

# Diplomarbeit

## Gesichtserkennung



AUTOR: REI HOXHA

# Inhaltsverzeichnis

<b>1</b>	<b>Gesichtserkennung</b>	<b>1</b>
1.1	Allgemeines . . . . .	1
1.2	Technische Lösung . . . . .	2
1.2.1	Hardware und Aufbau . . . . .	2
1.2.2	Software . . . . .	5
1.3	Herausforderungen, Probleme, und deren Lösung . . . . .	14
1.4	Projektmanagement und Controlling . . . . .	14
1.5	Ergebnisse . . . . .	16

# Kapitel 1

## Gesichtserkennung

Am Beginn wird eine allgemeinere Übersicht dieses Diplomarbeit Kapitels gemacht bzw. dargestellt. Damit wird es verstanden, worum es überhaupt in diesem Teil geht.

### 1.1 Allgemeines

Die Gesichtserkennung ist der Teil des Systems, mit dem ich mich beschäftige. Es wird durch einen Vergleich überprüft, ob die Person vorher schon registriert worden ist oder nicht, und ob sein Gesichtsdaten schon am Server existieren oder nicht. Nur wenn die Vergleichsergebnisse positiv sind, darf die Person rein kommen. Der Person werden die Ergebnisse durch Anzeigern kommuniziert. Dieses Teil des Projektes erfordert eine Arbeit mit Datenbanken, Gesichtsvergleichsalgorithmen und mit vielen System Tests. Teil meiner Aufgabe ist auch der Aufbau des Systems und alles was mit Hardware zu tun hat. Alle Hardware Komponenten werden in den folgenden Kapiteln klar, verständlich und deutlich erklärt. Eine grobe Skizze des Systems ist in Abb.1.1 zu sehen.

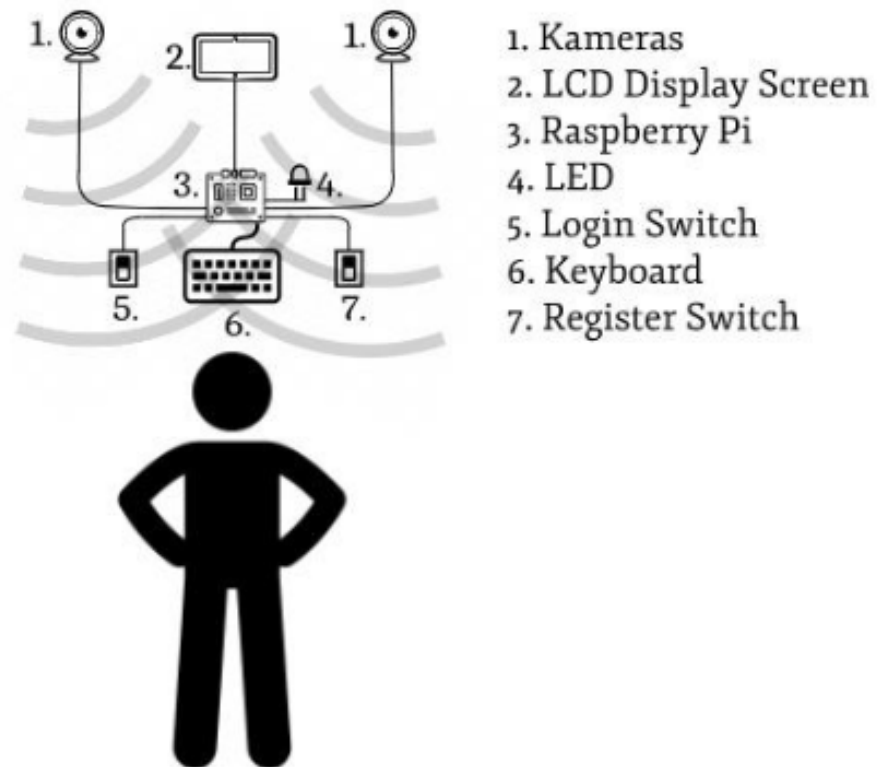


Abbildung 1.1: Grobe Skizze des Systems

## 1.2 Technische Lösung

In diesem Section wird eine feinere und detaillierte Übersicht im Bezug auf der technischen Lösung gemacht. Alles was mit Technik zu tun hat, wird hier erklärt und umfasst.

### 1.2.1 Hardware und Aufbau

Als erstens wird der Aufbau des Systems beschrieben, zusammen mit allen verwendeten Komponenten, die zu verwenden sind. Ohne die Hardware würde nichts funktionieren, weil die Software ohne die Hardware einfach nicht funktionieren kann.

#### Bauteile und HW-Komponenten

Die Bauteile, die für dieses System verwendet geworden sind, sind die folgende:

##### 1. Steckboard bzw. Steckplatine

Die Steckplatine ist die wichtigste Komponente der Hardware unseres Systems. Sie wird für die Entstehung der elektrischen Verbindung von verschiedenen elektrischen Bauteile benötigt, um elektrische Schaltungen zu bauen oder um ver-

schiedene Tests und Experimenten zu machen. In dieser Steckplatine werden alle anderen Komponenten platziert, damit die Verbindung erstellt werden kann und damit das System laufen kann.

## 2. *Kabel bzw. Leiter*

Damit die verschiedenen Komponenten, die in der Steckplatine platziert sind, miteinander verbunden werden können, braucht man unbedingt Kabel. Mithilfe von Kabeln können elektrische Impulse und Signale fließen, damit die Energie und die Information übertragen werden. Die verwendeten Kabel sind aus Kupfer und vom Typ Male-Male als auch vom Typ Male-Female. Die Kabeln vom Typ Male-Female werden verwendet, um die Verbindungen zwischen die Elemente in der Steckplatine und den Raspberry Pi zu ermöglichen. Auf der anderen Seite werden die Male-Male Kabeln verwendet um die Verbindungen innerhalb der Steckplatine zu ermöglichen.

## 3. *LEDs*<sup>1</sup>

LEDs sind elektronische Halbleiter Elemente, die Licht produzieren können, wenn sie Spannung kriegen. Ein LED besteht aus zwei Beinen. Das längere Bein ist die Anode, die den Pluspol symbolisiert. Das andere Bein ist die Kathode, und symbolisiert den Minuspol. Durch die Beine wird der Kontakt mit der Steckplatine erstellt.

## 4. *Widerstand*

Ein Widerstand ist ein elektrisches Bauteil, das zur Reduzierung von Strom verwendet wird, damit ein Gleichgewicht zwischen Strom und Spannung gesichert werden kann. Die Einheit ist Ohm.

## 5. *Taster*

Ein Taster wird wie ein Button gedrückt, mit dem Zweck Impulse oder Signale zu schicken. Im Gegenteil zu einem Schalter wird der Taster nach der Betätigung wieder in der Basiszustand zurückgestellt. Ein Plusleiter, Minusleiter und ein Datenleiter sind bei einem Taster vorhanden.

