SHKOLLA AUSTRIAKE

Höhere technische Schule fur Informationstechnologie
Shkolla e mesme profesionale private për teknologji informacioni

Österreichiche Schule Peter Mahringer
Shkolla Austriake Shkodër

Gesichtsregistrierung und Gesichtserkennung

Diplomarbeit Nr. 17.06

Klasse 5A, Schuljahr 2019/20

Ausgeführt von: Aron Terzeta
Rei Hoxha
Egli Hasmegaj
Jordi Zmiani

Projektbetreuer1: Matthias Maurer
Projektbetreuer2: Dominik Stocklasser
Projektbetreuer3: Andreas Kucher

Shkoder, 6. März 2020

Eidesstattliche Erklärung

Wir versichern, dass wir die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt haben. Wir haben uns keiner anderen als der im beigefügten Quellenverzeichnis angegebenen Hilfsmittel bedient. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht.

Rei Hoxha

Ort, Datum Unterschrift

Egli Hasmegaj

Ort, Datum Unterschrift

Approbation Datum u. Unterschrift	PrüferIn	IT-Koordinator/Direktion

Sämtliche in dieser Diplomarbeit verwendeten personenbezogenen Bezeichnungen sind geschlechtsneutral zu verstehen.

Kurzfassung

Vom Auftraggeber wird ein System gefordert, das Gesichter erkennt, um einen kontrollierten Zugang in der Schule zu ermöglichen und die Sicherheit der Schule wird dadurch erhöht. Alle Gesichter sollen von den aktuellen Schülern und Lehrer erkannt werden. Es soll auch zwischen einer reellen Person und einem Foto den Unterschied berücksichtigt werden.

Wir haben uns für diese Idee entschieden, weil Sicherheit heute hoch interessant und relevant ist. Es geht hier um einen kontrollierten Zugang in Institutionen mittels Gesichtserkennung zu ermöglichen, da Gesichter eindeutig für jede Person sind. Die größten Herausforderungen und Voraussetzungen des Projekts befinden sich in dem Planungsprozess. Eine andere Voraussetzung ist das Gebrauch von zwei Kameras, damit der Unterschied zwischen einer reellen Person und einem Foto berücksichtigt wird.

Abstract

This paper represents the face detection and recognition system that enables the detection of a human face and is able to identify it. It is thought to improve the security system of an institution while controlling the access of certain locations, rooms. Further it achieves the goal of differentiating between a real person and a photo being identified.

Ky punim paraqet sistemin e zbulimit dhe njohjes së fytyrës që mundëson zbulimin e një fytyre njerëzore dhe është në gjendje ta identifikojë atë. Mendohet se përmirëson sistemin e sigurisë së një institucioni ndërsa kontrollon hyrjen në disa lokacione, dhoma. Më tej ajo arrin qëllimin e diferencimit midis një personi të vërtetë dhe një fotografie që identifikohet.

Dieses Dokument stellt das Gesichtserkennungs- und -erkennungssystem dar, mit dem ein menschliches Gesicht erkannt und identifiziert werden kann. Es wird angenommen, dass es das Sicherheitssystem einer Institution verbessert und gleichzeitig den Zugang zu bestimmten Orten und Räumen kontrolliert. Ferner wird das Ziel erreicht, zwischen einer realen Person und einem identifizierten Foto zu unterscheiden.

Danksagung

Inhaltsverzeichnis

1 Allgemeines **1**

2 Planung **2**

2.1 Projektziele 2

2.1.1 Ziele 2

2.1.2 Nicht Ziele 3

2.1.3 Optionale Ziele 3

2.2 Projektplanung 3

2.3 Projektmanagmentmethode 4

3 Gesichtsregistrierung **6**

3.1 Umsetzung 6

3.1.1 Allgemein 6

3.1.2 Verwendete Technologien 6

3.1.3 Technische Lösung 7

3.1.4 Herausforderungen 13

3.1.5 Qualitätssicherung und Controlling 15

3.2 Ergebnisse 17

3.2.1 Implementierung 17

4 Umsetzung - Jordi **18**

4.1 Allgemeine Beschreibung 18

4.2 Technische Lösungen 21

4.2.1 Datenkbank 21

4.3 Herausforderungen 24

4.3.1 Datenkbank - MariaDB 24

4.3.2 Python Scripting 24

4.3.3 OpenCV 25

4.4 Qualitätssicherung 25

5 Ergebnisse - Jordi **27**
5.1 Datenbank . . . 27
5.2 Tieferkennung . . . 27

6 Bildverarbeitung **28**
6.1 Allgemeines . . . 28
6.1.1 Entwicklungsumgebung und Technologien . . . 28
6.1.2 Frameworks und Bibliotheken . . . 29
6.2 Technische Lösungen . . . 30
6.2.1 Lösungsweg - Structed Software design . . . 30
6.2.2 Gesichtsdetektion . . . 32
6.2.3 Normalisierung . . . 35
6.2.4 Zuschneiden von Gesichter . . . 38
6.2.5 Gesichtsschlüsselpunkte Extraktion . . . 39
6.3 Herausforderungen, Probleme und deren Lösung . . . 41
6.3.1 Lösungen . . . 42
6.4 Qualitätssicherung, Controlling . . . 42
6.5 Ergebnisse . . . 43

7 Gesichtserkennung - Rei **44**
7.1 Allgemeines . . . 44
7.2 Technische Lösung . . . 45
7.2.1 Hardware und Aufbau . . . 45
7.2.2 Software . . . 48
7.3 Herausforderungen, Probleme, und deren Lösung . . . 63
7.4 Projektmanagement und Controlling . . . 65
7.5 Ergebnisse . . . 66
7.6 Was wurde gelernt und **welche Verbesserungs möglichkeiten gibt es** 67
7.7 Ausblick . . . 69
7.7.1 Wie geht es weiter ? . . . 69
7.7.2 Was passiert mit dem Ergebnis? . . . 69
7.7.3 was passiert mit dem Prototyp? . . . 69
7.7.4 Gibt es Folgeprojekte? . . . 69
7.7.5 Gibt es Interessanten? . . . 69

Kapitel 1

Allgemeines

Unten werden die Idee, das Thema und die Aufgabenstellung dieser Diplomarbeit verfasst.

Die Idee, ein System zu entwickeln dass Gesichter erkennt und registriert, ist daraus entstanden wegen folgenden Grunds. Es wurde vom Auftraggeber dieses System gefordert, um einen kontrollierten Zugang in der Schule zu ermöglichen. Es ist gedacht, die Sicherheit der Schule dadurch zu erhöhen und die Überwachung effizienter machen. Hauptziel ist es, alle Gesichter von den aktuellen Schülern und Lehrer zu registrieren und zu erkennen. Das System sollte auch den Unterschied zwischen einer reellen Person und einem Foto berücksichtigen. Es ist auch gefordert, dass die betreffende Person keine Maske, Brille oder Hüte bei der Gesichtserkennung trägt. Die Erkennung von Gesicht erfolgt auch nicht beim Bewegen von Person. Das Team besteht aus Aron Terzeta, Egli Hasmegaj, Rei Hoxha und Jordi Zmiani. Aufgaben sind wie folgend geteilt.

- Aron beschäftigt sich hauptsächlich mit der Gesichtsregistrierungsteil und Tiefenschärfe des Bildes herauszuholen.
- Egli kümmert sich um die wichtigsten Gesicht Daten zu extrahieren(Größe und Form der Augenhöhlen, Nase, Wangenknochen und Kiefer). Position/ Verhältnisse der Hauptmerkmale relativ zueinander herausholen. Aufbereitung der Daten für Abgleich.
- Rei: User-Gesichtsdaten von Bildverarbeitung-Funktion holen, Vergleichen von Gesichtsdaten, System aufbauen.
- Jordi: Datenbankdesign: Eine DB einrichten, Entwurf der Struktur der DB, DB in MySQL implementieren, Zugriffsberechtigungen festlegen, Error-checking.

  Wir sind dafür hoch motiviert, dieses Projekt richtig umzusetzen.

Kapitel 2

Planung

Dieses Kapitel beschreibt im Detail wie die Diplomarbeit gestaltet und abgegrenzt ist. Die Abgrenzung der Arbeit ist entscheidend wegen der hohen Komplexität des Projektes. Sie erfolgt durch Ziele, nicht-Ziele und optionale Ziele. Das ist im Unterkapitel 2.1 genau verfasst. Weiter folgt die Planung im Kapitel 2.3. Es werden hier das Lösungskonzept und die Projektmanagement erklärt. Es wird nun spezifiziert welche Projektmanagementmethode eingesetzt wurde.

2.1 Projektziele

Ziele, nicht Ziele und optionale Ziele

2.1.1 Ziele

Ziele sind wesentlich für jedes Projekt. Deshalb wurden die Ziele dieses Projekts in drei Kategorien geteilt. In der ersten Kategorie gehören Ziele, die unbedingt erfüllt werden müssen. Anderfalls wurde das Projekt scheitern.

1. Live vs. Foto unterscheiden. (3-dimensionale Erkennung an Gesicht machen. Tiefe messen damit zwischen einer Person und einem Foto differiert wird.)
2. Gesichts-Schlüsselpunkt-Extraktion, um ein Gesicht zu identifizieren.
3. Größe und Form der Augenhöhlen, Nase, Wangenknochen und Kiefer analysieren.
4. Position/Verhältnisse der Hauptmerkmale relativ zueinander herausholen.
5. Bilderdaten in Vektoren umwandeln mithilfe eines Algorithmus.
6. Abstimmung (Vergleichen mit den anderen Fotos in der Datenbank, um zu sehen, ob die Person schon registriert wurde).
7. Max. 500 Personen in einer Datenbank speichern.
8. 10 Tests, jeder Test in einer anderen Raumkondition, um alle Betriebskonditionen zu testen.

9. Datenbankdesign
10. Error checking
11. Safe Mode (eine Batterie, Back-ups in einem lokalen Server)
12. Min. Arbeitsvorbereitung (Min. Gesichtsdetektionszeit)
13. Admin account (Register-Rechte nur für Schüler und Lehrer eingeben)

2.1.2 Nicht Ziele

Hier sind die Nicht-Ziele definiert, damit das Projekt begrenzt ist und damit nichts gemacht wird, was nicht angefordert war.

1. Mehr als ein Gesicht gleichzeitig erkennen.
2. Maske, Brille, Hüte tragen.
3. Gesicht in Bewegung erkennen.
4. Person ins Profil oder andere Position sein.
5. Thermische Kamera einsetzen.

2.1.3 Optionale Ziele

Hier gehören Ziele, die optional sind. Das heißt sie sind nicht zwingend und wurden eingesetzt nur nachdem alle wichtigen und primären Ziele erfüllt sind.

1. Öffnung der Haustüren oder jeder anderen Tür mit Gesichtserkennung.
2. LCD-Display Implementation.
3. Integration in dem Infotainment-System.
4. Licht neben der Kamera (Night Vision implementieren damit die Erkennung/Registrierung auch dann funktioniert, wenn es dunkel ist.)

2.2 Projektplanung

Unsere Big Picture ist unser erstes grobes Design, das die Lösungsskizze des Projekts beschreibt. Es gibt bestimmte Gründe, warum Big Picture und Structed Design verwendet wurden, um die Software zu beschreiben. Diese Methode ermöglicht eine sehr gute Darstellung und Beschreibung des Lösungswegs. Ist schnell und leicht zu machbar. Alles ist klar sichtbar und nicht kompliziert. Big Picture und Structed Design folgt das Top-Down Prinzip, das heißt die Funktionen werden hierarchisch zerlegt (Jede Funktion wird in die folgenden Ebenen detaillierter beschreibt).

Structed Design und Big Picture haben keine Begrenzung. Dort können eindeutig alle Funktionen, Schnittstellen, Signalen und Daten beschreibt werden, sodass von allem leicht zu verstehen ist. Sehen Sie auf Abb. 2.1

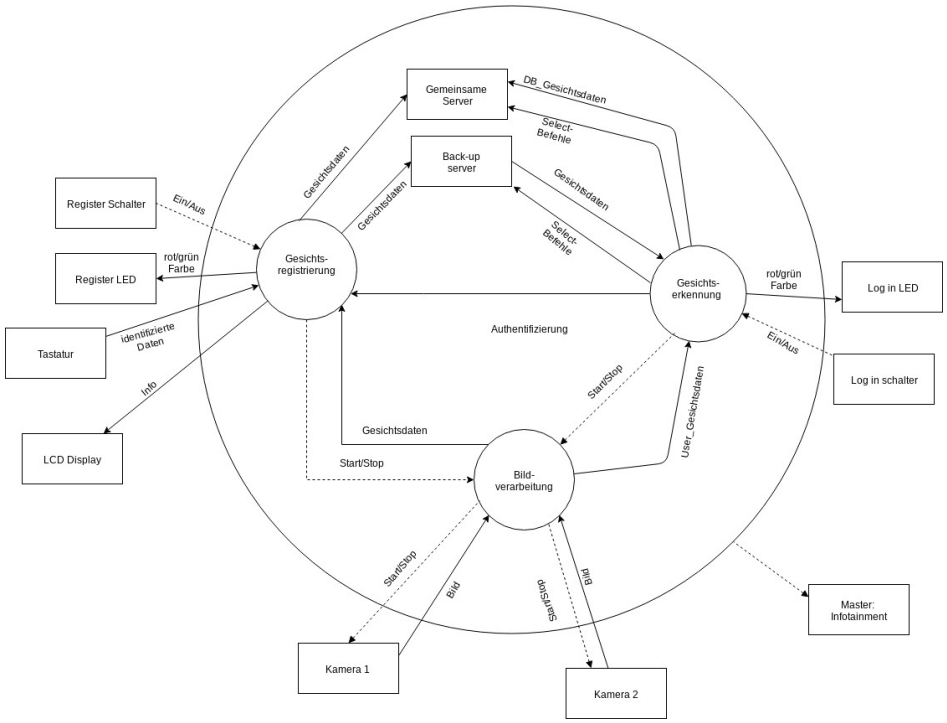


Abbildung 2.1: Big picture

2.3 Projektmanagmentmethode

Als Projektplanmethode haben wir Scrum, eine agile Methode, gewählt, weil es die Möglichkeit bietet, komplexe Projekte mit einem kleinen Personenkreis zu verwalten. Scrum ist ideal für Software- bzw. Hardware-Entwicklungsteams, weil das Team während des Projekts verschiedene Änderungen an seinem Plan vornehmen muss. Aus diesem Grund ist es besser, tägliche Zielvorgaben zu haben und in einem kurzen Zeitraum von 1 bis 4 Wochen so genannte Sprints durchzuführen, bei denen das Ziel am Ende dieser Springs ein Prototyp ist. Verschiedene Prototypen herzustellen und am Ende den richtigen auszuwählen, ist die beste Wahl für die Projektmanagementmethode zur Gesichtserkennung. Es gibt auch tägliche Pläne, in denen sich das Team zusammensetzt und entscheidet, was die Ziele für den Tag sind und was sie tun müssen. Sehen Sie auf Abb. 2.2

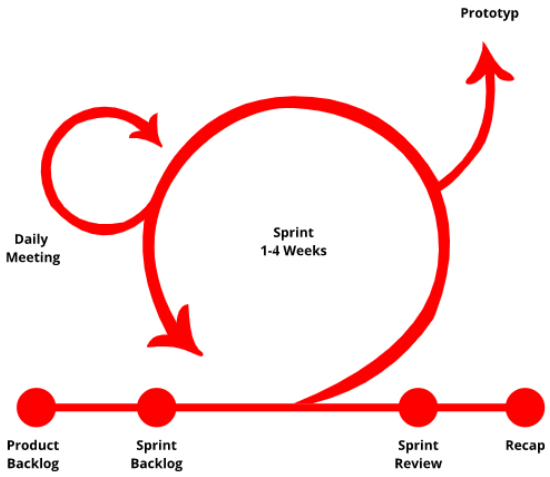


Abbildung 2.2: Scrum

Kapitel 3

Gesichtsregistrierung

In diesem Kapitel wird genauer beschrieben, wie der Gesichtsregistrierungsteil funktioniert.

3.1 Umsetzung

Hier wird erklärt, wie der Gesichtsregistrierungsteil implementiert wird.

3.1.1 Allgemein

Sicherheit ist heutzutage hoch interessant und relevant in vielen technischen und nicht technischen Bereichen. Das System, das entwickelt wird, hat mit Gesichter von Personen zu tun. Alle wissen, dass das Gesicht für jede Person anders ist. Jede Person wird mit ihrem Gesicht authentifiziert, weil es einzigartig ist. Das Gesicht hat Merkmale, die von verschiedenen Algorithmen herausgeholt werden können, um diese für die Authentifizierung der Personen zu verwenden

Das System ist in zwei Teile geteilt. Es gibt den Registrierungsteil und den Erkennungsteil. Bei dem Registrierungsteil wird die komplette Registrierung der Schüler und Schülerinnen, der Lehrer und Lehrerinnen durchgeführt. Bei dem Erkennungsteil wird die Authentifizierung der Schüler und Schülerinnen gemacht.

3.1.2 Verwendete Technologien

Hier werden die Technologien beschrieben, die verwendet wurden. Das andere Paket namens "git"könnte als eine Backup-Strategie verwendet, wenn das System abstürzt. Git ist ein Versionsverwaltungssystem, das verschiedene Versionen einer Software auf einem Server speichert. Auf dem Server gibt es dann verschiedene Versionen des Systems und die Daten können von einer bestimmten Version dann zurückgeholt werden. Es wird meistens bei der Implementierung-Phase verwendet, um die Veränderungen des Source-Codes zu verwalten.

cmake[1] ist ein Paket, das gebraucht wird, wenn man mit OpenCV[2] arbeitet. Es muss das System so konfiguriert sein, damit das OpenCV-Framework in einem C++ Programm verwendet werden kann. Deshalb brauchen wir spezielle cmake Befehle, die dies ermöglichen.

"CMake wird verwendet, um den Softwarekompilierungsprozess mithilfe einfacher plattform- und compilerunabhängiger Konfigurationsdateien zu steuern und native Makefiles und Arbeitsbereiche zu generieren, die in einer Compilerumgebung Ihrer Wahl verwendet werden können."[8]

Das gleiche passiert auch, wenn z.B. Python statt C++ verwendet wird. Die anderen Pakete wie z.B. libgtk2.0-dev pkg-config, libavcodec-dev, libavformat-dev, libswscale-dev, sind benötige Paketen, damit man das OpenCV-Framework verwenden kann.

Linux als Betriebssystem [21]
Linux ist das meist verwendete Betriebssystem der Welt. Es ist eine open-source Software. Linux ist flexibel, man kann die einzelnen Modulen wegnehmen, ohne dass das Betriebssystem abstürzt. Der Benutzer kann auch die Kernkomponenten wählen wie z.B. welcher System-Grafiken angezeigt werden, bzw. die ganzen Komponenten der Benutzeroberfläche können ausgewählt werden. Ich habe Linux aus verschiedene Gründe gewählt. Linux ist für eingebettete Systeme sehr geeignet. Es ist sicher gegen Schadprogrammen, Viren und Trojanern. Linux ist einfacher. Vorher war es ein kompliziertes System, seit den Bemühungen der Ubuntu-Fundationen und der Ubuntu-Distribution ist es sehr einfach verwendbar.
Python als Programmiersprache
"Python ist eine Programmiersprache, die 1991 veröffentlicht wurde. Python besitzt eine einfache Lesbarkeit und eine eindeutige Syntax. Python lässt sich leicht erlernen und unter UNIX, Linux, Windows und Mac OS verwenden."[11] Warum Python gewählt wurde, hat verschiedene Gründe. Python hat weniger Schlüsselwörter, reduziert die Sytax auf das Wesentliche und optimiert die Sprache. Ein Programm, das in Python geschriebt ist, ist vom Betriebssystem unabhängig. Das bedeutet, es kann Plattform unabhängig interpretiert werden.

3.1.3 Technische Lösung

In diesem Unterabschnitt wird alles was mit Technik zu tun hat erklärt. Es wird eine detalierte Beschreibung im Bezug auf die technischen Lösungen gemacht.

Das System besteht aus verschiedenen sogennanten Terminatoren. Ein Terminator befindet sich außerhalb des zu definierenden Systems. Es kann eine andere Person, System oder eine Organisation sein. "Terminatoren können von unserem System Informationen, Nachrichten, Materialien oder Energie erhalten oder das System empfängt

[1] ein erweiterbares Open-Source-System, das den Erstellungsprozess in einem Betriebssystem und auf compilerunabhängige Weise verwaltet.
[2] eine Bibliothek von Programmierfunktionen, die hauptsächlich auf Computer Vision in Echtzeit abzielen.

diese."[24]

Der wichtigste Terminator ist der "Register-Schalter". Er initialisiert das ganze System. Schalter in der Technik ist nichts anderes, nur ein Gerät zum Ein- und Ausschalten des Stroms oder zum Leiten des Stromflusses. Wenn der Schalter gedrückt wird, bekommt das System einen Input, transformiert ihn und gibt dann einen Output. Das System ist sehr einfach verwendbar.

Die Register-LED[3] dient als eine Anzeige. Wenn mit dem System etwas nicht stimmt, dann wird mit rot geleuchtet. Wenn alles passt, dann wird mit grün geleuchtet. Eigentlich hat die normale LED nur eine Farbe, aber es wird eine spezielle LED verwendet, namens RGB LED[4]. Eine RGB LED hat drei Grundfarben, rot, grün und blau. Mit diesen drei Farben kann man alle Farben erstellen. Es könnte auch zwei LEDs geben, rot und grün, aber es ist effektiver, ein RGB LED zu verwenden.

Eine spezielle Eigenschaft des Systems ist die Verwendung einer Tastatur. Sie wird verwendet, weil die einzelnen Personen ihren Namen, bzw. Email eingeben müssen. Die andere spezielle Eigenschaft ist die Verwendung eines LCD-Screens[5]. Da werden z.B. Errors oder die Login-Daten angezeigt. Es ist für den Benutzer einfach zu erkennen, wenn es ein Problem mit dem System gibt. Eigentlich ist die Hauptfunktion des LCD-Screens, den Input des Benutzers anzuzeigen. Warum es so geplant ist? Das Problem besteht darin, dass, wenn der Benutzer seine Email-Adresse schreibt, Fehler passieren können, weil er nicht sieht, was er schreibt. Um das zu vermeiden, wird der LCD-Screen verwendet. Der Benutzer kann sehen, was er schreibt.

Um registrierte Personen mit ihren Gesichtsdaten zu speichern, braucht das System einen Server. Auf diesem Server läuft ein Datenbank Management System. Die Datenbank ist so konfiguriert, dass die Person mit ihren Infos gespeichert werden können.

Das System hat auch einen Backup-Server. Die Daten werden parallel bei dem Hauptserver sowie bei dem Backup-Server gespeichert, damit die Daten noch verfügbar sind, wenn der Hauptserver ein Problem hat. Die Verwendung des Backup-Servers ist zustande gekommen, weil das System 24/7 arbeiten muss. Wenn der Hauptserver Wartungen oder Probleme hat, kann der Backup-Server die Arbeit übernehmen. Auf Abb. 3.1 wird gezeigt, wie der Gesichtsregistrierung-Teil arbeitet.

[3]Light-emitting Diode, fließt Strom durch, strahlt sie Licht aus
[4]rot, grün,blau LED
[5]Liquid Crystal Display, präsentiert die elektrischen Signale in Form von visuellen Bildern.

