

# 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 wie wurden sie gelöst . . . . .	14
1.4	Projektmanagement und Controlling . . . . .	14
1.5	Ergebnisse . . . . .	16

# Kapitel 1

## Gesichtserkennung

### 1.1 Allgemeines

Heutige Zeit ist die Technologie ein der wichtigsten Themen für den Welt. Mithilfe der Technologie hat die Menschheit wesentliche Fortschritte gemacht im Bereich der Lebensqualitätsverbesserung und Erleichterung. Mehr als jemals zuvor kommen intelligente Systeme im Einsatz, die um den Arbeitsaufwand und die Arbeitskomplexität zu reduzieren dienen. Diese Systeme werden in die wichtigsten Bereiche des Lebens verwendet, wie zum Beispiel: Menschlichen Beziehungen, Business, Transport, Banken, Medizin, Bildung und Kommunikation. Selbstverständlich gibt es zahlreiche Versuche von verschiedenen Menschen, die als Hauptziel den Zugriff auf diesen Systemen zu ermöglichen haben. Deshalb ist die Sicherheit in diesem Systemen eine der aktuellsten und wichtigsten Themen. Der unkontrollierte Zugriff auf solchen wichtigen und delikaten Systemen wurde zu unreparierbare Konsequenzen führen. Auf dieser Gründe muss es sehr genau und detailliert geplant werden, welche Sicherheit Strategien die klügsten bzw. die besten sind.

Der Hauptthema dieses Projekts ist genau die Sicherheit der Systeme, und zwar die Ermöglichung des kontrollierten Zugriffs von Menschen in verschiedenen Gebäude. Es wurde die Verantwortlichkeit übernommen, dass der Zugriff in der Österreichischen Schule “Peter Mahringer” überwiesen und kontrolliert ist. Zu diesem Zweck wird in diesem Projekt ein intelligentes System entwickelt, dass die Aufgabe hat, menschliche Gesichtern (Gesichtern von Schülern, Lehrern, Eltern usw.) zu registrieren und zu erkennen. Es wird höchstwahrscheinlich in dem Schultür integriert, weil vom dort werden alle unbedingt reinkommen. Dieser Projekt wird große Hilfe für die Schule sein, aufgrund der vom Kameras und komplexen Algorithmen angebotenen Möglichkeiten. Beispielweise wird es überprüft, ob der Person dass reinkommt ein Student oder Lehrer ist, oder wird es notiert, wer die Schule wann verlässt oder besucht. Hat ein Person kein Zugriff, wird er nicht reinzukommen erlaubt.

Um diesem Projekt zu realisieren, wird das Projekt in drei große und wichtige Teile zerlegt. Das erste Teil heißt Bildverarbeitung und hat die Aufgabe, Gesichtsbildern vom Kamera zu normalisieren und verarbeiten, damit sie in dem geeignete Format für die Gesichtsvergleichung sind. Das zweite Teil beschäftigt sich mit der Registrierung der Gesichter von Personen im Server. Das dritte Teil ist die Gesichtserkennung. Das

ist genau das Teil für dem ich verantwortlich bin. Hier geht es um die Erkennung der Gesichtern der Benutzern des Systems. Es wird durch eine Vergleichen überprüft, ob der Person vorher schon registriert worden ist oder nicht, und ob sein Gesichtsdaten schon im Server existieren oder nicht. Nur wenn die Vergleichsergebnisse positiv sind, darf der Person reinkommen. Dem 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 wurde sowie im Abb.1.1 zu sehen.