## 6. *Raspberry Pi*

Raspberry Pi ist ein Minicomputer, der in diesem Projekt den normalen Computer ersetzt. Der verwendete Raspberry, Version 3, hat 4 USB-Anschlüsse, einem Netzteil, eine SD-Karte, 16 GPIO<sup>2</sup> Pins und einem VGA Schnittstelle. Die 32-Bit SD-Karte ist das wichtigste Element, weil dort alle Daten und Informationen gespeichert sind.

## 7. *Bildschirm*

Ein Bildschirm ist eine Anzeige, die für die visuelle Darstellung von verschiedenen Informationen oder Daten(wie Videos, Fotos, Statistiken usw.) verwendet wird. Ein Bildschirm wird zu den heutigen Zeiten sehr häufig verwendet, aufgrund der hohen Benutzerfreundlichkeit, die angeboten wird.

---

<sup>1</sup>Light Emitting Diode

<sup>2</sup>Generated Input Output

## 8. Tastatur

Eine Tastatur ist ein Input Gerät, dass durch das Eindrücken von Tastern den Benutzer die Eingabe von Daten oder Befehle ermöglicht.

## 9. Kamera

Letztens werden 2 Kameras benötigt, die die Fotos der Gesichter der Personen machen werden. Sie werden auch im Raspberry integriert bzw. mit dem Raspberry verbunden. Die Kameras sind auf Typ Aukey.

## Schaltplan und Erklärung des Aufbaus

Die Hardware Komponenten werden in einer Steckplatine platziert. Dort werden die Verbindungen mit den anderen Komponenten sowie mit dem Raspberry Pi erstellt. Die elektrische Schaltung wird durch einen Schaltplan beschrieben. Dieser Schaltplan wurde mit Hilfe eines Programms, das Fritzing heißt, erstellt und spielt eine sehr wichtige Rolle bei der Organisation und Planung des Schaltkreises. Der Schaltplan ist auf Abb.1.2 zu sehen.

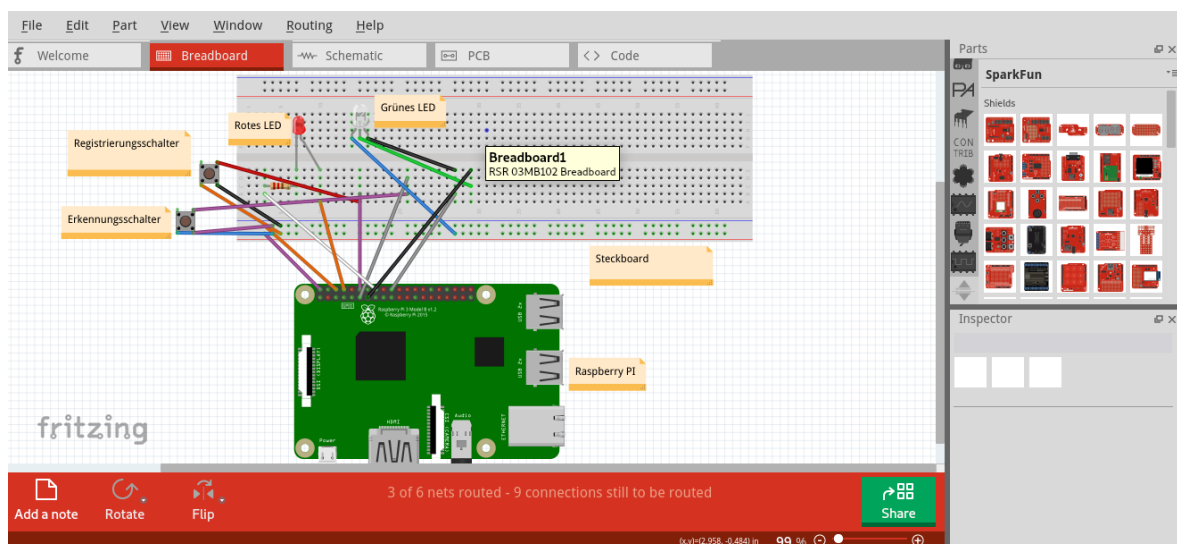


Abbildung 1.2: Schaltplan des Systems

Wie der Schaltplan zeigt, besteht das System aus zwei LEDs, zwei Tastern, einem Widerstand, eine Steckplatine und einem Raspberry Pi. Sehr wichtig für den Aufbau der Schaltung sind die GPIO Pins. Diese Pins sind in dem Raspberry Pi platziert und können als Input, als Output oder Spannung Pins verwendet werden. Der erste Taster dient für die Gesichtsregistrierung und besteht aus einem Pluspol, einem Datenleiter und aus dem Minuspol. Der Pluspol Anschluss wird mittels Steckplatine mit dem 5-Volt Pin des Raspberry Pi verbunden, während der Minuspol Anschluss mit dem Minuspol der Steckplatine verbunden wird. Der Datenanschluss ist mit dem GPIO18 Pin verbunden. Wie bei dem Registrierungs-Taster wird auch bei dem Erkennungstaster<sup>3</sup> der Minuspol Anschluss mit einem Pin des Minusbereichs der Steckplatine verbunden. Der Pluspol Anschluss gehört zu dem 5-Volt Pin des Raspberry PI, während der

<sup>3</sup>Auch als Login Taster genannt

Datenanschluss mit dem GPIO17 Pin verbunden sind. Die rote LED wird verwendet wenn die Registrierung oder Erkennung der Benutzer im System nicht erfolgreich war. Das längere Bein (die Anode) wird mit dem GPIO23 Pin des Raspberry verbunden, während die Kathode mit dem Minuspol Bereich der Steckplatine verbunden wird. Die grüne LED ist eine RGB (Red Green Blue) LED. Diese LED kann die Farbe ändern, und wird im Fall einer erfolgreichen Registrierung oder Erkennung des Benutzers im System verwendet. Sie hat im Gegensatz zu der normalen LED 3 Anschlüsse. Der Minus Anschluss wird in dem Minuspol Bereich der Steckplatine eingeschlossen und der Pluspol Anschluss wird mit dem Pluspol Bereich der Steckplatine verbunden. Mit dem GPIO27 Pin des Raspberry wird der Datenanschluss verbunden. Die GPIO Pins sind extrem wichtig für die Integration der Tastern, LEDs und der anderen Bauteile in dem technischen und logischen Teil bzw. in der Software und in den verwendeten Skripten.