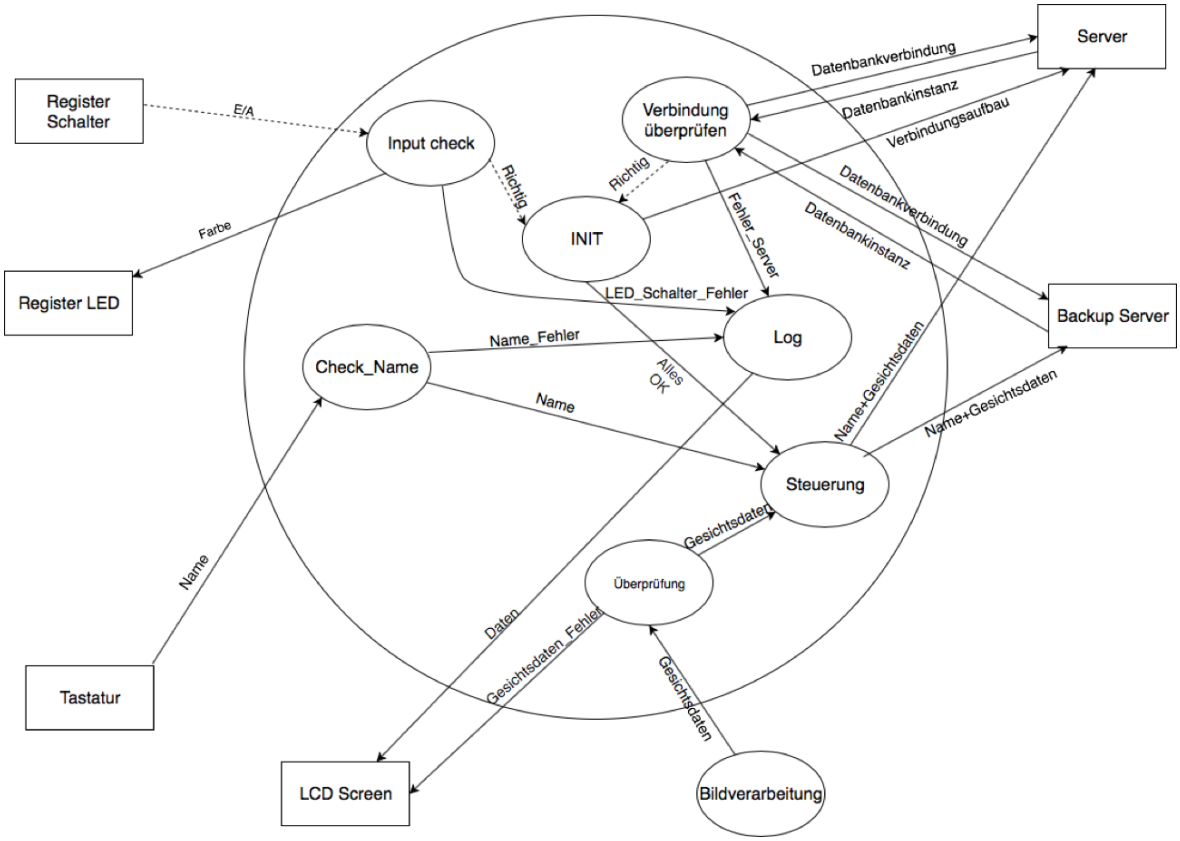


Abbildung 3.1: Structed Software Design bzw. erste Ebene

Das System hat verschiedene Terminatoren. Die Terminatoren sind im Kapitel 3.1.3 erklärt. Auf Abbildung 3.1 können 6 Terminatoren identifiziert werden:

- Register Schalter - Externer Trigger Schalter zum Starten des Registrierungsteils
- Register LED - Anzeiger
- Tastatur - Input Hardware für die Datenerfassung(Name,Email,Rolle)
- LCD Screen - Anzeiger
- Server bzw. Backup Server - eigener Server für die Speicherung der Daten

Auf der erste Ebene können 7 Hauptprozesse identifiziert werden:

- Input check - Kontrolliert in einer unendlichen Schleife, ob der Register Schalter gedrückt wird.
- Check_Name - Kontrolliert, ob der Name in einem richtigen Format (Nur Buchstaben) eingegeben wird.
- Verbindung überprüfen - Überprüft, ob die Server im Betrieb sind und, ob die Verbindung richtig aufgebaut ist
- Log - Speichert die Fehler
- Steuerung - Übernimmt die zentrale Steuerung der Andwendung
- INIT - Initialisiert das ganze System
- Überprüfung - Die Gesichtsdaten bzw. Gesichtspunkte werden kontrolliert und entweder bei der Steuerung weitergeschickt oder auf dem LCD Screen angezeigt, wenn einen Fehler auftritt.

Um Schalter und LEDs im System verwenden zu können, brauchen wir ein spezielles Paket namens "RPi.GPIO". Dieses Paket macht es möglich, den Raspberry PI mit Hardware (LED und Schalter) verbinden zu können. Dafür werden GPIOs verwendet. Der Schalter hat 3 Anschlüsse. Einer wird mit 5V verbunden, der andere mit Ground und der dritte ist für Daten. Dies wird dann mit einem GPIO-Port in Raspberry PI verbunden. Das gleiche gilt auch für die LED, damit sie vom Raspberry PI kontrolliert verwenden kann, wird sie mit einem GPIO-Port verbunden. Mithilfe dieser GPIO-Ports bekommt das System eine Information, wenn der Schalter gedrückt wird. Wert "1" bedeutet, dass der Schalter gedrückt wird und die verschiedenen Skripten aufgerufen werden.

Schritte:

1. Am Beginn des Skripts wird diese Zeile schreiben: "#!/usr/bin/python". Es gibt zwei Gründe, warum diese Zeile geschreibt wird. Der erste Grund ist, dass dieses Program mit einem Python-Interpreter ausgeführt wird, der zweite ist, der Inhalt der Datei wird von der Python-Binärdatei unter /usr/bin/python interpretiert

2. Alle Paketen importieren. Sehen Sie Abb. 3.1

Listing 3.1: Packages zu importieren

```
import RPi.GPIO as GPIO
import time
import os
import subprocess
import sys
```

- RPi.GPIO ist ein Paket, das verwendet wird, um Zugriff auf die sogennanten GPIO-Ports[6] zu haben. Vorher habe ich erwänt, wenn wir Zugriff auf die HW-Komponenten haben wollen, die mit Raspberry PI verbunden sind, brauchen wir die GPIO-Ports. Um diese GPIO-Ports in Python zu verwenden, brauchen wir das sogennante Paket "RPi.GPIO". Es gibt verschiedene Pakete, die einen Zugriff zwischen GPIO-Ports und Python ermöglichen, wie z.B. rpi.gpio, GPIOZero usw. Es wird das rpi.gpio Paket verwendet, weil es leicht verständlich, programmierfreundlich und einfach zu verwenden ist. [25]
- time ist ein Paket in Python. Von diesem Paket wird nur die Funktion "sleep" verwendet. Diese Funktion pausiert das python-Programm. [15]

[6]Eingehende und Ausgehende digitale Signale, am Eingangsport können verwendet werden, um digitalen Messwerte mitzuteilen, die von Sensoren empfangen werden

- os ist das wichtigste Paket in unserem Skript. Es erlaubt uns, dass wir in einem Python-Skript andere Skripte aufrufen können. Es ist egal, in welcher Programmiersprache diese Skripten geschrieben sind. Es gibt auch verschiedene Methoden, wie man verschiedene Skripte in einem Python-Skript aufrufen kann. Man kann mit dem subprocess Paket, eine Main-Funktion in dem Skript starten und die verschiedenen Funktionen des anderen Skripts aufrufen.
- Das "subprocess" Paket dient zur Verbindung zwischen verschiedenen Prozessen, in meinem Fall, den Aufruf eines Skriptes.
- Das sys Paket wird verwendet, um Console Parameter zu bekommen. Das bedeutet, wenn das Skript aufgerufen wird, wie z.B. login.py dann kann man login.py einen Parameter mitgeben: login.py <parameter >

3. GPIO-Ports Datenrichtung einrichten. Datenrichtung für LED ist "out", weil das LED als ein Output für unseres System dient. Datenrichtung für Schalter ist "in", weil der Schalter als ein Input für unseres System dient. Auf Abb. 3.2 ist der Python-Code.

Listing 3.2: GPIO-Ports Konfiguration

```
GPIO.setmode(GPIO.BCM)
GPIO.setup(23,GPIO.OUT)
GPIO.setup(18,GPIO.IN)
GPIO.setup(17,GPIO.IN)
GPIO.setup(27,GPIO.OUT)
```

Es gibt verschiedene Betriebsarten für GPIO wie BCM und Board. Ich verwende BCM[7], weil ich das Paket RPi.GPIO verwende. Mit diesem Paket darf nur die Betriebssart "BCM "verwendet werden. [16] Für die Registrierung der Schüler und Schülerinnen bzw. Lehrer und Lehrerinnen ist es nötig, dass der Admin sich bei dem System einloggt. Die Überprüfung, ob der Admin da ist oder nicht, wird mit einem Vergleich von zwei Bildern gemacht. Ein Bild von Admin ist gespeichert, das andere wird gemacht. Dazu wird das Skript, das Bilder macht, aufgerufen, und dann vergleicht man es mit einem Skript mit dem anderen Bildern. Es gibt "matched" zurück, wenn die Gesichter bei den beiden Bildern übereinstimmen und "not matched"wenn die Gesichter nicht übereinstimmen.

[7]Broadcom Pin Number

4. Dann kommt der Teil “Input check”. Hier verwende ich dann die Methode “input”. Die Methode befindet sich im Paket “RPi.GPIO” und gibt entweder true oder false zurück. Im Verzeichnis /sys/class/gpio/gpio<GPIO-PORT>”gibt es zwei Dateien, value and direction. Direction für den Port des Schalters ist IN und für den Port des LEDs ist OUT. Mit der Methode “input” hole ich den Wert (value) der Schalter-Port. Wenn der Schalter gedrückt wird, wird der Wert “1” herausgekommt , 1 repräsentiert “true”. Das bedeutet, dass die Input-methode “true” zurückliefert und das Programm läuft weiter.

   Nachdem der Schalter gedrückt wird, wird ein Skript aufgerufen. Dieses Skript dient zur Registrierung der Person in der Datenbank. Für die Registrierung der Schüler und Schülerinnen bzw. Lehrer und Lehrerinnen ist es nötig, dass der Admin sich bei dem System einloggt. Ist der Admin da, können die Personen mit der Registrierung beginnen. Diese Person wird nach ihrem Vornamen, Nachnamen, Email und Rolle gefragt. Mit der Rolle ist gemeint, die Funktion bzw. die Stelle dieser Person in der Schule, ob die Person ein Schüler, Lehrer oder ein Admin ist. Die Rolle wird vom Admin geschrieben. 1 steht für Admin, 2 für Schüler und 3 für Lehrer und Lehrerinnen. Sie wird eigentlich nur für die Verwaltung des Systems verwendet. Wenn der Admin wissen möchtet, wie viele Lehrer oder Schüler schon registiert sind, oder ob es noch nicht registrierte Personen gibt, dann schaut er in der Datenbank nach. Die E-Mail speichern wir dann in einer Variable und diese Variable übergeben wir dann an ein anderes Skript. Dieses Skript erstellt mit der Kamera eine Verbindung und macht ein Bild. Der Name des Bildes ist gleich mit der Email der Person. Es ist so gewählt, weil es für das Einfügen der Daten in der Datenbank und bei der Speicherung des Paths des Bildes in der Datenbank mit dem gleichen Namen wie die E-Mail einfacher ist.

   Es wird die E-Mail verwendet, weil es eine performantere Suche in der Datenbank ermöglicht. Die E-Mail ist einzigartig, nur einmal für jede Person. Von 1000 Datensätze wird nur der Datensatz ausgegeben, in dem die E-Mail mit der eingetippten E-Mail vom Benutzer übereinstimmt.

   In der Datenbank gibt es nur eine Tabelle. In dieser Tabelle wird alles gespeichert: Name, E-Mail, Rolle, Path des Bildes und alle Gesichtspunkte. Für die Registrierung der Personen in der Datenbank werden zwei Schritte durchgeführt. Der erste Schritt ist das Einfügen der neuen Person mit ihrem Vornamen, Nachnamen, E-Mail und Rolle in der Datenbank. Der zweite Schritt ist das Update der Gesichtspunkte. Es wird ein Bild gemacht, von diesem Bild werden die Gesichtspunkte extrahiert. Danach werden diese Gesichtspunkte in den zuständigen Spalten der Datenbank gespeichert. Dieser Schritt wird durch eine Update-Anweisung gemacht. Nachdem die Update-Anweisung erfolgreich ist und die Person mit ihrem Vornamen, Nachnamen, E-Mail, Rolle, und ihren Gesichtspunkten gespeichert ist, wird die LED grün angesteuert. Wenn ein Fehler auftritt, wird die LED rot angesteuert. Es gibt einige Fehler, die bei der Registrierung auftreten können:

   (a) Auf dem Bild gibt es kein Gesicht oder mehr als ein Gesicht.

   (b) Das Gesicht ist schlecht positioniert und es ist nicht möglich, die Gesichtspunkte zu extrahieren.

(c) Die Person ist schon registriert.

Wenn ein von den oben erwänten Fehler auftritt, wird das auf dem Bildschirm angezeigt und die LED wird rot angesteuert.

3.1.4 Herausforderungen

Das Projekt hatte für mich viele Herausforderungen. Das lag davon, dass es ein ziemlich großes Projekt war, und ich neue Technologien verwendet habe, die ich vorher nie verwendet hatte. Ich habe z.B. keine Erfahrung mit OpenCV, Python und andere Dinge, die ich später erwähnen werde. Ich habe von diesen Herausforderungen und Problemen viel gelernt. Einerseits bin ich froh, anderseits bin ich frustiert, weil dies das Enddatum des Projekts verzögert hat. Die Herausforderungen waren:

- **opencv zu installieren**. Es hat 3 Woche gedauert, bis ich es installiert hatte.
- **Beginn des Projektes**. Der Beginn eines Projektes ist immer schwierig. Die Koordination im Team war sehr schwierig. Ich, als Projektleiter, musste allen sagen, wie sie arbeiten sollen, wo sie die Dateien finden können usw. Das war die größte Herausforderung.
- Git repository, Einrichtung von git. Manche Teammitglieder wussten sehr wenig von git und ich musste es ihnen erklären. Manchmal gab es merge conflicts, weil sie pull gemacht haben, ohne dass Sie die Änderungen commited haben. Ich sollte dies lösen, weil ich mehr Erfahrung mit git hatte.
- Python als Programmiersprache. Wir wollten vorher mit C++ arbeiten, aber es war sehr schwierig, OpenCV in Visual Studio zu installieren. Manche von uns wollten in Windows arbeiten und der einzige Weg war, mit Visual Studio zu arbeiten. Es hat nicht funktioniert, deshalb sind wir zu Python gewechselt. Wir haben Python gewählt, weil opencv in Python sehr einfach installierbar ist. Mit Python hatten wir keine große Erfahrung. Alles, was ich mit Python gemacht habe, ist eine Verbindung mit der Datenbank und Aufbau der SQL-Abfragen (Insert,Select,Update,Delete). Alle andere Wissen sollte ich selbst von Bücher, Internet, Tutorials lernen. Die große Herausforderung hier war, die richtigen Quellen zu finden.
- Verwendung der Kamera und verbinden mit Python. Ich wusste nicht, welche Funktionen man verwendet, um die Verbindung mit der Kamera zu erstellen.
- Bei der älteren Version von Raspbian heißt das Paket, das python mit dem Datenbank Managment System(MySQL) verbindet, 'python-mysqldb' und jetzt heißt es 'python-mariadb'. Ich wusste das nicht und dies hat mir etwas Zeit gekostet.
- Abhängigkeiten zwischen einzelnen Arbeitsteilen. Die Aufgaben waren so geteilt, dass es Abhängigkeiten zwischen ihnen gab. Das hat dann zu einer Verspätung der Projektabgabe geführt, weil die Teammitglieder aufeinander warten mussten.

- Die großen Teile meiner Planung haben gepasst, nur wenige Kleinigkeiten musste ich ändern. Sie sind erst in der Implementierungsphase angezeigt.
- Problem mit dem Zugriff auf die Elementen eines Numpy-Arrays.
- Verbindung der Skripte miteinander und Verknüpfung der Variablen, die sich in verschiedenen Skripten befinden.
- Aufteilen der Gesichstpunkte in zwei Arrays. In einem Array nur die X-Werte und in dem anderen Array nur die Y-Werte.
- Die Aufgabenteilung zwischen den Teammitgliedern war unausgewogen.

Ich habe diese Lösungen für die Herausforderungen gefunden:

1. Ich habe viel Tutorials geschaut, Websites gelesen, wie opencv in Raspberry PI installiert werden kann. Ich habe viele verschiedene Methoden ausprobiert, aber keine guten Ergebnisse erzielt. Nach vielen Versuchen wurde es erfolgreich installiert. Dannach habe ich verschiedene Skripte in Python gemacht, um es zu testen. Manche der Skripte haben funktioniert, manche nicht. Eine kleine Herausforderung war für mich, dass ich die Skripte, die nicht richtig funktioniert haben verbessert. Anschließend habe ich herausgefunden, dass das Problem bei dem Kompilieren von opencv aufgetreten sind (cmake). Ich habe es noch einmal vom Beginn kompiliert.Das hat das Problem gelöst, alle Skripte funktionieren, es gibt keine Fehler mehr, die mit dem opencv Paket zu tun haben.
2. Ein Treffen mit meiner Gruppe vor dem Beginn des Projektes war notwendig. Ich habe Ihnen gesagt und erklärt, in welchen Verzeichnise sie arbeiten sollten, die Struktur der Dokumentation, welcher Kommunikationskanal verwenden wir, um Probleme, Herausforderungen usw. zu besprechen. Jede Person hatte ihre Vorschläge, um Aufgaben zu lösen, und dieses Treffen hat sehr lange gedauert, bis alle verstanden hatten, wie, wo,was, wann gemacht werden soll. Aber auch nach dem Treffen gab es zwischendurch Missverständisse bzw. Probleme mit der Kommunikation, z.B. wurde nicht im richtigen Verzeichnis gearbeitet usw.
3. Ein Git-Repository zu erstellen und einzurichten war einfach. Ich hab es online in github.com erstellt, einen Name eingegeben und dann die anderen Teammitglieder als Collaborators hinzugefügt. Um strukturierter zu arbeiten, habe ich verschiedene Branches angelegt. Es gab mit dem Befehl “push” und “pull” Probleme. Das habe ich gelöst, indem ich allen Teammitgliedern gesagt habe, dass, wenn sie in einem Github-Repository arbeiten möchten vor dem Beginn der Arbeit einen “pull” machen müssen, damit die Änderungen, die von anderen in dem Repository gemacht wurden, mit der Version am Computer synchronisiert wird. Sie wussten nie, was die anderen in diesem Repository gemacht haben. Sie machen “push”, ohne zu sagen, dass sie einen “push” gemacht haben. Das führt dann zu merge-Probleme usw.

4. Ich habe jedem Teammitglieder gesagt, er muss mindestens zwei Wochen mit dem Lernen von Python verbringen. Tutorials ansehen, Beispiele selbst probieren, die Quellen dafür selbst finden.
5. Für die Verbindung der Kamera mit OpenCV, gibt es einen Skript in der offiziellen Website-Dokumentation von OpenCV. Da habe ich alle Funktionen gesucht und gefunden, die ich brauchte, um die Kamera in Python verwenden zu können.
6. Um die Abhängigkeiten zu minimieren,hat jedes Teammitglied andere Aufgaben bekommt, als die, die in der Dokumentation stehen. Ich war gezwungen, diese Änderung zu machen, sonst hätte das Projekt viel länger gedauert.
7. Bei der Implementierung sind Kleinigkeiten herausgekommen, die bei der Planung nicht berücksichtigt waren. Die habe ich direkt in der Implementierung verbessert, ohne dass ich nocheinmal die Planung machte. Aber ich habe diese Kleinigkeiten zur Kenntnis genommen, damit ich in der Zukunft keinen solchen Fehler(Kleinigkeiten) mehr in der Planungsphase machen werde.
8. Ich habe die Dokumentation von Numpy Array nicht gut gelesen. Ich bat meinem Betreuer um Hilfe, weil ein Problem mir viel Zeit genommen hat. Der Betreuer hat mir dann geholfen, die Lösung zu finden.
9. Alle Teammitglieder haben ihre Skripts erstellt. Es war meine Aufgabe, alle diese Skripts mit einem Prototyp-Skript zu verbinden. Es war keine wissenschaftliche Arbeit, man musste viel testen, während man den Prototyp erstellte. Nach vielen Tests, habe ich erfolgreich einen Prototyp erstellt.
10. Eigentlich sollte Egli die Gesichtspunkte in ein Array speichern. Sie konnte dieses Ziel nicht erfolgreich abschließen und ich sollte ihr helfen. Zuerst wollte ich alle Gesichtspunkte in ein 2D Array speichern. Nach vielen Versuchen kam ich zu dem Schluss, dass es schwierig war und viel Zeit kosten würde. Stattdessen habe ich die Gesichtspunkte in zwei Arrays gespeichert. In einem Array habe ich nur die x-Werte gespeichert und in dem andere Array habe ich nur die y-Werte gespeichert. Das war eigentlich die einzige Möglichkeit, wie ich dieses Problem lösen konnte.
11. Am Beginn der Diplomarbeit hat das Team eine Aufgabenteilung gemacht. Mit der Zeit haben die Teammitglieder bemerkt, dass nicht jedes Teammitglied die gleiche Menge an Aufgaben hat. Dannach haben wir uns dazu entschlossen, eine neue Aufgabenteilung zu machen. Zuerst hatte ich auch den 2D vs 3D Unterschied zu realisieren. Mit der neuen Aufgabenteilung ist es ein Ziel von Jordi und nicht mehr ein Ziel von mir.

3.1.5 Qualitätssicherung und Controlling

Risiko ist meistens eine Einschätzung, was es einem Unternehmen kostet, wenn die Projektziele nicht erreicht werden. Ich, als Projektleiter, musste das machen. Eine Risikoanalyse zu erstellen ist sehr schwierig, weil es mit der Zukunft zu tun hat. Zuerst musste ich an die Zukunft denken, bei welchen Bauteilen können Fehler auftreten,

welche Programme können ausfallen. Das bedeutet, einen Überblick über die Zukunft zu haben und einzuschätzen, was für Fehler und Risiken es geben kann. Dann schätzte ich die Wahrscheinlichkeit ihres Eintretens ab und am Ende die Maßnahmen. Dahinter versteckt sich eine größe Arbeit.[13] Auf Abb. 3.2 wird die Risikoanalyse dargestellt. Wahrscheinlichtkeit, Kosten usw. sollen beachtet werden.