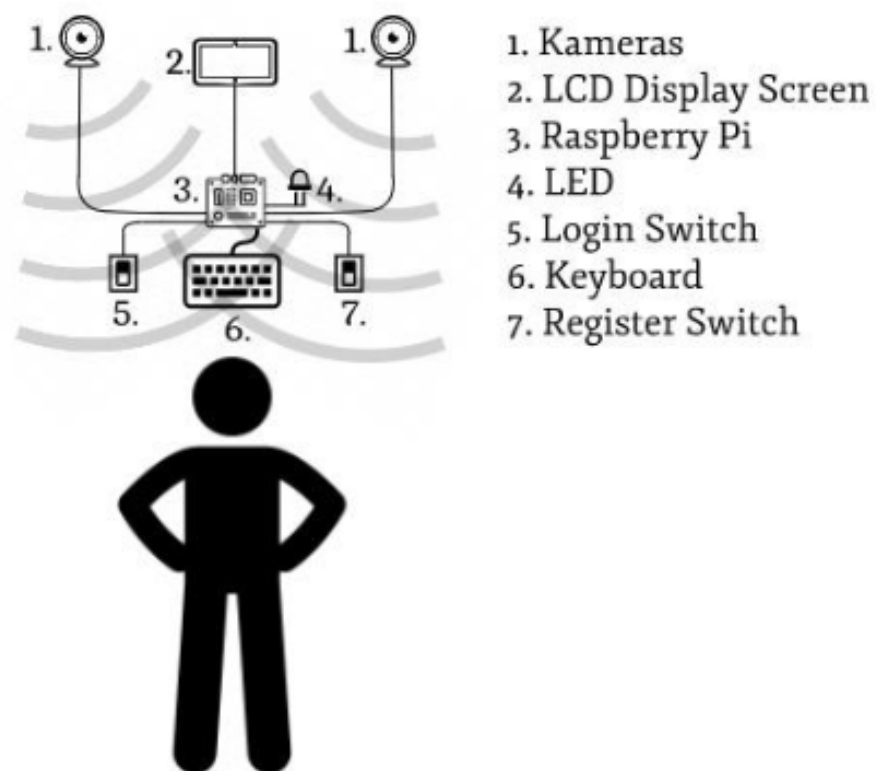


Abbildung 1.1: Grobe Skizze des Systems

## 1.2 Technische Lösung

### 1.2.1 Hardware und Aufbau

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

## Bauteile und HW-Komponente

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

1. *Steckboard bzw. Steckplatine*

Die Steckplatine ist der wichtigste Komponente der Hardware unseres Systems. Sie wird für die Entstehung der elektrischen Verbindung von verschiedenen elektrischen Bauteile, um elektrische Schaltungen zu bauen oder um verschiedene Tests und Experimenten zu machen. In dieser Steckplatine werden alle andere Komponenten platziert, damit die Verbindung erstellt werden kann und damit das System laufen kann.

2. *Kabeln bzw. Leitern*

Damit die verschiedene Komponenten, die in der Steckplatine platziert sind, miteinander verbunden sein können, braucht man unbedingt die Kabeln. Mithilfe von Kabeln können elektrische Impulse und Signale fließen, damit die Energie und die Information überträgt werden. Die verwendete Kabeln sind aus Kupfer und sind sowie 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 die Verbindungen innerhalb der Steckplatine machen.

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

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

4. *Widerstand*

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

5. *Tastern*

Ein Taster ist 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 gefunden.

6. *Raspberry Pi*

Raspberry Pi ist ein Minicomputer, dass in diesem Projekt den normalen Computer ersetzt. Der verwendete Raspberry ist Version 3 und 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.

---

<sup>1</sup>Light Emmitting Diode

<sup>2</sup>Generated Input Output

### 7. Bildschirm

Ein Bildschirm ist nur eine Anzeige, dass um die visualische Darstellung von verschiedenen Informationen oder Daten(wie Videos, Fotos, Statistiken usw) verwendet wird. Ein Bildschirm wird heutige Zeiten sehr häufig verwendet, aufgrund der hohen Benutzerfreundlichkeit dass angeboten wird.

### 8. Tastatur

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

### 9. Kamera

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

## 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 mithilfe eines Programms, dass Fritzing heißt, gemacht und spielt eine sehr wichtige Rolle bei der Organisation und Planung des Schaltkreises. Der Schaltplan ist auf Abb.1.2 zu sehen.

