController::class,
5          ↪ 'index'])->name('users.index');
6      Route::get('users/create', [UserController::class,
7          ↪ 'create'])->name('users.create');
8      Route::post('users', [UserController::class,
9          ↪ 'store'])->name('users.store');
10 });

```

Code 23: web.php

10.3.6.2 Blade Templates

Blade ist eine Template Engine, die mit Laravel mitgeliefert wird. Blade erleichtert und vereinfacht das Verwenden von PHP Code in HTML, dabei wird der Blade Syntax in normalen PHP Code kompiliert. Blade Dateien werden mit der Extension `.blade.php` erstellt. Damit der Inhalt der einzelnen Seiten dynamisch ist müssen Daten übergeben werden, dies erfolgt über die Controller.

Um nun Inhalt innerhalb eines Blade Files anzuzeigen verwendet Blade doppelt Geschweifte Klammern.

```

1  Hallo, {{ $user->full_name }}.

```

Code 24: example.blade.php

Im Code 24 wird nun auf das übergebene Model User zugegriffen und der Name ausgegeben.

10.3.6.3 Controllers

Es besteht theoretisch die Möglichkeit die komplette Logik für die bearbeitung von Anfragen direkt in den Route Files zu platzieren, dies ist jedoch sehr unübersichtlich und deshalb ist es sinnvoll

diese Logik in Controller Klassen auszulagern. Dabei werden ähnliche Anfragen zusammengeführt, so ist es sinnvoll alle Anfragen wie in Code 23 im gleichen Controller zusammenzuführen.

```

1  <?php
2  namespace App\Http\Controllers;
3
4  use Illuminate\Http\Request;
5  use App\Models\User;
6  use Illuminate\Support\Facades\Hash;
7  use App\Http\Requests\StoreUser;
8  use App\Http\Requests\UpdateUser;
9  use App\Models\Role;
10 use Image;
11
12 class UserController extends Controller
13 {
14     public function index()
15     {
16         $this->authorize('index', User::class);
17         $users = User::orderBy('id', 'asc')->paginate(25, ['*'],
18             ↪ 'users');
19         return view('users.index')->with('users', $users);
20     }
21     . . .

```

Code 25: UserController.php

10.3.6.4 Artisan CLI

Artisan ist ein Kommandozeilen Tool von Laravel und zentraler Bestandteil und stellt eine Fülle an sehr nützlichen Befehlen dem Entwickler zur Verfügung. Mithilfe von Artisan können neue Controller, Models, Migrations und vieles sehr einfach erstellt werden. Da es sehr viele Befehle gibt folgen hier

nur die wichtigsten Befehle:

- **php artisan list**
Zeigt eine Liste aller verfügbaren Befehle an
- **php artisan serve**
Startet einen PHP Development Server unter dem Port 8000
- **php artisan make:model <name>**
Erstellt ein neues Model
- **php artisan make:migration <name>**
Erstellt ein neue Migrations
- **php artisan make:controller <name>**
Erstellt ein neuer Controller
- **php artisan migrate**
Führt die Datenbank Migration Files aus

10.3.6.5 Migrations

Datenbank Migrations sind praktisch Vorlagen wie einzelne Tabellen in der Datenbank auszusehen haben. Migrations erleichtern die Handhabung von SQL Datenbank im Team mithilfe von Source Control wie Git um einiges.

```

1  <?php
2  class CreateUsersTable extends Migration
3  {
4      public function up()
5      {
6          Schema::create('users', function (Blueprint $table) {
7              $table->id();
8              $table->string('first_name');
9              $table->string('last_name');
10             $table->string('email')->unique();
11             $table->string('avatar')->default('default.png');
12             $table->timestamp('email_verified_at')->nullable();
13             $table->string('password');
14             $table->string('last_login_ip')->nullable();
15             $table->timestamp('last_login_at')->nullable();
16             $table->rememberToken();
17             $table->timestamps();
18         });
19     }
20
21     public function down()
22     {
23         Schema::dropIfExists('users');
24     }
25 }
```

Code 26: create_users_table.php

Im Code 26 sind zwei Methoden zu erkennen, up und down, in der up Methode werden neue Tabellen und Spalten in der Datenbank erstellt, bei der down Methode wird alles von der up Methode rückgängig gemacht, in diesem Fall wird die komplette Tabelle gelöscht. Um nun die Datenbank

zu migrieren muss in der Artisan CLI Befehl `php artisan migrate` ausgeführt werden, dabei werden alle Migrations im Verzeichnis `database/migrations` ausgeführt.

10.3.6.6 Eloquent ORM

Eloquent ist ein Object–relational mapping (ORM) d.h. um auf die SQL Datenbank zuzugreifen müssen keine Raw SQL Queries ausgeführt werden. Im Code erscheint die Datenbank als objektorientierte Datenbank und erleichtert somit dem Umgang mit der Datenbank. Dabei entspricht besitzt jede Tabelle in der Datenbank einem dazugehöriges Model, dieses Model wird verwendet um im Code mit der Datenbank zu interagieren. Ein weiter Vorteil ist, dass der Code sehr übersichtlich und leicht lesbar ist, jedoch wird die Ausführungszeit leicht erhöht, was jedoch keine große Rolle bei kleinen- bis mittelgroßen Webseiten spielt.

```
1 <?php
2 User::where('first_name', 'Philipp')->first();
```

Code 27: Eloquent Query