Risikoanalyse: Gesichtsregistrierung und Gesichtserkennung

Risikotyp	Nr.	Wahr-sch.	Aus-wirk.	Ampel	Manager	Beschreibung	Behandlung und Kontrolle	Termin / Nächster Schritt
Standardrisiken								
Ressourcen	1	1	8	8	Rei Hoxha	Ausfall von Ressourcen	Ressourcen im Voraus sichern (Reserven an Mitarbeitern, an HW, an Zeit)	Projekt gleich abschließen
Planung	2	4	8	32	Aron Terzeta	Schlechte Planung (verschiedene Eintrittsfälle nicht berücksichtigt)	Nocheinmal mit der Planung beginnen	Den Projektantrag ablehnen
Technik	3	3	3	9	Egli Hasmegaj	Nicht eindeutig Gesichtspunkte-Extraktion	Fertige Skripts aus dem Internet holen	Die Implementierung zu anderen Firmen zulassen
Staat	4	8	7	56	Aron Terzeta	Rechtliche Aspekte (Persönliche Informtionen) + Datenschutz	Mit dem Staat vor der Implementierung sprechen	Geld zu Staat bezahlen, damit der Staat nicht in Schwierigkeiten uns bringt
Planung	5	2	4	8	Jordi Zmiani	Mehr Benutzer als geplant (Datenbankdesign)	ein skalierbares Design der Datenbank	nocheinmal Datenbank erstellen
Zeit	6	3	5	15	Alle	Spät mit Arbeit begonnen, z.B. die Installation von OpenCV sollte 4 Stunden aber ist 2 Tage	Schneller dann arbeiten oder vorher dem Kunden sagen, dass es ein bisschen spät das Produkt fertig ist.	Kunden sagen, dass entweder wartet bis zum Ende(eine große Verspätung) oder nimmt das nicht fertige Produkt zurück
Staat	7	4	0	0	Niemand	Wächter wird seinen Job verlieren, weil System viel Know-How braucht	-	-
Kommunikation	8	4	8	32	Alle	Wenn das Team keine gute Beziehung zwischen den Mitgliedern hat	Versuchen, einen gemeinsamen Weg und Sprache zu finden, wenn nicht, neue Teammitglieder	Team wechseln
Technik	9	3	3	9	Rei Hoxha	HW nicht genug, keine Know-How, wie man die speziellen Bauteile verwenden kann	Reserven, Bedienungsanleitungen lesen(auch in Internet suchen)	Experten fragen, Hilfe von Experten bekommen
Technik	10	5	5	25	Alle	Mangelnde Einfahrung	Tutorials sehen	Teile von Algorithmen und Skripts von Internet holen
Technik	11	3	3	9	Aron Terzeta	Nachträgliche Änderungswünsche des Systems	Vorher planen (Skalierbarkeit und Erweiterung)	Nocheinmal Planung
Technik	12	2	3	6	Aron Terzeta	Veränderung am kritischen Weg	Schnell den neuen kritischen Weg finden	Projekt ohne kritischen Weg(sehr gefährlich)
Technik	13	1	3	3	Aron Terzeta	fehlende Terminüberwachung	-	-
Zeit	14	2	2	4	Alle	Zeitprognose unterschätzen	Sagen, dass es eine Verspätung gibt. Der Kunde und der Chef muss es wissen	-
Zeit	15	3	4	12	Jordi Zmiani	Mangelnde Puffer in der Kalkulation	Mehr Stunde daran bis zum Ende arbeiten	Hilfe von außen bekommen
Ressourcen	16	1	1	1	Rei Hoxha	Mangelhafte Kontrolle der Projektkosten	Selbst dann den Mangel bezahlen	-
Ressourcen	17	2	4	8	Rei Hoxha	fehlende Ausrüstung	Direkt mit dem Projektleiter sprechen, und dann er entscheidet, ob es gekauft, ausgeliehen oder ... wird	-
Planung	18	0	1	0	Aron Terzeta	Geringe Personalkapazitäten	Entweder bleiben wir mit diesen Personalkapazität und das Produkt später fertig machen oder neue Personal einstellen, damit das Produkt in time fertig zu machen	-
Ressourcen	19	1	4	4	Alle	Ausfall einzelner Projektglieder	Ein anderer Projektglieder diese Arbeit machen	Einem neuen Team das Projekt einrichten

Abbildung 3.2: Risikoanalyse in Excel

Es gibt eine Priorität bei den Maßnahmen. Je höher die Zahl in der Spalte „Ampel“ ist, desto wichtiger die Maßnahmen und die Risiken sind. Die, in denen die rote Farbe ist, werden zuerst behandelt. Wenn es grün ist, bedeutet es, dass das Risiko eine sehr kleine Rolle auf das System spielt. Sie werden am letztens behandelt, wenn alle andere Maßnahmen für die roten und gelben Risiken getroffen werden. Spalte “Ampel”:

- 0 : das Risiko spielt keine Rolle oder eine sehr kleine Rolle. Die Maßnahmen für diese Risiken können getroffen werden.
- 1-3 : das Risiko hat eine Bedeutung. Die Maßnahmen für diese Risiken sollen getroffen werden.

- 3-100 : das Risiko spielt eine große Rolle auf das System. Die Maßnahmen für diese Risiken müssen getroffen werden.

3.2 Ergebnisse

Die Zeit für den Abschluss der Diplomarbeit ist vorbei. Jetzt sollen alle Ziele erfolgreich abgeschlossen werden. Meine Ziele sind eigentlich erfolgreich abgeschlossen. Eine Person kann sich mit ihrem Vornamen, Nachnamen, E-Mail und Rolle registrieren lassen. Die Gesichtspunkte werden von dem Gesicht dieser Person extrahiert und in der Datenbank zusammen mit Vornamen, Nachnamen, E-Mail und Rolle gespeichert. Das System hat zwei Administratoren. Die Administratoren loggen sich mit einem Passwort ein. Ohne das Einloggen des Administrators kann keine Person registriert werden. Am Beginn war geplant, dass das System ein LCD-Screen enthält. Jetzt geht es sich mit der Zeit nicht aus, dieses LCD-Screen zu programmieren und im System zu implementieren. Statt des LCD-Screens wird einen Bildschirm verwendet. Es wird nun alles auf dem Bildschirm angezeigt. Diese Veränderung hat eigentlich keine Bedeutung für das System.

3.2.1 Implementierung

Nachdem einer umfangreichen und guten Arbeit meinerseits, kann ich einen Prototyp mit einigen Funktionen vorweißen. Die Funktionen, die in diesem Prototyp integriert sind, gehören zu den Zielen, die ich realisieren sollte. Diese Funktionen sind:

- Admin-Konto. Die Registrierung einer neuen Person geht nicht ohne ein Admin-Konto. Der Admin muss bei dem System eingeloggt sein, damit der Registrierungsteil funktioniert. Der Admin loggt sich mit einem Passwort bei dem System ein.
- Eine Person wird mit ihrem Vornamen, Nachnamen, E-Mail und Rolle in der Datenbank gespeichert
- Ein Bild von einer Person wird gemacht
- Von diesem Bild werden die Gesichtspunkte extrahiert
- Die Gesichtspunkte werden in den Arrays gespeichert.
- Ich hole dann die Werte von diesen Arrays und speichere sie in der Datenbank.

Kapitel 4

Umsetzung - Jordi

4.1 Allgemeine Beschreibung

Ein wichtiger Teil jedes Projekt ist die Datenbank, und dass ist der Teil wo ich am meisten konzentriert bin. Andere Teilebereiche wo ich teilnehmen habe, sind bei der Extraktion der Gesichtspunkte, wo mein Job, den Tiefe von Bildern zu finden, damit ein 2D vs. 3D unterschied geben sollte, ist.

Technologie	Beschreibung	Lizenz
MySQL	Datenbankverwaltungssystem	Kostenlos
MariaDB	Datenbankverwaltungssystem	Kostenlos
Python	Objekte-orientierte Programmiersprache	Kostenlos
OpenCV	Bildverarbeitung Programmbibliothek	Kostenlos

Tabelle 4.1: Technologien

MySQL

MySQL ist eine echte Multi-User, Multi-Treaded SQL Datenbank und wird von allen großen Providern oder auch Suchmaschinenbetreibern eingesetzt. MySQL ist eine CLi-

Abbildung 4.1: MySQL Logo
[2]

ent/Server Implentierung, die aus einem Server-Dämon mysqld und vielen Client Programmen, sowie Bibliotheken für PERL, PHP/3, PHP/4 sowie ASP besteht. SQL ist eine standardisierte Datenbanksprache, die das Speichern, Updaten und den Zugriff auf Informationen erleichtert. Beispielsweise kann man Produktinformationen eines Kunden auf einem WWW-Server speichern und abrufen. MySQL ist äußerst schnell und flexibel genug, um sogar Bilder und Log-Dateien darin abzulegen. In der Praxis ist MySQL sehr viel schneller, als z.B. ORACLE oder INFORMIX.[23]

MySQL hat ein breites Anwendungsspektrum und wird meistens in Verbindung mit PHP, Linux, Python usw. verwendet. Der Grund zu der Auswahl von MySQL steht bei der einfachen Bedienung und Verwaltung von Datenbank. Da es sich bei DBS um eines der am häufigsten verwendeten DBS handelt, gibt es eine Vielzahl von Tools, die zum Verwalten der Datenbank verwendet werden können.

MariaDB

MariaDB ist aktuell die am schnellsten wachsende Open-Source-Datenbanklösung. Sie wird hauptsächlich von der MariaDB Corporation entwickelt und ist ein Fork von MySQL. Mittlerweile bietet das Datenbankverwaltungssystem mit seinen diversen kostenfreien Features vieles, was MySQL nicht oder nur kostenpflichtig zur Verfügung stellt (z.B. eine Speicher-Engine zur performanten Verarbeitung von riesigen Datenmengen; ein Datenbank-Proxy zur sicheren und hoch-verfügbaren Verwaltung skalierbarer Installationen u.v.m.). Im Gegensatz zu MySQL verfügt MariaDB jedoch nicht über einen eigenen Client wie die Workbench. Eine gute kostenfreie Alternative stellt HeidiSQL dar, jedoch verfügt diese über kein Dashboard, welches z.B. die Funktionsweise des Servers darstellt und damit Optimierungsentscheidungen erleichtert.[12]

Abbildung 4.2: MariaDB Logo
[3]

Es ist hauptsächlich in Linux implementiert, da es das Standard-DBS ist. Es bietet eine einfache Verbindung mit Linux-Dateien wie Python, C ++ und vielen anderen Programmiersprachen. Da dies der Standard-Linux-DBS ist und unsere Software unter Linux mit Python-Skripten geschrieben wurde, ist MariaDB unsere Wahl für DBS.

Python

Python ist eine Programmiersprache, die dank ihrer klaren Syntax und einfachen Lesbarkeit leicht zu erlernen ist und sich sehr vielseitig einsetzen lässt. Für die gängigen

Betriebssysteme ist Python frei verfügbar. Die üblichen Programmierparadigmen wie die objektorientierte oder funktionale Programmierung werden unterstützt.
Bei Python handelt es sich um eine Programmiersprache mit einer klaren Syntax und guten Lesbarkeit. Sie gilt als leicht zu erlernen und ist in den gängigen Betriebssystemen interpretierbar.Python unterstützt mehrere Paradigmen der Programmierung wie die funktionale, objektorientierte oder aspektorientierte Programmierung und ist auch als Skriptsprache nutzbar.[20] Die Sprache weist ein offenes, gemeinschaftsbasier-

Abbildung 4.3: Python Logo
[4]

tes Entwicklungsmodell auf, das durch die gemeinnützige Python Software Foundation gestützt wird, die de facto die Definition der Sprache in der Referenzumsetzung CPython pflegt.

Der Grund, warum die Software in Python geschrieben ist, ist, dass Python eine einfache und schnelle Möglichkeit bietet, Datenbankskripte und Gesichtserkennungsprogramme zu erstellen.

OpenCV

OpenCV (Open Source Computer Library) ist eine Open-Source-Bibliothek für Computer Vision und Machine-Learning-Software. OpenCV wurde entwickelt, um eine gemeinsame Infrastruktur für Computer Vision-Anwendungen bereitzustellen und die Nutzung der maschinellen Wahrnehmung in den kommerziellen Produkten zu beschleunigen. Als BSD-lizenziertes Produkt macht OpenCV es Unternehmen leicht, den Code zu nutzen und zu ändern.Es verfügt über C++, Python, Java und MATLAB-Schnittstellen und unterstützt Windows, Linux, Android und Mac OS.[6]

Abbildung 4.4: OpenCV Logo
[5]

Obwohl es in C ++ geschrieben wurde, bietet es viele Schnittstellen zu anderen Programmiersprachen wie Python, die in diesem Projekt implementiert werden. OpenCV ist eine plattformübergreifende Bibliothek und da es viele Deep-Learning-Algorithmen und Frameworks enthält, musste es unsere Wahl für die Objekterkennung sein.

4.2 Technische Lösungen

4.2.1 Datenkbank

Wie bereits erwähnt, haben wir uns für MySQL und MariaDB als DBS entschieden, wobei meistens das zweite mehr zum Einsatz kommt. Es ist wichtig zu beachten, dass die Datenbank nicht über die Befehlszeile oder die MySQL/MariaDB-Konsole implementiert wurde, sondern mithilfe von Python-Skripten. In diesem Fall würde jeder gegebene Befehl in den Python-Dateien gespeichert, und wenn etwas schief ging, könnten die Skripte einfach verbessert und ein Fehler behoben werden.

Um eine Datenbank zur Gesichtserkennung und -identifikation zu entwerfen, mussten einige Nachforschungen angestellt werden, bei denen es hauptsächlich darum ging, Beispiele für vorhandene Datenbanken zu finden. Auf diese Weise könnten wir uns eine Vorstellung davon machen, wie die Datenbanken aussehen. Obwohl die meisten Datenbanken nicht kostenlos waren und einige Unterlagen unterschrieben werden mussten, um die vollständigen Details zu erhalten, war dies mehr als ausreichend, um einen Ausgangspunkt zu haben. Einige der gefundenen Datenbanken sind:

- 3D Mask Attack Dataset
- The AR Face Database, The Ohio State University
- Caltech Faces
- CAS-PEAL Face Database
- The Color FERET Database, USA

Auf den ersten Blick scheint es nicht sehr schwierig zu sein, eine Gesichtserkennungsdatenbank zu erstellen, aber da sich das Projekt entwickelt, müssen viele Änderungen vorgenommen werden, da sie an die Gesichtsdaten der Person weitergegeben werden müssen.

Der nächste Schritt bestand darin, in Python nachzuforschen, wie die Schnittstelle mit MariaDB verbunden wird. Da es von vielen Linux-Benutzern empfohlen wurde, Python als beste Wahl für die Verbindung zu MariaDB zu wählen, wurde der Programmierer aufgefordert, den Code mithilfe von Methoden und try and except-Anweisungen zu schreiben, um einige saubere und perfekte Skripte zu erstellen. Auf diese Weise wäre der Code sehr einfach zu analysieren und zu verbessern, aber am wichtigsten wäre es, Fehlermeldungen zu beseitigen.

Beschreibung der Tabellen

Nach der anfangs Planung wurden drei Tabellen definiert, die folgenden Tabellen:

- Person
- Info
- Log

Tabelle Person

Die erste Tabelle, Person, enthält alle User mit ihren grundlegenden Informationen, wie Vorname, Nachname, Rolle, Email. Die Spalte Rolle ist damit geeignet, damit ein Unterschied zwischen Personen geben sollte, weil es zwei Gruppen, die Admins und des normalen Users. Die Admin bekommen die Zahl 1 bei der Rolle, wahrend die anderen Users 0. Bei den Spalten, wo der Datentyp varchar verwendet wurde, gibt es ein Grund wiese varchar und nicht char. Das ist so denn char den ganzen Platzt reserviert, wo anderseits der Datentyp varchar nur die Lange des gespeicherten Strings reserviert, obwohl die Spalte ein varchar mit der Lange 50 sein kann.

Die Spalte idP ist eine eindeutige Spalte, die es zu der Identifizierung von der Datenzeile gilt, deshalb wird auch als Primar Schlüssel definiert. Es wird auch auto increment festgelegt, damit die nächste Zeile automatisch die nächste Zahl bekommt.

Die Spalte Nachname enthält der Nachname der Person, deshalb wird als Datentyp Varchar mit der Lange 50 gespeichert, weil Varchar zum Speichern von Strings geeignet ist. bekommt.

Die Spalte Email enthält der Email der Person, deshalb wird als Datentyp Varchar mit der Lange 50 gespeichert, weil Varchar zum Speichern von Strings geeignet ist. bekommt.

Die Spalte Rolle enthält der Rolle der Person, deshalb wird als Datentyp Char mit der Lange 1 gespeichert, weil Char zum Speichern von Strings geeignet ist. Um eine einfache Nummer zu speichern, ist hier auch den Datentyp int verwendbar.

Tabelle Info

Die zweite Tabelle enthält die Bilder der Personen, die wahrend der Zeit gemacht wurden, und auch die Grund Information der Bilder, wie Punkten und der Path von dem Bild. Die Spalten der folgenden Tabellen sind:

Die Spalte idF ist eine eindeutige Spalte, die es zu der Identifizierung von der Datenzeile gilt, deshalb wird auch als Primar Schlüssel definiert. Es wird auch auto increment festgelegt, damit die nächste Zeile automatisch die nächste Zahl bekommt.

Die Spalte idP ein fremder Schlüssel, der als Primar Schlüssel für die Person Tabelle geeignet ist. Diese fremden Schlüssel sind dazu, damit eine Verbindung zwischen diese zwei Tabellen geben kann. Diese zwei Tabelle haben eine 1 zu mehreren Beziehungen, wo eine Person mehrere Bilder haben kann. Das wurde so geplant, damit je mehr Bilder eine Person hat, desto sicher, besser und funktionierbar der Vergleich sein kann.

Jeder Punkt besteht aus eine X und Y Koordinate, und von jedem Bild gibt es einen Startpunkt, wo die anderen Punkten relativ zu diesem Punkt gespeichert werden. Für diese Punkte wird den Datentyp dezimal werden, weil die Koordinaten kommastellen Zahlen sind. Nur für den ersten Punkt werden genau die Koordinaten gespeichert, für die anderen zeigt die X- und Y- Koordinate einen Vektor, das heißt es wird der Distanz von diesem Startpunkt aufgenommen.

Die Spalte imagePath speichert die Path der Bilder und wird mit dem Datentyp varchar mit einer Länge von 50 gespeichert, weil Varchar zum Speichern von Strings geeignet ist. Die Bilder wurden als Path und nicht als ganze Image gespeichert, weil ob diese Bilder in der Datenbank gespeichert werden, wurde die Datenbank überfüllt mit Daten. Alternativ was medium blob aber dieser Datentyp brauch zu viel Speicherplatz von unserer Datenbank. Es konnte eine Overflow von der Datenbank geben.

Log Tabelle

Die dritte Tabelle ist nichts anders als eine Tabelle, wo alle Änderungen in der Datenbank protokoliert werden. In diesen Tabellen sollten die Grund Informationen gespeichert werden müssen, wie z.B wer hat was, wann gemacht. Diese frage muss die Tabelle beantwortet.

Die Spalte idL ist eine eindeutige Spalte, die es zu der Identifizierung von der Datenzeile gilt, deshalb wird auch als Primar Schlüssel definiert. Es wird auch auto increment festgelegt, damit die nächste Zeile automatisch die nächste Zahl bekommt.

Die Spaltet User soll der Person speichert, wer die Änderungen durchgeführt hat. Hier wird der Datenbank User gespeichert, weil um etwas zu verändern soll die Person zuerst in der Datenbank einloggen. Die Spalte hat den Datentyp Varchar mit der Lange 50, weil die Funktion Current User, da eingetragen wird.

Die Spalte Date als der Name zeigt, kümmert um die Speicherung von der Zeit, wann eine SQL Statement ausgeführt wird. Die Spalte hat den Datentyp datetime, und da wird die Funktion now(), die die aktuelle Zeit darstellt, gespeichert.

Die Spalte Info, informiert den User, was für eine Änderung gemacht wurde, z.B neue User wurde eingelegt. Diese Spalte bekommet den Datentyp Varchar, weil Varchar zum Speichern von Strings geeignet ist. Die Werten in dieser Spalte werden durch Triggers definiert, also wenn etwas in der Datenbank passiert, kümmern die Triggers um die eintragen von Daten.

4.3 Herausforderungen

Jedes Projekt hat seine Herausforderungen, und wenn etwas schief geht sollten Alternativen gefunden werden. Auch in diesem Projekt gab es solche Probleme, die durch Hilfe von Tutorials und Internet Blogs gelöst werden.

4.3.1 Datenkbank - MariaDB

Die meisten Problemen sind bei der Datenbank getroffen, das ist so passiert wegen der kleinen Unterschiede zwischen MySQL und MariaDB. Manchmal wurde mit dem Progammierregeln von MySQL, obwohl MariaDB in Verwendung war. Zwischen diese zwei Datenbankverwaltungssystem gibt es kleine Unterschiede, die zu Problemen fuhren. Den getroffenen Fehlern werden hinunter besprochen:

ERROR 1452: Cannot add or update a child row: a foreign key constraint fails. Das Problem hier war es, dass es keine Tabelle geändert werden konnte, wegen die Beziehungen von Tabellen. In Prinzip sollte zuerst das Kind Tabelle gelöscht werden, und dann die Eltern Tabelle, weil sonst wird eine Konflikt geben, wozu bei der Kind Tabelle einen Fremd Schlüssel definiert ist, der jetzt nicht mehr existiert.

You have an error in your SQL syntax; check the manual that corresponds to your mariadb server version for the right syntax to use near .. At line 1
Dieser Fehler wurde getroffen, wenn einen Trigger Erstellung Script ausgeführt wurde. Der Fehler hat hier beim Delimiter gestanden, weil wenn ein Trigger durch ein Script erstellet wird, braucht man keine Delimiter einsetzen, sondern nur wenn es manuell durch der Command Line sollte der Delimiter festgelegt. Hier kann man sehen, die Änderungen zwischen MariaDB und MySQL, in MySQL sollte dieser Code ohne Probleme ausgeführt werden.

Mysql-Server is missing// Dieser Fehler zeigte, dass keine MySQL Server auf dem Betriebssystem installiert war. Dieses Problem ist getroffen, wegen die Anderungen in Linux System. MySQL Server wurde nicht mehr supportiert, und jetzt war im Betrieb MariaDB, und statt mysql-server zu installieren, sollte mariadb-server durch den Befehl *sudo apt-get install mariadb-server* installiert.

ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/run/mysqld/mysqld.sock' (2 "No such file or directory")
Dass war nur ein kleiner Fehler, weil wenn man in Kali Linux programmiert und gearbeitet hat, musste den MariaDB Server immer noch einmal durch den Befehl *service mariadb restart* gestartet werden. Im Vergleich zu Debian Betriebssystem, die auf RaspberryPi installiert war, musst den Server nicht gestartet werden, weil es automatisch im Betrieb war.

4.3.2 Python Scripting

Python ist beliebige Programmiersprachen von vielen Users, aber Python unterschiedet sich von anderen Programmiersprachen durch den Code Struktur. Die meisten Fehlern

wurde bei der Ubetragung von den Parametern zwischen den verschieden Funktionen.

TypeError: not all arguments converted during string formatting python
Der folgende Code nimmst als Parameter die Daten von dem neuen User, damit der User in der Datenbank registriert wird. Aber wenn man nur die Methode zu probieren mochte, und gibt als Parameter nur der Name, als eine einfache Registrierung, dann wurde dieser Fehler getroffen.

insertvalues1(curs,'jordi','zmiani','jorzmi14@htl-shkoder.com',1)

Bei der ersten Möglichkeit sollten zwei Parameter gegeben, denn nur bei einem String Parameter wurde dieser Fehler generiert. Bein der zweiten Möglichkeit musst diese Parameter geparst werden durch die Funktion format().

TypeError: 'str' object is not callable (Python)
Str in Python ist eine spezielle Methode, und diese Methode sollte nicht überschrieben werden, das heißt, dass es sollte keine Variable bzw. Funktion mit diesem Namen erstellet werden. Aber trotzdem wurde dieser Fehler generiert, weil wenn ein Parameter auf den falschen Datentyp geparst wird, dann wird dieses Problem dargestellt.

4.3.3 OpenCV

OpenCV Biblotheken konnen manchmal schwierig zu installiert werden, weil es so viele Moglichkeit zum Einrichten von OpenCV gibt, die aber von dem Betriebsystem abhangen. Am meisten Fallen wenn die Installation Fehler produziert, musste OpenCV von Beginn noch einmal eingerichtet werden.