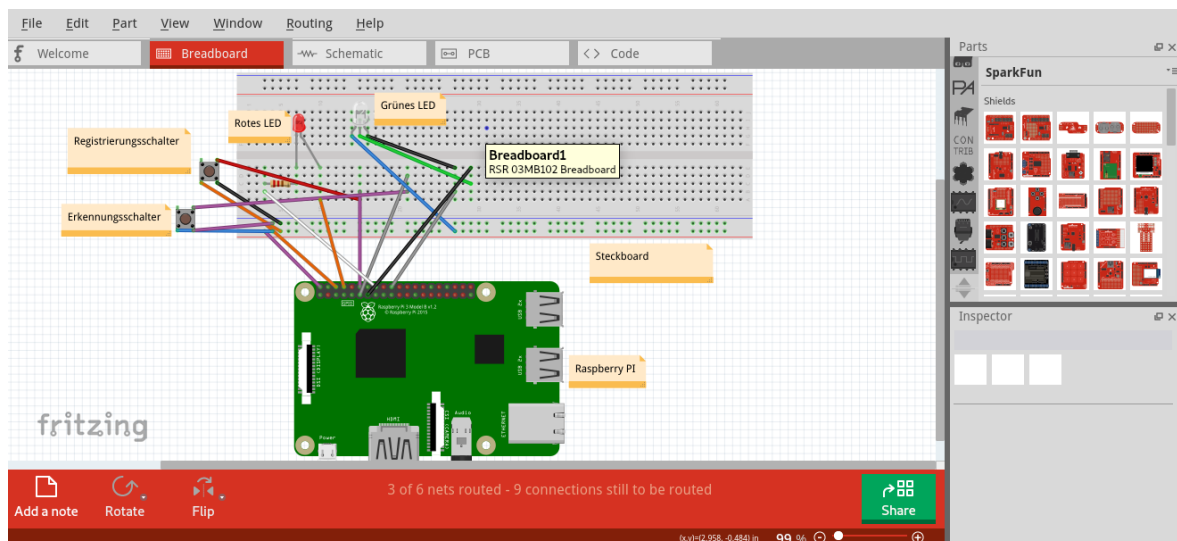


Abbildung 1.2: Schaltplan des Systems

Wie der Schaltplan zeigt, besteht das System aus zwei LEDs, zwei Tastern, einem Widerstand, eine Steckplatine und aus 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 verbindet sich mit dem Minuspol der Steckplatine. Der Datenanschluss ist mit dem GPIO18 Pin verbunden. Wie bei der Registrierung-Taster wird auch bei der 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, während der Datenanschluss mit dem GPIO17 Pin sich verbindet. Die rote LED wird verwendet wenn die Registrierung oder Erkennung der Benutzer im System nicht erfolgreich war. Die längere Beine oder die Anode wird mit dem GPIO23 Pin des Raspberry verbunden, während die Kathode wird in dem Minuspol Bereich der Steckplatine eingeschlossen. Die grüne LED ist eine RGB(Red Green Blue) LED. Diese LED kann die Farbe ändern. Wird im Fall einer erfolgreichen Registrierung oder Erkennung des Benutzers im System verwendet. Hat im Gegensatz zu der normalen LED 3 Anschlüsse. Der Minus Anschluss wird in dem Minuspol Bereich der Steckplatine eingeschlossen und der Pluspol Anschluss verbindet sich mit dem Pluspol Bereich der Steckplatine. 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 der technischen und logischen Teil bzw. in der Software und in der verwendete Skripte.

### 1.2.2 Software

Nachdem der Aufbau und die verwendete Hardware des Systems beschrieben wurde, ist die Software dran. In diesem Unterkapitel wird alles was mit der logische Teil der Umsetzung hat: die verwendete Betriebssystemen, Programmiersprachen, Frameworks, Technologien und Planungsmethoden. Es wird jede Aufgabe zusammen mit der zugehörigen Lösung im Details beschrieben und jeder programmierter Skript erklärt.

#### **Verwendete Betriebssystemen, Programmiersprachen, Frameworks, Technologien und Planungsmethoden**

In diesem Teil des Projekts dass 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. Dieses Betriebssystems wichtigste Bestandteile sind der Bootloader, der Kernel, und das Init System. Linux würde 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 Kostenlosigkeit

Linux wurde genutzt, weil es komplett kostenlos benutzt werden kann und weil es ein Open Source Betriebssystem ist.

---

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

- Hohe Sicherheit  
Für dieses Projekt ist Sicherheit sehr wichtig, und Linux bietet eine hohe Sicherheit an. Es gibt keine Viren und keine Abstürze. Backup ist auch sehr leicht machbar.
- Vielfältigkeit an Möglichkeiten und Benutzerfreiheit  
Linux wurde auch aufgrund der Vielfältigkeit der Möglichkeiten dass angeboten wird und aufgrund der Freiheit, die die Benutzern haben, verwendet. Beliebige Systemkonfigurationen, verschiedene Gesichtserkennungspaketen(OpenCV<sup>4</sup>) und andere Programmen oder Technologien(Fritzing) sind vom Linux besser und günstiger als bei anderen Betriebssystemen gefunden und installiert. Endlich ist der Raspberry auch mehr kompatibel mit Linux. Als Linuxderivat wurde aufgrund der großen Menge angebotenen der Paketen, der hohen Geschwindigkeit und der schnellen Korrigieren des Fehlers Debian gewählt.[9]



Abbildung 1.3: Debian

[1]

- Python  
Die gewählte Programmiersprache ist Python. Der wichtigste Grund dieser Wahl ist die hohe Leichtigkeit der Verwendung der OPEN-CV Pakete für die Gesichtserkennung Algorithmen. Im Vergleich zum C zum Beispiel werden die OpenCV Bibliotheken und Programmen viel praktischer und schneller gemacht. Python hat auch Vorteile im Bezug der Skripten Einbindung oder der Skripten Aufruf, als ein oder zwei Output Parametern genug sind um zwei Skripten miteinander zu verbinden. Version 3 wurde verwendet.