## 1.2.2 Software

Nachdem der Aufbau und die verwendete Hardware des Systems beschrieben wurden, ist die Software dran. In diesem Unterkapitel wird alles was mit der logische Teil der Umsetzung zu tun hat: das verwendete Betriebssysteme, Programmiersprachen, Frameworks, Technologien und Planungsmethoden. Es wird jede Aufgabe zusammen mit der zugehörigen Lösung im Detail beschrieben und jedes programmiertes Skript erklärt.

### Verwendete Betriebssysteme, Programmiersprachen, Frameworks, Technologien und Planungsmethoden

In diesem Teil des Projekts, das Gesichtserkennung heißt, werden für die Umsetzung verschiedene Technologien, Frameworks und Planungsmethoden verwendet.

#### 1. Betriebssystem und Programmiersprache

- Linux (Debian)

Linux ist ein von die wichtigsten Betriebssystemen die zur Verfügung stehen. Linux wird meistens für Programmierungs- und Konfigurationszwecke. Die wichtigsten Bestandteile eines Betriebssystems sind der Bootloader, der Kernel, und das Init System. Linux wurde für die Umsetzung dieses Projekts gewählt, obwohl es auch andere Möglichkeiten wie zum Beispiel Microsoft Windows gab. Die Gründe dieser Wahl sind die folgende:

- Open Source und freie Software

Linux wurde genutzt, weil es komplett kostenlos benutzt werden kann und weil es ein Open Source Betriebssystem ist. Vorteile eines Open Source Betriebssystems sind die niedrige Kosten, die hohe Sicherheit, die Transparenz und die Zuverlässigkeit. Was noch erwähnt zu werden ist, ist die gute Reputation der Open Source Software, weil nur hoch qualifizierte Entwickler mit ihnen sich beschäftigen.

- Hohe Sicherheit

Für dieses Projekt ist Sicherheit sehr wichtig, und Linux bietet eine hohe

Sicherheit. Es gibt keine Viren und sehr kleine Absturz Wahrscheinlichkeiten. Ein Backup ist auch sehr leicht machbar.

- Vielfältigkeit an Möglichkeiten und Benutzerfreiheit

Linux wurde auch aufgrund der angebotenen Benutzerfreiheit und Benutzerfreundlichkeit verwendet. Beliebige Systemkonfigurationen, verschiedene Gesichtserkennungspakete(OpenCV<sup>4</sup>) und andere Programme oder Technologien(Fritzing) sind von Linux leichter und günstiger erhältlich als bei anderen Betriebssystemen. Endlich ist der Raspberry PI auch kompatibler mit Linux. Als Linux Distribution wurde Debian gewählt. Gründe dafür sind die große Menge der angebotenen Pakete, die hohe Geschwindigkeit und die schnelle Fehlerkorrektur. [9]



Abbildung 1.3: Debian[1]

- Python

Die gewählte Programmiersprache ist Python. Der wichtigste Grund dieser Wahl ist die einfache Verwendung der OPEN-CV Pakete für die Gesichtserkennung Algorithmen. Im Vergleich zu C werden die OpenCV Bibliotheken und Programmen viel leichter und schneller eingebunden. Python hat auch Vorteile im Bezug auf Skripten Einbindung oder Skripten Aufruf, weil ein oder zwei Output Parametern reichen würden, um zwei Skripten miteinander zu verbinden(Variablen, Funktionen usw. importieren). Es wurde Python Version 3 verwendet.



Abbildung 1.4: Python[2]

## 2. Frameworks

In diesem Projekt werden einige Frameworks verwendet, die weiter unten beschrieben werden.

- Open CV

Das wichtigste Framework für dieses Projekt ist OpenCV. OpenCV ist eine Softwarebibliothek, die für Computer-Vision und maschinelles Lernen

---

<sup>4</sup>Open Computer Vision



verwendet wird. Die Bibliothek verfügt über mehr als 2500 optimierte Algorithmen, die sowohl klassische als auch moderne Computer Vision- und maschinelle Lernalgorithmen umfassen. Diese Algorithmen können verwendet werden, um Gesichter zu registrieren und zu erkennen, um Objekte zu identifizieren, menschliche Handlungen in Videos zu klassifizieren und Kamerabewegungen zu verfolgen. Dieses Framework wurde deswegen gewählt, weil die Vielfältigkeit der angebotenen Optionen und Paketen einfach größer ist als bei anderen Frameworks. Ein anderer Vorteil ist das OpenCV Open-Source ist. Die größte Herausforderung ist die lange und komplizierte Installation auf Linux. [10]



Abbildung 1.5: Open CV[3]

- Git  
Git ist ein Versionsverwaltungssystem, das für das Back-Up des Systems verwendet wurde. Dieses Framework ist ein verteiltes Versionsverwaltungssystem, das heißt, die unterschiedlichen Versionen der Dateien werden effizient gespeichert, in dem die ganze Repository vom Server kopiert wird. Dieses erlaubt dem Benutzer lokal zu arbeiten, und alle Änderungen nachträglich am Server zu aktualisieren. Nicht nur das, sondern auch die hohe Geschwindigkeit der Fehlerfindung und die perfekte Unterstützung von großen Projekten, in dem alle Teammitgliedern auf alle Daten Zugriff haben, waren die Gründe, warum Git für das Back-Up des Systems gewählt wurde. [12]



Abbildung 1.6: Git[4]

### 3. Technologien

Das System braucht auch ein paar Technologien für die Umsetzung. Unter Technologien versteht man die Programme, die Systeme und die spezielle Komponenten, die benutzt worden sind.

- Raspberry Pi

Wie es vorher erwähnt wurde in dem Gesichtserkennung Teil des Projekts statt ein Computer ein Raspberry Pi verwendet. Was ein Raspberry Pi ist wurde in dem Kapitel 7.2.1 erklärt. Die wichtigsten Gründe dieser Wahl werden aber erst jetzt genannt. Der kleinere Platzbedarf und die hohe Flexibilität des Raspberry PI sind wesentliche Vorteile. Die Bestandteile bewegen sich nicht, und es gibt keinen Lärm. Was sehr beliebt ist, ist, dass der Raspberry PI gleichzeitig für die Hardware(GPIO PINS) und die Software(wie ein normaler Computer) verwendet werden kann. Er unterstützt Linux(Raspbian) sehr gut. Ein der wichtigsten Gründe dieser Wahl ist auch, dass im Fall eines Fehlers ein Raspberry PI sehr schnell und problemlos geändert oder repariert werden kann. (Gibt es ein Problem mit der SD-Karte, dann muss sie geändert werden, und dann kann die Hardware wieder ganz normal genutzt werden. Wenn es ein Hardware Problem gibt, dann muss die alte SD-Karte in eine neue SD-Karte kloniert werden und nur ein anderes Raspberry verwendet werden. Somit sind alle Daten gespeichert und es ist möglich, mit Hilfe des anderen Raspberry Pi, sie zuzugreifen)



Abbildung 1.7: Raspberry Pi[5]

- Fritzing

Fritzing ist ein Programm das für die graphische Darstellung des Schaltkreises des Systems verwendet wurde. Durch dieses Programm wurde auch der Schaltplan des Systems erstellt(Abb.1.2). Fritzing bietet eine sehr große Menge von elektronischen Komponenten und ein PCB, eine Steckplatine und einen Schematic View. In diesem Fall wurde die Breadboard View gewählt, weil dort alles übersichtlicher ist. Was noch gut ist, ist das Fritzing kostenlos im Linux angeboten wird, was für Windows nicht der Fall ist.