Unable to locate package libjasper-dev
Diese Bibliotheken ermöglich ein paar Funktion von den OpenCV, das ist eine von vielen Bibliotheken, die eingerichtet werden sollten. In diesem Fall konnte diese Bibliothek nicht installiert werden, weil es die Repository geändert hat, und deshalb musste die Bibliotheken von einer älteren Version heruntergeladen werden. Mit den folgenden Befehlen wurden diese Bibliotheken installiert.

echo "deb http://us.archive.ubuntu.com/ubuntu/ yakkety universe sudo tee -a /etc/apt/sources.list
sudo apt update
sudo apt install libjasper1 libjasper-dev

4.4 Qualitätssicherung

Einer der wohl bekanntesten Berater/-innen, Lehrer/-innen und Autoren/Autorinnen (über 200 Veröffentlichungen) zum Thema Qualität ist der Amerikaner W. Edwards Deming. Deming entwickelte auf der Basis der allgemeinen Problemlösungsmethode den sogenannten Plan-Do-CheckAct-Zyklus (PDCA-Zyklus). Auf der Plan-Phase werden

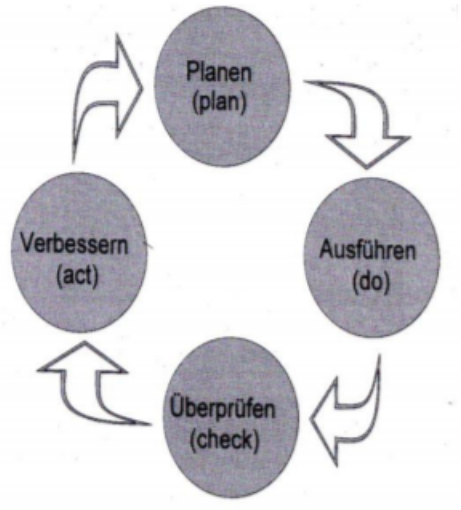


Abbildung 4.5: Plan-Do-Check-Act-Zyklus
[1]

Probleme betrachtet und Lösungsmaßnahmen erarbeitet. Die Do-Phase ist die Phase der Umsetzung bzw. Ausführung der zuvor gefundenen Lösungen. In der Check-Phase wird bewertet, ob die Maßnahmen zum Erfolg geführt haben. Innerhalb der nachfolgenden Act-Phase findet eine Standardisierung der erfolgreichen Maßnahmen statt, die fortan als Basis für weitere Verbesserungen dient.[1]

Auf dem Bild 4.5 wird die PDCA Zyklus dargestellte, und es ist ganz sichtbar dass die Qualitätsmethode sehr gut für die Verbesserung von Software und Programmen ist. Deshalb wurde auch diese Methode gewählt, weil durch Error und Checking konnten die Fehler gefunden, analysiert und verbessert. Das passt wirklich gut beim Team, wenn sie keine Erfahrungen mit bestimmenden Programmiersprachen bzw. Softwareprogrammen haben.

Oben bei den Herausforderungen war klar, dass wenn Probleme getroffen wurden, wurden sie durch Hilfe von Tutorials bzw. Net Blogs gelost. Durch verschiedene Zyklen beachtet man auch bei den Risiken, weil wenn etwas nicht funktionier, wird schnell das Problem gefunden und später wird eine andere Losungswegen bestimmt.

Kapitel 5

Ergebnisse - Jordi

5.1 Datenbank

Für den Bereich Datenbank wurde ordentlich jede Woche programmiert, wo seit dem Anfang wurde es Recherche über verschiedenen Arten von Gesichtserkennung Datenbank, und später hat die Planung und Design von Datenbankentwurf begonnen. Es gab auch viele Änderungen abhängig von anderen Teilbereichen des Projektes. Die Datenbank musst immer geändert werden, und nach den Gesichtsdaten anpassen, damit alles in Ordnung ist. Die folgenden Punkten der Datenbank wurden realisiert:

- Datenbankdesign
- Erstellen von Tabellen
- Log Table
- Triggers
- Zugriffsrechte

Die Datenbank sollte fast fertig sein, nur kleine Anderungen konnten durchgefuhrt werden, wie fur die Backup von Daten.

5.2 Tieferkennung

Für den anderen Bereich Tieferkennung wurde gearbeitet aber nicht mit einem hohen Niveau wegen der Probleme beim Gesicht Erkennung, weil Tieferkennung ein letztes Ziel von dem Projekt ist. Hier wurde natürlich Recherche gemacht, und ein kleines stuck Programm geschrieben, nur zum Ausprobieren von Bibliothek und OpenCV.

Kapitel 6

Bildverarbeitung

6.1 Allgemeines

Diese Diplomarbeit besteht aus vielen unterschiedlichen Modulen, die um bestimmte Ziele bzw. Aufgaben zu lösen gut aufgeteilt sind.
Sehr wichtiger Teil dieses Projektes ist der Bildverarbeitungsteil.
In dem folgenden Abschnitt der Ausarbeitung wird es genau erklärt und beschrieben wie diese Aufgabe gelöst wurde und was dafür verwendet wurde.
Für die Umsetzung wurden folgende Technologien gebraucht.

6.1.1 Entwicklungsumgebung und Technologien

Die gewählte Entwicklungsumgebung war grundsätzlich eine virtuelle Maschine die Linux auf einem Windows System geboten hat.
Linux im Gegensatz von Windows ermöglicht volle Kontrolle über Updates und Upgrades und dadurch könnten komplexe Aufgaben einfacher erlediget werden.
Alles wird leicht und bequem durch Konsole eingegeben.
So wurden die Entwicklung, das Testen von den genutzten Algorithmen und anderen Technologien vielmals erleichtert. [17]

Zusätzlich ist Raspberry Pi als Backup System verwendet worden das auch mit einem lauffähigen Linux Betriebssystem (Debian) funktioniert.
Raspberry Pi ist klein, funktioniert aber wie ein normaler Rechner und ist kostengünstig.
Für technische/elektronische Projekte wie das Betreffende, ist es perfekt geeignet.
Für die Implementierung des Codes wurde hauptsächlich die Programmiersprache Python verwendet.
Python ist eine allgemeine Programmiersprache auf hohem Niveau. Dies bedeutet dass es näher an menschlichen Sprachen ist.
Also ist ein in Python geschriebener Code sehr leicht von einem Mensch zu lesen, zu verwalten und zu warten.

Es bietet zahlreiche Modulen durch seine große und robuste Standardbibliothek an, von denen man ruhig, abhängig vom Bedarf, auswählen kann. Sehr leicht kann es in der Dokumentation der Python-Standardbibliothek nachgesehen werden um sich besser mit den gezielten Funktionalitäten auszukennen.
[22]

"Git ist ein freies und Open Source verteiltes Versionskontrollsystem, das entwickelt wurde, um alles von kleinen bis zu sehr großen Projekten mit Geschwindigkeit und Effizienz abzuwickeln."[7]
Die Versionierung der ganzen Software Änderungen erfolgte durch Git.
Es hat sich nützlich erweist indem die Arbeit dadurch zwischen die Projektmitgliedern sehr gut koordiniert und verwaltet wurde.

6.1.2 Frameworks und Bibliotheken

Schlüsselwort von unserem Projekt war das Framework bzw. die Bibliothek „OpenCV".
OpenCV ist eine open-source Bibliothek für Computer Vision. Also, ganz allgemein kann sie als eine Bibliothek für Bildverarbeitung betrachtet werden.
Sie kümmert sich unter anderem um die Manipulation von Bildern, die Analyse von denen und um die daraus bestimmte Muster bzw. Objekte, für verschiedenen Zwecken einzusetzen. Ganz berühmte Anwendungsgebieten sind Gesichtserkennung und Stereo Vision.
Stereo Vision bedeutet die Extrahierung von Informationen aus einem Bild in eine 3-dimensionalen Ebene.
OpenCV wurde unter anderen Bibliotheken aufgrund seiner vielen guten Eigenschaften und seiner Flexibilität ausgewählt. Es umfasst hunderte von Computer-Vision-Algorithmen und besteht aus strukturierten Einheiten bzw. Module die als feststehende Bibliotheken implementiert sind.
Es ist Cross-Platform, in C/C++ geschrieben und unterstützt auch Python.
[14]

SciPy hat sich nützlich, beim Lösen von gewisse mathematische Angabenund bei zusätzliche Installationen von Bibliotheken zu helfen.
SSciPy ist eine kostenlose und Open-Source-Python-Bibliothek für wissenschaftliches und technisches Rechnen."[27]
Dies sind einige der Kernpakete von SciPy die nutzbar für das Projekt waren:

NumPy legt die Basis für das wissenschaftliche Rechnen mit Python.
Es unterstützt große mehrdimensionale Arrays und Matrizen. Es enthält viele Mathematische Funktionen auf hoher Ebene um die Arrays zu bearbeiten.
NumPy kann also als leistungsfähiger mehrdimensionaler Behälter für generische Daten verwendet werden.
Es können beliebige Datentypen definiert werden.

Dlib ist ein so genanntes Toolkit, das Algorithmen für Machine Learning und Tools

Abbildung 6.1: dlib logo [19]

zum Erstellen komplexer Software zur Lösung von der realen Welt getauchte Probleme, beinhaltet.
Es wird vielseitig eingesetzt, in Robotik, Mobilgeräte und unter anderem in Computer Vision.[19]
Es wurde prinzipiell bei der Extrahierung der so genannten „Facial Landmarks“ gebraucht.

6.2 Technische Lösungen

Bei der Gesichtsregistrierung und Gesichtserkennung Diplomarbeit war es die schwierigste und herausforderndste Aufgabe, sie sorgfältig zu planen, um die Effektivität bei der Arbeit zu steigen und das Endprodukt zufriedenstellend zu machen.

Wichtig war es die Ziele richtig zu setzen und sie gut abzugrenzen damit es zu keine Lücken bei der Implementierung kommt.

Aus diesem Grund musste die Arbeitsteilung gut geregelt werden, damit jedes Teammitglied sich auf einen bestimmten Modul der Arbeit konzentrieren konnte.
Es wurde auch das berücksichtigt, dass jedes Teammitglied das machte was ihm/ihr am besten gefällt und was ihm/ihr am leichtesten fällt.

Die Planung der betreffenden Bereiche der Projektarbeit hat durch die Methode „Structed Design“ erfolgt.
Structed Design ist eine Erweiterung von der „Big Picture“ Methode, die dazu dient, ein technisches System mit ihren Schnittstellen mit außen grob zu beschreiben.
Es ist in verschieden Ebenen unterteilt, von außen beginnend.

Unten wird die erste Ebene der Structed Design von der Bildverarbeitungsteil dargestellt.

6.2.1 Lösungsweg - Structed Software design

Wie es in der Abb.6.2 sichtbar ist, ist die Bildverarbeitungsteil in diese Einheiten unterteilt:

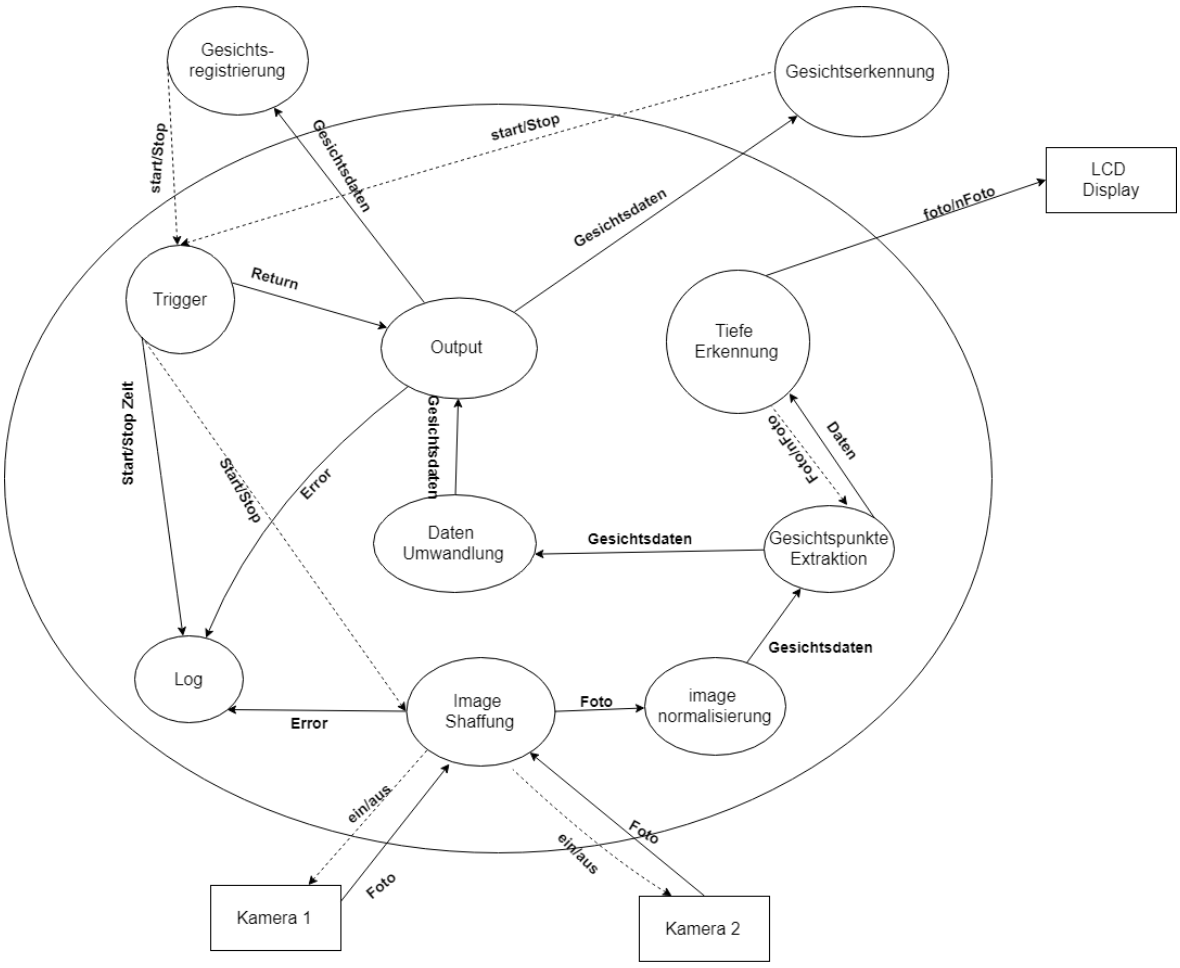


Abbildung 6.2: Bildverarbeitung Structed Design

1. Bildaufnahme
2. Image Normalisierung
3. Gesichtschlüsselpunkten Extrahierung
4. Log
5. Output
6. Trigger

Die Schnittstellen von außen sind das Gesichtserkennungsmodul und die Gesichtsregistrierungsmodul.
Diese schicken eine Anforderung an den Trigger der dann die Bildaufnahme Modul initialisiert. Unabhängig von welchen von diesen beiden Schnittstellen die Anfrage kommt, macht das Bildaufnahme Modul aus der zwei Kameras ein Foto.
Implementierungsnah wurde bis jetzt nur eine Kamera verwendet, da die Stereo Vision noch nicht funktional ist.
Nachdem das Foto gemacht wird folgt die Image Normalisierung bzw. das „Image Prozessing".
Wenn ein Bild zu klein ist wird die Größe angepasst, wenn ein Gesicht verdreht ist wird es gerade rotiert. Mehr dazu wird in dem Normalisierung Abschnitt erklärt.
Danach folgt die Erkennung von Gesichtern, das Ausschneiden von denen und die Extrahierung der Gesischtsschlüsselpunkte.
Sie werden dann zur Abgleich oder Registrierung je nach Bedarf, bereitet und geschickt.

Hinweis: Bei der Umsetzung wurden bestimmte Bereiche mit einander verknüpft, was zu Folgendes führte: was bei der Planung steht, stimmt nicht vollig mit der Methoden zur Umsetzung überein. Weitere Unterschiede werden im Punkt 13b. genauer beschrieben.

6.2.2 Gesichtsdetektion

Das erste Ereignis, dass erfüllt worden war bei der Gesichtserkennung ist das Finden von Gesichtern, also die Feststellung, wo sich die Gesichter im Bild befinden.
Dafür ist der „Haar Cascade Classifiers" vorgefertigter Klassifikator benutzt worden.

Gesichtsdetektion durch „Haar" Merkmale ist eine sehr effektive Methode die von Paul Viola und Michael Jones entwickelt wurde, indem sie Merkmale gruppiert haben und die Erkennung von diesen dadurch schneller gemacht haben.

Die Arbeit heißt „Rapid Object Detection using a Boosted Cascade of Simple Features". [26]

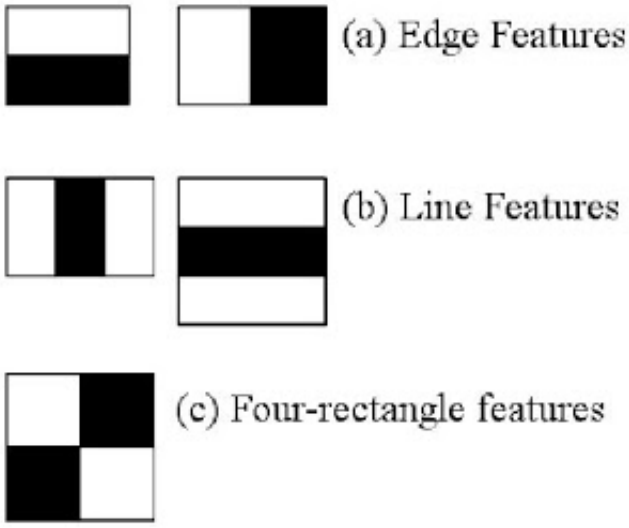


Abbildung 6.3: Haar Features[26]

Unten wird eine kurze technische Übersicht über diese Klassifikator gegeben.

Es geht hier um Maschinelles Lernen, wie diese Kaskadenfunktionen durch viele negative(Bilder ohne Gesichter) und positive(Bilder mit Gesichter) Bilder trainiert wurden um Objekte zu erkennen.
In diesem Fall arbeiten wir selbstverständlich mit Gesicht Objekten. Dafür wurden die Haar Merkmale benutzt. Auf Abbildung 6.3 sind sie dargestellt. Jedes Haar Merkmal ist nichts anders als ein Wert, durch das Subtrahieren der Summe der Pixel unter dem weißen Rechteck von der Summe der Pixel unter dem schwarzen Rechteck, erhält.
Diese Summen berechnet man durch integrale Bilder, die zur Vereinfachung zu Summenberechnungen dient.

Man muss aufpassen, da nicht alle Merkmale von Nutzen sein könnten. Man kann es zum Beispiel deutlich auf Abbildung **??** sehen, wie die Nase viel heller als die Region bei den Augen ist. Die Selektion von den besten Merkmalen wird durch das Adaboost Algorithmus berechnet.

Mit dieser Absicht setzen wir alle Funktionen auf alle Trainingsbilder ein. Für jedes Merkmal wird der beste Schwellenwert ermittelt, der die Gesichter in positive und negative klassifiziert.

Umsetzung in Code

Unten ist ein Code Abschnitt der Gesichtsdetektion dargestellt.

Listing 6.1: Kern Code für Gesichtsdetektion

```
face_cascade =
        cv2.CascadeClassifier('haarcascade_frontalface_default.xml')

image = cv2.imread('gesicht.jpg')

gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)

faces = face_cascade.detectMultiScale(gray,1.1,4)

for(x, y, w, h) in faces:

cv2.rectangle(image, (x, y), (x+w, y+h), (255,0,0), 2)
```

Es wurde die trainierte Klassifikator-XML-Datei (haarcascade_frontalface_default.xml) die im GitHub-Repository von OpenCV zu befinden ist, herunterladen.
In den Zeilen **1** und **2** wurde sie geladen und in die Variable face_cascade gespeichert.
In die Zeile **4** wird das Bild vom den Funktion *imread* gelesen.
Die Erkennung funktioniert nur bei Graustufenbildern. Daher wurde das Farbbild in Graustufen umgewandelt, wie in der Zeile **6** ersichtlich.
Der Funktion *detectMultiScale* in der Zeile bf erkennt die Gesichter im Bild. Dieser Funktion ist sehr wichtig und braucht 3 Argumente – das Eingabebild, der Skalierungsfaktor (scaleFactor) und *minNeighbours*.

scaleFactor gibt die an, um wie viel wird das Bild vergrößert bzw. verkleinert. Und *minNeighbours* gibt an, wie viele Nachbarn jedes Kandidatenrechteck haben muss, um es beizubehalten. Dieser Parameter bestimmt die Qualität der erkannten Gesichter: Ein höherer Wert führt zu weniger Erkennungen, jedoch zu einer höheren Qualität. Es ist bei uns nicht so entscheidend viele Gesichter zu erkennen, sondern die Qualität, ist hier der Wert 4 eingesetzt.

In Zeile 10 beginnend, wird es durch die Gesichter *faces* iteriert. x und y entsprechen die Koordinaten vom Bild, w, h bezeichnen die Breite und Höhe *width*, *height*.

Schließlich wird in Zeile **12** mit dem Funktion *cv2.rectangle* ein Rahmen auf dem Gesicht gezeichnet.
Der Funktion bekommt diese Parameter: *image* für die Bildeingabe, x, y ergeben den Startpunkt, w, h den Endpunkt, die Farbe und die Dicke der Rahmen werden in den 2 letzten Parameter gegeben.

Dieser Stammcode wurde so umgeändert, dass es in der Lage ist, durch mehrere *Predictors* bzw. Klassifikatoren, die Genauigkeit der Gesichtsdetektion zu erhöhen.

Auf Abbildung 6.4 sieht man deutlich das erkannte Gesicht.

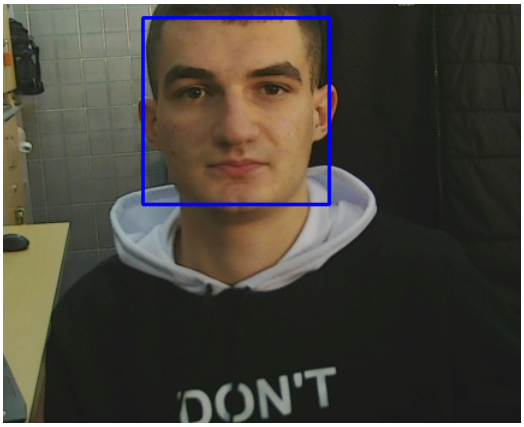


Abbildung 6.4: Output: Box auf Gesicht

6.2.3 Normalisierung

In dieser Kapitel wird ein sehr wichtiger Bereich, der sogenannte Normalisierung der Daten, genau erklärt.

Was ist Normalisierung?

Normalisierung in *Computer Vision* ist der Prozess der Bereitstellung von Daten, die Korrektur der "falschen"Daten, die die Genauigkeit unseres Systems beschädigen können.
Normalisierung heißt anders auch Ausrichtung. Die Ausrichtung der Gesichter, die im Bild "falsch"positioniert sind.
Also entspricht Normalisierung den Prozess in dem man zuerst die geometrische Struktur von Gesichtern in digitalen Bildern identifiziert, das Versuch, eine maßgebliche Ausrichtung des Gesichts basierend auf Skalierung und Rotation zu erhalten.

Methoden für Normalisierung

Es gibt viele Methoden mit dem man Bilder normalisieren könnte. Einige von denen basieren auf irgendwelche pre-definierte 3D-Modelle und transformieren dadurch die Eingabe Bilder, dass die Gesichtsschlüsselpunkte der Eingabe Bild die von dem 3D Modell übereinstimmen.
Die in dieser Diplomarbeit verwendete Methode, eine eher einfachere, verlässt sich nur auf die Gesichtspunkte selbst, um eine normalisierte Rotation und Skalen Darstellung des Gesichts durch Affine und Ähnlichkeit Transformation zu erhalten. Diese Methode wurde deswegen implementiert, weil es sehr effizient ist und genau gut zu dieser Diplomarbeit passt.

Warum Normalisierung?

Der Grund warum in dieser Arbeit Normalisierung von Daten durchgeführt wird, liegt genau in der Tatsache, dass viele Gesichtserkennungsalgorithmen einschließlich unserer,

von der Anwendung von dieser Ausrichtung viel profitieren können. Es wird dadurch der Präzision der Gesichtserkennung erhöht.

Implementation

Die hier verwendete Methode für die Normalisierung, wie oben kurz erwähnt wurde, wendet Ähnlichkeitstransformation bei zwei Paaren entsprechender Punkte ein. Die Punkten sind die von Dlib extrahierten Gesichtsmerkmalen. Siehe Kapitel 6.2.5.

OpenCV benötigt aber 3 Punkte zur Berechnung der Ähnlichkeitsmatrix. Wir nehmen somit als dritten Punkt, der dritte Punkt des gleichseitigen Dreiecks mit diesen beiden gegebenen Punkten an.

Was eine Ähnlichkeitstransformation lässt die Mathematik Theoremen beschreiben. Eine Ähnlichkeitstransformation ist eine der mehreren starre Transformationen wie z.b.: Reflexion, Rotation oder Translation gefolgt von einer Dilatation.