Abbildung 1.4: Python

[2]

---

<sup>4</sup>Open Computer Vision



## 2. Frameworks

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

- Open CV

Die wichtigste Framework ist OpenCV. OpenCV ist eine Softwarebibliothek dass für Computer-Vision und maschinelles Lernen verwendet wurde. 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 erkennen, um Objekte zu identifizieren, menschliche Handlungen in Videos zu klassifizieren und Kamerabewegungen zu verfolgen. Dieser Framework wurde deswegen gewählt, weil die Vielfältigkeit der angebotenen Optionen und Paketen einfach größer als bei anderen Frameworks ist. Ein anderer Vorteil ist dass 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, dass für den Back-Up des Systems verwendet wurde. Dieser Framework ist ein verteiltes Versionsverwaltungssystem, das heißt, die unterschiedliche Versionen der Dateien werden funktional gespeichert, in dem die ganze Repository vom Server kopiert wird. Dieses erlaubt dem Benutzer lokal zu arbeiten, als alle Änderungen nachträglich im Server aktualisiert werden. Nicht nur das, sondern auch die hohe Geschwindigkeit der Fehlerfindung und die perfekte Unterstützung vom großen Projekte, in dem alle Teammitgliedern auf alle Daten Zugriff haben, waren die Gründe, warum Git für den 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 sind die Programme, die Systeme und die spezielle Komponenten, dass benutzt worden sind.

- Raspberry Pi

Wie es vorher erwähnt wurde in dem Gesichtserkennungteil des Projekts statt ein Computer ein Raspberry Pi verwendet. Was ein Raspberry Pi ist wurde in dem Kapitel Hardware erklärt. Die richtige Gründe der Wahl sind aber erst jetzt genant zu werden. Das kleinere Platzbedarf und die hohe Flexibilität dass der Raspberry anbietet sind wesentliche Vorteile. Die Bestandteile bewegen sich nicht, und es gibt gar keinen Lärm. Was sehr beliebt ist ist dass der Raspberry kann gleichzeitig für die Hardware(GPIO PINS) und die Software(wie ein normaler Computer) verwendet werden. Der unterstützt Raspberry Linux sehr gut. Ein der wichtigsten Gründe dieser Wahl ist auch dass und dass im Fall eines Fehlers oder irgendwann wenn es notwendig ist, wird ein Raspberry sehr schnell und problemlos geändert oder repariert(einfach SD-Karte kopieren und dann ändern, alles wird gespeichert und die Informationen werden zuverlässig in einem anderen Raspberry transportiert).



Abbildung 1.7: Raspberry Pi

[5]

- Fritzing

Fritzing ist ein Programm das für die graphische Darstellung der Schaltkreis 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, Steckplatine und Schematic View. In diesem Fall wurde die Breadboard View gewählt, weil dort alles übersichtlicher ist. Was noch gut ist, ist dass 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 damit die Benutzerfotos und Benutzerdaten gespeichert werden können, wurde die berühmteste Open-Source relationales Datenbank System, MySQL. Zusammen mit MySQL wurde auch die Pakete MariaDB auf dem System installiert, damit Python mit dem MySQL-Python Pakete besser umsetzen kann. Mithilfe der MariaDB werden die ganzen Tabellen mit ihren Daten gesehen.



Abbildung 1.9: Mysql

[7]



Abbildung 1.10: Mariadb

[8]

#### 4. Planungsmethode

- Structed Design Structed Software Design ist die Methode, die um das System zu planen 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 wichtigste Elemente die in der Structed Design beschrieben werden sind:

- Bubbles  
Die wichtigste Arbeitsbereiche des Systems(Gesichtsregistrierung, Gesichtserkennung, Bildverarbeitung)
- Schnittstellen  
Sind die Interfaces, die die Verbindung zwischen die Bestandteile machen /
- Nachrichten  
Sind die geschickte Signalen innerhalb des Systems.
- Flüsse  
Zeigt die Daten die innerhalb des Systems fließen.
- Modulen  
Bestandteil des Systems, dass innerhalb liegt.
- Terminatoren  
Ähnlich mit der Modulen, aber befinden sich außerhalb des Systems und bewirken es
- definierte Abgrenzungen  
Mithilfe der Abgrenzungen wird vom Systemhersteller genau definiert, was er nicht machen wird, sei es für Zeit-,Schwierigkeits-, oder Komplexitätsverbundene Gründe. Diese Abgrenzungen dienen damit der Auftraggeber genau weißt was er erwarten muss, damit es keine Missverständnisse gibt.[11]

Mithilfe der Structed Design wurde die BigPicture des Systems und die Erste Ebene der Gesichtserkennung erstellt.