Abbildung 1.8: Fritzing[6]

- MySQL und MariaDB

Um die Testdatenbanken zu erstellen und die Benutzerfotos und Benutzerdaten zu speichern, wurde die berühmteste Open-Source relationales Datenbank System, MySQL verwendet. Zusammen mit MySQL wurde auch die Pakete MariaDB auf dem System installiert, damit Python mit MySQL durch einer Schnittstelle verbunden wird. Mithilfe der MariaDB werden die ganzen Tabellen mit Daten gefüllt.



Abbildung 1.9: Mysql[7]



Abbildung 1.10: Mariadb[8]

#### 4. Planungsmethode

- Structed Software Design

Structed Software Design ist die Methode, die zur Planung des Systems verwendet wurde. Diese Software-Architektur ermöglicht den Entwurf, das Implementierungskonzept und das Integrationskonzept des Systems zu planen. Obwohl Structed Design großer Aufwand benötigt, hat diese Methode viele Vorteile, die bei der Wahl der Planungsmethode eine große Rolle gespielt haben: Structed Design verwendet Logik, ist von allem verstehbar, beschreibt alles ohne Abgrenzungen und ganz detailliert. Die wichtigsten Elemente die in der Structed Design beschrieben werden sind:

- Bubbels

Die wichtigsten Arbeitsbereiche des Systems (Gesichtsregistrierung, Gesichtserkennung, Bildverarbeitung)

- Schnittstellen

Sind die Interfaces, die die Verbindung zwischen die Bestandteile machen

- Nachrichten

Sind die geschickten Signale innerhalb des Systems.

- Flüsse  
Zeigen die Daten die innerhalb des Systems fließen.
- Modulen  
Bestandteil des Systems, das innerhalb liegt.
- Terminatoren  
Ähnlich mit der Modulen, aber befinden sich außerhalb des Systems und beeinflussen es von außen.
- definierte Abgrenzungen  
Mithilfe der Abgrenzungen wird vom Systemhersteller genau definiert, was er nicht machen wird, sei es auf Zeit-, Schwierigkeits-, oder Komplexitätsverbundene Gründe. Diese Abgrenzungen dienen dazu, dem Auftraggeber genau zu zeigen, was er erwarten kann, damit es keine Missverständnisse gibt. [11]

Mithilfe der Structed Software Design wurden das Big Picture des Systems zusammen mit seine notwendige Iterationen, die mehr an das Gesichtserkennungsteil konzentriert sind, erstellt. (1.Ebene des Gesichtserkennungsteils)

### **Lösungsweg- Beschreibung und Erklärung**

Der Lösungsweg für die Umsetzung der Aufgaben der Gesichtserkennung ist streng mit einer guten vorherigen Planung verbunden. Deshalb wurde nicht nur das Big Picture(Großes Sicht des Systems nach Außen), die in die vorherigen Kapiteln beschrieben wurde erklärt, sondern auch die Erste Ebene des Gesichtserkennungsteils. Die Big Picture sieht man unter Abb.1.11

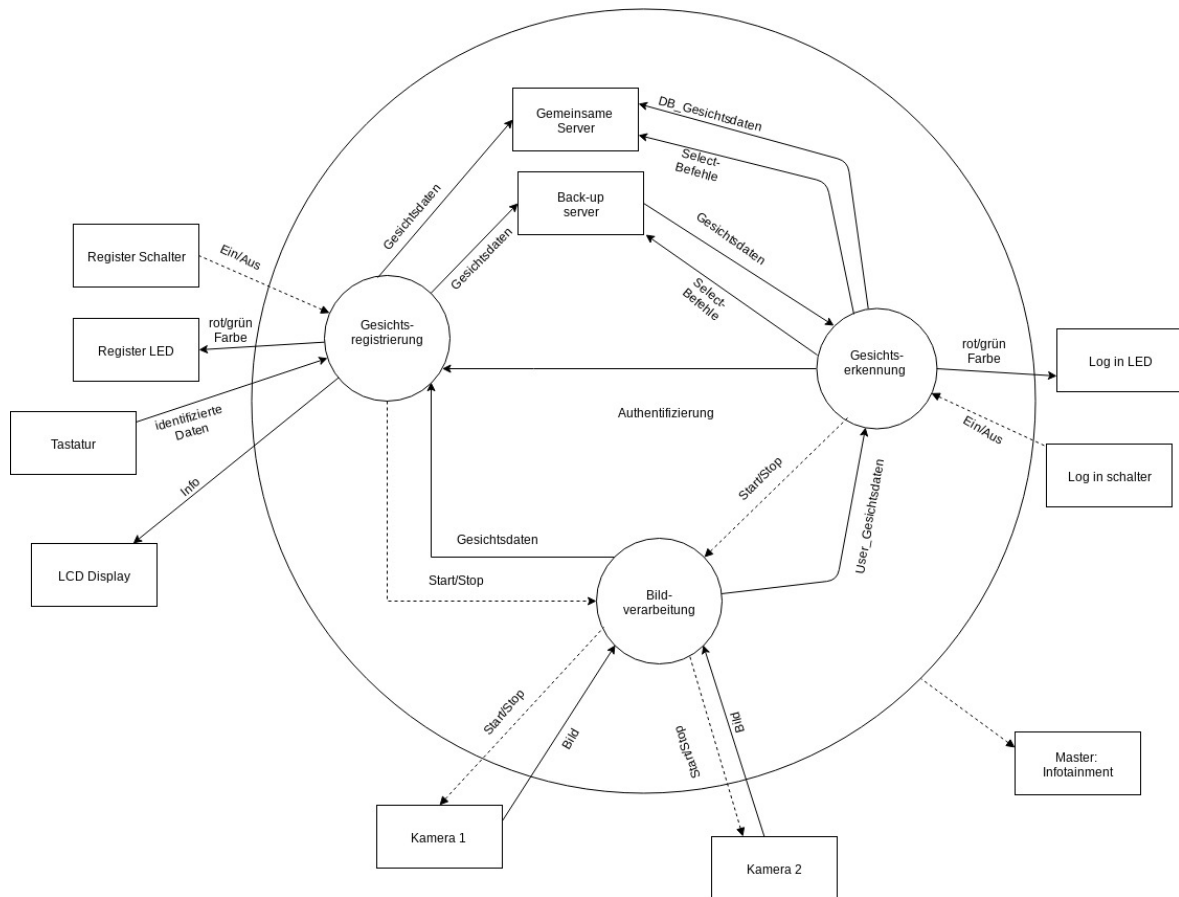


Abbildung 1.11: Big Picture

Die erste Ebene sieht man unter Abb.1.12. Sie ist eine detaillierte Version des Big Picture, das sich nur auf den Erkennungsteil konzentriert. Man spricht von einer Iteration, die hier passiert ist. Folgend wird es die Vorgehensweise und den Lösungsweg dieser Aufgabe mit Hilfe der 1.Ebene erklären und beschreiben.