Wenn eine Figur durch eine Ähnlichkeitstransformation transformiert wird, wird ein Bild erstellt, das der ursprünglichen Figur ähnlich ist. Mit anderen Worten, zwei Figuren sind ähnlich, wenn eine Ähnlichkeitstransformation die erste Figur zur zweiten Figur trägt.

Solches Prinzip ist auch in unsere Methode eingesetzt. Es wird eine zweite, dem ersten Bild ähnlichen Figur, erstellt.

Listing 6.2: Implementation Normalisierung

```
def similarityTransformMat(initialPoints, destinationPoints):
        ...
        tform = cv2.estimateAffinePartial2D(np.array([initialPoints]), np
        return tform[0]
```

Im Code Abschnitt 6.2 wird die Methode gezeigt, die als Parameter die Anfangs- und die Zielpunkte nimmt und aus den beiden jeweils einen dritten Punkt berechnet.

Sie werden dann in Arrays gespeichert und aus denen wird doch die Ähnlichkeitstransformation von den in OpenCV eingebetteter Funktion *cv2.estimateAffinePartial2D* berechnet.

Ausgabe ist eine 2D affine Transformation, also eine 2x3 Matrix oder leere Matrix, wenn die Transformation nicht geschätzt werden konnte.
"Die Funktion schätzt eine optimale affine 2D-Transformation mit 4 Freiheitsgraden, die auf Kombinationen aus Translation, Rotation und gleichmäßiger Skalierung beschränkt sind. Verwendet den ausgewählten Algorithmus für eine robuste Schätzung."[14]

Die nächste Funktion, genau die wichtigste, macht die Ausrichtung der Gesicht im Bild. Es bekommt ein, von Funktion *cv2.imread* gelesenes Bild, die gewünschte Größe, und die Dlib Gesichtsmerkmale als Parameter.

Listing 6.3: Gesichtsausrichtung

```
def faceAlign(image, size, faceLandmarks):
        (h, w) = size
        initialPoints = []
        destinationPoints = []

        initialPoints = [faceLandmarks[36], faceLandmarks[45]]

        destinationPoints = [(np.int(0.3*w), np.int(h/3)), (np.int(0.7*w)

        similarityTransform = similarityTransformMat(initialPoints, desti

        faceAligned = np.zeros((image.shape), dtype=image.dtype)

        faceAligned = cv2.warpAffine(image, similarityTransform, (w, h))

        return faceAligned
```

Im Code Abschnitt 6.3 in Zeile **6** wird die Position der linken Ecke des linken Auges und der rechten Ecke des rechten Auges von den Eingabebild genommen. In Zeile **8** wird die Position der linken Ecke des linken Auges und der rechten Ecke des rechten Auges im ausgerichtete Bild berechnet.

In Zeile **10** und **11** wird eben die Ähnlichkeitstransformation durch den vorher erstellten Funktion *similarityTransformMat* berechnet.

In Zeile **12** wird das ausgerichtete Gesicht in einem Tupel gespeichert.

Schließlich wird die Ähnlichkeitstransformation durch die Methode *cv2.warpAffine* angewendet.
Diese Methode bekommt den Tupel vom Bild, *similarityTransform*, und die Größen von Bild als Parameter. Das wird dann vom Funktion zurückgegeben. (Zeile **16**)

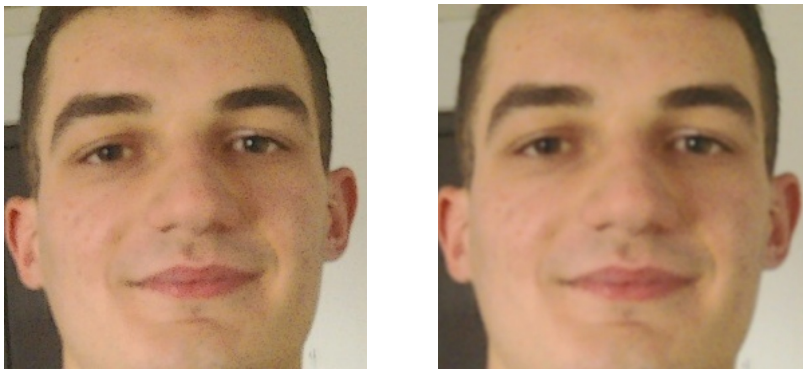
Eine affine Transformation ist jede Transformation, die die Kollinearität (d. H. Alle auf einer Linie liegenden Punkte, die anfänglich nach der Transformation noch auf einer Linie liegen) und die Entfernungsverhältnisse (z. B. der Mittelpunkt eines Liniensegments bleibt der Mittelpunkt nach der Transformation) bewahrt.
Eine affine Transformation wird auch als Affinität bezeichnet.[**affine**]

Zusätzlich wurde noch eine Glättungsfilter angewendet. Dies geschieht durch die Funktion *cv.2filter2d* die als Parameter ein Bild bekommt und der die Daten zur Filter. Kernel ist auf Code abschnitt 6.4 genau sichtbar.

Listing 6.4: Glättungsfilter

```
kernel = np.ones((5,5),np.float32)/25

output = cv2.filter2d(output,-1, kernel)
```



(a) Nicht normalisiert (b) Normalisiert

Abbildung 6.5: Unterschied: normalisiert, nicht normalisiert

(a) Nicht normalisiert (b) Normalisiert

Abbildung 6.6: Unterschied: normalisiert, nicht normalisiert .2

Auf Abbildungen 6.6 und 6.5 sieht man ganz klar die Unterschiede die die Einsetzung der Normalisierung in Bilder verursacht.

6.2.4 Zuschneiden von Gesichter

Nachdem Gesichter gefunden werden, muss man daraus eigene Bilder machen, die das Extrahieren der Schlüsselpunkte erleichtern.

Listing 6.5: Code Abschnitt: Gesicht Zuschneiden

```
if nrFace > 0:
        for face in faces:
                for(x, y, w, h) in faces:
                        r = max(w, h) /2
                        centerx = x + w /2
                        centery = y + h /2
```

```
                    nx = int(centerx - r)
                    ny = int(centery - r)
                    nr = int(r * 2)
                faceimg = image[ny:ny+nr, nx:nx+nr]
                filenam = input("Give new filename for cropped photo: \n"
                image2 = cv2.imwrite(filenam,faceimg)
elif nrFace <= 0:
        print("no faces found")
```

nrFace bestimmt die Anzahl der Gesichter, die von den Kaskadenklassifikator gefunden wurden. Nur wenn dieser Wert größer als 0 ist soll der Programm weiterlaufen. Es wird durch jedes Gesicht iteriert.

Zeilen **1-6** berechnen das Zentrum von Bild neu und in Zeile **7** wird ein neues Image mit den berechneten Parametern angelegt.
Mit imwrite wird das Image gespeichert (Zeile **12**). Auf Abb.6.7 sieht man deutlich nur das geschnittene Gesicht.

Abbildung 6.7: Output: Abgeschnittenes Gesicht

6.2.5 Gesichtsschlüsselpunkte Extraktion

Nun kommt ist es zu dem wichtigsten Punkt meines Teils der Diplomarbeit, die Gesichtsschlüsselpunkte Extraktion.

Dazu wurden die Gesichtsmerkmale *„Facial Landmarks“* von dlib verwendet.
Die Gesichtsmarkierungen werden verwendet, um Bereiche des Gesichts zu finden und darzustellen, wie zum Beispiel: die Augen, die Augenbrauen, die Nase, den Mund und den Kiefer.

Wie macht man das? Wie erkennt man Gesichtsmerkmale?
Das ist eine Teilmenge des Problems der Formvorhersage.
Bei einem vorgegebenen Eingabebild (und normalerweise einer ROI[8], die das interessierende Objekt angibt) versucht ein Formvorhersager, wichtige Punkte entlang der Form zu lokalisieren.
Dieser Formvorhersager die im Dlib integriert ist wurde von Kazemi and Sullivan in

[8]Region of interest, Region von Interesse

ihrem Paper: *One Millisecond Face Alignment with an Ensemble of Regression Trees* entwickelt.

Dieser Methode werden viele Trainingsdaten hinzugefügt womit es ein Kombination von Regressionskurven trainiert wird um die Positionen der *Facial Landmarks* direkt aus den Pixelintensitäten selbst zu beurteilen.[18]

Dadurch werden unsere gewünschten 68 Gesichtsmarkierungen mit hohen Vorhersagbarkeit extrahiert und als x, y Koordinaten weitergegeben. Zusätzlich ist es auch so gemacht worden, dass diese gefundene Koordinaten den Dimensionen vom Bild nicht abhängig sind. Die Indizes der 68 Koordinaten sind in der Abb.6.8 dargestellt:

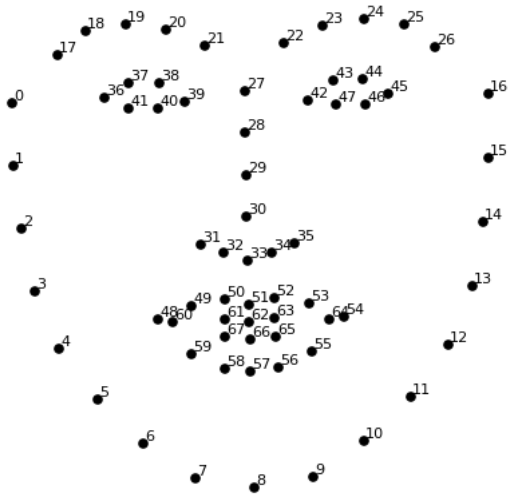


Abbildung 6.8: Gesichtsschlüsselpunkte [18]

Nachdem wir der „Predictor“ geladen haben speichern wir sie in einem Formobjekt mit 68(x,y) Koordinaten.
Es wird hier nicht die resize Funktion verwendet, weil es an Qualität verliert und es rechenintensiver ist, je größer das Eingabebild wird.
Jede Koordinate läuft in einer Schleife durch und entspricht dem spezifischen Gesichtsmerkmal im Bild.

0. 0.425 0.1359375
1. 0.42916666666666664 0.1625
2. 0.4375 0.190625
3. 0.44583333333333336 0.215625
4. 0.46041666666666664 0.2390625
5. 0.48333333333333334 0.259375
6. 0.5125 0.2734375
7. 0.5479166666666667 0.2859375
8. 0.5833333333333334 0.2875
9. 0.61875 0.2828125
10. 0.65 0.2703125
11. 0.675 0.253125
12. 0.6916666666666667 0.23125
13. 0.7 0.20625
14. 0.7041666666666667 0.1796875
15. 0.7104166666666667 0.153125
16. 0.7125 0.125
17. 0.44583333333333336 0.11875
18. 0.4625 0.103125
19. 0.4875 0.0984375
20. 0.5166666666666667 0.1
21. 0.5416666666666666 0.1078125
22. 0.5895833333333333 0.10625
23. 0.6166666666666667 0.0984375
24. 0.6458333333333334 0.09375
25. 0.6729166666666667 0.096875
26. 0.6916666666666667 0.1125
27. 0.56875 0.1265625
28. 0.5708333333333333 0.1453125
29. 0.5729166666666666 0.1640625
30. 0.5729166666666666 0.184375
31. 0.5479166666666667 0.196875
32. 0.5625 0.1984375
33. 0.5770833333333333 0.2
34. 0.5895833333333333 0.1984375
35. 0.6041666666666666 0.1953125
36. 0.47708333333333336 0.1328125
37. 0.49375 0.1265625
38. 0.5125 0.125
39. 0.53125 0.1328125
40. 0.5125 0.1375
41. 0.49375 0.1375
42. 0.6083333333333333 0.13125
43. 0.625 0.121875
44. 0.64375 0.121875
45. 0.6604166666666667 0.128125
46. 0.6458333333333334 0.1328125
47. 0.6270833333333333 0.134375
48. 0.525 0.2265625
49. 0.54375 0.2203125
50. 0.5625 0.2171875
51. 0.5770833333333333 0.21875
52. 0.5916666666666667 0.215625
53. 0.6104166666666667 0.21875
54. 0.6291666666666667 0.2234375
55. 0.6125 0.23125
56. 0.5958333333333333 0.2359375
57. 0.5791666666666667 0.2375
58. 0.5645833333333333 0.2359375
59. 0.5458333333333333 0.2328125
60. 0.5333333333333333 0.2265625
61. 0.5625 0.225
62. 0.5791666666666667 0.2265625
63. 0.59375 0.225
64. 0.6229166666666667 0.2234375
65. 0.59375 0.2234375
66. 0.5791666666666667 0.225
67. 0.5645833333333333 0.2234375

Abbildung 6.9: Gesichtsdaten

Die Merkmale werden in einer numpy Array gespeichert für den Zweck der Speicherung in Datenbank.

6.3 Herausforderungen, Probleme und deren Lösung

Die größte Herausforderung lag bei der Planung von der Software und die Methoden die zum Extrahieren von den Gesichtsschlüsselpunkten dienten.
Es hat mich viel Zeit gekostet bis ich eine geeignete Lösung gefunden habe und das hat viel Stress gemacht.
Eine weitere Herausforderung war das Verknüpfen von den Entwicklungsumgebungen und die Kooperation zwischen den Teammitgliedern.
Die verwendete Systeme waren all zu unterschiedlich und es konnte keine Standardisierung zwischen ihnen gefunden werden. Also ist viel Zeit beim Installieren und Konfigurieren investiert worden. Ein Grund dafür ist die mangelnde Erfahrung mit den neuen Technologien.

Viele Sachen, wie z.b.: die Einteilung der Arbeit, die genaue Spezifikationen, also prinzipiell die sehr grobe Planung, waren am Beginn auch wegen den Kommunikationslücken unklar.

Als das Projekt weiter entwickelte, hat die Frage nach Genauigkeit und Präzision auch enorm gestiegen. Es hat immer wieder Fälle gegeben in dem die fertig gestellten neuronale Netze nicht so gut funktioniert haben beim Finden von Gesichter.

Es war keine gute Idee, nur von einem Prädiktor *Predictor* sich abhängig zu machen. Herausforderung war es deutlich, die Einsetzung von mehreren neuronalen Netzen und die Erhöhung der Treffsicherheit.

Eine einfache Normalisierung genügte auch nicht (die Umwandlung der Bilder in Graustufe), sondern es war eine komplexere Lösung dafür viel notwendiger und brauchbar. Bis das fertig geschrieben wurde, bis es richtig gut funktionierte, hat es viel Zeit und Bemühung gekostet. Dafür waren auch relativ fortgeschrittene mathematische Kenntnisse verlangt. Dabei habe ich viel "Try und Errorëingewendet.

Es hat auch Schwierigkeiten bei der Verwaltung von Qualität gegeben. Die Erfahrung fehlte, somit waren gewisse Maßstäbe, Standarte auch nicht bekannt.

6.3.1 Lösungen

Während der Arbeit habe ich viel recherchiert und mich genau über alles Mögliche informiert.

Es wurde viel herumprobiert und experimentiert. Es wurde auch sehr viel getestet, damit die Ziele auch qualitativ erfüllt wurden.

Die Kommunikation hat sich mit Bedarf während der Zeit stark verbessert und dadurch sind auch die Unklarheiten abgeklärt worden.

6.4 Qualitätssicherung, Controlling

Die Qualität wurde durch verschiedene Methoden gesichert. Eine von denen war die Methode 5xWarum.

„Fünf warum“ ist eine iterative Methode, die Fragen als Basis hat und die Beziehungen zwischen die Ursachen und die Probleme. Es geht hier um die Verschachtelung der Ursachen und das Herausfinden von denen durch iterative Fragetechnik, da viele Probleme nicht nur eine einzige Ursache haben. Die Methode ruft jedes Mal eine andere Folge von Fragen auf.[10]

Die aufgetauchten Probleme haben viele Ursachen, die nicht mit dem ersten Blick sichtbar sind. 5x warum hilft durch diese verschachtelten Ursachen das grundlegende Problem zu entdecken.

Eine Problemstellung: „Segmentation fault“ beim Finden von Keypoints mit FAST[9].“

1. Was ist überhaupt ein „Segmentation Fault“? Ein Segmentierungsfehler tritt auf, wenn ein Programm versucht, auf einen Speicherort zuzugreifen, auf den es nicht

[9]Features from Accelerated Segment Test

zugreifen darf, oder wenn versucht wird, auf einen Speicherort auf nicht zulässige Weise zuzugreifen (z. B. beim Versuch, an einen schreibgeschützten Speicherort zu schreiben, oder einen Teil des Betriebssystems zu überschreiben).

2. Warum passiert das, warum versucht mein Program auf einen Speicherort zuzugreifen, auf den es nicht zugreifen darf?

   Problem ergibt sich in dieser Zeile: “kp = fast.detect(image, None)”

   Wahrscheinlich durften die Daten die fast.detect ergibt nicht in kp gespeichert werden.

   ODER

   Das Paket in dem die FAST Algorithmus drinnen ist ist nicht (richtig) installiert worden. Warum? Alle nötigen Pakete wurden in einer gesamten Installation geholt und FAST war nicht da.

3. Warum dürfen die Daten die fast.detect ergibt nicht in kp gespeichert werden ?

   Die Daten die fast.detect ergibt, dürfen nicht in kp gespeichert werden, weil kp kein Array ist.

4. Warum ist kp kein Array?

   Kp ist kein array, weil man es in Python manuell angeben muss.

5. Warum wurde es nicht manuell angegeben?

   Es wurde nicht manuell gegeben weil, die Methode fast.detect nicht gut recherchiert wurde. Man sollte mehr darüber in die Dokumentation nachschauen.

   Konklusion
   Durch die 5x Warum Methode ist es zu den Ergebnis gekommen: Es musste ja mehr untersucht werden bevor man eine solche Methode implementiert. Die „5xWarum“ Methode haben dabei geholfen dass die eigentliche Ursache des Problems herausgefunden worden ist.
   Die nächste Schritte sind: entweder eine andere Methode, die schon installiert ist, verwenden oder die benötigten Pakete für FAST manuell installieren.

6.5 Ergebnisse

Ich habe ungefähr 180 Stunden Zeit ins gesamt für die Diplomarbeit investiert und folgendes erreicht.
Gesichtsdetektion und das Zuschneiden von Gesichter wurden fertig implementiert.
Bilder sind ausführlich normalisiert worden.
Gesichtsschlüsselpunkte wurden extrahiert.

Kapitel 7

Gesichtserkennung - Rei

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

7.1 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, wird die Person erkannt. Der Person werden die Ergebnisse durch Anzeiger 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.7.1 zu sehen.

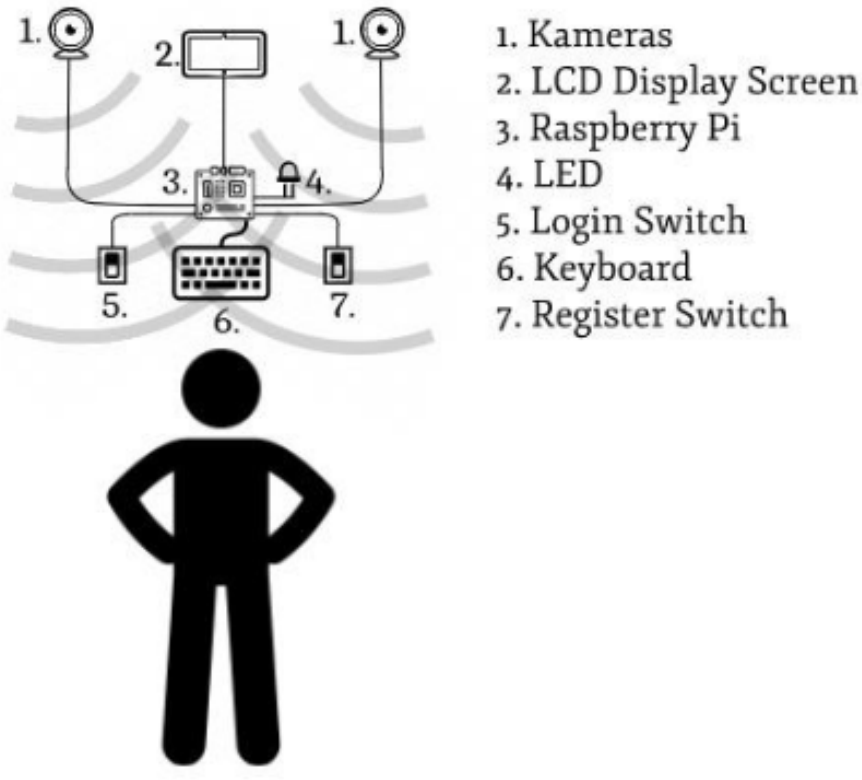


Abbildung 7.1: Grobe Skizze des Systems

7.2 Technische Lösung

In diesem Abschnitt wird eine feinere und detaillierte Übersicht im Bezug auf die technischen Lösung gegeben. Alles was mit Technik zu tun hat, wird hier erklärt.

7.2.1 Hardware und Aufbau

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

Bauteile und HW-Komponenten

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

1. *Steckboard bzw. Steckplatine*
   Die Steckplatine ist eine Komponente der Hardware des Systems. Sie wird für die Entstehung der elektrischen Verbindung von verschiedenen elektrischen Bauteile benötigt, um elektrische Schaltungen zu bauen oder um verschiedene 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*[10]
   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 hergestellt.

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 Schalter 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.[9]

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[11] Pins und einem VGA Schnittstelle. Die 3.-Bit SD-Karte ist ein wichtiges 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.

[10]Light Emitting Diode
[11]Generated Input Output

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

9. *Kamera*
   Es werden 2 Kameras benötigt, die die Fotos der Gesichter der Personen machen. Sie werden auch im Raspberry integriert bzw. mit dem Raspberry verbunden. Die Kameras sind vom 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 hergestellt. Die elektrische Schaltung wird durch einen Schaltplan beschrieben. Dieser Schaltplan wurde mit Hilfe eines Programms, "Fritzing", erstellt und spielt eine sehr wichtige Rolle bei der Organisation und Planung des Schaltkreises. Der Schaltplan ist auf Abb.7.2 zu sehen.

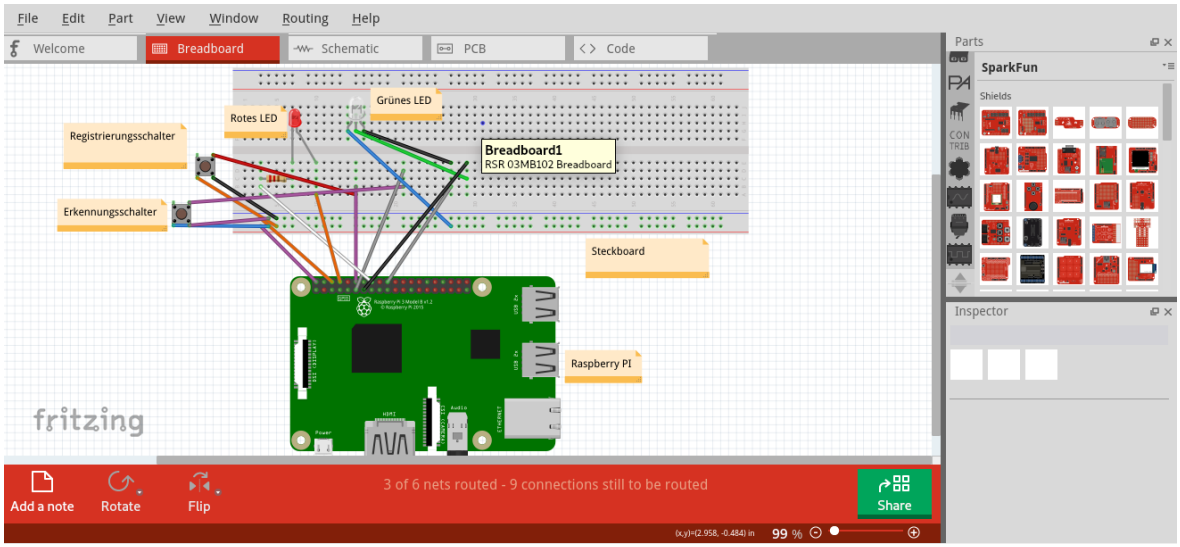


Abbildung 7.2: Schaltplan des Systems

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[12] 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

[12] Auch als Login Taster genannt

Datenanschluss mit dem GPIO17 Pin verbunden ist. 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.