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

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

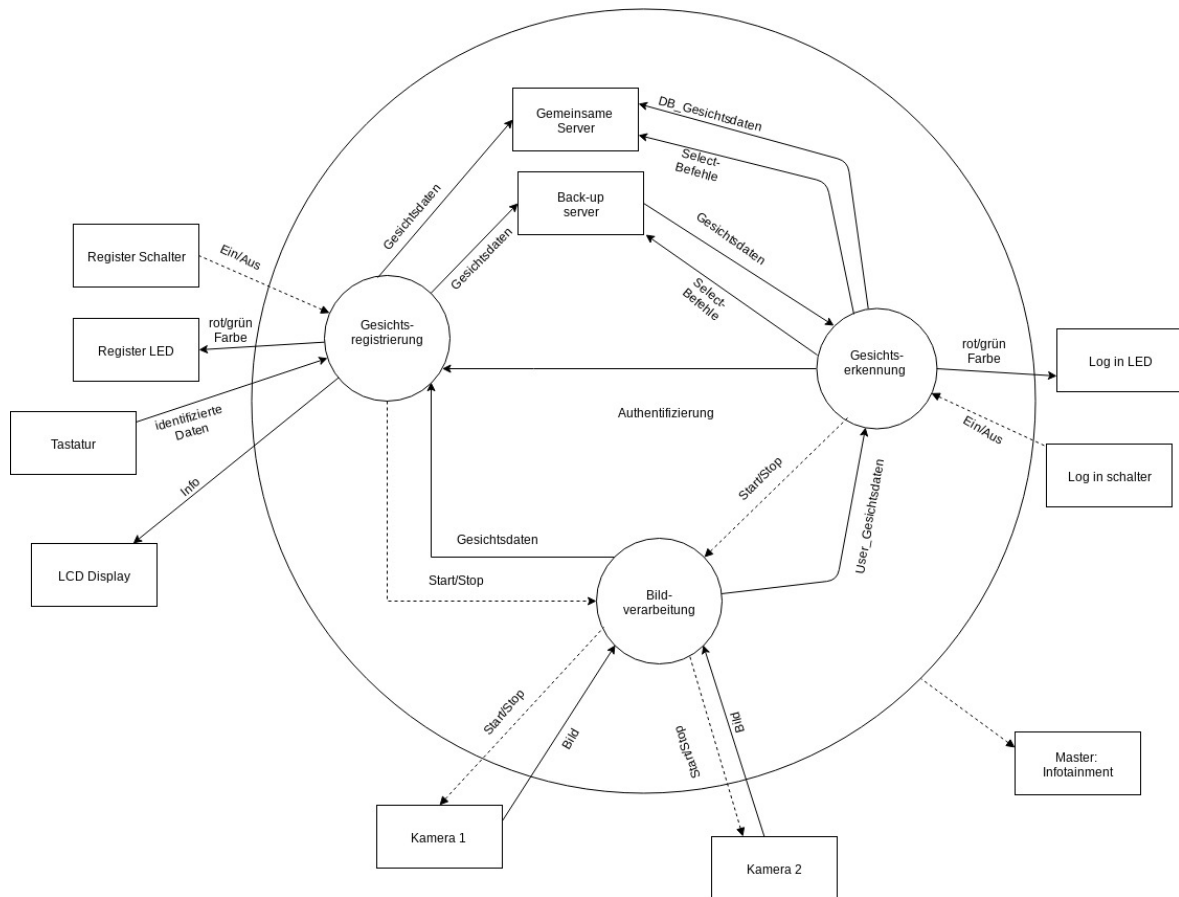


Abbildung 1.11: Big Picture

Die erste Ebene, die steht unter Abb.1.12. Sie ist ein mehr detailliertes Version der Big Picture, dass sich nur auf dem Erkennungsteil konzentriert. Man spricht von einer Iteration, dass hier passiert. Folgend wird es die Vorgehensweise und der Lösungsweg dieser Aufgabe mithilfe der Ersten Ebene erklärt und beschrieben.