```
1 SELECT * FROM `users` WHERE `first_name` = Philipp
```

Code 28: Raw SQL Query

Beim Vergleich zwischen dem Code 27 und Code 28 ist es erkenntlich, dass der Eloquent Query verständlicher ist. Besonders bei komplexeren Queries ist Eloquent den RAW SQL Queries im Bezug auf die Lesbarkeit stark überlegen.

10.3.6.7 Laravel Sanctum

Laravel Sanctum ist ein First-Party-Package von den Entwicklern von Laravel und ermöglicht eine einfache Authentifizierung mithilfe von API-Tokens. Dabei wird der vom Benutzer erstellte API-Token im HTTP Header `Authorization` mitgeliefert und somit autorisiert. Da es sich hierbei um ein Sicherheitskritisches Modul handelt ist es besonders wichtig, dass dieser Code nahezu Fehlerfrei ist und dies wird durch kontinuierliche Updates und Patches garantiert.

10.4 Lokale Entwicklungsumgebung mit Laragon

Für die Programmierung des Webinterfaces müssen zuerst einige Vorkehrungen getroffen werden, dazu zählt zu einem die Installation von benötigter Software und deren konfiguration.

10.4.1 Benötigte Software

- **Laragon** (<https://laragon.org>)
Beinhaltet mehrere Softwarepakete die für die Entwicklung notwendig sind.
 - Apache HTTP Server
 - MySQL
 - PHP
- **phpMyAdmin** (<https://www.phpmyadmin.net>)
Webinterface für MySQL
- **Composer** (<https://getcomposer.org>)
Paketmanager für PHP
- **Git** (<https://git-scm.com>)
Versionskontrolle
- **Visual Studio Code** (<https://code.visualstudio.com>)
Quelltext-Editor

10.4.2 Konfiguration von PHP

Um PHP Befehle von der Kommandozeile auszuführen muss die Installation zuerst in den Windows Path Variables hinzugefügt werden.

Dies erfolgt durch die Advanced System Settings ► Environment Variables ► System Variables. Dort kann nun die Path Variable editiert werden und der Pfad hinzugefügt werden in welchem die `php.exe` liegt.

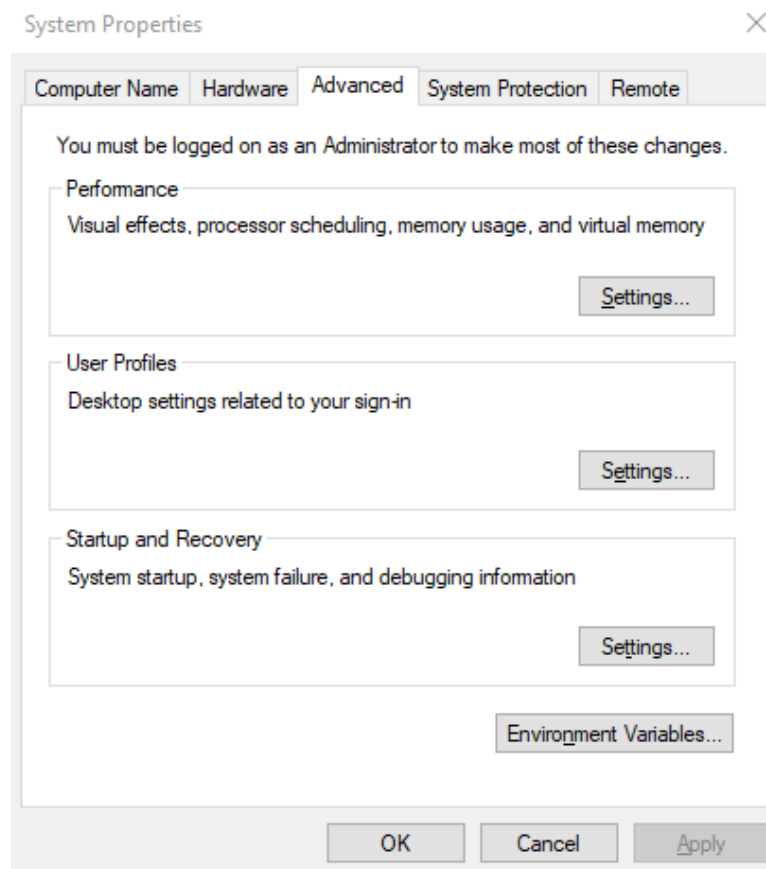


Abbildung 23: Advanced System Settings

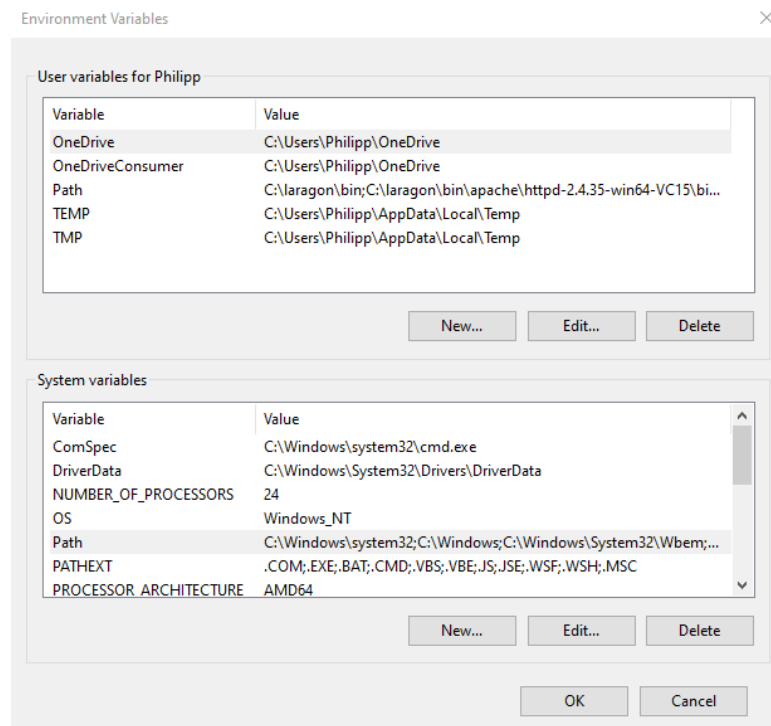


Abbildung 24: System Variables

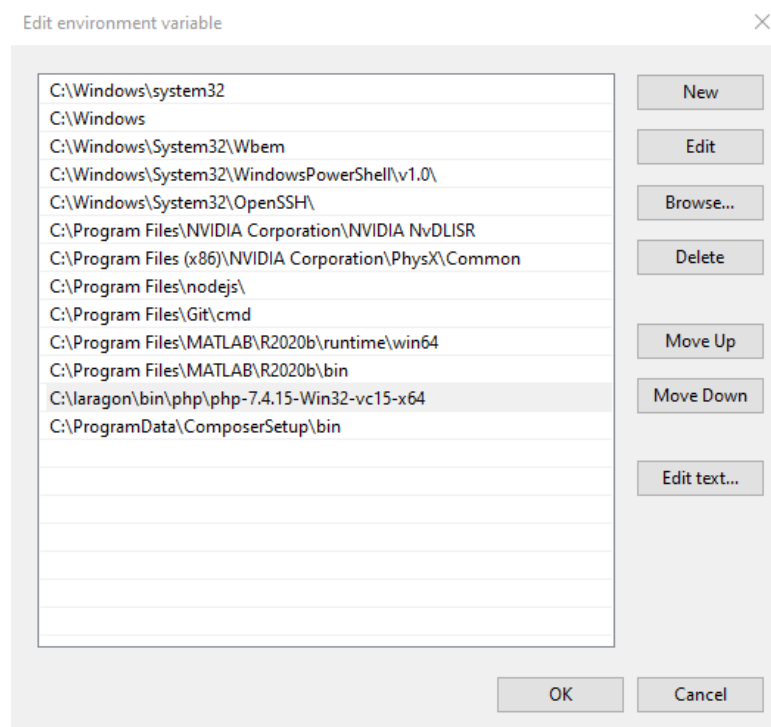


Abbildung 25: Environment Variables

Die korrekte konfiguration kann durch die Kommandozeile geprüft werden, dort muss das Befehl

php -v ausgeführt werden. Dabei ist zu beachten, dass nach dem hinzufügen der Path Variable die gewählte Kommandozeile neu gestartet werden muss.

```
C:\Users\Philipp>php -v
PHP 7.4.15 (cli) (built: Feb  2 2021 20:47:45) ( ZTS Visual C++ 2017 x64 )
Copyright (c) The PHP Group
Zend Engine v3.4.0, Copyright (c) Zend Technologies
```

Abbildung 26: PHP Version

Somit ist PHP korrekt konfiguriert.

10.4.3 Installation von phpMyAdmin

phpMyAdmin ist ein Tool, welches den Umgang mit MySQL Datenbanken mit einem Webinterface erleichtert. Die aktuellste Version lässt sich von <https://www.phpmyadmin.net/downloads> downloaden. Dieses Archiv muss entpackt werden und ausgehend vom Laragon Root Verzeichnis in das Verzeichnis /etc/apps kopiert werden. Um die Installation zu überprüfen muss der Apache HTTP Server und der MySQL Server gestartet werden, nun sollte bei einer korrekten Installation das Webinterface von phpMyAdmin unter <http://localhost/phpmyadmin> erreichbar sein.

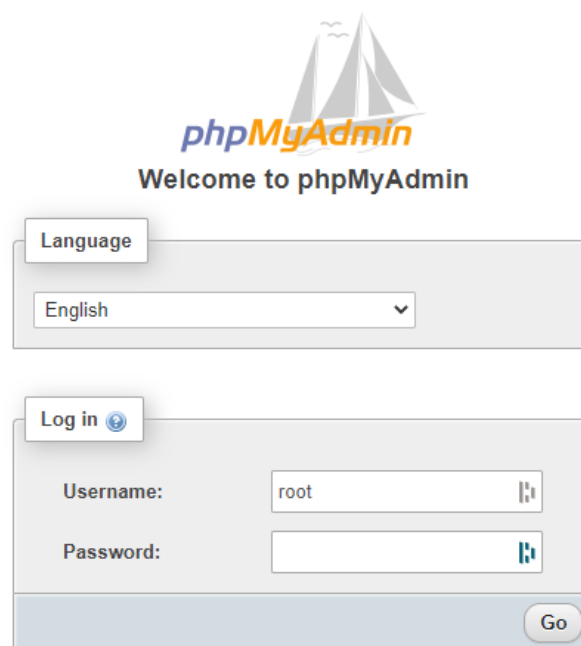


Abbildung 27: phpMyAdmin Webinterface

Es ist nicht notwendig ein Passwort einzugeben, da Standardmäßig kein Passwort gesetzt wird.

10.5 Lokale Entwicklungsumgebung mit WSL und Docker

10.5.1 Benötigte Software

- **Docker** (<https://www.docker.com>)
Ermöglicht Isolation von Anwendungen mit Containervirtualisierung
- **WSL** (<https://docs.microsoft.com/en-us/windows/wsl>)
Kompatibilitätsschicht für Linux Anwendungen unter Windows 10
- **Visual Studio Code** (<https://code.visualstudio.com>)
Quelltext-Editor

10.5.2 Installation von WSL

Zuerst müssen einige Einstellungen in Windows getroffen werden um später eine Linux Distribution herunterzuladen können. Diese Befehle können über die Kommandozeile mit Administrativen Rechten ausgeführt werden.