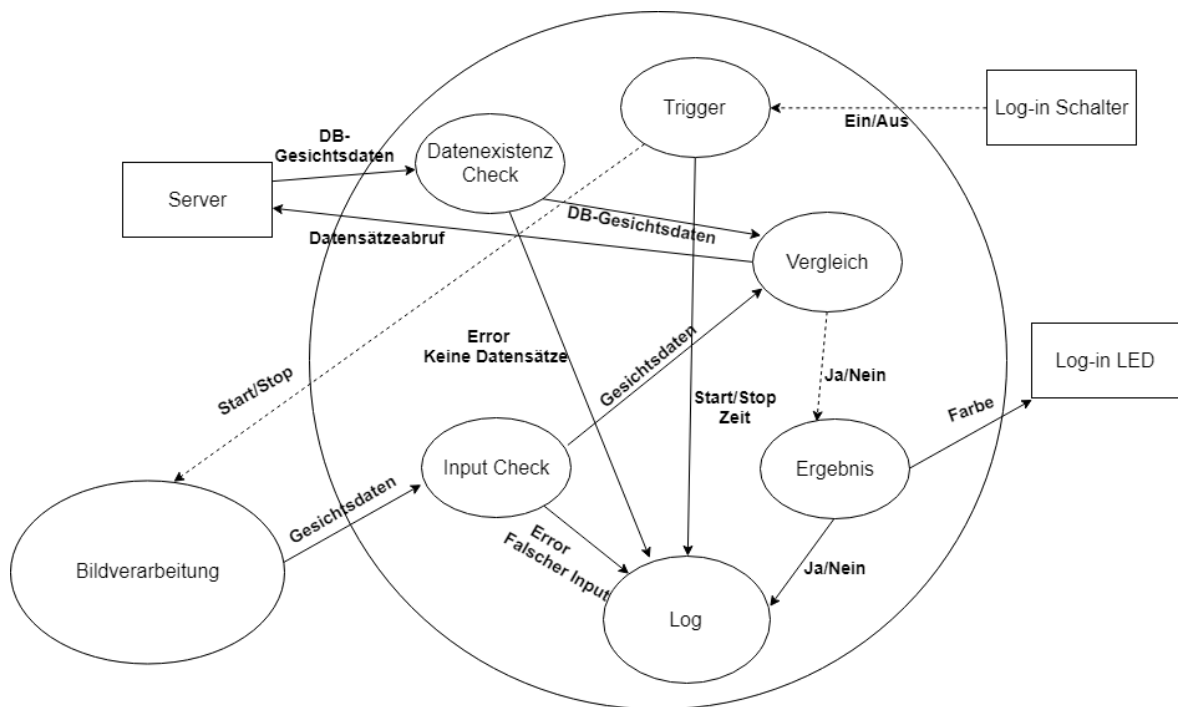


Abbildung 1.12: Erste Ebene

### Einzelne Arbeitsschritte:

#### 1. Erkennungstaster wird gedrückt

Will der Benutzer sich im System einloggen, muss er zuerst den Erkennungstaster drücken. Wird dieser Taster gedrückt, dann kriegt der Benutzer eine Anmeldung, bei der er seine Emailadresse eingeben muss. Um diese Aufgabe zu erledigen, muss das Paket RPi.GPIO importiert werden. Auf diese Weise ist der Zugriff auf den GPIO Pins möglich, damit der Taster Zustand, oder der LED Zustand erkannt werden. Das Paket mysqldb wird auch benötigt, um Zugriff auf die Datenbank zu haben.

```
if GPIO.input(17):
    exec(open('Existiert_nExistiert.py').read())
```

#### 2. Benutzer gibt seine Emailadresse ein

Es ist wichtig zu erwähnen, dass für die Umsetzung dieses Systems es notwendig ist, die verschiedenen Skripten miteinander zu verbinden. Deshalb müssen sie innerhalb anderer Skripten angerufen werden. Das wird durch Variablen gemacht. Deshalb ist das Importieren des Pakets sys nötig. Der Benutzer gibt seine Emailadresse an, die danach verwendet wird, um in der Datenbank schneller auf die Benutzerbilddaten zuzugreifen.

#### 3. Bild wird gemacht und temporär gespeichert

Gleich nachdem der Benutzer sein Email eingegeben hat, macht die Kamera das Bild. Dieses Bild wird temporär gespeichert, weil es nach dem Vergleich mit dem

Bild, das schon in der Datenbank liegt, nicht mehr benötigt wird. Sonst würde es extrem viele Bilderdaten in der Datenbank geben, die nur einmal verwendet werden. Stattdessen wird nur der Pfad des Bildes zusammen mit den extrahierten Punkten in der Datenbank gespeichert.

4. *Die Existenz derselben Emailadresse des Benutzers in der Datenbank wird geprüft*  
Dieser Schritt ist deswegen wichtig, weil falls die eingegebene Email nicht in der Datenbank existiert, muss das System die Bildvergleiche nicht machen.
5. *Wird dieselbe Emailadresse in der Datenbank gefunden, werden die zugehörige Bildinformationen geladen*

Falls aber die vom Benutzer eingegebene Emailadresse in der Datenbank ist, weißt das System Bescheid, dass ein Bilderdatenvergleich stattfinden muss. Um das zu erreichen, werden alle Bildinformationen geladen. Unter Bildinformationen sind die extrahierten Gesichtspunkte, mindestens 16, zu verstehen. Der kurze Codeabschnitt unten zeigt genau wie dies funktioniert.

```
if b=="existiert":
    mycursor.execute(
        """select * from info i \
        join person p \
        on i.idP=p.idP \
        where p.email='%s';""" % var1)
    myresult=mycursor.fetchall()
    for x in myresult:
        print(x)
```