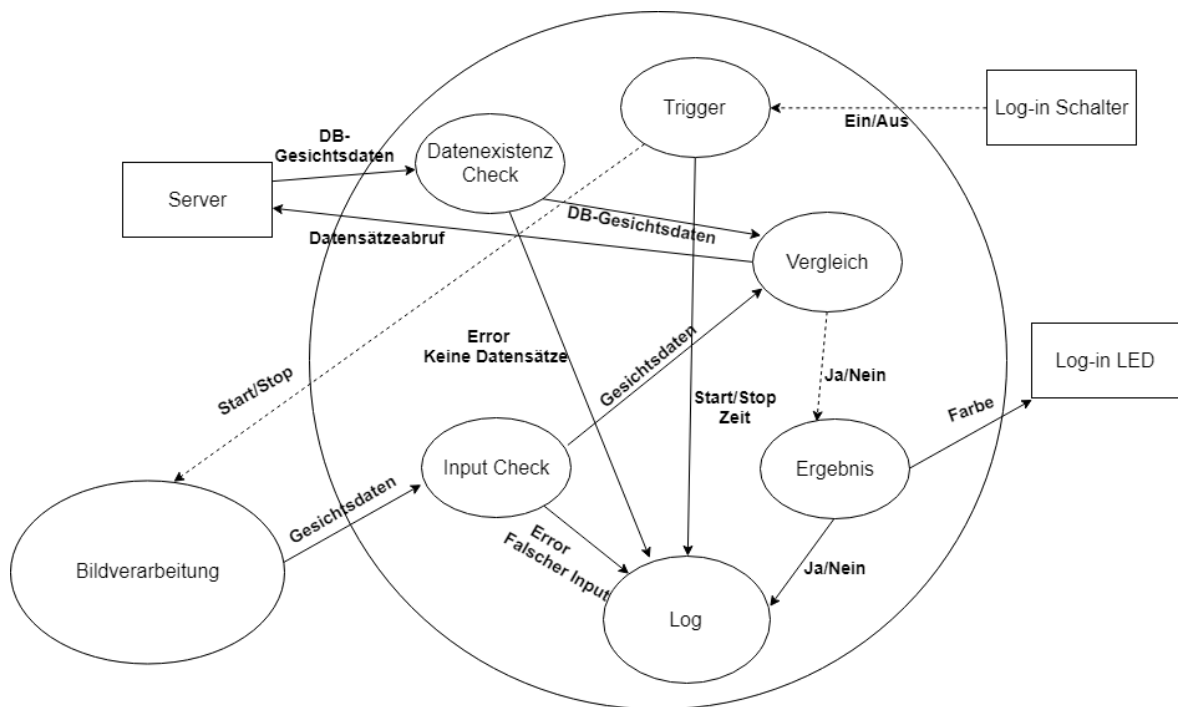


Abbildung 1.12: Erste Ebene

### Einzelne Arbeitsschritte:

#### 1. Erkennungstaster wird eindrückt

Will der Benutzer im System einloggen um reinzukommen, muss er zuerst den Erkennungstaster eindrücken. Wird dieser Taster gedrückt, dann kriegt der Benutzer eine Anmeldung, dass er seine Emailadresse eingeben muss. Um diese Aufgabe zu erledigen, muss die Pakete RPi.GPIO importiert werden. In dieser Weise ist der Zugriff auf den GPIO Pins ermöglicht, damit die Taster- Zustand, oder die LED Zustand erkannt werden. Die Pakete mysqldb wurde auch in der Zukunft benötigt, um Zugriff an 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 das Skripten Anruf innerhalb anderer Skripten extrem wichtig ist. Das wird durch Variablen gemacht. Deshalb ist das Importieren der Pakete sys nötig. Der Benutzer ergibt seine Emailadresse an, dass verwendet wird, um in der Datenbank schneller auf den Benutzerbilddaten zu suchen.

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

Gleich nachdem der Benutzer sein Email gegeben hat, wird die Kamera das Bild machen. Dieses Bild wird deswegen temporär gespeichert, weil nachdem der Vergleich von diesem Bild mit den anderen Bilderdaten, dass schon in der Datenbank

liegen, ist die Verwendung dieses Bildes nicht sinnvoll. Sonst würde es extrem viele Bilderdaten in der Datenbank, dass nur einmal verwendbar sind. Stattdessen wird nur das Path des Bildes zusammen mit der extrahierten Punkten in der Datenbank gespeichert.