```
1  dism.exe /online /enable-feature  
    ↪ /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

Code 29: WSL Feature Feature aktivierens

10.5.2.1 2. Schritt: Virtual Machine Aktivieren

```
1  dism.exe /online /enable-feature /featurename:VirtualMachinePlatform /all  
    ↪ /norestart
```

Code 30: Virtual Machine Feature aktivieren

Nach diesem Schritt ist ein Neustart des Computers notwendig.

10.5.2.2 3. Schritt: Linux Kernel Update

Nun muss ein Linux Kernel Update installiert werden, die aktuelle Version ist unter <https://aka.ms/wsl2kernel> zu finden.

10.5.2.3 4. Schritt: WSL 2

Nach dem Neustart des Computers sollte es nun möglich sein WSL 2 als Version auszuwählen.

```
1 wsl --set-default-version 2
```

Code 31: WSL 2 auswählen

10.5.2.4 5. Schritt: Linux Distribution herunterladen

Zuletzt kann eine Linux Distribution aus dem Windows Store heruntergeladen werden, in diesem Fall Debian (<https://www.microsoft.com/de-de/p/debian>).

10.5.3 Installation von Docker

Die aktuellste Version von Docker Desktop für Windows lässt sich am einfachsten über die offizielle Website von Docker herunterladen (<https://docker.com>). Nach der Installation muss noch die WSL Integration aktiviert werden. Dazu muss in den Einstellungen unter Resources ► WSL Integration und dort muss der Haken bei `Enable integration with my default WSL distro` gesetzt werden und die installierte Linux Distribution muss unten aktiviert werden.

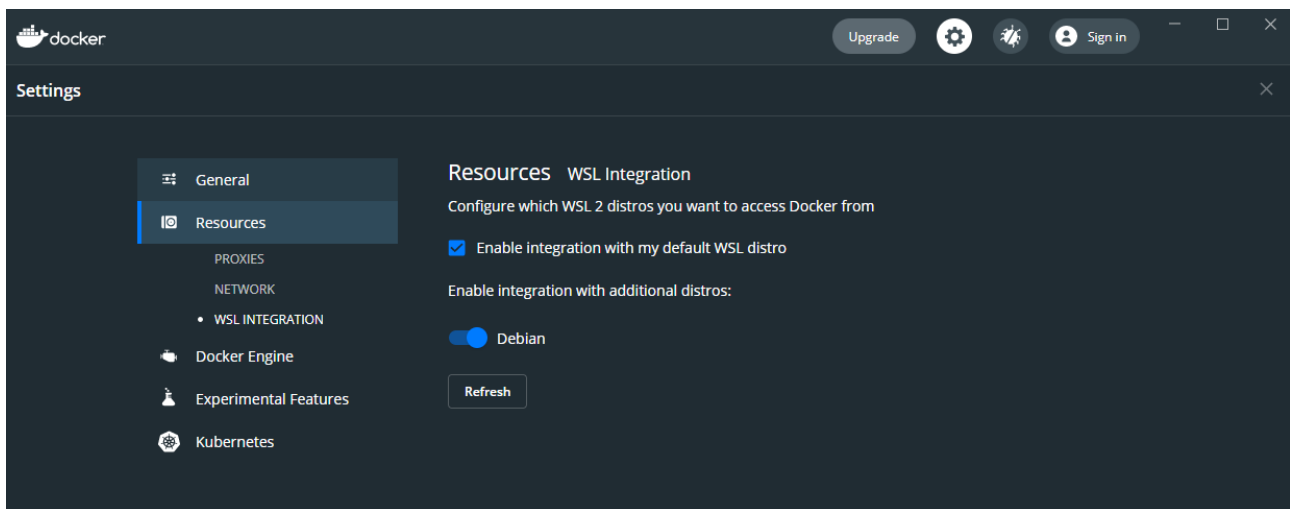


Abbildung 28: Docker WSL Integration

Somit ist die Lokale Entwicklungsumgebung mit WSL und Docker abgeschlossen, die benötigte Software wird später automatisch durch Laravel Sail in einem Docker Container installiert.

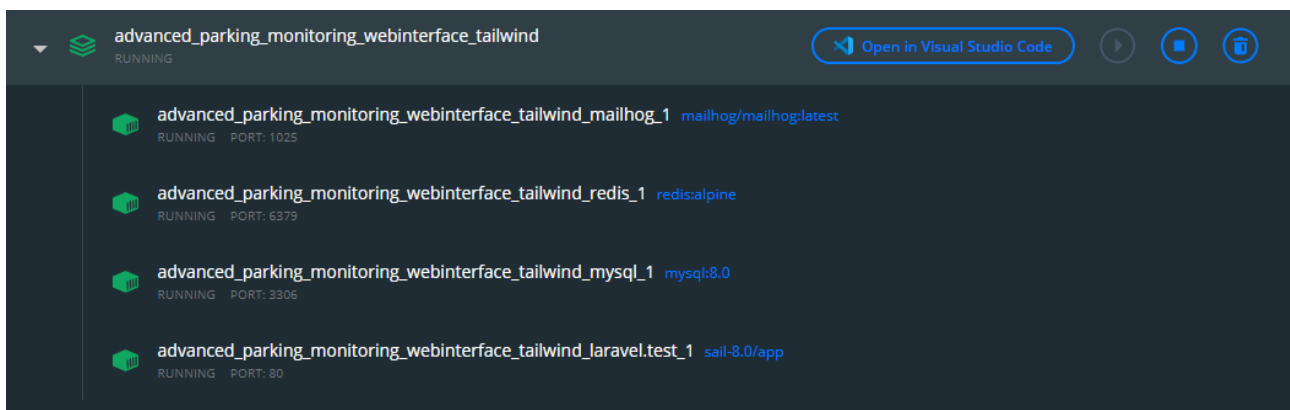


Abbildung 29: Docker Container Steuerung

Es ist somit möglich die Services welche im Container in der Linux Distribution laufen über die Docker Desktop Anwendung zu steuern.

10.6 Production Server

Der Production Server bzw. der Live Server ist der Server wo sich die Webanwendung befindet und die Endbenutzer zugreifen, dieser Server wird auch einfach mit Production abgekürzt. Der Production Server ist in diesem Fall ein Virtual Private Server mit dem Betriebssystem Debian 10, welcher bei einem Internet-Hosting Unternehmen mit Sitz in Deutschland gehostet wird.

10.6.1 Benötigte Software

Für den Live Server wird der sogenannte „LAMP“ Stack verwendet. LAMP steht dabei für die Software **L**inux, **A**pache, **M**ySQL und **P**HP.

- **Apache Web Server** (<https://httpd.apache.org>)
HTTP Server
- **MariaDB** (<https://mariadb.org>)
Fork von MySQL
- **PHP** (<https://www.php.net>)
Hypertext Preprocessor (PHP)
- **phpMyAdmin** (<https://www.phpmyadmin.net>)
Webinterface für MySQL
- **Composer** (<https://getcomposer.org>)
Paketmanager für PHP
- **Git** (<https://git-scm.com>)
Versionskontrolle

10.6.2 Installation des LAMP Stacks

Bevor die Software Pakete installiert werden sollte die Linux Software/Update Repository geupdatet werden.

```
1 apt-get update && apt-get upgrade
```

Code 32: Respositorys updaten

10.6.2.1 Apache

Nun kann der Apache Web Server installiert werden.

```
1 apt install apache2
```

Code 33: Apache installieren

Die Installation kann nun leicht überprüft werden indem man im Browser die IP bzw. die dazugehörige Domain öffnet, in diesem Fall: (<http://dev.philipp-kraft.com>).

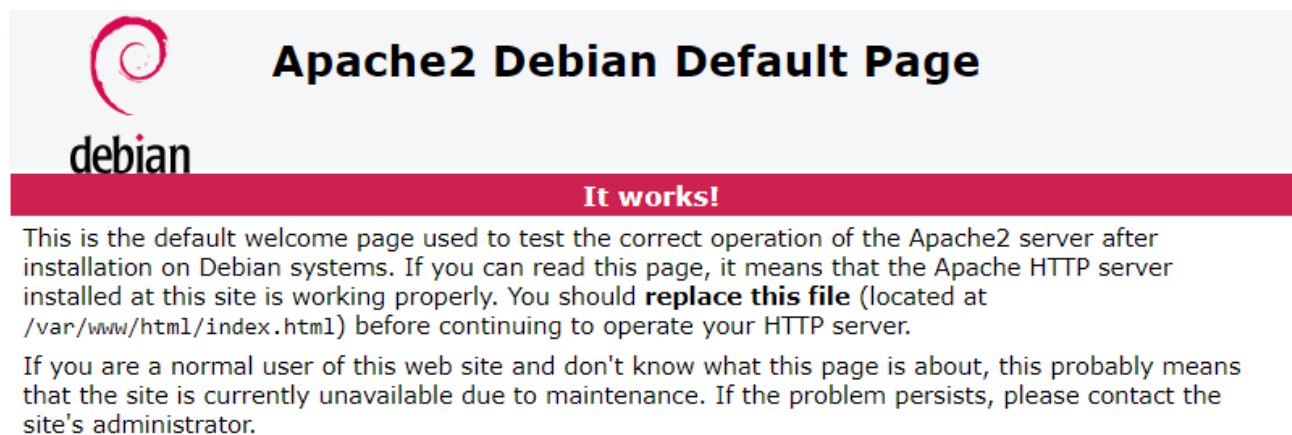


Abbildung 30: Debian Default Page

Erscheint die Debian Default Page ist Apache korrekt installiert.