6. *Die Bildinformationen der beiden Bildern werden verglichen*

Die wichtigste Aufgabe ist der Vergleich der Bildinformationen. Es werden die extrahierten Punkte der bereits gemachten Bilder mit den Punkten der schon in der Datenbank existierenden Bildern verglichen. Dazu wird das wichtigste Paket, Open CV, benötigt. Nachdem dieses Paket importiert wurde, kann der Vergleich beginnen. In jedes Gesichtsbild wird ein Startpunkt bzw. Origin für den Vergleich bestimmt. Dieser Punkt ist der zentrale Punkt des Gesichts. Es wurde so gedacht, dass der Abstand von jedem anderem Gesichtspunkt mit dem Origin kalkuliert wurde. Dieser Abstand wird dann mit dem Abstand der Origin des zweiten Bildes und seinem anderen Punkt. Die zwei Punkte, dessen Abstand mit den jeweiligen Startpunkten der Bilder kalkuliert ist, müssen miteinander einpassen (zum Beispiel ein Punkt der rechten Augen im ersten Bild muss mit dem zugehörigen Punkt in dem zweiten Bild verglichen werden). In dieser Weise wird mit Hilfe vielen Tests eine Standard Toleranz kalkuliert. Passen die kalkulierten Toleranzen für jeder der 16 Punkte am Ende des Vergleichs zu der Standard Toleranz, stimmt die Vergleiche, und die Bilder passen miteinander.

7. *Passen die beiden Bildern zusammen, wird der Benutzer eingeloggt.*  
Wenn das Vergleichsergebnis positiv ist, wird die grüne LED leuchten und der Benutzer darf weitermachen. Sonst wird die rote LED eingeschaltet.

## 1.3 Herausforderungen, Probleme, und deren Lösung

Während dieser Arbeit musste man mit einigen Problemen und Herausforderungen konfrontiert werden.

- *OpenCV Installation*

Die größte Probleme hat es bei der Installation von OpenCV gegeben. Diese Installation hat sehr lang gedauert und es war sehr schwer zu bestimmen, welche Paketen ausgelassen werden sollten und welche nicht. Dieses Problem wurde nach vielen Tests gelöst, durch das Kopieren von einer anderen SD-Karte, auf der OpenCV bereits installiert war. CMake und Make waren sehr wichtige Pakete, mit denen diese Aufgabe erledigt war.

- *Fritzing*

Das Problem beim Fritzing war folgendes, dass jedes Mal wenn das Programm beendet wurde, konnte es nicht mehr geöffnet werden. Es fehlten entweder die Pfade oder die Bauteile, die für um die Schaltung zu bauen notwendig waren. Die Lösung war eigentlich sehr leicht. Das Programm wurde komplett gelöscht und dann wieder im System installiert, und danach wurde die ganze Schaltung gemacht, ohne das Programm zu beenden. Sonst wären dieselben Probleme wieder aufgetreten.

- *Kurzschluss beim Raspberry Pi*

Was noch passiert ist, ist das es beim Anfassen vom Raspberry PI ein Kurzschluss gab. Der Raspberry PI war kaputt und musste ersetzt werden. Glücklicherweise konnte die SD-Karte erfolgreich in kloniert werden und deshalb sind alle Daten und Informationen gespeichert.

- *Pakete FaceRecognition Installation*

Die Installation des Paketes FaceRecognition konnte nicht erfolgreich gemacht werden, weil die dlib Pakete auch gebraucht wird. Wahrscheinlich aufgrund des zu geringen RAMs kann diese Pakete nicht installiert werden. Dieses Problem wurde noch nicht gelöst.

- *Probleme bei der Aufruf der Skripten*

Am Beginn war das Umgehen mit der Skripten Aufrufe sehr schwer. Es war schwer zu verstehen wie das genau funktionierte, weil die große Variablenanzahl die Arbeit kompliziert machte. Dieser Anzahl wurde reduziert und es wurden viele Recherchen an der korrekten Verwendung von der sys Pakete gemacht um das Problem zu lösen.

## 1.4 Projektmanagement und Controlling

In Bezug auf Projektmanagement und Controlling wurde die Methode des Fehlerbaums verwendet. Diese ist eine berühmte Methode um die Aufwandschätzung zu kalkulieren und um eine Fehlerursache zu finden. Es wird der Problem in kleinen Teilen geschnitten damit es klarer wird. Es wurde auch eine detaillierte Soll-Ist Analyse gemacht, die bei