4. *Die Existenz der selben Emailadresse des Benutzers in der Datenbank wird geprüft*

Dieser Schritt ist deswegen wichtig, weil falls das gegebene Email nicht in der Datenbank schon existierte, braucht das System gar nicht besorgen, um die Bildvergleichung zu machen.

5. *Wird die selbe Emailadresse in der Datenbank gefunden, werden die zugehörige Bildinformationen gekriegt*

Falls aber die vom Benutzer gegebene Emailadresse die gleiche mit einer Adresse in der Datenbank ist, weißt das System Bescheid, dass eine Bilderdatenvergleichung stattfinden muss. Um das zu erreichen, werden alle Bildinformationen gekriegt. Unter Bildinformationen sind die extrahierte Gesichtspunkte, die mindestens 16 sein werden, zu verstehen. Die kurze Codeabschnitt unten zeigt genau wie dieser Information Aufnahme erreicht wurde.

```
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 vergleicht*

Die wichtigste Aufgabe ist die Vergleichen der Bildinformationen. Es werden die extrahierte Punkten der bereit gemachten Bildern mit der Punkten der schon in der Datenbank existierte Bildern vergleicht. Dazu wird die wichtigste Pakete, dass Open CV heißt, benötigt. Nachdem diese Pakete importiert wurde, kann die Vergleichen beginnen. Es wurde so gedacht, dass jeder Punkt von einem Bild mit dem zugehörigen Punkt des anderen Bildes vergleicht wurde. In dieser Wiese wird eine Toleranz kalkuliert. Danach wird der Skript dass die Bilder vergleicht angerufen, dass die Toleranz als Output gibt. Passt sie zu der kalkulierten Toleranz, stimmt die Vergleichen, die Bilder passen miteinander.

7. *Passen die beiden Bildern zusammen, darf der Benutzer reinkommen*

Wenn die Vergleichsergebnisse positiv sind, wird die grüne LED leuchten und der Benutzer darf weitermachen. Sonst wird die rote LED eingeschaltet.

## 1.3 Herausforderungen, Probleme, und wie wurden sie gelöst

Während dieser Arbeit wurden einige Problemen und Herausforderungen eingetreten.

- *OpenCV Installation*

Die größte Probleme hat die Installation der OpenCV gegeben. Sie dauerte sehr lang und es war sehr schwer zu bestimmen, genau was ausgelassen werden sollte und was nicht. Dieses Problem wurde nach vielen Proben gelöst, durch das Kopieren von einer anderen SD-Karte, dass OpenCV schon installiert hatte. CMake und Make waren sehr wichtige Paketen um diese Aufgabe zu erledigen.