10.6.2.2 PHP

Neben PHP werden auch einige PHP Extensions benötigt.

```
1 apt install wget php php-cgi php-mysqli php-pear php-mbstring php-gettext
  ↪ libapache2-mod-php php-common php-phpseclib php-mysql
```

Code 34: PHP installieren

Die Installation kann einfach mit dem Befehl `php -v` überprüft werden.

10.6.2.3 MariaDB


```
1 apt install mariadb-server
```

Code 35: MariadB installieren

Nun muss MariaDB noch konfiguriert werden.

```
1 mysql_secure_installation
```

Code 36: MariaDB Secure Installation

Dabei wird dem root MySQL User ein Passwort gesetzt, Anonyme Benutzer gelöscht und es werden Test Datenbanken gelöscht.

Nun wird ein neuer Benutzer mit root Berechtigungen erstellt.

```
1 mysql
2 GRANT ALL ON *.* TO 'admin'@'localhost' IDENTIFIED
3 BY 'password' WITH GRANT OPTION;
4 flush privileges;
5 exit
```

Code 37: MariaDB konfiguration

10.6.2.4 phpMyAdmin

Die aktuelle Version von phpMyAdmin kann von (<https://www.phpmyadmin.net/downloads>) bezogen werden und mit dem wget Befehl heruntergeladen werden.

```
1 wget https://files.phpmyadmin.net/phpMyAdmin/5.0.4
2 /phpMyAdmin-5.0.4-all-languages.tar.gz
```

Code 38: phpMyAdmin Download

Anschließend muss das Archiv entpackt werden.

```
1 tar xvf phpMyAdmin-5.0.4-all-languages.tar.gz
```

Code 39: phpMyAdmin Entpacken

Als nächstes muss das entpackte Archiv in einen anderen Pfad verschoben werden und zusätzlich müssen einige Rechte und Verzeichnisse angepasst werden.

```
1 mv phpMyAdmin-5.0.4-all-languages /usr/share/phpmyadmin
2 mkdir -p /var/lib/phpmyadmin/tmp
3 chown -R www-data:www-data /var/lib/phpmyadmin
4 mkdir /etc/phpmyadmin/
```

Code 40: phpMyAdmin Rechte und Verzeichnisse

Nun muss eine Konfigurations Datei erstellt werden und dort muss ein Blowfish Secret²² angegeben werden und den Pfad für ein Temporäres Verzeichnis.

```
1 cp /usr/share/phpmyadmin/config.sample.inc.php
  ↪ /usr/share/phpmyadmin/config.inc.php
```

Code 41: phpMyAdmin Konfigurationsdatei erstellen

und am Ende dieser Datei müssen folgende zwei Zeilen eingefügt werden.

```
1 $cfg['blowfish_secret'] = 'H20xcGXxflSd8JwrwVlh6KW6s2rER63i';
2 $cfg['TempDir'] = '/var/lib/phpmyadmin/tmp';
```

Code 42: phpMyAdmin Blowfish Secret und TempDir

Zuletzt muss der Apache Web Server konfiguriert werden.

²²32 Zeichen String für Cookie-Authentifizierung

Im Verzeichnis `/etc/apache2/sites-available` muss eine neue Konfiguration angelegt werden `phpmyadmin.conf`.

```
1 Listen 9000
2
3 <VirtualHost *:9000>
4     ServerName localhost
5
6     <Directory /usr/share/phpmyadmin>
7         AllowOverride None
8         Require all granted
9     </Directory>
10
11     DocumentRoot /usr/share/phpmyadmin
12
13     ErrorLog ${APACHE_LOG_DIR}/phpmyadmin.error.log
14     CustomLog ${APACHE_LOG_DIR}/phpmyadmin.access.log combined
15 </VirtualHost>
```

Code 43: phpmyadmin.conf

Nun kann diese Virtual Host Konfigurations Datei aktiviert werden und danach muss der Apache Web Server neu gestartet werden.

```
1 a2ensite phpmyadmin
2 systemctl restart apache2
```

Code 44: Virtual Host aktivieren

Diese Konfiguration ermöglicht es, dass das Webinterface von phpMyAdmin über den Port 9000 (`http://dev.philipp-kraft.com:9000`) erreichbar ist und nicht wie Standardmäßig vorgesehen über das Verzeichnis `/phpmyadmin` (`http://dev.philipp-kraft.com/phpmyadmin`), dies bietet

einen Sicherheitsvorteil.

10.6.2.5 Webinterface Virtual Host

Nun muss noch eine Virtual Host Konfiguration für das Webinterface selbst erstellt werden.

```
1 <VirtualHost *:80>
2   ServerAdmin webmaster@localhost
3   DocumentRoot /var/www/apm/public
4
5   <Directory />
6     Options FollowSymLinks
7     AllowOverride All
8   </Directory>
9
10  <Directory /var/www/apm>
11    Options Indexes FollowSymLinks MultiViews
12    AllowOverride All
13    Order allow,deny
14    allow from all
15  </Directory>
16
17  ErrorLog ${APACHE_LOG_DIR}/error.log
18  CustomLog ${APACHE_LOG_DIR}/access.log combined
19 </VirtualHost>
```

Code 45: apm.conf

Diesmal wird auf den Standard HTTP Port 80 gehört und dieser führt in das Verzeichnis `/var/www/apm/public`. Zusätzlich werden noch Directives gesetzt, damit das Standard `.htaccess` File von Laravel richtig funktionieren kann.

10.6.2.6 Installation von Composer

Die Installation von Composer gestaltet sich relativ einfach.

```
1 wget -O composer-setup.php https://getcomposer.org/installer
```

Code 46: Download Composer Installer

Nun muss das Setup ausgeführt werden und damit das `composer` Befehl Global verfügbar ist wird Composer in den Pfad `/usr/local/bin` verschoben.

```
1 php composer-setup.php --install-dir=/usr/local/bin --filename=composer
```

Code 47: Composer Setup

10.6.3 Deployment mit Github Actions

Unter Deployment versteht man die automatische Installation von Software, in diesem Fall auf einem Linux Server. Erreicht wird das durch zwei Bash Scripts und mit Github Actions (<https://github.com/features/actions>).

10.6.3.1 Git Setup

Sollte auf dem Server noch kein Git installiert sein, lässt sich das wie folgt installieren.

```
1 apt install git
```

Code 48: Git Installation

Im Verzeichnis `/var/www/apm` soll sich später das Webinterface befinden, deshalb muss in diesem Pfad Git konfiguriert werden. Dazu wird die Remote URL konfiguriert.

```
1 git config remote.origin.url 'https://{TOKEN}@github.com/  
2 Philipp-Kraft/Advanced_Parking_Monitoring_Webinterface.git'
```

Code 49: Git Remote Origin

Da beim Github Account eine Zwei-Faktor-Authentisierung verwendet wird muss die Authentifizierung mit einem Personal access token erfolgen. Dieser kann unter <https://github.com/settings/tokens> erstellt werden, der erstellte Token kann dann einfach in der URL eingefügt werden.