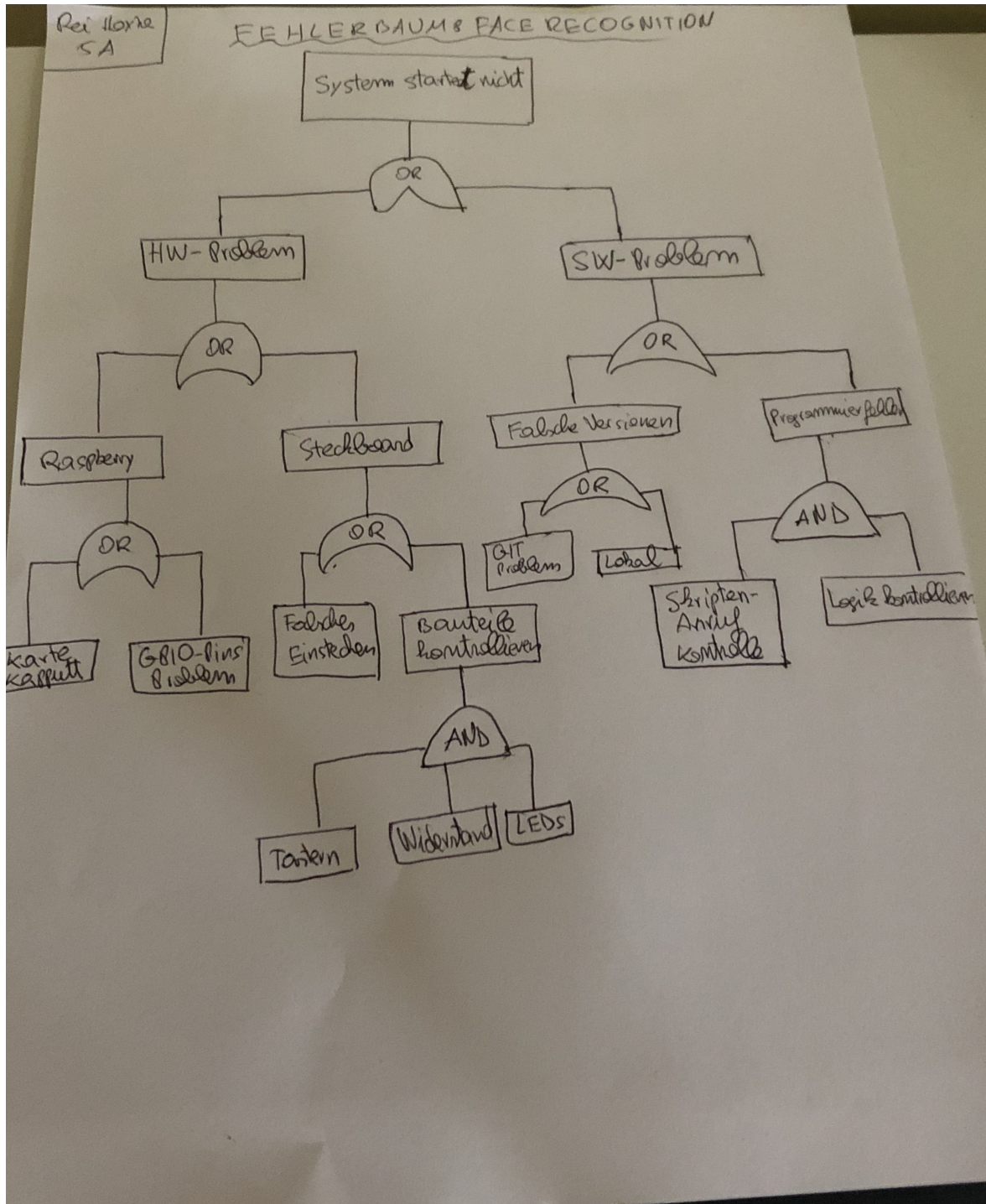


Abbildung 1.13: Fehlerbaum

der neuen Aufgabeteilung sehr geholfen hat, und ein Balkendiagramm, das für die Organisation und Festlegung der Arbeitstermine dient.

## 1.5 Ergebnisse

Die bisherigen Ergebnisse sind die folgenden:

1. Systemaufbau  
Das ganze System wurde zusammen mit der ganzen Hardware aufgebaut.
2. Digitale Darstellung des Systemaufbaus  
Der Schaltplan des Systems wurde durch Fritzing digital dargestellt.
3. Erkennung des Admins  
Der Admin wurde sofort vom System erkannt, mithilfe eines Skripts, dass OpenCV verwendet um das Adminbild mit den anderen Bildern zu vergleichen.
4. Aufnahme der Benutzer Gesichtsdaten mithilfe der Emailadresse  
Gibt der Benutzer seine Emailadresse ein, werden die zugehörigen Gesichtsdaten von der Datenbank geladen, um den Vergleich zu machen.

# Abbildungsverzeichnis

1.1	Grobe Skizze des Systems . . . . .	2
1.2	Schaltplan des Systems . . . . .	4
1.3	Debian[1] . . . . .	6
1.4	Python[2] . . . . .	6
1.5	Open CV[3] . . . . .	7
1.6	Git[4] . . . . .	7
1.7	Raspberry Pi[5] . . . . .	8
1.8	Fritzing[6] . . . . .	8
1.9	Mysql[7] . . . . .	9
1.10	Mariadb[8] . . . . .	9
1.11	Big Picture . . . . .	11
1.12	Erste Ebene . . . . .	12
1.13	Fehlerbaum . . . . .	15

# Tabellenverzeichnis

# Literatur

## Aus dem Netz

- [1] URL: [https://www.google.com/search?rlz=1C1GGRV\\_enAL753AL753&biw=1366&bih=576&tbm=isch&sxsrf=ACYBGNQ9s-sOUP2K5YJ\\_TPdkR9G7feLepw%3A1576022986105&sa=1&ei=yjPwXZz7BaXgkgWTv4eICw&q=debian&oq=debian&gs\\_l=img.3.0.0i6716j0l4.10804.12283..14300...1.0..0.166.825.0j6...0...1..gws-wiz-img.....35i39.fbB2uG6SnFE#](https://www.google.com/search?rlz=1C1GGRV_enAL753AL753&biw=1366&bih=576&tbm=isch&sxsrf=ACYBGNQ9s-sOUP2K5YJ_TPdkR9G7feLepw%3A1576022986105&sa=1&ei=yjPwXZz7BaXgkgWTv4eICw&q=debian&oq=debian&gs_l=img.3.0.0i6716j0l4.10804.12283..14300...1.0..0.166.825.0j6...0...1..gws-wiz-img.....35i39.fbB2uG6SnFE#).
- [2] URL: [https://www.google.com/imgres?imgurl=https%3A%2F%2Fictslab.com%2Fwp-content%2Fuploads%2F2019%2F03%2Fd1326ca6cca8038cd115a061b4e2b3bc-840x430.png&imgrefurl=https%3A%2F%2Fictslab.com%2Fen%2F6-reasons-why-python-is-the-future-programming-language%2F&docid=UKwQ11G91v7GOM&tbnid=0aCNC7utSq-sOM%3A&vet=10ahUKEwjz\\_T5pqzmAhXQsKQKHVOPDN4QMwh7KAkwCQ.i&w=840&h=430&bih=576&biw=1366&q=python%20programming&ved=0ahUKEwjz\\_T5pqzmAhXQsKQKHVOPDN4QMwh7KAkwCQ&iact=mrc&uact=8](https://www.google.com/imgres?imgurl=https%3A%2F%2Fictslab.com%2Fwp-content%2Fuploads%2F2019%2F03%2Fd1326ca6cca8038cd115a061b4e2b3bc-840x430.png&imgrefurl=https%3A%2F%2Fictslab.com%2Fen%2F6-reasons-why-python-is-the-future-programming-language%2F&docid=UKwQ11G91v7GOM&tbnid=0aCNC7utSq-sOM%3A&vet=10ahUKEwjz_T5pqzmAhXQsKQKHVOPDN4QMwh7KAkwCQ.i&w=840&h=430&bih=576&biw=1366&q=python%20programming&ved=0ahUKEwjz_T5pqzmAhXQsKQKHVOPDN4QMwh7KAkwCQ&iact=mrc&uact=8).
- [3] URL: [https://www.google.com/imgres?imgurl=https%3A%2F%2Fopencv.org%2Fwp-content%2Fuploads%2F2019%2F02%2Fopencv-logo-1.png&imgrefurl=https%3A%2F%2Fopencv.org%2F&docid=bk2cK6wdUxgMYM&tbnid=BHGj0QUCoS4CjM%3A&vet=10ahUKEwiT\\_IPsqKzmAhVM-aQKHQNcAjqQMwh3KAAwAA.i&w=512&h=512&itg=1&bih=576&biw=1366&q=opencv&ved=0ahUKEwiT\\_IPsqKzmAhVM-aQKHQNcAjqQMwh3KAAwAA&iact=mrc&uact=8](https://www.google.com/imgres?imgurl=https%3A%2F%2Fopencv.org%2Fwp-content%2Fuploads%2F2019%2F02%2Fopencv-logo-1.png&imgrefurl=https%3A%2F%2Fopencv.org%2F&docid=bk2cK6wdUxgMYM&tbnid=BHGj0QUCoS4CjM%3A&vet=10ahUKEwiT_IPsqKzmAhVM-aQKHQNcAjqQMwh3KAAwAA.i&w=512&h=512&itg=1&bih=576&biw=1366&q=opencv&ved=0ahUKEwiT_IPsqKzmAhVM-aQKHQNcAjqQMwh3KAAwAA&iact=mrc&uact=8).
- [4] URL: [https://www.google.com/search?rlz=1C1GGRV\\_enAL753AL753&biw=1366&bih=576&tbm=isch&sxsrf=ACYBGNTout4ebIbjJD3KkDdfub0V5SFv1w%3A1576022753156&sa=1&ei=4TLwXbyLCcTWkWX-7pvQBQ&q=git&oq=git&gs\\_l=img.3..0110.56666.56955..57700...0.0..0.132.490.0j4.....0...1..gws-wiz-img.....0i67j35i39j0i3.Qugf0\\_VouHo&ved=0ahUKEwj8--ipqzmAhVE66QKHx73BloQ4dUDCAc&uact=5#](https://www.google.com/search?rlz=1C1GGRV_enAL753AL753&biw=1366&bih=576&tbm=isch&sxsrf=ACYBGNTout4ebIbjJD3KkDdfub0V5SFv1w%3A1576022753156&sa=1&ei=4TLwXbyLCcTWkWX-7pvQBQ&q=git&oq=git&gs_l=img.3..0110.56666.56955..57700...0.0..0.132.490.0j4.....0...1..gws-wiz-img.....0i67j35i39j0i3.Qugf0_VouHo&ved=0ahUKEwj8--ipqzmAhVE66QKHx73BloQ4dUDCAc&uact=5#).
- [5] URL: <https://www.google.com/imgres?imgurl=https%3A%2F%2Fwww.element14.com%2Fcommunity%2Fcommunity%2Fimage%2F2475%2F2.png&imgrefurl=https%3A%2F%2Fwww.element14.com%2Fcommunity%2Fcommunity%2Fraspberrypi&docid=SexjLuPxEC3YHM&tbnid=h-hlStmFtIy0qM%3A&vet=10ahUKEwiWwYbLpqzmAhUMyaQKHfamDZAQMwiTAihQMFA..i&w=320&h=320&bih=576&biw=1366&q=raspberrypi&ved=0ahUKEwiWwYbLpqzmAhUMyaQKHfamDZAQMwiTAihQMFA&iact=mrc&uact=8>.