- *Fritzing*

Das Problem beim Fritzing war folgendes, dass jedes Mal dass der Programm beendet wurde, konnte er nicht mehr geöffnet werden. Es fehlten die Paths oder die Bauteile, die für die Schaltung zu verwenden waren. Die Lösung war eigentlich sehr leicht. Der Programm wurde komplett gelöscht und dann wieder im System installiert, und danach wurde die ganze Schaltung gemacht, ohne den Programm zu beenden. Sonst wurden die selben Probleme wieder auftreten.

- *Kurzschluss beim Raspberry Pi*

Was noch passiert ist, ist dass beim Anfassen vom Raspberry ein Kurzschluss gab. Der Raspberry war kaputt und musste ersetzt werden.

- *Pakete FaceRecognition Installation*

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

- *Probleme bei dem Skripten Aufruf*

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

## 1.4 Projektmanagement und Controlling

Im 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 alles klarer wird. Es wurde noch eine detaillierte Soll-Ist Analyse gemacht, die bei der neuen Aufgabeteilung sehr geholfen hat, und einen Balkendiagramm, dass für die Organisation und Festlegung der Arbeitstermine dient.



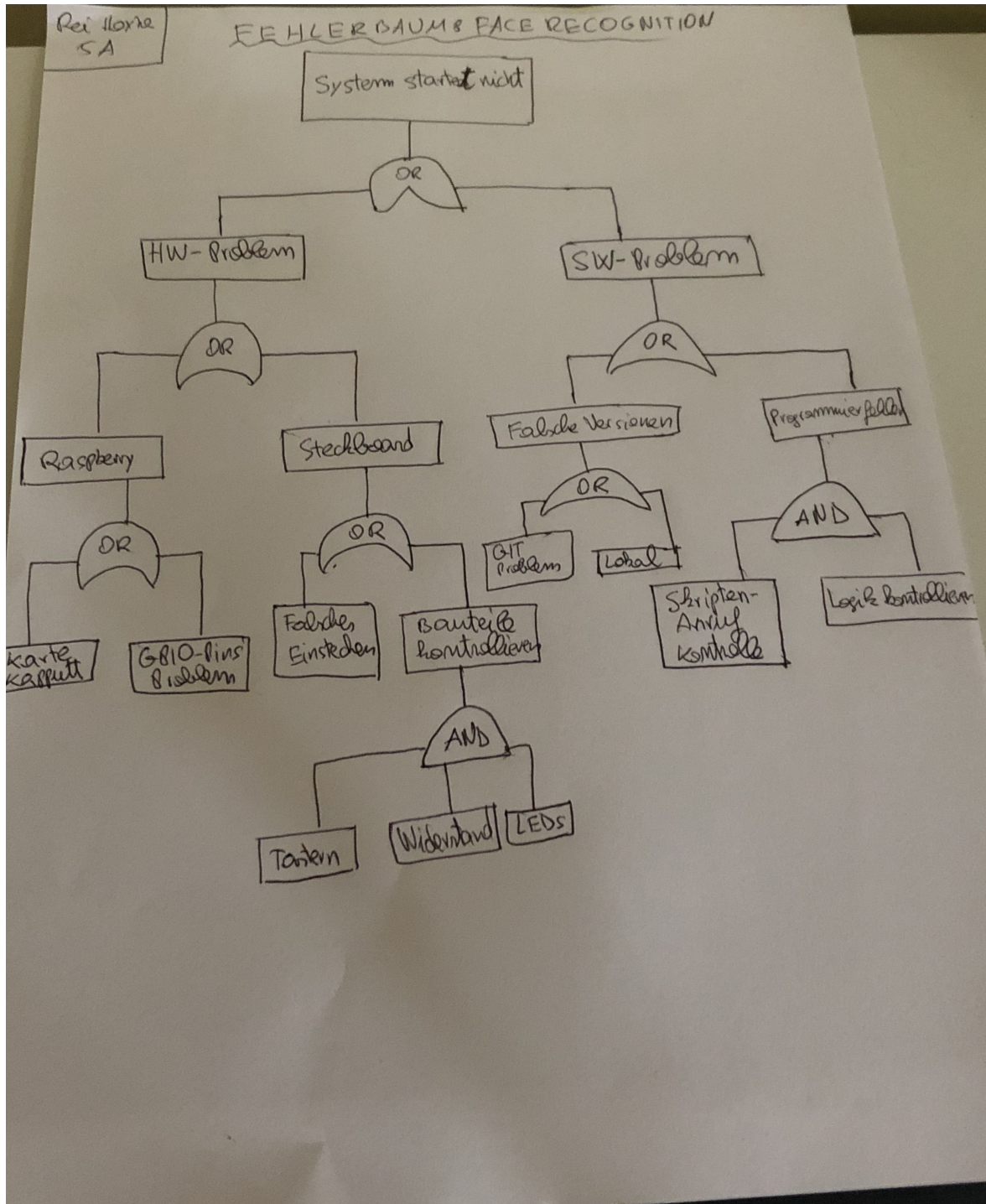


Abbildung 1.13: Fehlerbaum

## 1.5 Ergebnisse

Die bisherige Ergebnisse sind die folgende:

1. Systemaufbau  
Das ganze System wurde aufgebaut zusammen mit der ganze Hardware.
2. Digitale Darstellung des Systemaufbaus  
Der Schaltplan des Systems wurde durch Fritzing digital dargestellt.
3. Erkennung der Admin  
Der Admin wurde sofort vom System erkannt, mithilfe einer Skript dass OpenCV verwendet um das Adminbild mit der anderen Bilder zu vergleichen.
4. Aufnahme der Benutzer Gesichtsdaten mithilfe der Emailadresse  
Gibt der Benutzer seine Emailadresse ein, werden die zugehörige Gesichtsdaten von der Datenbank selektiert und gekriegt, um den Vergleich zu beginnen.

# Abbildungsverzeichnis

1.1	Grobe Skizze des Systems . . . . .	2
1.2	Schaltplan des Systems . . . . .	4
1.3	Debian . . . . .	6
1.4	Python . . . . .	6
1.5	Open CV . . . . .	7
1.6	Git . . . . .	7
1.7	Raspberry Pi . . . . .	8
1.8	Fritzing . . . . .	9
1.9	Mysql . . . . .	9
1.10	Mariadb . . . . .	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".