10.6.3.2 Deploy Script

Das Deploy-Script wird auf der Lokalen Entwicklermaschine ausgeführt. Das Script wechselt in den Production Branch und merged mit dem Main Branch, dieser Push in den Production Branch löst dann die Github Action aus.

```
1  #!/bin/sh  
2  set -e  
3  
4  #vendor/bin/phpunit  
5  
6  (git push) || true  
7  
8  git checkout production  
9  git merge main  
10  
11 git push origin production  
12  
13 git checkout main
```

Code 50: deploy.sh

10.6.3.3 Server Deploy Script

Das Server Deploy Script `server_deploy.sh` versetzt die Laravel Applikation in den Wartungsmodus und lädt sich vom `deploy` Branch den Code auf den Server herunter, danach werden einige Befehle ausgeführt.

```
1  #!/bin/sh
2  set -e
3
4  echo "Deploying application ..."
5
6  # Enter maintenance mode
7  php artisan down
8
9  # Update codebase
10 git fetch origin deploy
11 git reset --hard origin/deploy
12
13 # Install dependencies based on lock file
14 composer install --no-interaction --prefer-dist --optimize-autoloader
15
16 # Migrate database
17 php artisan migrate:refresh --seed
18
19 # Clear cache
20 php artisan optimize
21
22 # Exit maintenance mode
23 php artisan up
24
25 echo "Application deployed!"
```

Code 51: serverdeploy.sh

10.6.3.4 Github Action

Github Actions ist ein Projekt von Github, welches es ermöglicht Automatisierungen in den Bereichen Entwicklung, Testing und Deployment durchzuführen.

Als erstes muss ein Workflow erstellt werden, dieser wird im Root-Verzeichnis des Projekts erstellt APM\.github\workflows\main.yml.

```
1  name: Deploy Laravel app
2
3  on:
4    push:
5      branches: [ production ]
6
7  jobs:
8    deploy:
9      runs-on: ubuntu-latest
10     steps:
11       - uses: actions/checkout@v2
12         with:
13           token: ${ secrets.PUSH_TOKEN }
14       - name: Set up Node
15         uses: actions/setup-node@v1
16         with:
17           node-version: '12.x'
18       - run: npm install
19       - run: npm run production
20       - name: Commit built assets
21         run: |
22           git config --local user.email "action@github.com"
23           git config --local user.name "GitHub Action"
```



```

24     git checkout -B deploy
25     git add -f public/
26     git commit -m "Build front-end assets"
27     git push -f origin deploy
28 - name: Deploy to production
29   uses: appleboy/ssh-action@master
30   with:
31     username: root
32     host: dev.philipp-kraft.com
33     password: ${ secrets.SSH_PASSWORD }
34     script: 'cd /var/apm && ./server_deploy.sh && chown -R
      ↪ www-data:www-data /var/apm && chmod -R 755 /var/apm && chmod
      ↪ -R 777 /var/apm/storage'
```

Code 52: main.yml

Dieser Workflow setzt einen Ubuntu Server in der Cloud auf und baut dort die Assets zusammen, damit die Downtime auf dem Production Server möglichst gering ist. Danach wird eine SSH Verbindung zum Production Server aufgebaut und dort wird das Bash-Script `server_deploy.sh` ausgeführt. Gleichzeitig werden einige Berechtigungen angepasst.

In der Repository muss nun noch das SSH-Passwort und der Personal access token hinterlegt werden. Dies geschieht über **Settings ► Secrets**. Dort können nun über **New repository secrets** die Secrets hinterlegt werden.



Abbildung 31: Action Secrets

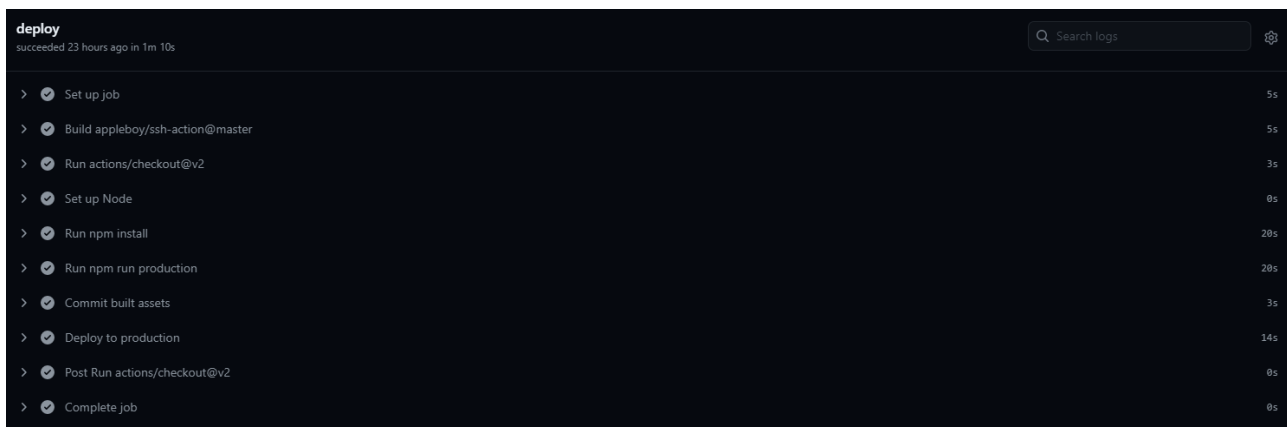


Abbildung 32: Github Action Übersicht

10.7 Grundlegender Aufbau Frontend

Da das Ziel ist, dass das Frontend²³ des Webinterfaces möglichst Modular aufgebaut ist und so wenig Code wie möglich wiederholt wird. Ermöglicht wird dies durch das aufbauen der Seite mithilfe von Components.

10.7.1 Components

Components sind praktisch kleine Bausteine aus denen die komplette Seite aufgebaut ist. Components sind kein natives Feature von HTML/CSS oder PHP, diese Funktion wird von der Template Engine Blade bereitgestellt, deshalb werden diese auch oft Blade Components genannt.

10.7.1.1 Anonymous Components

Es gibt viele verschiedene Möglichkeiten Components zu erstellen und verschiedene Konventionen am, einfachsten sind aber die Anonymous Components, diese haben den Vorteil, dass diese in einer Datei verwaltet werden können und somit sehr einfach zu handhaben sind.

10.7.1.2 Components erstellen

²³Präsentationsebene in Form der grafischen Benutzeroberfläche

Das Erstellen von einem Component wird nun anhand eines Buttons gezeigt. Da dieser als Anonymous Component angelegt wird muss dieser mit keiner Klasse assoziiert werden, es wird einfach im Pfad `resources\views\components` ein Ordner mit dem Namen `buttons` angelegt und darin ein Blade File mit dem Namen `primary.blade.php`. Dort kann nun der gewünschte HTML Code platziert werden.

```
1 <button type="submit" class="inline-flex items-center px-4 py-2  
  ↪ bg-apm-blue...">  
2   {{ $slot }}  
3 </button>
```

Code 53: primary.blade.php

Im Code 53 ist eine Variable mit dem Namen `$slot` verwendet worden. Diese Variable wird später beim verwenden automatisch mit dem Inhalt zwischen dem HTML Element ersetzt.

10.7.1.3 Components verwenden

Es stellt sich nun die Frage wie man dieses erstellte Component nun verwendet. Die Blade Components verwenden den gleichen Syntax wie ein normales HTML Element, mit dem Unterschied dass ein `x-` vor dem Namen des Components angeführt werden muss. Da sich der vorhin erstellte Component in einem Ordner befindet muss das auch angegeben werden, dabei wird kein Slash wie üblich um einen Pfad anzugeben verwendet sondern ein Punkt, es muss auch keine Extensions angegeben werden.

```
1 <x-buttons.primary>Press me!</x-buttons.primary>
```

Code 54: Verwendung eines Button Components

10.7.1.4 Attribute übergeben

Auch wenn viele Components ohne Problem überall ohne Veränderung verwendet werden können, ist es gewünscht bei manchen Components beispielweise eine zusätzliche Klasse anzugeben um die Größe des Elements zu verändern. Erreicht wird das mit der `attributes` Variable im Blade File des Components, da aber oft schon Attribute definiert sind ist es möglich mit der `merge` Methode die Attribute zusammenzuführen.

```
1 <button {{ $attributes->merge(['type' => 'submit', 'class' => 'inline-flex  
  ↳ items-center px-4 py-2 bg-apm-blue...']) }}>  
2   {{ $slot }}  
3 </button>
```

Code 55: Modularer Button Component

Somit ist dieser Button Component nun vollständig Modular.

10.7.2 Layouts

Da auf den meisten Seiten des Webinterfaces fast das gleiche Layout beibehaltet ist es sinnvoll diesen Inhalt in ein Component umzuwandeln. Auch Layouts sind Components.