7.2.2 Software

Nachdem der Aufbau und die verwendete Hardware des Systems beschrieben wurden, wird nun die Software beschrieben. In diesem Unterkapitel wird alles was mit dem logische Teil der Umsetzung zu tun hat besprochen: 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 Technologien

In diesem Teil des Projekts, das Gesichtserkennung heißt, werden für die Umsetzung 2 verschiedene Technologien verwendet, die in den anderen Teilen nicht oder anders verwendet sind.

- Open CV
  Das wichtigste Framework für dieses Projekt ist OpenCV. OpenCV ist eine Softwarebibliothek, die für Computer-Vision und maschinelles Lernen 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. [0]

- 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.7.2). 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 in Linux angeboten wird, was für Windows nicht der Fall ist.[0]

Lösungsweg- Beschreibung und Erklärung

Der Lösungsweg für die Umsetzung der Aufgaben der Gesichtserkennung ist streng mit der 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 erste Ebene sieht man unter Abb.7.3. Sie ist ein detaillierte Version des Big Picture, das sich nur auf den Erkennungsteil konzentriert. Man spricht von einer Iteration, die hier passiert ist. Folgend wird die Vorgehensweise und der Lösungsweg dieser Aufgabe mit Hilfe der 1.Ebene erklären und beschrieben.

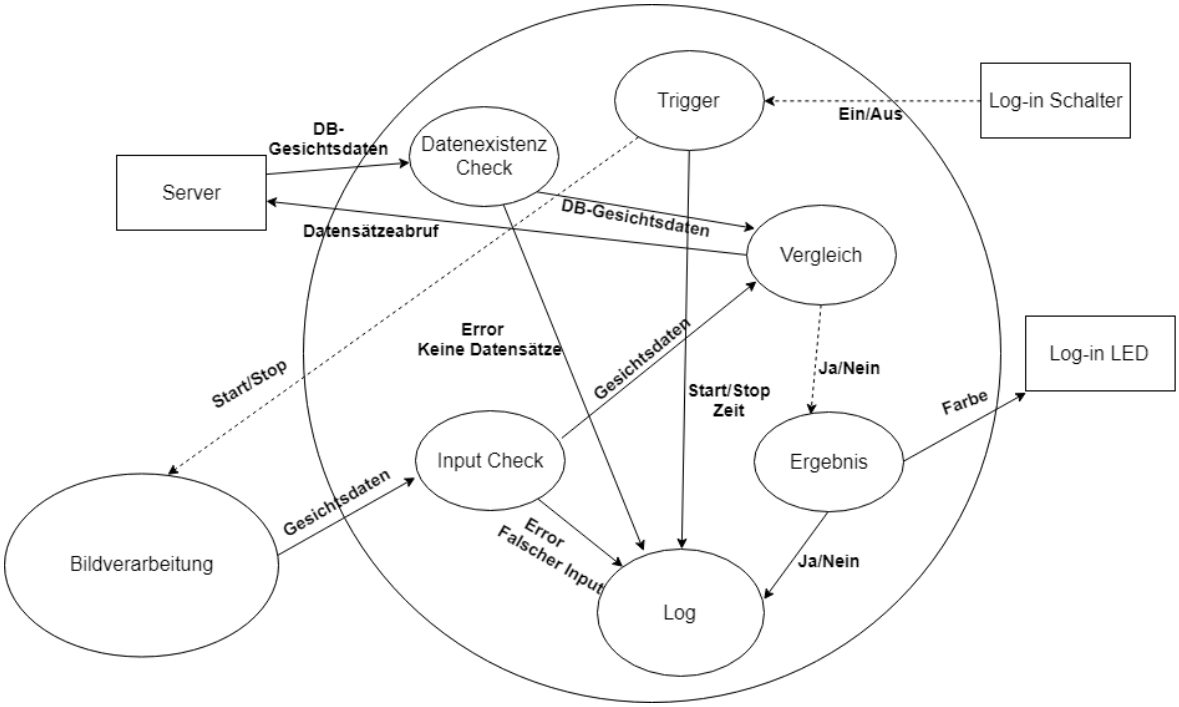


Abbildung 7.3: Erste Ebene

Die wichtigste Elemente der ersten Ebene sind auf der Abbildung 7.4 erklärt:

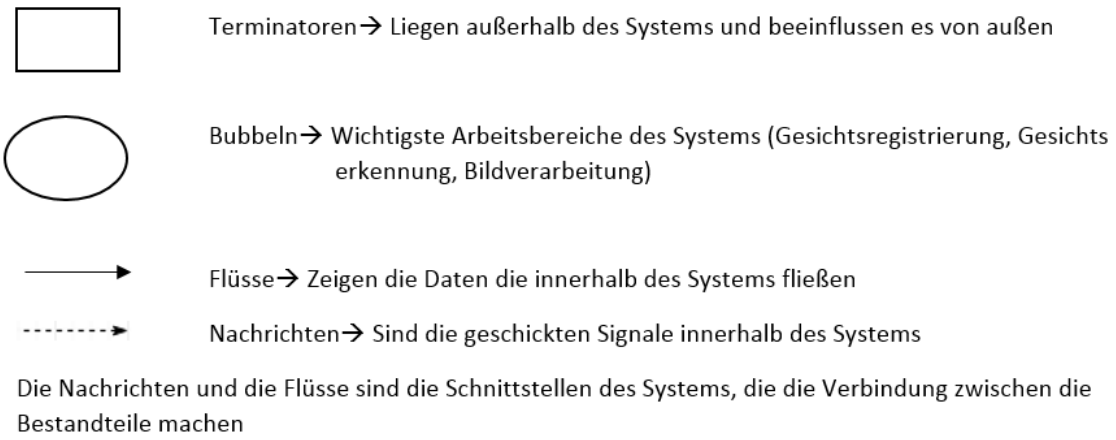


Abbildung 7.4: Erste Ebene Erklärung

Einzelne Arbeitsschritte des Erkennungsprozesses:

In diesem Unterkapitel wird der Vergleich der Gesichter der Personen detailliert erklärt. Es wurden die einzelne Schritte der Arbeit sowie die verwendeten Methoden beschrieben.

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, es dass für die Umsetzung dieses Systems 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. Es wurde deshalb die Emailadresse(und nicht Vorname, Nachname usw.) gewählt, weil diese Adresse immer eindeutig ist, das

heißt, es ist unmöglich, dass zwei Benutzern dieselbe Adresse haben, und dasselbe kann für die andere Benutzerdaten(Vorname, Nachname usw.) nicht gesagt werden.

3. *Die Existenz der eingegebenen Emailadresse des Benutzers in der Datenbank wird geprüft*

```
for x in myresult:
if x[2]==rei.emailiperkrahasim:
bool=True
else:
bool=False
os.system('./Log_Erkennung_Email_Nicht_Gefunden.py')
```

Wie es vorher erwähnt wurde, ist dieser Schritt deswegen wichtig, weil falls die eingegebene Email nicht in der Datenbank existiert, muss das System den Bildvergleich nicht machen. Der Benutzer kriegt stattdessen eine Anmeldung, die ihm sagt, dass er noch einmal versuchen kann seine Emailadresse einzugeben. Wird diese neue Adresse auch nicht in der Datenbank gefunden, kann danach dieser Benutzer mit dieser Adresse sich nicht mehr versuchen in System anzumelden.

4. *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 Bilddaten 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.

5. *Gemachtes Bild wird zugeschnitten*

Das von der Kamera gemachtes Bild ist noch nicht bereit zum Vergleich. Um die Punkte richtige zu extrahieren, muss das Bild zuerst zugeschnitten werden. Das heißt, die Dimensionen des Bildes werden reduziert, und somit werden die andere Objekten auf dem Bild nicht berücksichtigt(Sie würden die Performance der Vergleichsprozess negativ beeinflussen). Das Zuschneiden des Bildes wurde mit Hilfe eines Programms, das Teil der Arbeit der Bildverarbeitungsgruppe ist, gemacht. Wird das Bild zugeschnitten, wird es automatisch auch umbenannt. Das Bild nimmt das Schlüsselwort „New“+die Emailadresse der Person als Name. Der Datentyp ist .jpg .

6. *Die Gesichtspunkte werden extrahiert*

Das zugeschnittene Bild des Gesichts der Person besteht aus verschiedene Punkte. Insgesamt sind es 68 Gesichtspunkte. Alle diese Punkte müssen für den Vergleichsprozess extrahiert werden. Das heißt, jeder Punkt muss ein X und ein Y Wert

haben. Diese Koordinaten müssen ermittelt werden. Dies ist auch eine Aufgabe der Bildverarbeitungsbranche, und ist extrem wesentlich für das Funktionieren des Erkennungsprozesses.

7. *Wird dieselbe Emailadresse in der Datenbank gefunden, werden die zugehörige Bildinformationen geladen*

   Falls aber die vom Benutzer eingegebene Emailadresse schon in der Datenbank liegt, weißt das System Bescheid, dass ein Bilderdatenvergleich stattfinden muss. Um das zu erreichen, werden alle Bildinformationen geladen. Unter Bildinformationen sind die 68 extrahierten Gesichtspunkte 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)
```

   Das heißt, die extrahierte Punkte des von der Kamera gemachten Bildes und die extrahierte Punkte, die in der Datenbank gespeichert sind und die den Person mit derselbe Emailadresse wie die vom Benutzer eingegebene Adresse gehören , stehen jetzt endlich zur Verfügung. Ab hier wird angenommen, dass die eingegebene Emailadresse mit mindestens einer in der Datenbank vorhandenen Emailadresse gleich ist, damit der Vergleichsprozess erklärt werden kann.

8. *Die extrahierten Punkte werden in 5 verschiedene Bereiche geteilt*

   Bevor der Vergleichsprozess beginnt, müssen die 68 extrahierten Punkte geteilt werden. Die 5 großen Bereiche sind das Gesicht, die Nase, das rechte Auge, das linke Auge und der Mund. Jeder Bereich hat seine eigenen Punkte. Die erste 28 Punkte gehören dem Gesicht, die nächste 9 gehören der Nase, die nächste 6 dem rechten Auge, die nächste 6 dem linken Auge und die letzte 19 gehören dem Mund. Durch dieser Prozess wird die Vergleichsqualität stark erhöht, weil die Punkte der Nase werden genau mit der Punkte der anderen Nase verglichen, und nicht mit dem von dem Augen beispielsweise.
   Der untere Codeabschnitt zeigt wie dieser Schritt für die ersten 27 Punkte programmiert wird(nur um eine Idee zu haben, wie es funktioniert):

```
for pika in range(0,27):
 dis=math.sqrt(abs(( (vleratx.item(pika)-float(res[vx])))))*
 abs( (vleratx.item(pika)-float(res[vx])) ) + (
```

```
abs((vleraty.item(pika)-float(res[vy])))*
abs((vleraty.item(pika)-float(res[vy]))  )  ))
```

9. *Die Bildinformationen der beiden Bilder werden verglichen*

Die wichtigste Aufgabe ist der Vergleich der Bildinformationen. Es werden die extrahierten Gesichtspunkte der bereits gemachten Bilder mit der Gesichtspunkte in der Datenbank verglichen. Selbstverständlich müssen wie es vorher erwähnt wurde die beiden Personen, deren Gesichtspunkte verglichen werden dieselbe Emailadresse haben. Das wichtigste Paket, Open CV, wird für diesen Vergleich benötigt. Nachdem dieses Paket importiert wurde, kann der Vergleich beginnen. Es gibt verschiedene Schritte, die erfolgen, und die nun erklärt werden:

- Startpunkt bzw. Origin wird festgelegt

  In jeden Gesichtsbild wird ein Startpunkt bzw. eine Origin für den Vergleich festgelegt. Dieser Punkt ist der Punkt mit der Koordinaten (0/0). Es wird der Abstand von jedem anderen Gesichtspunkt zu dieser Origin berechnet. Dieser Abstand wird dann mit dem entsprechenden Abstand des zweiten Bildes verglichen. Diese Abstände müssen miteinander übereinstimmen(zum Beispiel ein Punkt vom rechten Augen im ersten Bild muss mit dem zugehörigen Punkt im zweiten Bild verglichen werden, und nicht mit einem Punkt von dem Augen in ersten Bild).

- Die Abstände jedes anderen Punktes des gemachten Bildes vom Startpunkt werden berechnet.
  Nachdem der Startpunkt festgelegt wurde, beginnt der Vergleich der Abstände des gemachten Bildes(Punkt-Startpunkt) und der Abstände, die in der Datenbank gespeichert sind.

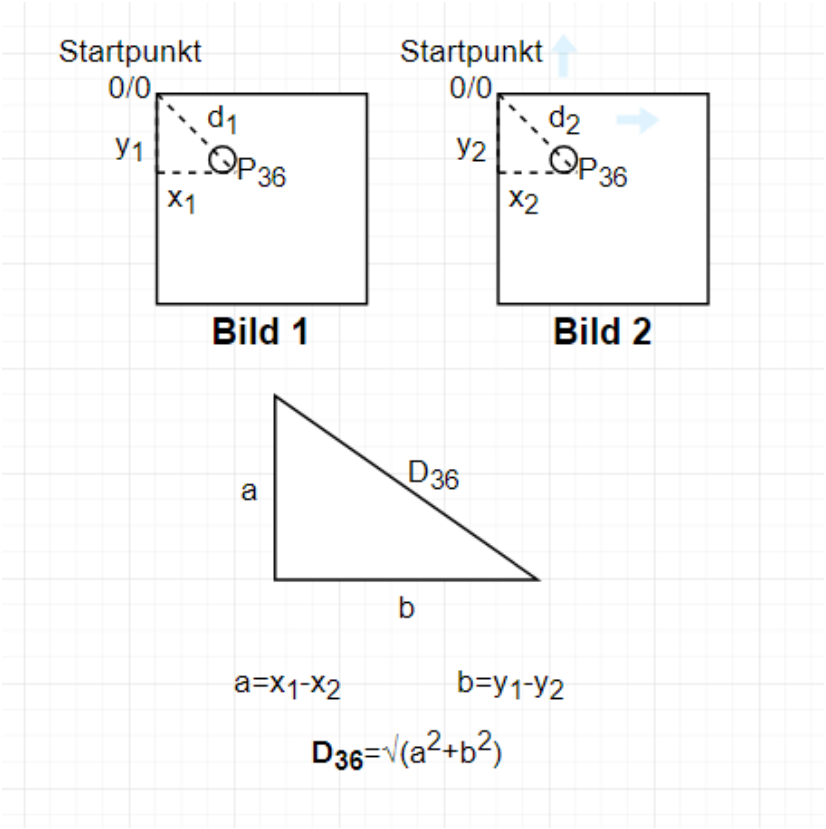


Abbildung 7.5: Vergleichsmethode zweier Punkte

Auf Abbildung 7.5 wird die verwendete Vergleichsmethode dargestellt. Der Startpunkt hat die Koordinaten (0/0) und befindet sich links oben. Jeder Punkt hat seine bestimmten Koordinaten, zum Beispiel, der Punkt 36 des ersten Bildes hat die Koordinaten x1 und y1, und den 36.Punkt des anderen Bildes hat die Koordinaten x2 und y2. Um den Abstand des Punkt 36 zu berechnen, wird die pythagoreische Formel verwendet. Die Differenz der x-Koordinaten und die Differenz der beiden y-Koordinaten sind die beide Katheten(a und b auf der Abbildung). Um die Hypotenuse zu finden wird das Quadrat beider Katheten addiert, und am Ende wird die Wurzel des Ergebnisses berechnet. Das Ergebnis ist der Abstand des Punktes 36 vom Startpunkt. Dieselbe Methode wird für jeder Punkt verwendet. Das obige Beispiel vergleicht die Abstände der Punkte von 2 verschiedenen Bildern. Das Vergleichsprinzip ist dasselbe wie bei dem Vergleich der Punkteabstände des gemachten Bildes und der in der Datenbank gespeicherten Punkteabstände. Es ist einfach leichter erklärbar und verstehbar, wenn 2 Bildern vergleicht wurden. Die Abstände aller Punkt aller Bereiche wurde in einer Variable gespeichert.

- Es wird der maximale Abstand für jeden einzelnen Bereich berechnet

  Nachdem die oben erklärten Abstände kalkuliert wurden, wird der maximale Abstand für jeden Bereich(Gesicht, Nase, linke Auge, rechte Auge, Mund)

benötigt. Diese Maximumwerte sind wesentlich für den Vergleich. Warum das so ist wird später erklärt.
Der untere Codeabschnitt zeigt wie dieser Schritt für die ersten 27 Punkte implementiert ist:

```
MAX = [0,0,0,0,0]
for pika in range(0,27):
dis=math.sqrt(abs(( (vleratx.item(pika)-float(res[vx])))))*
abs((vleratx.item(pika)-float(res[vx])) ) + (
abs((vleraty.item(pika)-float(res[vy])))*
abs((vleraty.item(pika)-float(res[vy])) ) ))
if(dis < min):
        min = dis
if(dis > max):
        max = dis
global MAX
MAX[0]=max
print("Max:\%s\n"\%(MAX[0]))
min=10.0
max=0.0
dis=0
```

Zuerst wurde eine Array MAX erstellt, die 5 Elemente enthält. Jedes Element gehört zu einem der 5 Bereiche. Beispielsweise der Maximumwert an der Stelle 0(erste Element) ist der Maximumabstand, der bei dem Gesichtsbereich gefunden wurde. Die Funktion Max in Python, die den größte Wert automatisch findet, wurde verwendet. Jeder Abstand wird mit dem Maximumwert verglichen(Maxwert beginnt am Anfang bei 0). Ist er größer, dann wird dieser Abstand der neue Maximumwert. So wird der Maximumwert ermittelt.

- Es wird das geometrische Mittel der Abstände aller Bereiche berechnet

"Das geometrische Mittel der Abstände aller Punkte (nicht in einzelnen Bereichen) ist auch wichtig für den Vergleich. Es gibt 68 Gesichtspunkten insgesamt, das heißt, 68 Abstände von dem Startpunkt. Um das geometrische Mittel von n Zahlen x1,x2,...xn zu ermitteln, muss man deren Produkt bilden und von diesem die n-te Wurzel ziehen."[0]

Auf Abbildung 7.6 wird die Formel, die sich ergibt gezeigt:

$$G(x_1, x_2, \ldots, x_n) = \overline{x}_{geom} = \sqrt[n]{x_1 \cdot x_2 \cdots x_n}$$

Abbildung 7.6: geometrisches Mittel

Der untere Codeabschnitt zeigt wie dieser Schritt für die ersten 27 Punkte

implementiert ist(nur um eine Idee zu haben, wie es funktioniert):

```
for pika in range(0,27):
dis=math.sqrt(abs(( (vleratx.item(pika)-float(res[vx])))) * abs(
(vleratx.item(pika)-float(res[vx])) ) + (
abs((vleraty.item(pika)-float(res[vy])))*
abs((vleraty.item(pika)-float(res[vy])) ) ))
gm *= dis
vx+=2
vy+=2
gm=gm**(1/27)
print("GM:%s\n"%(gm))
gm=1
dis=0
```

- Die ermittelte Maximumwerte und die geometrischen Mittel der Abstände der Punkte des gemachten Bildes werden mit der Maximumwerten und den geometrische Mitteln der Abstände der, in der Datenbank, gespeicherten Punkte verglichen

  (a) Beschreibung

  Wenn die angesprochene Maximumwerte und die geometrische Mitteln der Abstände der Punkte des gemachten Bildes und die geometrische Mitteln Abstände des in der Datenbank gespeicherten Punkte gleich oder sehr ähnlich sind, dann wird von der selben Person gesprochen und die Erkennung ist erfolgreich. Deshalb werden einige Abgrenzungen bestimmt, wann die Erkennung als erfolgreich gemacht wird und wann nicht.

  (b) Mögliche Fälle

  Insgesamt gibt es 5 Bedingungen bzw. 5 Fälle, die eintreten können und die wichtig sind. Diese Bedingungen wurden mit Hilfe des vielen gemachten Tests, darüber später gesprochen wird, bestimmt. Ist einer dieser fünf Bedingungen wahr, ist die Erkennung nicht erfolgreich und es wird nicht von derselben Person gesprochen. Diese Fälle werden unten mit Hilfe von Bildern erklärt. Es werden die Maximumwerten jeder Bereiche(Gesicht, Nase, linke Auge, rechte Auge, Mund, also 5 insgesamt), sowie das geometrische Mittel aller 68 Punkten berücksichtigt.

  i. 1.Fall: Mindestens 2 Maximumwerte sind größer als 0.06

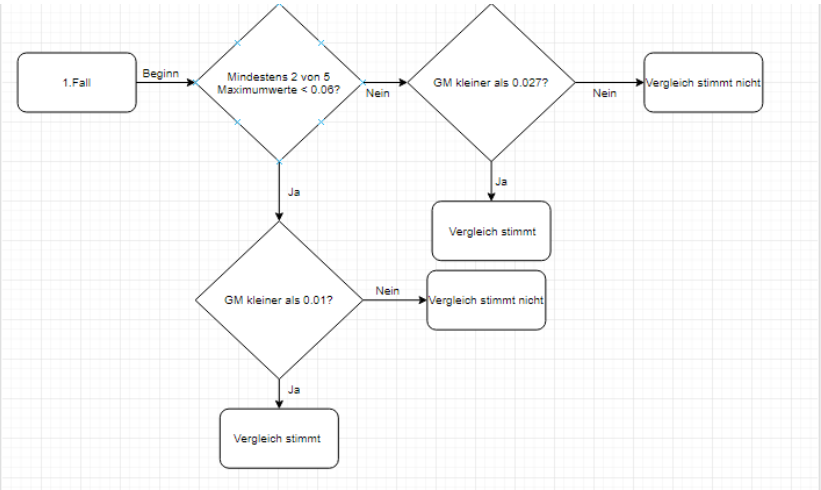


Abbildung 7.7: 1_Fall

Sind 2 oder mehrere Maximumwerte größer als 0.06, wird das geometrische Mittel betrachtet. Ist der Wert größer als 0.01, gibt es keine Übereinstimmung. Wenn weniger als 2 Werte größer als 0.06 sind, muss auch das geometrische Mittel kleiner als 0.027 sein, um eine erfolgreiche Erkennung zu haben.

Auf Abbildung 7.7 ist ein Flussdiagramm dargestellt, das die Abfolge dieser Bedingung beschreibt.

Unten ist der Codeabschnitt, der zeigt wie das funktioniert, ersichtlich:

```
global count
count = 0
for i in range(0,len(MAX)):
if MAX[i] > 0.06:
count+=1
status1=True
status11=True
status2=True
status22=True

#Wenn JA: Nur wenn gmT kleiner als 0.01 sind sie richtig,
sonst FALSCH
if count ==1:
global status11
status11=False
if count >= 2:
global status22
status22=False
if gmT > 0.01:
```