- [6] URL: [https://www.google.com/imgres?imgurl=https%3A%2F%2Fpbs.twimg.com%2Fprofile\\_images%2F378800000547620716%2Ff4a72078c6bdb3d379894da3ebd09270\\_400x400.png&imgrefurl=https%3A%2F%2Ftwitter.com%2Ffritzingorg&docid=QHWNrnpTdZnKkM&tbnid=\\_LdE\\_o8j5JTS\\_M%3A&vet=10ahUKEwj8\\_--ipqzmAhVE66QKHX73BloQMwimASgjMCM..i&w=400&h=400&bih=576&biw=1366&q=fritzing&ved=0ahUKEwj8\\_--ipqzmAhVE66QKHX73BloQMwimASgjMCM&iact=mrc&uact=8](https://www.google.com/imgres?imgurl=https%3A%2F%2Fpbs.twimg.com%2Fprofile_images%2F378800000547620716%2Ff4a72078c6bdb3d379894da3ebd09270_400x400.png&imgrefurl=https%3A%2F%2Ftwitter.com%2Ffritzingorg&docid=QHWNrnpTdZnKkM&tbnid=_LdE_o8j5JTS_M%3A&vet=10ahUKEwj8_--ipqzmAhVE66QKHX73BloQMwimASgjMCM..i&w=400&h=400&bih=576&biw=1366&q=fritzing&ved=0ahUKEwj8_--ipqzmAhVE66QKHX73BloQMwimASgjMCM&iact=mrc&uact=8).
- [7] URL: [https://www.google.com/imgres?imgurl=https%3A%2F%2Fcdn.worldvectorlogo.com%2Flogos%2Fmysql.svg&imgrefurl=https%3A%2F%2Fworldvectorlogo.com%2Flogo%2Fmysql&docid=tIz-HXUbbG9S5M&tbnid=Bgugh\\_pvo8X-JM%3A&vet=10ahUKEwiDtYDsqazmAhVNKuWkHRzPBGEQMwhuKAAwAA..i&w=2500&h=1733&bih=576&biw=1366&q=mysql%20logo&ved=0ahUKEwiDtYDsqazmAhVNKuWkHRzPBGEQMwhuKAAwAA&iact=mrc&uact=8](https://www.google.com/imgres?imgurl=https%3A%2F%2Fcdn.worldvectorlogo.com%2Flogos%2Fmysql.svg&imgrefurl=https%3A%2F%2Fworldvectorlogo.com%2Flogo%2Fmysql&docid=tIz-HXUbbG9S5M&tbnid=Bgugh_pvo8X-JM%3A&vet=10ahUKEwiDtYDsqazmAhVNKuWkHRzPBGEQMwhuKAAwAA..i&w=2500&h=1733&bih=576&biw=1366&q=mysql%20logo&ved=0ahUKEwiDtYDsqazmAhVNKuWkHRzPBGEQMwhuKAAwAA&iact=mrc&uact=8).
- [8] URL: <https://www.google.com/imgres?imgurl=https%3A%2F%2Fmariadb.org%2Fwp-content%2Fuploads%2F2015%2F10%2Fmariadb-usa-inc.png&imgrefurl=https%3A%2F%2Fmariadb.org%2F&docid=E6-SRGvYzOUkdM&tbnid=Flj2ERTV48QWWM%3A&vet=10ahUKEwjv-NmDqqzmAhWEzqQKHTiWB8IQMwhkKAIwAg..i&w=340&h=175&bih=576&biw=1366&q=mariadb%20logo&ved=0ahUKEwjv-NmDqqzmAhWEzqQKHTiWB8IQMwhkKAIwAg&iact=mrc&uact=8>.
- [9] *Gründe für die Wahl von Debian*. URL: [https://www.debian.org/intro/why\\_debian.de.html](https://www.debian.org/intro/why_debian.de.html).
- [10] *OpenCV*. URL: <https://opencv.org/about/>.

## Der ganze Rest

- [11] Dominik Stocklasser. “Systemtechnikscherpunkt 4.KlasseSystementwicklungVersion 01.10Autor: Dominik Stocklasser”.
- [12] Dominik Stocklasser. “Systemtechnikscherpunkt 4.KlasseVersionskontrolleVersion”.