Es gibt im Webinterface folgende Layouts:

- **app.blade.php**

Layout für Allgemeine Seiten

- **admin.blade.php**

Layout für die Administrativen Seiten mit einer Sidebar und Page Header

- **display.blade.php**

Besonderes Layout für die Display Seiten

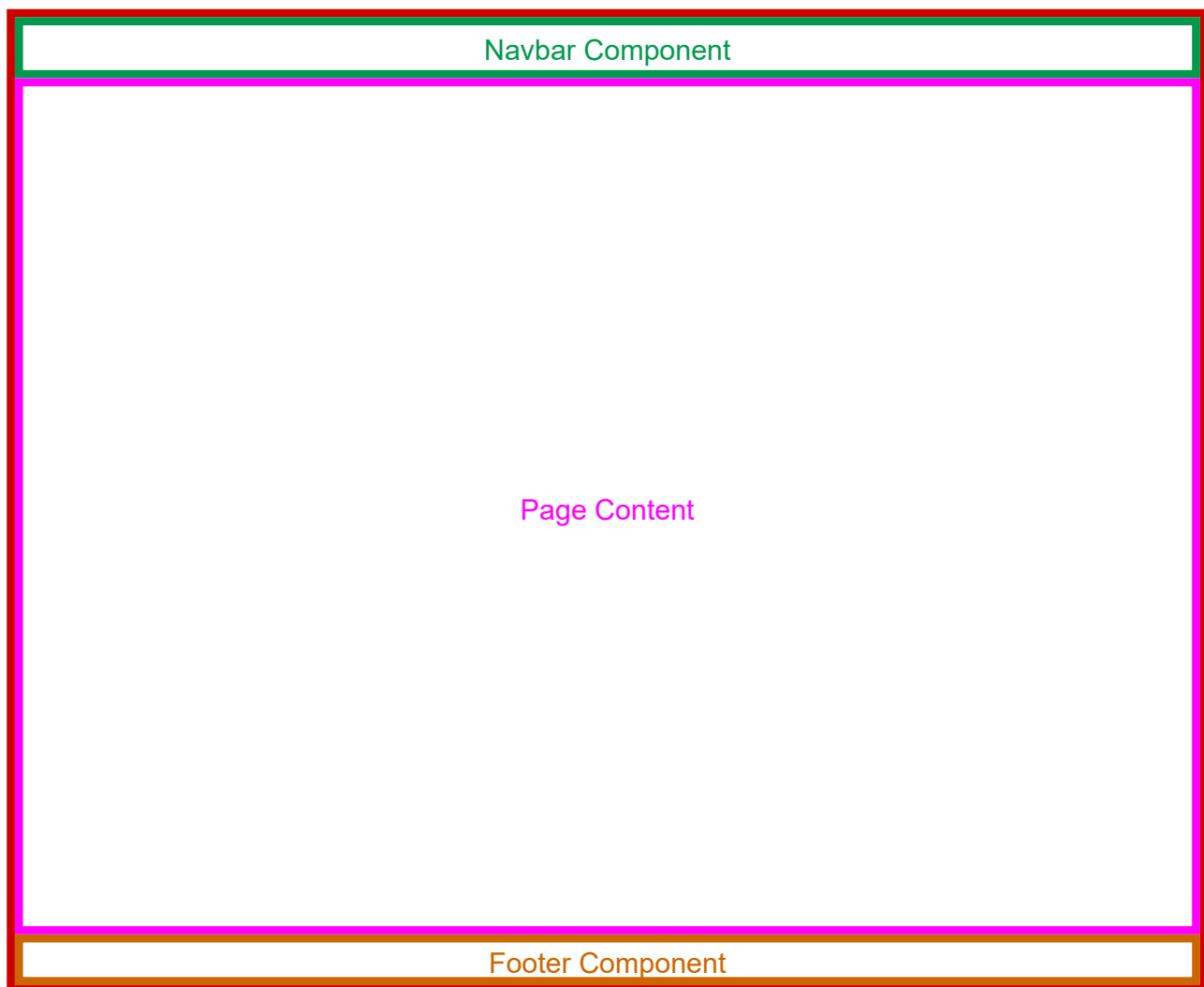
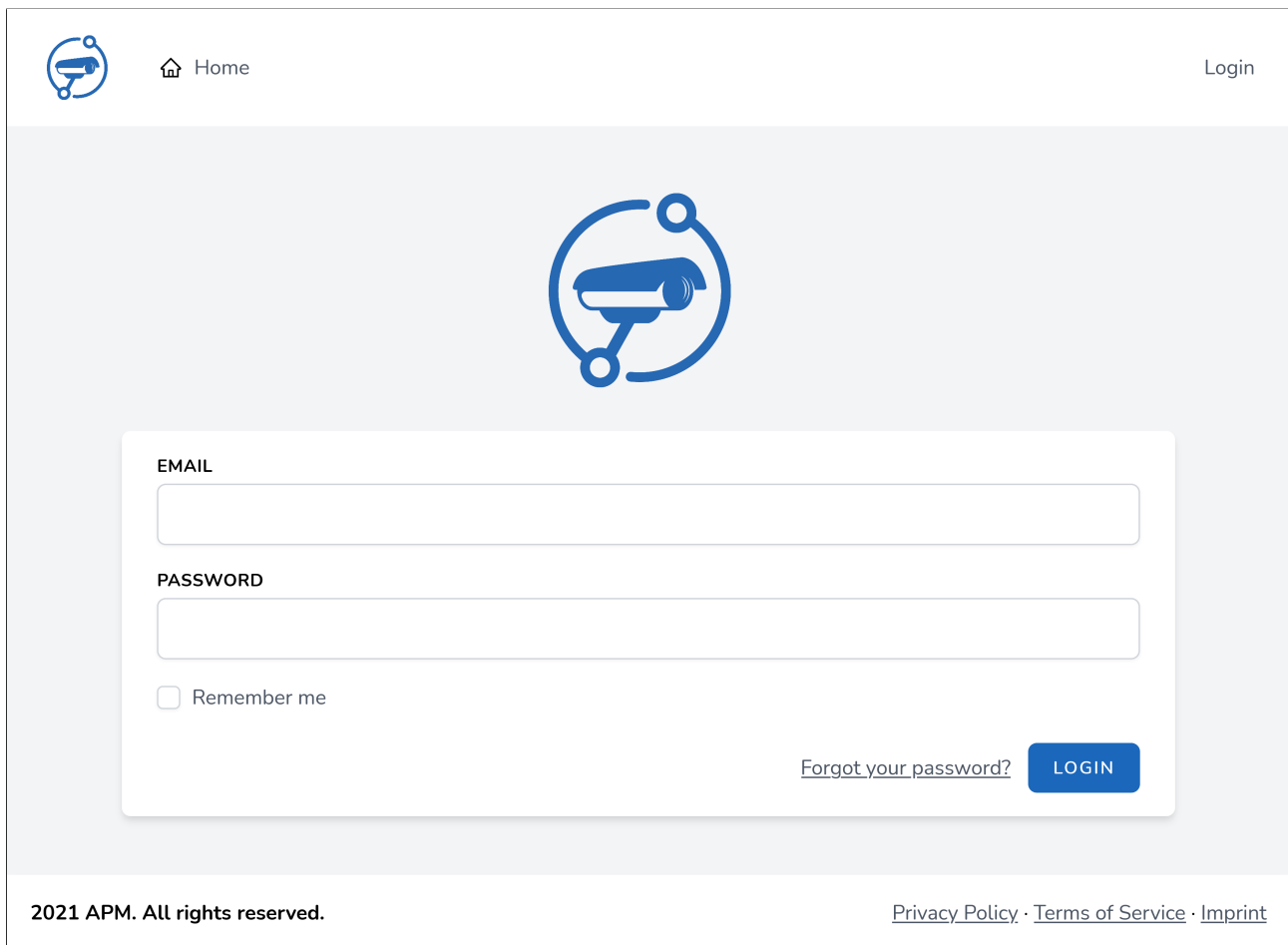


Abbildung 33: App Layout

Das App Layout ist das Standardmäßige Layout, es besteht aus drei Components, der Navigation Bar, dem Page Content und aus einem dynamische Footer.



The image shows a login page with an app layout. At the top, there is a header bar with the HTL Rankweil logo on the left, the text 'HTBLuVA Rankweil' and 'Höhere Lehranstalt für Elektronik und Technische Informatik' in the center, and the htl logo with 'bildung mit zukunft' on the right. Below the header, the main content area has a light gray background. On the left, there is a small circular icon with a camera and a magnifying glass. In the center, there is a large circular icon with a camera and a magnifying glass. On the right, there is a 'Login' link. Below the large icon, there is a white login form with the following elements: an 'EMAIL' label above a text input field, a 'PASSWORD' label above a text input field, a 'Remember me' checkbox, a 'Forgot your password?' link, and a blue 'LOGIN' button. At the bottom of the page, there is a footer bar with the text '2021 APM. All rights reserved.' on the left and links for 'Privacy Policy', 'Terms of Service', and 'Imprint' on the right.