```
global status1
status1 = False

#Wenn NEIN: gmT muss kleiner als 0.027 sein,
                                    um RICHTIG zu sein
else:
if gmT > 0.027:
global status2
status2 = False
```

ii. 2.Fall: Genau einen Maximumwert größer als 0.07 ist und kleiner als 0.084

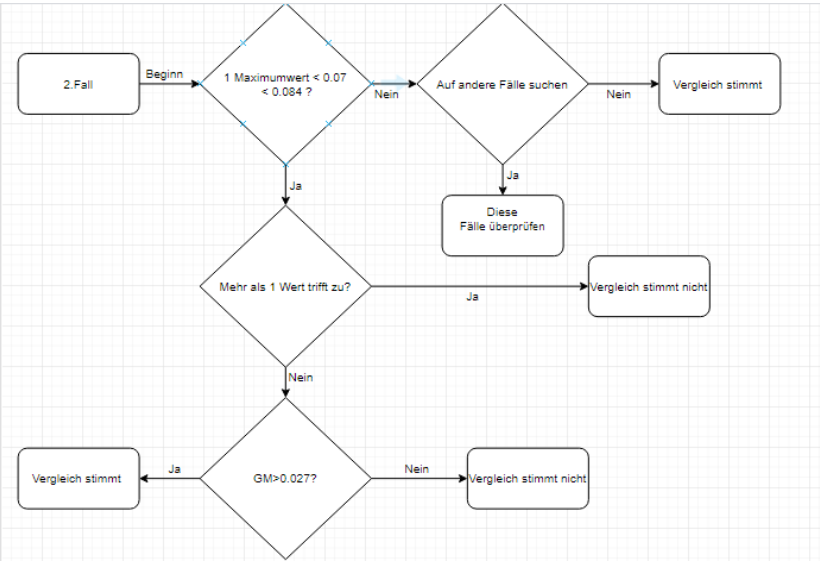


Abbildung 7.8: 2_Fall

Stimmt dieser Bedienung, wird wieder überprüft, ob das geometrische Mittel kleiner als 0.027 ist oder nicht. Wenn ja, dann gibt es keine Übereinstimmung. Ist aber dieser Bedienung wahr für mehr als einen Wert, gibt es keine Übereinstimmung.

Auf Abbildung 7.8 ist ein Flussdiagramm dargestellt, das die Abfolge dieser Bedingung beschreibt.

Unten ist der Codeabschnitt, der zeigt wie das funktioniert, ersichtlich:

```
count = 0
for i in range(0,len(MAX)):
if MAX[i] > 0.07 and MAX[i]< 0.084:
count+=1
status3=True
```

```
status33=True
status4=True
#Wenn JA :Wenn gmT kleiner als 0.027 ist passt,
sonst FALSCH
if count == 1:
global status33
status33=False
if gmT > 0.027:
global status3
status3 = False
#Wenn 2 oder mehr: FALSCH
elif count > 1:
global status4
status4 = False
```

iii. 3.Fall: Mindestens einen Maximumwert ist größer als 0.084

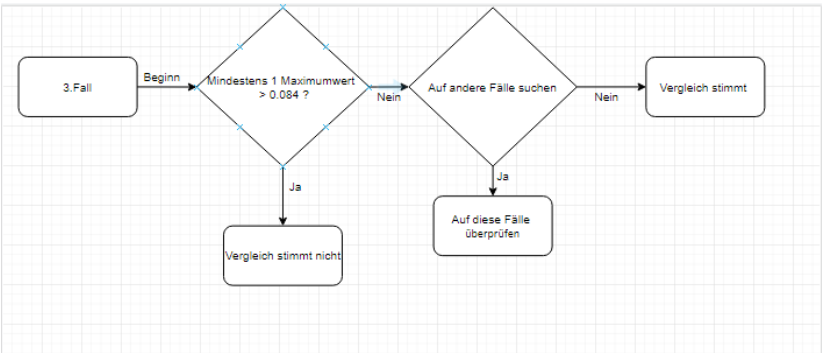


Abbildung 7.9: 3_Fall

Trifft dieser Bedienung zu, stimmt die Erkennung nicht und der Person wird nicht vom System erkannt.

Auf Abbildung 7.9 ist ein Flussdiagramm dargestellt, das die Abfolge dieser Bedingung beschreibt.

Unten ist der Codeabschnitt, der zeigt wie das funktioniert, ersichtlich:

```
for i in range(0,len(MAX)):
if MAX[i] > 0.084:
count+=1
status5=True
#Wenn JA: FALSCH
if count >=1:
global status5
status5=False
```

(c) Andere Abgrenzungen und Entscheidungen

Trifft eine von diesen 2 unterstehenden Kombinationen zu, ist weitere Überprüfung (geometrisches Mittel) nicht mehr nötig und es gibt keine Übereinstimmung.

i. 1 Maximumwert größer als 0.06 und 1 Maximumwert größer als 0.07

ii. 2 Maximumwerte größer als 0.06 und 1 Maximumwert größer als 0.07

Unten ist der Codeabschnitt, der zeigt wie das funktioniert, ersichtlich:

```
if status1==False or status2==False
or status3==False or status4==False
or status5==False or
(status11==False and status33==False)
or (status22==False and status33==False):
print("Vergleich nicht erfolgreich ,
sie sind nicht eingeloggt !!!")
os.system
('./Log_Erkennung_Person_Existiert_Nicht.py')
```

- *Ist die Erkennung erfolgreich, kriegt der Benutzer eine Information, dass er im System erfolgreich angemeldet ist.*

Wenn keinen Fall bzw. Bedienung zutrifft, ist die Authentifizierung erfolgreich. Der Benutzer kriegt danach eine Information, die ihm Bescheid gibt, dass die Authentifizierung erfolgreich war und dass er weitermachen darf. Dazu wurde eine gelbe LED verwendet, die leuchtet.

Unten ist der Codeabschnitt, der zeigt wie das funktioniert, ersichtlich:

```
else:
b="existiert"
print("Vergleich erfolgreich , Sie sind eingeloggt !!!")
os.system('./Log_Erkennung_Person_Existiert.py')
```

Andere wichtige Punkte und Abgrenzungen

Es gibt auch einige Punkte, die erwähnt werden müssen, und die zur Verwendung des Systems wesentlich sind:

1. Distanz der Benutzer von der Kamera

Damit der Vergleich und die Erkennung der Gesichter erfolgreich und so genau wie möglich sind, gibt es eine Distanz, die der Benutzer von der Kamera stehen muss. Diese Distanz ist 1 Meter, dem Benutzer wird das durch einer Linie auf dem Boden angezeigt.

2. Unbekannte Emailadresse eingegeben

   Wenn der Benutzer nach dem Drücken des Tasters eine falsche Emailadresse eingibt, wird er informiert, dass er eine zweite Chance bekommen wird, die richtige Emailadresse einzugeben(Vielleicht hat er zum Beispiel ein Fehler beim Eintippen gemacht). Ist die Adresse noch immer unbekannt oder inkorrekt, beginnt der Vergleichsprozess nicht und der Benutzer kann sich nicht anmelden.

3. Kein eindeutiges Gesicht von der Kamera nach der Bildaufnahme gefunden

   Wenn nach der Bildaufnahme kein eindeutiges Gesicht erkannt wird, wird ein zweites Bild gemacht. Der Benutzer wird durch eine Meldung erinnert, dass er gerade und an der Linie stehen muss.

4. Mehrere Gesichter von der Kamera nach der Bildaufnahme gefunden

   Wenn nach der Bildaufnahme mehrere Gesichter erkannt werden, wird wieder ein zweites Bild gemacht. Werden wieder mehrere Gesichter erkannt, kann das System nicht überprüfen, ob such der Benutzer anmelden darf oder nicht (Nicht Ziel).

5. Benutzer vom System nicht erkannt

   Ist die Authentifizierung nicht erfolgreich, leuchtet eine rote LED um den Benutzer zu zeigen, dass er sich nicht anmelden darf. Er kann aber alles nochmals probieren(Erkennungstaster drücken usw. . . .), oder er kann sich registrieren.

6. Verarbeitungszeit des Systems

   Jedes intelligente System braucht einer Verarbeitungszeit. Der Erkennungsprozess braucht ungefähr 7 bis 8 Sekunden. Das ist aber auch von der Geschwindigkeit des Benutzers abhängig.

Das Testen

Eine der wichtigsten Arbeitspakete des Erkennungsteils war das Testen des Vergleichsverfahrens. Um die Vergleichsbedingungen und Begrenzungen, die weiter oben erwähnt wurden, zu finden und zu bestimmen, war es wesentlich und extrem notwendig, viele Tests zu machen. Zuerst hat der Prozess des Testens mit verschiedene Bilder einer Person begonnen. Danach wurden verschieden Personen verwendet, um eine Toleranz zu finden, die in der Bedienungen verwendet werden kann. Nachdem die Gesichtspunkte

von 2 Bilder getestet und verglichen waren, sind die Ergebnisse auch mit den in der Datenbank gespeicherten Punkten getestet werden. Am Ende des Testens sind die vorher erklärte Bedingungen definiert und formuliert.

Das Logging

Jedes große System braucht Logging. Mit Hilfe einer Logdatei (englisch: log file) können alle oder bestimmte Aktionen von Prozessen auf einem Computersystem automatisch protokolliert werden. Das heißt, diese Logdatei kann vom Benutzer verwendet werden, um Fehlermeldungen zu erhalten, um den Fehler zu verstehen, um den Zeitpunkt verschiedener wichtiger Ereignissen zu erfahren und vieles mehr. Alle diese Vorteile und Merkmale des Logging waren mehr als genug zum entscheiden, dass die Protokollierung der Ereignisse des Systems ein wichtiges Ziel sind. Das war der Grund, warum der Gesichtserkennungsteil auch protokolliert.
Es wurden die wichtigsten Ereignisse mitprotokolliert:

1. Der Zeitpunkt, wann der Erkennungstaster gedrückt wurde

   Es ist sehr wichtig für die Admins des Systems zu wissen, wann der Erkennungstaster gedrückt wurde, bzw. wann die Gesichtserkennungsprozess beginnt. Diese Zeit kann dafür verwendet werden, um die totales Dauer des Erkennungsprozesses zu bestimmen.

2. Der Zeitpunkt, wann der Benutzer erfolgreich angemeldet war

   Dieses Ereignis ist ein sehr wichtiges, weil es der Schlusspunkt des Erkennungsteils ist. Es kann auch als Beweis verwendet werden, dass der Benutzer im System schon registriert war.

3. Der Zeitpunkt, wann der Benutzer sich nicht erfolgreich anmeldet

   Wie vorher erwähnt, ist eine Log-Datei eine sehr gute Methode, wichtige Fehlermeldungen sehr schnell zu erhalten. In diesem Fall wird gespeichert, wann der Benutzer sich nicht erfolgreich anmeldet.

4. Der Zeitpunkt, wann die eingegebene Emailadresse in der Datenbank nicht gefunden wird

   Gibt der Benutzer eine falsche oder eine nicht existierende Emailadresse ein, wird dieses Ereignis geloggt. Das ist extrem wichtig für die Administratoren, weil ohne eine vom Benutzer eingegebene Emailadresse, die in der Datenbank schon existiert, beginnt die Vergleichsprozess für die Erkennung gar nicht. Auf diesem Weg wissen die Administratoren, wo der Fehler ist, und der Benutzer wird darüber informiert.

Wie wurde das Logging gemacht?
Das Logging des Gesichtserkennungsprozesses war nicht sehr komplex in Python. Es gab eine Tabelle in der Datenbank, die für dieser Funktion geeignet war. In dieser Tabelle wurden alle Informationen, die protokolliert werden müssen, gespeichert. Mit Hilfe einer Python Skript wurde dann die Verbindung aller Skripten, die die notwendige Informationen senden mit dieser Log Tabelle, die in der Datenbank liegt, gemacht. Es wurde danach ein Programm für jeder Ereignis, das oben erwähnt war, erzeugt. Diese Programme ermöglichten die Verbindung mit der Datenbank, und zeigten eine Anmeldung. Beispielweise wurde in der Datenbank notiert, dass es ein Problem bei Gesichtserkennung gab(Taster nicht gedrückt..). In der Hauptskript wurde dann dieses einzelne Programm aufgerufen.

Unten ist der Codeabschnitt, der zeigt wie das funktioniert, ersichtlich.

Der Taster wird gedrückt:

```
if GPIO.input(17):
os.system('./Log_Erkennung_TasterGed.py')
```

Zeitpunkt wird gespeichert:

```
logging.basicConfig(filename=log_file_path)
if(log_to_db):
logging.getLogger('').addHandler(logdb)
log=logging.getLogger('Gesichtserkennung')
log.setLevel(log_error_level)
test_var='Taster gedruckt'
log.error('This event occurred: %s' % test_var)
```

7.3 Herausforderungen, Probleme, und deren Lösung

Während dieser Arbeit musste man einigen Problemen und Herausforderungen lösen.

- *OpenCV Installation*

  Die größten 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 wurde.

- *Fritzing*

  Das Problem beim Fritzing war, dass jedes Mal wenn das Programm beendet wurde, es nicht mehr geöffnet werden konnte. Es fehlten entweder die Pfaden oder

die Bauteile, die für die Schaltung 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 einen Kurzschluss gab. Der Raspberry PI war kaputt und musste ersetzt werden. Glücklicherweise konnte die SD-Karte erfolgreich in kopiert werden und deshalb sind alle Daten und Informationen gespeichert.

- *Paket FaceRecognition Installation*

  Die Installation des Paketes FaceRecognition konnte nicht erfolgreich gemacht werden, weil die dlib Pakete auch gebraucht wurden. 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 dem Aufruf der Skripten 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 über die korrekten Verwendung des *sys* Paketes gemacht um das Problem zu lösen.

- Schlechtere Vergleichsqualität

  Am Beginn des Testens des Vergleichs waren die Ergebnisse nicht stabil. Es konnte keinen Zusammenhang gefunden werden. Beispielsweise waren die Maximumwerte der Abstände bei Bilder, die derselben Person gehörten größer als bei Bildern, die verschiedenen Personen gehörten. Nach viele Tests wurde der Grund verstanden. Die 68 Punkte mussten in 5 verschiedene Teile(Gesicht, Nase, linke Auge, rechte Auge, Mund) zerlegt werden. In dieser Weise wird ein Punkt, der dem Mund gehört nicht mit einem Punkt, der der Nase gehört verglichen.

- Probleme bei Einteilung der Gesichtspunkte in 5 Bereichen

  Der Prozess der Einteilung der Gesichtspunkte in dem verschiedenen Bereiche war auch schwierig. Ein großes Array mit 68 Elementen wurde verwendet. Es war sehr kompliziert, auf einem Index dieses Array zu zugreifen. Deshalb wurde dieses große Array in 5 kleinere Arrays zerlegt. Auf diese Weise war es viel leichter zu verstehen, wie die Indizien funktionieren und wie man dort zugreifen kann.

- Probleme bei der Berechnung des geometrischen Mittels

Die Berechnung des geometrischen Mittels war auch nicht leicht. Nachdem die Formel gefunden war, musste es in dem Programm integriert werden. Die Schwierigkeit war bei dem Ziehen von der N-Wurzel. Sollte das n 67 oder 68 sein? Das war von dem Index abhängig. Am Ende wurde verstanden, dass das Produkt aller Abstände hoch 1/68, die richtige Lösung ist.

- Probleme mit der Datenbank

  Probleme hat es noch beim Zugriff auf der in der Datenbank gespeicherten Gesichtspunkte gegeben. Es war sehr kompliziert zu verstehen, wie und wo die Punkte genau gespeichert waren(Auf welchem Index die X-Werte, auf welchem die Y-Werte). Es wurde deshalb Hilfe von dem Mitarbeiter, der sich mit der Datenbank beschäftigt hätte, gebraucht, und danach war alles klarer und verständlich.

- Probleme mit GIT

  Am Ende müssen die Probleme mit Git erwähnt werden. Es gab viele Fälle, wo es Konflikten mit der Dateien gab. Man muss immer zuerst Pull machen, und danach Push. Das ist aber oft nicht passiert. Es wurde in einer Datei gearbeitet, die aber von einem anderen Person schon geändert wurde. Deshalb gab es Probleme bei den Push. Oft mussten die ganze Dateien gelöscht werden, und danach wieder vom Git geholt werden. Es wurde am Ende zwischen der Gruppe darüber gesprochen und das Problem gelöst.

7.4 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 das Problem in kleinen Teile geschnitten damit es klarer wird. Es wurde eine detaillierte Soll-Ist Analyse gemacht, die bei der neuen Aufgabeteilung sehr geholfen hat, und eine Fehlerbaumanalyse zu erstellen.

Die Fehlerbaumanalyse wird verwendet, um die Zuverlässigkeit von technischen Systemen zu testen. Die Fehlerbaumanalyse nimmt als Ausgangspunkt nicht eine einzelne Systemkomponente, sondern das potenziell gestörte Gesamtsystem. Die Fehlerbaumanalyse baut auf der sogenannten negativen Logik auf. Das heißt, der Fehlerbaum beschreibt eine Ausfallsfunktion die bei dem Zustand logisch-1 einen Ausfall ausdrückt, bei logisch-0 liegt ein funktionsfähiges System vor. Sie gehört zu den “Top-Down”-Analyseformen im Risikomanagement. In einem ersten Schritt wird daher das Gesamtsystem detailliert und exakt beschrieben. Darauf aufbauend wird analysiert, welche primären Störungen eine Störung des Gesamtsystems verursachen oder dazu beitragen können. Ausgangspunkt ist hierbei zunächst ein einziges unerwünschtes Ereignis, welches an der Spitze des Fehlerbaums steht, das sogenannte Top-Ereignis. Das Top-Ereignis resultiert in der Regel aus einer Risikoanalyse bzw. Szenarioanalyse. In der einfachsten Form besteht er aus folgenden Elementen: Entscheidungsknoten (E), die

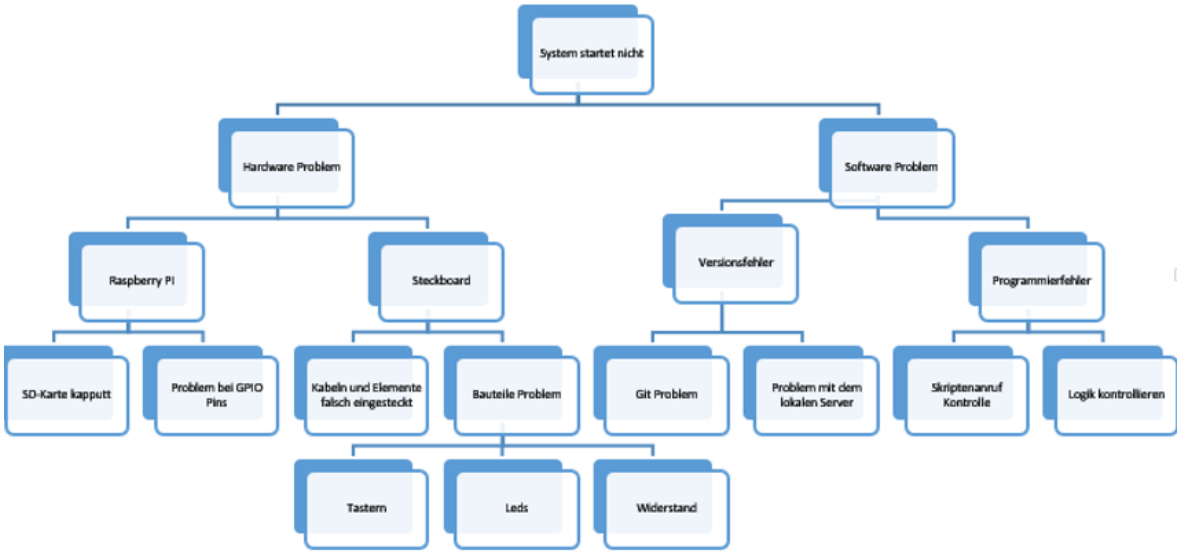


Abbildung 7.10: Fehlerbaum

Entscheidungen kennzeichnen, Zufallsknoten, die den Eintritt eines zufälligen Ereignisses darstellen sowie aus Ergebnisknoten (R), die das Ergebnis von Entscheidungen oder Ereignissen darstellen.

Auf der Abbildung 7.10 wird ein Fehler in den Gesichtserkennungsteil des Systems beschrieben. Zuerst muss es überprüft werden, ob der Fehlern bei der Hardware oder Software liegt.

Hardware Problem
Ist es ein Problem in Hardware, wird der Raspberry Pi und die Steckboard getestet. Beim Raspberry kann es entweder Probleme mit dem SD-Karte oder mit der GPIO Pins geben. Bei dem Steckboard sind entweder die Kabel und die Verbindungen falsch eingesteckt, oder die Bauteile sind kaputt(Taster, Widerstand, LED, Kamera).

Software Problem
Wenn es ein Softwareproblem ist, sind entweder die Versionen vermischt(Problem mit GIT), oder es gibt Programmierfehlern. Diese Programmierfehlern können bei der Logik(Algorithmen) oder bei der Syntax auftreten(Skripten Aufruf, Variablen usw.)

7.5 Ergebnisse

Die finale Ergebnisse sind die folgende:

1. Systemaufbau

Das ganze System wurde zusammen mit der Hardware aufgebaut.

2. Digitale Darstellung des Systemaufbaus
   Der Schaltplan des Systems wurde durch Fritzing digital dargestellt.

3. Erkennung der Gesichter
   Es wurde eine Gesichtserkennung erreicht, die in 80 Prozent der Fälle funktioniert.

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.

7.6 Was wurde gelernt und welche Verbesserungs möglichkeiten gibt es

Während dieser Diplomarbeit wurde viele wichtiges gelernt und verstanden. Die wichtigsten Lektionen werden unten genannt und erklärt.

- Zeitmanagement

  Die Wichtigkeit der Zeit wurde wirklich verstanden. Man hatte nicht mehr die Möglichkeit, Dinge immer für den letzte Moment zu lassen, weil es einfach zu viel ist. Ohne einen detaillierten Zeitplan und ohne eine systematische Arbeit würde dieses Projekt nicht fertig sein.

- Wie man mit OpenCV und mit Bilderdaten arbeitet

  Es wurde gelernt, wie man überhaupt mit Bildinformationen arbeitet, und wie das Paket OpenCV funktioniert. In Zukunft wenn es um Arbeiten mit Bilderdaten geht, ist das Basiswissen vorhanden.

- Wichtigkeit des Testens des Systems

  Wie wichtig der Test des Prozesses ist wurde auch hier verstanden. Vorher wurde gedacht, dass das Testen des Systems nur eine zusätzliche und unwichtige Arbeit ist. Es war aber nur mit Hilfe des Testens möglich, den Vergleichsprozess erfolgreich zu realisieren und Fehler zu finden. Ein System, das vorher nicht getestet ist, kann kein gutes System sein.

- Gruppenarbeit und Verantwortlichkeiten übernehmen

  Die Rolle der Gruppe ist auch sehr wichtig. Geht es den Mitarbeitern miteinander gut und sie sind gut gelaunt, sind sie auch mehr motivierter zum Arbeiten. Die Kommunikation soll nie fehlen und die Entscheidungen des Projektleiters sollen

akzeptiert werden. Man muss auch seine Verantwortlichkeiten selbst übernehmen und seine Arbeit mit seiner Stärken erledigen. Das wurde alles auch während der Arbeit gelernt.

- Wichtigkeit der Planung und der Begrenzungen

  Es wurde schon gelernt, dass die Planung eines Projekts sehr wichtig war, aber mit Hilfe dieser Diplomarbeit wurde ganz genau verstanden, warum das so ist. Ohne die Big Pictures und die Erste Ebene wurde der Aufwand der Arbeit viel grösser, und die Zeit würde sicherlich nicht genügen. Die Begrenzungen sind auch wichtig, weil ohne die wurde man Zeit an etwas verlieren, das am Ende nicht realisierbar oder nicht nötig ist.

- Wie man mit Fehlern umgeht

  Außerdem wurde verstanden und gelernt, wie man mit Fehlern umgehen soll. Pessimismus und das Aufgeben sind nicht die Lösungen. Man muss optimistisch sein und viele verschiedene Wege suchen, wie man den Fehlern finden und dann reparieren kann. Fremde Hilfe ist auch irgendwann notwendig.

Verbesserungsmöglichkeiten sind auch erkannt werden. Diese sind:

- Besseres Zeitmanagement

  Die Arbeit musste ein bisschen früherer begonnen worden, und es muss besser verteilt werden. Es muss verstanden werden, dass hinter einem schlechten Projekt eine schlechte Zeitmanagement versteckt.

- Backup Verfahren

  Nächstes Mal wird die Arbeit öfter gespeichert. Back-Ups werden täglich gemacht, damit nichts verloren geht. Man denkt, dass das lokale Speichern genug ist. Was ist aber, wenn der PC kaputt wird? Das wird in die nächste Arbeiten mehr berücksichtigt.

7.7 Ausblick

In diesem Kapitel wird kurz erklärt, was mit dem Projekt in der Zukunft passieren wird?

7.7.1 Wie geht es weiter ?

Dieses Projekt wird nicht in der Schule implementiert, weil die Schule es nicht angenommen hat. Dieses Projekt war zu kompliziert, um in die Schulinfrastruktur implementiert zu werden. Nach der Diploma Präsentation wird dieses Projekt abgeschlossen.

7.7.2 Was passiert mit dem Ergebnis?

Das Ergebnis wird von der Schule betrachtet, aber nichts besonders anders und nicht weiterentwickelt von dieser Gruppe.

7.7.3 was passiert mit dem Prototyp?

Der Prototyp wird im Lager der Schule eingeschlossen, da kein Interesse daran hat.

7.7.4 Gibt es Folgeprojekte?

Momentan gibt es keine, aber die anderen Gruppen mit dem Schwerpunkt Systemtechnik können bzw. haben die Möglichkeit mit dem Projekt weiterzumachen.

7.7.5 Gibt es Interessanten?

Sowohl von der Schule als auch von anderen Unternehmen gibt es keine Interessengruppen, die dieses Projekt finanzieren möchten.

Abbildungsverzeichnis

2.1 Big picture 4
2.2 Scrum 5

3.1 Structed Software Design bzw. erste Ebene 9
3.2 Risikoanalyse in Excel 16

4.1 MySQL Logo 18
4.2 MariaDB Logo 19
4.3 Python Logo 20
4.4 OpenCV Logo 20
4.5 Plan-Do-Check-Act-Zyklus 26

6.1 dlib logo [19] 30
6.2 Bildverarbeitung Structed Design 31
6.3 Haar Features[26] 33
6.4 Output: Box auf Gesicht 35
6.5 Unterschied: normalisiert, nicht normalisiert 38
6.6 Unterschied: normalisiert, nicht normalisiert .2 38
6.7 Output: Abgeschnittenes Gesicht 39
6.8 Gesichtsschlüsselpunkte [18] 40
6.9 Gesichtsdaten 41

7.1 Grobe Skizze des Systems 45
7.2 Schaltplan des Systems 47
7.3 Erste Ebene 49
7.4 Erste Ebene Erklärung 50
7.5 Vergleichsmethode zweier Punkte 54
7.6 geometrisches Mittel 55
7.7 1_Fall 57
7.8 2_Fall 58
7.9 3_Fall 59

7.10 Fehlerbaum 66

Tabellenverzeichnis

4.1 Technologien 18

Datenblätter und Applikationsschriften

[1] MA BSc Bekim Alibali. *ITP QM ALB Skript SchülerTeil1*.

Aus dem Netz

[2] URL: https://www.mysql.com/.

[3] URL: https://mariadb.org/.

[4] URL: https://www.python.org/.

[5] URL: https://opencv.org/.

[0] URL: https://fritzing.org/learning/.

[0] URL: https://de.serlo.org/mathe/stochastik/daten-datendarstellung/daten-kenngroessen/geometrisches-mittel.

[0] *Einführung in Computer Vision mit OpenCV und Python*. URL: https://blog.codecentric.de/2017/06/einfuehrung-in-computer-vision-mit-opencv-und-python/.

[6] *Einsteigerguide: Was ist OpenCV?* URL: https://viscircle.de/einsteigerguide-was-ist-opencv/.

[7] *Git*. URL: https://git-scm.com/docs.

[8] Kitware inc. *CMake*. 2000. URL: url.

[9] *Unterschied Taster als Schließer, Taster als Wechsler und Schalter*. URL: https://dein-elektriker-info.de/taster-als-schliesser-und-taster-als-wechsler/.

Der ganze Rest

[10] *5x Warum*. https://www.quality.de/lexikon/5xwarum/. [Online; accessed 2019]. 2017.

[11] Stephan Augsten. *Was ist Python?* `https://www.dev-insider.de/was-ist-python-a-843060/`. [Online; accessed 2019]. 12/7/2019.

[12] Martin H. Badicke. "MariaDB-Monitor?" In: (7. März 2017).

[13] MA BSc Bekim Alibali. "Projektmanagement Teil5".

[14] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000).

[15] Jackson Cooper. *Python's time.sleep() – Pause, Stop, Wait or Sleep your Python Code*. `https://www.pythoncentral.io/pythons-time-sleep-pause-wait-sleep-stop-your-code/`. [Online; accessed Tuesday 23rd July 2013]. 2013.

[16] Ben Croston. *RPi.GPIO Python Module*. `https://sourceforge.net/p/raspberry-gpio-python/wiki/Home/`. [Online; accessed 2014]. 2014.

[17] EDUCBA. *Differences Between Linux vs Windows*. `https://www.educba.com/linux-vs-windows/`. [Online; accessed 2019]. 2017.

[18] Vahid Kazemi und Josephine Sullivan. "One millisecond face alignment with an ensemble of regression trees". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014), S. 1867–1874.

[19] Davis E. King. "Dlib-ml: A Machine Learning Toolkit". In: *Journal of Machine Learning Research* 10 (2009), S. 1755–1758.

[20] Stefan Luber. "Was is Python?" In: *Nico Litzel* (4. Juli 2018).

[21] CHIP Digital GmbH Niels Held. *Linux-Umstieg: So einfach gelingt der Windows-Wechsel*. `https://www.chip.de/artikel/Linux-Umstieg-So-einfach-gelingt-der-Windows-Wechsel-2_140047889.html`. [Online; accessed 2007]. 2007.

[22] Mindfire solutions. *Python: 7 Important Reasons Why You Should Use Python*. `https://medium.com/@mindfiresolutions.usa/python-7-important-reasons-why-you-should-use-python-5801a98a0d0b/`. [Online; accessed 2019]. 3/10/2017.

[23] Guido Stepken. "MySQL Datenbankhandbuch". In: *Abruf: http://www. littleidiot. de/mysql (am 23.04. 2008)* (1999).

[24] Dominik Stocklasser. "Architektur".

[25] Jeff Tranter. *Control Raspberry Pi GPIO Pins from Python*. `https://www.ics.com/blog/control-raspberry-pi-gpio-pins-python`. [Online; accessed Wednesday, July 31, 2019]. 2019.

[26] Paul Viola und Michael Jones. "Robust Real-time Object Detection". In: *International Journal of Computer Vision*. 2001.

[27] Pauli Virtanen u. a. "SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python". In: *arXiv e-prints*, arXiv:1907.10121 (Juli 2019), arXiv:1907.10121. arXiv: `1907.10121 [cs.MS]`.