Home Login

EMAIL

PASSWORD

☐ Remember me

[Forgot your password?](#) **LOGIN**

2021 APM. All rights reserved. [Privacy Policy](#) · [Terms of Service](#) · [Imprint](#)

Abbildung 34: Login Seite mit App Layout

In der Abbildung 34 wird das App Layout verwendet. Dabei sind die drei Components erkennbar.

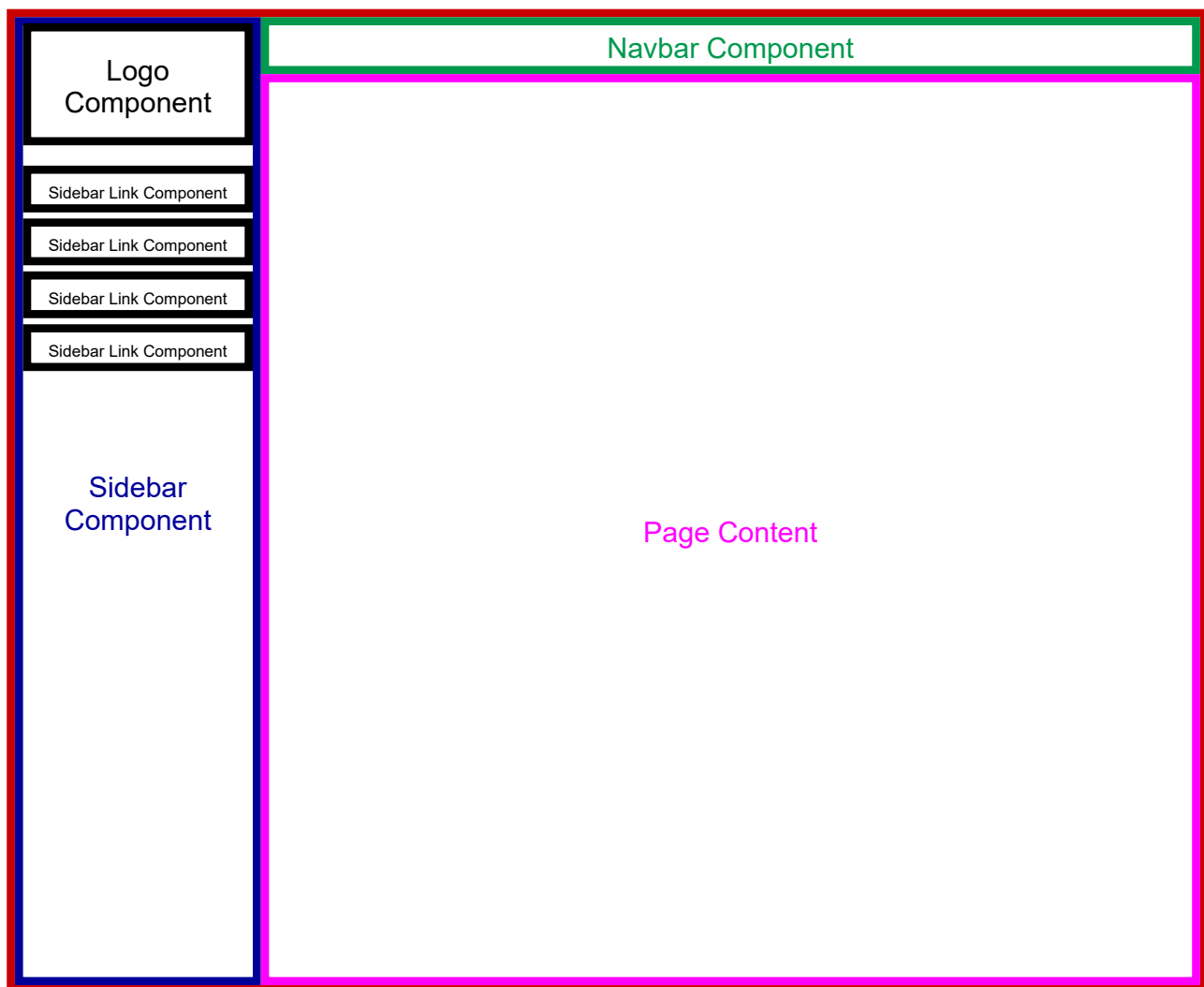


Abbildung 35: Admin Layout

Das Admin Layout besitzt eine zusätzliche Sidebar mit Links zu verschiedenen Administrativen Seiten, im Vergleich zum Standardmäßigen besitzt das Admin Layout keinen Footer und eine leicht veränderte Navbar²⁴.

²⁴Navigation Bar engl. für Navigationsleiste

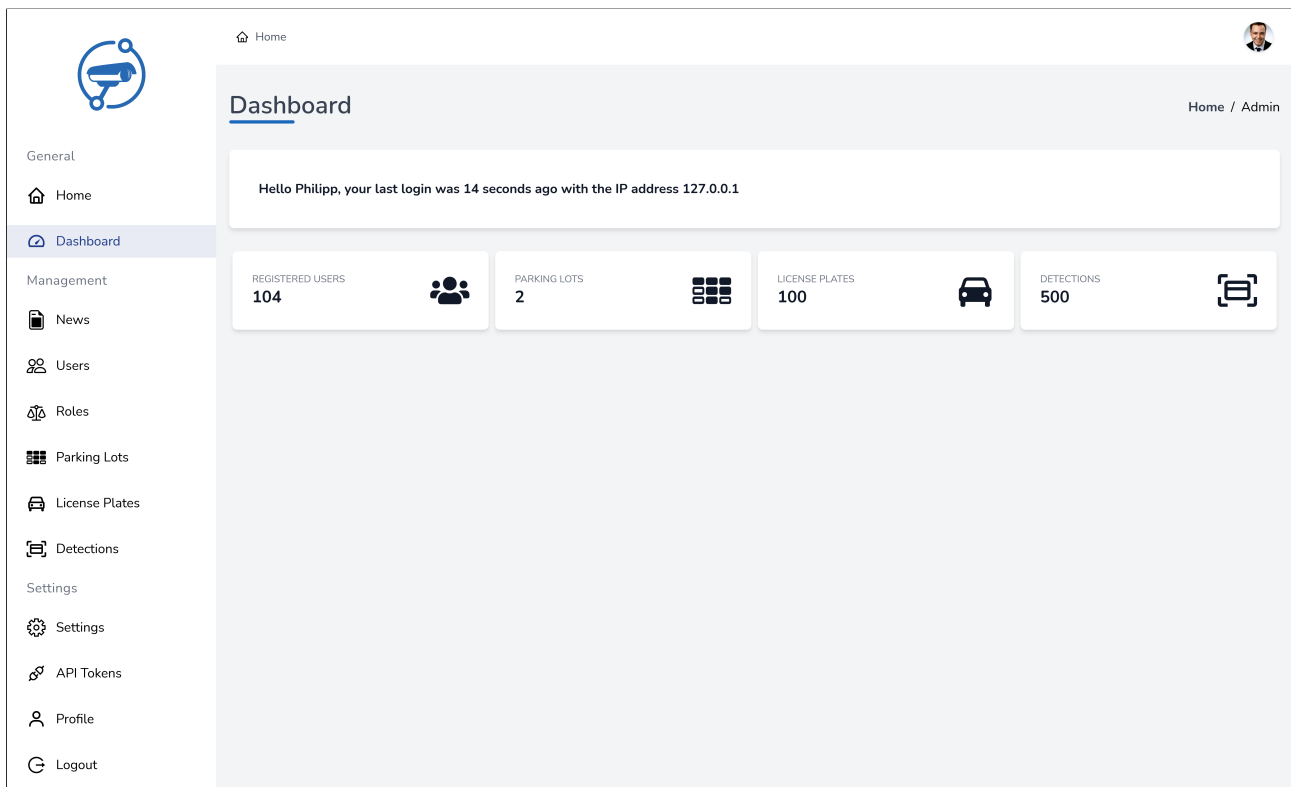


Abbildung 36: Dashboard mit Admin Layout

Im Dashboard wird das Admin Layout verwendet, somit befindet sich Links vom Page Content nun eine Sidebar mit weiteren Links zu anderen Administrativen Seiten.

Um nun die Funktion besser zu verstehen wird nun auf den Code einzelner Components genauer eingegangen.

10.7.3 Navbar

Die Navigationsleiste auch Navbar genannt ist ein zentrales Element der Webseite um verschiedene Seiten anzusteuern. Dabei veränderte sich diese dynamisch, je nachdem ob man eingeloggt ist oder ob ein bestimmtes Feature wie das Registrieren von neuen Benutzern deaktiviert ist.


```

1  @if (!(\Request::is('admin/*') || \Request::is('admin')))
2      <a class="flex title-font font-medium items-center text-gray-900 mr-8">
3          <x-application-logo class="max-h-12" />
4      </a>
5  @if (!Auth::guest())
6      <a href="{{ route('dashboard') }}">{{ __('Dashboard') }}</a>
7  @endif
8  @endif

```

Code 56: Ausschnitt 1 navigation.blade.php

Der Ausschnitt 1 regelt das Aussehen der Navigationsleiste wenn ein Benutzer eingeloggt ist. Dabei wird kein Logo angezeigt wenn der Benutzer eine administrative Seite öffnet, da das Logo bereits in der Sidebar vorhanden ist. Ebenfalls wird auf nicht administrativen Seiten ein zurück zum Dashboard Link angezeigt.

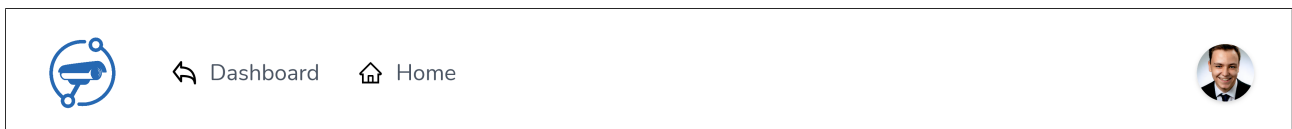


Abbildung 37: Navbar Variante 1



Abbildung 38: Navbar Variante 2

```

1  @if (Auth::guest())
2      <div class="flex items-center">
3          <a href="{{ route('login') }}">{{ __('Login') }}</a>
4  @if (Route::has('register'))
5      <a href="{{ route('register') }}">{{ __('Register') }}</a>
6  @endif
7  </div>
8  @endif

```

Code 57: Ausschnitt 2 navigation.blade.php

Im zweiten Ausschnitt geht es um die Login/Register Links, diese werden nur nicht eingeloggten Benutzern angezeigt. Ob der Register Link angezeigt wird ist abhängig von den getätigten Einstellungen in den Seiten Einstellungen.



Abbildung 39: Navbar Variante 3

10.7.4 Sidebar

Die Sidebar ist das zentrale Element für administrative Benutzer des Webinterfaces. Über die Sidebar sind alle wichtigen Seiten auf einen Blick zu sehen und erreichen, dabei spielt es keine Rolle ob auf dem Smartphone oder auf dem Desktop, denn die Sidebar wurde so implementiert, dass sie auf kleineren Auflösungen eingeklappt wird und nur auf Wunsch mit einem Klick auf das Hamburger-Menü-Symbol²⁵ in der Navigationsleiste geöffnet, somit wird wichtiger Platz gespart.

²⁵Drei waagrechte Striche ähneln einem Hamburger

10.7.5 Footer

10.8 Funktionen

10.8.1 Dashboard

10.8.2 Login- und Registersystem

10.8.3 News

10.8.4 Benutzerverwaltung

10.8.5 Rechte- und Rollenverwaltung

10.8.6 Parkplatzverwaltung

10.8.7 Kennzeichenverwaltung

10.8.8 Erkennungsverlauf

10.8.9 Seiten Einstellungen

10.8.10 API Schlüssel

10.8.11 Displays

10.8.12 Notifications

10.8.13 Profil

10.8.14 Lokalisierung

10.9 Performance und Sicherheit

10.9.1 Form Validation

10.9.2 Authentifizierung

10.9.3 Authentisierung

10.9.4 SQL Injection

10.9.5 Cross-Site-Scripting

10.9.6 Lighthouse

11 Zusammenfassung und Ausblick

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

12 Anhang

Abbildungsverzeichnis

1	Vor Bilateraler Filterung	18
2	Nach Bilateraler Filterung	18
3	Graustufenbild	19
4	Binäres Bild nach Thresholding	19
5	Binäres Bild nach Thresholding	20
6	Nach Erosion	20
7	Ablaufdiagramm der Kennzeichenerkennung	24
8	Beispiel einer Anwendung von Matplotlib	30
9	GPIO-Pins des Raspberry Pi	35
10	Raspberry Pi	37
11	Raspberry Pi mit Kamera	38
12	Installation der Spule	41
13	LTSPice Blockbild zweier RL-Glieder	43
14	Stromkurven zweier RL-Glieder	44
15	RL-Oszillator mit NE555	45
16	RL-Oszillator mit NE555 Zeitsignale	46
17	Vergleich des NE555-Oszillators bei unterschiedlichen L	47
18	Eddy Currents in einem Leiter	48
19	Parallelschwingkreis in LTSpice	49
20	Gedämpfte Schwingung	50
21	Einfache HTML Seite	55
22	Einfache HTML Seite mit CSS	57
23	Advanced System Settings	67
24	System Variables	68
25	Environment Variables	68
26	PHP Version	69
27	phpMyAdmin Webinterface	69
28	Docker WSL Integration	72
29	Docker Container Steuerung	72

30	Debian Default Page	74
31	Action Secrets	83
32	Github Action Übersicht	84
33	App Layout	87
34	Login Seite mit App Layout	88
35	Admin Layout	89
36	Dashboard mit Admin Layout	90
37	Navbar Variante 1	91
38	Navbar Variante 2	91
39	Navbar Variante 3	92

Tabellenverzeichnis

Codeverzeichnis

1	PIP Installation von Jupyter Notebook	26
2	PIP Installation von Numpy	27
3	PIP Installation von OpenCV	28
4	Installation benötigter Tools und Bibliotheken für OpenCV	28
5	Klonen von OpenCV von GitHub	29
6	Kompilieren von OpenCV	29
7	Abschließende Installation von OpenCV	29
8	PIP Installation von Matplotlib	30
9	PIP Installation von Keras	31
10	PIP Installation von Tensorflow	31
11	Benötigte Tools für Tensorflow	32
12	Tensorflow von GitHub herunterladen	32
13	Entpacken von Tensorflow	32
14	Zusätzlich erforderliche Librarys	32
15	Kompilieren von Tensorflow	33
16	Abschließen der Installation von Tensorflow	33
17	PIP Installation von Scikit-learn	34
18	PIP Installation von Requests	34
19	PIP Installation von Gpiozero	35
20	PIP Installation von Picamera	36
21	index.html	54
22	style.css	56
23	web.php	61
24	example.blade.php	61
25	UserController.php	62
26	create_users_table.php	64
27	Eloquent Query	65
28	Raw SQL Query	65
29	WSL Feature Feature aktivierens	70

30	Virtual Machine Feature aktivieren	70
31	WSL 2 auswählen	71
32	Respositorys updaten	73
33	Apache installieren	74
34	PHP installieren	74
35	Mariadb installieren	75
36	MariaDB Secure Installation	75
37	MariaDB konfiguration	75
38	phpMyAdmin Download	75
39	phpMyAdmin Entpacken	76
40	phpMyAdmin Rechte und Verzeichnisse	76
41	phpMyAdmin Konfigurationsdatei erstellen	76
42	phpMyAdmin Blowfish Secret und TempDir	76
43	phpmyadmin.conf	77
44	Virtual Host aktivieren	77
45	apm.conf	78
46	Download Composer Installer	79
47	Composer Setup	79
48	Git Installation	79
49	Git Remote Origin	80
50	deploy.sh	80
51	serverdeploy.sh	81
52	main.yml	83
53	primary.blade.php	85
54	Verwendung eines Button Components	85
55	Modularer Button Component	86
56	Ausschnitt 1 navigation.blade.php	91
57	Ausschnitt 2 navigation.blade.php	92

Abkürzungsverzeichnis

PK Philipp Kraft

DK Dennis Köb

SB Samuel Bleiner

APM Advanced Parking Monitoring

HTML Hypertext Markup Language

W3C World Wide Web Consortium

CSS Cascading Style Sheets

PHP Hypertext Preprocessor

JS JavaScript

DOM Document Object Model

MVC Model-View-Controller

ORM Object–relational mapping

CLI Command-line interface

SQL Structured Query Language

Literaturverzeichnis

Einstein, Albert. »Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]«. In: *Annalen der Physik* 322.10 (1905), S. 891–921. doi: <http://dx.doi.org/10.1002/andp.19053221004>.