```
#!/usr/bin/python3

import cv2
cam = cv2.VideoCapture(0)
cv2.namedWindow("test")

img_counter = 0
while True:
    ret, frame = cam.read()
    cv2.imshow("test", frame)
    if not ret:
        break
    k = cv2.waitKey(1)
    if k%256 == 27:
        print("Escape hit, closing...")
        break
    elif k%256 == 32:
        #space
        img_name = "frame{}.png".format(img_counter)
        cv2.imwrite(img_name, frame)
        print("{} written!".format(img_name))
        img_counter+=1

cam.release()

cv2.destroyAllWindows()
```

```
#!/usr/bin/python3
import MySQLdb
import time
import logging
db_server='localhost'
db_user='aronterzeta'
db_password='aronterzeta'
db_dbname='test'
db_tbl_log='log'
log_file_path='/home/pi/diplomarbeit/python_scripts/fehlerlog.txt'
log_error_level='DEBUG'
log_to_db=True

class LogDBHandler(logging.Handler):
    def __init__(self,sql_conn,sql_cursor,db_tbl_log):
        logging.Handler.__init__(self)
        self.sql_cursor=sql_cursor
        self.sql_conn=sql_conn
        self.db_tbl_log=db_tbl_log
    def emit(self,record):
```

```
            tm=time.strftime("%Y-%m-%d %H:%M:%S", time.localtime(record.creat
            self.log_msg=record.msg
            self.log_msg=self.log_msg.strip()
            self.log_msg=self.log_msg.replace('\'', '\'\'')
            sql='insert into '+self.db_tbl_log + ' (log_level, ' + \
                'log_levelname,log,created_at,created_by) ' + \
                'values (' + \
                '' + str(record.levelno) + ', ' + \
                '\'' + str(record.levelname) + '\', ' + \
                '\'' + str(self.log_msg) + '\', ' + \
                ' \'' + tm + '\', ' + \
                '\'' + str(record.name) + '\')'
            try:
                self.sql_cursor.execute(sql)
                self.sql_conn.commit()
            except MySQLdb.Error as e:
                print (sql)
                print("Critical DB Error")
if(log_to_db):
    log_conn=MySQLdb.connect(db_server,db_user,db_password,db_dbname,30)
    log_cursor=log_conn.cursor()
    logdb=LogDBHandler(log_conn,log_cursor,db_tbl_log)

logging.basicConfig(filename=log_file_path)
if(log_to_db):
    logging.getLogger('Aron').addHandler(logdb)
log=logging.getLogger('Gesichtsregistrierung')
log.setLevel(log_error_level)
test_var='Admin ist eingeloggt'
log.error('Die Nachricht ist: %s' % test_var)
```

```
#!/usr/bin/python3
import MySQLdb
import time
import logging
db_server='localhost'
db_user='aronterzeta'
db_password='aronterzeta'
db_dbname='test'
db_tbl_log='log'
log_file_path='/home/pi/diplomarbeit/python_scripts/fehlerlog.txt'
log_error_level='DEBUG'
log_to_db=True

class LogDBHandler(logging.Handler):
    def __init__(self,sql_conn,sql_cursor,db_tbl_log):
        logging.Handler.__init__(self)
```

```
            self.sql_cursor=sql_cursor
            self.sql_conn=sql_conn
            self.db_tbl_log=db_tbl_log
        def emit(self,record):
            tm=time.strftime("%Y-%m-%d %H:%M:%S", time.localtime(record.creat
            self.log_msg=record.msg
            self.log_msg=self.log_msg.strip()
            self.log_msg=self.log_msg.replace('\'', '\'\'')
            sql='insert into '+self.db_tbl_log + ' (log_level, ' + \
                'log_levelname,log,created_at,created_by) ' + \
                'values (' + \
                '' + str(record.levelno) + ', ' + \
                '\'' + str(record.levelname) + '\', ' + \
                '\'' + str(self.log_msg) + '\', ' + \
                ' \'' + tm + '\', ' + \
                '\'' + str(record.name) + '\')'
            try:
                self.sql_cursor.execute(sql)
                self.sql_conn.commit()
            except MySQLdb.Error as e:
                print (sql)
                print("Critical DB Error")
if(log_to_db):
    log_conn=MySQLdb.connect(db_server,db_user,db_password,db_dbname,30)
    log_cursor=log_conn.cursor()
    logdb=LogDBHandler(log_conn,log_cursor,db_tbl_log)

logging.basicConfig(filename=log_file_path)
if(log_to_db):
    logging.getLogger('Aron Terzeta').addHandler(logdb)
log=logging.getLogger('Gesichtsregistrierung')
log.setLevel(log_error_level)
test_var='Eine Person ist in der Datenbank registriert'
log.error('Die Nachricht ist: %s' % test_var)
```

```
#!/usr/bin/python3
import MySQLdb
import time
import logging
db_server='localhost'
db_user='aronterzeta'
db_password='aronterzeta'
db_dbname='test'
db_tbl_log='log'
log_file_path='/home/pi/diplomarbeit/python_scripts/fehlerlog.txt'
log_error_level='DEBUG'
log_to_db=True
```

```
class LogDBHandler(logging.Handler):
    def __init__(self,sql_conn,sql_cursor,db_tbl_log):
        logging.Handler.__init__(self)
        self.sql_cursor=sql_cursor
        self.sql_conn=sql_conn
        self.db_tbl_log=db_tbl_log
    def emit(self,record):
        tm=time.strftime("%Y-%m-%d %H:%M:%S", time.localtime(record.creat
        self.log_msg=record.msg
        self.log_msg=self.log_msg.strip()
        self.log_msg=self.log_msg.replace('\'', '\'\'')
        sql='insert into '+self.db_tbl_log + ' (log_level, ' + \
            'log_levelname,log,created_at,created_by) ' + \
            'values (' + \
            '' + str(record.levelno) + ', ' + \
            '\'' + str(record.levelname) + '\', ' + \
            '\'' + str(self.log_msg) + '\', ' + \
            ' \'' + tm + '\', ' + \
            '\'' + str(record.name) + '\')'
        try:
            self.sql_cursor.execute(sql)
            self.sql_conn.commit()
        except MySQLdb.Error as e:
            print (sql)
            print("Critical DB Error")
if(log_to_db):
    log_conn=MySQLdb.connect(db_server,db_user,db_password,db_dbname,30)
    log_cursor=log_conn.cursor()
    logdb=LogDBHandler(log_conn,log_cursor,db_tbl_log)

logging.basicConfig(filename=log_file_path)
if(log_to_db):
    logging.getLogger('Aron Terzeta').addHandler(logdb)
log=logging.getLogger('Gesichtsregistrierung')
log.setLevel(log_error_level)
test_var='Fehler bei der Registrierung einer neuen Person'
log.error('Die Nachricht ist: %s' % test_var)
```

```
#!/usr/bin/python3
import MySQLdb
import time
import logging
db_server='localhost'
db_user='aronterzeta'
db_password='aronterzeta'
db_dbname='test'
```

```
db_tbl_log='log'
log_file_path='/home/pi/diplomarbeit/python_scripts/fehlerlog.txt'
log_error_level='DEBUG'
log_to_db=True

class LogDBHandler(logging.Handler):
    def __init__(self,sql_conn,sql_cursor,db_tbl_log):
        logging.Handler.__init__(self)
        self.sql_cursor=sql_cursor
        self.sql_conn=sql_conn
        self.db_tbl_log=db_tbl_log
    def emit(self,record):
        tm=time.strftime("%Y-%m-%d %H:%M:%S", time.localtime(record.creat
        self.log_msg=record.msg
        self.log_msg=self.log_msg.strip()
        self.log_msg=self.log_msg.replace('\'', '\'\'')
        sql='insert into '+self.db_tbl_log + ' (log_level, ' + \
            'log_levelname,log,created_at,created_by) ' + \
            'values (' + \
            '' + str(record.levelno) + ', ' + \
            '\'' + str(record.levelname) + '\', ' + \
            '\'' + str(self.log_msg) + '\', ' + \
            ' \'' + tm + '\', ' + \
            '\'' + str(record.name) + '\')'
        try:
            self.sql_cursor.execute(sql)
            self.sql_conn.commit()
        except MySQLdb.Error as e:
            print (sql)
            print("Critical DB Error")
if(log_to_db):
    log_conn=MySQLdb.connect(db_server,db_user,db_password,db_dbname,30)
    log_cursor=log_conn.cursor()
    logdb=LogDBHandler(log_conn,log_cursor,db_tbl_log)

logging.basicConfig(filename=log_file_path)
if(log_to_db):
    logging.getLogger('Aron').addHandler(logdb)
log=logging.getLogger('Gesichtsregistrierung')
log.setLevel(log_error_level)
test_var='Der Schalter fuer Gesichtsregistrierung ist gedrueckt.'
log.error('Die Nachricht ist: %s' % test_var)

#!/usr/bin/python3.5‘
import MySQLdb

#def drop_tables(curs):
```

```
    #curs.execute("""Drop table if exists test1""")
#def create_tables(curs):
    #curs.execute("""create table test1(id int, name TEXT)""")
#def insert_value(curs,id1,name1):
 #    add_test=("insert into test1(id,name) values(%s,'%s');")
  #   data_test=(id1,name1)
   #  curs.execute(add_test%data_test)
    #return "inserted values"
#def read_value(curs):
    #curs.execute("select * from test1")
    #return curs.fetchall()
def insert_values1(curs,vorname,nachname,email,role):
    add_person=("Insert into person(vorname,nachname,email,role)"\
                "Values('%s','%s','%s',%s);")
    data_person=(vorname,nachname,email,role)
    curs.execute(add_person%data_person)
    #return "inserted values"
#def insert_values2(curs, spX, spY, v1X,v1Y,v2X,v2Y,v3X, v3Y,v4X,v4Y,v5X,
 #    add_info=("INSERT INTO info (spX, spY, v1X,v1Y,v2X,v2Y,v3X, v3Y,v4X,
  #                "VALUES(%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s,

   # data_info=(spX, spY, v1X,v1Y,v2X,v2Y,v3X, v3Y,v4X,v4Y,v5X,v5Y, v6X,v
    #curs.execute(add_info%data_info)
print ("Setting up database")
a=""
conn=MySQLdb.connect('localhost','aronterzeta','aronterzeta','test')
curs=conn.cursor()
#try:
    #drop_tables(curs)
    #conn.commit()
    #print ("Dropped tables")
#except:
    #conn.rollback()
    #print ("failed to drop")
#try:
    #create_tables(curs)
    #conn.commit()
    #print ("Created tables")
#except:
    #conn.rollback()
    #print ("failed to create")
print(a)
try:
    print("Vorname bitte eingeben: ")
    variable1=input()
    print("Nachname eingeben bitte:")
```

```
        variable2=input()
        print("email eingeben bitte:")
        variable3=input()
        print("Role eingeben bitte:")
        variable4=int(input())
        insert_values1(curs,variable1,variable2,variable3,variable4)
        conn.commit()
        os.system('./fehlerlog_Gesichtsregistrierung_Personregistration.py')
        a="inserted values"
        print (a)
        #print ("inserted values")
except:
        conn.rollback()
        os.system('./fehlerlog_Gesichtsregistrierung_notPersonregistration.py
        a="failed to insert"
        #print("failed to insert")
#try:
 #   for row in read_values(curs):
  #      print ('Rei')
  #  print ("Read values")
#except:
    # conn.rollback()
    # print ("failed to read values")

#!/usr/bin/python3.5
import RPi.GPIO as GPIO
import time
import os
import subprocess
import sys
from getpass import getpass
#from Person_Registrierung import (variable3)
#os.system("./enter.sh")
GPIO.setmode(GPIO.BCM)

GPIO.setup(23, GPIO.OUT)#AronLed
GPIO.setup(18, GPIO.IN)#AronSchalter
GPIO.setup(17, GPIO.IN)#ReiLED
GPIO.setup(27, GPIO.OUT)#ReiSchalter
a=""
b=""
correctpaswadmin="@r0nt3rz()"
#image2=sys.argv[1]
#foto2=sys.argv[1]
while True:
        if GPIO.input(18):
                os.system('./fehlerlog_Gesichtsregistrierung.py')
```

```
            print("Password for admin please :")
            inputpasw=getpass()
            if(inputpasw == correctpaswadmin):
                os.system('./fehlerlog_Gesichtsregistrierung_admin.py')
            #os.system('./Test '+(image2)+".jpg")
            #exec(open('./Vergleich_2_Fotos.py').read())
            #if(c=="matched"):
            #os.system('./connection.py')
                exec(open('Person_Registrierung.py').read())
                print(variable3)
                if (a=="inserted values"):
                    os.system('./Test '+(variable3)+".jpg")
                    #os.system('./selectID_UpdateBild.py '+(variable3))
                    os.system('./68fLandmarks.py %s'%(variable3))
                    GPIO.output(23,GPIO.HIGH)
                    time.sleep(1.5)
                    GPIO.output(23,GPIO.LOW)
                #exit()
                else:
                    print("Fehler")
                    GPIO.output(23,GPIO.LOW)
            #exit()
            else:
                print("Admin ist nicht da")
        else:
            #os.system('./connection.py')
            #GPIO.output(23,GPIO.LOW)
            print("Schalter fuer Gesichtsregistrierung nicht gedrueckt")
            time.sleep(0.5)
            #exit()
        if GPIO.input(17):
            os.system('./Log_Erkennung_TasterGed.py')
            exec(open('reitest3.py').read())
            if b=="existiert" and skaftyra=="face":
                GPIO.output(27,GPIO.HIGH)

                time.sleep(1.5)
                GPIO.output(27,GPIO.LOW)
            else:
                print("nicht existiert")
                GPIO.output(27,GPIO.LOW)
        else:
            print("Schalter fuer Gesichtserkennung nicht gedrueckt")
            time.sleep(0.5)

#!/usr/bin/python3
```

```
import MySQLdb
import sys
import os
import subprocess
variable3=sys.argv[1]
conn=MySQLdb.connect('localhost','aronterzeta','aronterzeta','test')
mycursor=conn.cursor()
conn.close()

#!/usr/bin/python3

import cv2
import numpy as np
import dlib
import sys
import faceDetect_crop_rei as fdcr
emailiperkrahasim=fdcr.emaili
nofaces=""
print(emailiperkrahasim)
def getPoints(Imagename):
    face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default
    image = cv2.imread(Imagename)
#image=cv2.imread('aroncropped2.jpg')
    if(image is None):
        print("Can't open image file")

#get image dimensions
    dimX = int(image.shape[0])
    dimY = int(image.shape[1])
    print("\tdimensionet e fotos",dimX,dimY)

#convert image into grayscale
    gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)


#load shape predictors to extract landmarks

    predictor = dlib.shape_predictor("shape_predictor_68_face_landmarks.d
    detector = dlib.get_frontal_face_detector()

#load face detectors and detect faces
    faces_cv = face_cascade.detectMultiScale(gray,1.1,4)
    faces = detector(gray)

#get number of faces detected
    nrFace = len(faces)
    print("\tDetected faces: %d" % nrFace)
```

```
    i = 0
    coords=[]
    z=[]
    vleratx1=np.zeros((68,1),dtype="float")
    vleraty1=np.zeros((68,1),dtype="float")
#get image dimensions
    height, width = image.shape[:2]


    if nrFace > 0:
        nofaces="face"
        for face in faces:

            x1 = face.left()
            y1 = face.top()
            x2 = face.right()
            y2 = face.bottom()


            i = 0
        #problemi!!
            landmarks = predictor(gray,face)
        #shape = shape_utils.shape_to_np(landmarks)
            coords = np.zeros((68,2),dtype="float")
        #right eye
            for d in range(0,68):
                x = float(landmarks.part(d).x / width)
                y = float(landmarks.part(d).y / height)
            #cv2.circle(gray2, (x, y), 4, (255,0,0), -1)
                i+1
                coords[d] = (x,y)
            #for index in range(len(coords[d]):
        #print(d,coords)
            #print(d,": ",x,y,"\n")
            z=np.hsplit(coords,2)
            vleraty1=z[1]
            vleratx1=z[0]
    elif nrFace <= 0:
        print("\tno faces found")

    ret = dict()
    ret['x'] = vleratx1
    ret['y'] = vleraty1

    return ret
```

```
vlera=getPoints(fdcr.filenampererkennung)

#!/usr/bin/python3
import cv2
import MySQLdb
import sys
import os
import math
import numpy as np
import dlib
import reitest as rei
conn=MySQLdb.connect('localhost','aronterzeta','aronterzeta','test')
mycursor=conn.cursor()
skaftyra=rei.nofaces
b=""
bool=False
bool=False
myresult=""
maxwert=0
counter=0
mycursor.execute("select * from person where email='%s';"%(rei.emailiperk
global myresult
myresult=mycursor.fetchall()
MAX = [0,0,0,0,0]
for x in myresult:
    if x[2]==rei.emailiperkrahasim:
        bool=True
    else:
        bool=False
        os.system('./Log_Erkennung_Email_Nicht_Gefunden.py')

#except:
    #print ("Select Statement nicht gut")
if bool:
    #os.system('./Log_Erkennung_PersonExistiert.py')
    for res in myresult:
        print(res[5])
        vx=5
        vy=6
        #ret = landmarks.getPoints(arrayFoto[f])
        vleratx = rei.vlera['x']
        vleraty = rei.vlera['y']
        min=10.0
        max=0.0
        summe=0.0
        gm=1.0
        dis=0.0
```

```
print("----------Ftyra Anash----------

for pika in range(0,27):
    dis=math.sqrt(abs(( (vleratx.item(pika)-float(res[vx])))) * a
    if(dis < min):
        min = dis
    if(dis > max):
        max = dis
    gm *= dis
    vx+=2
    vy+=2
gm=gm**(1/27)
global MAX
MAX[0]=max
print("Max:%s\t GM:%s\n"%(MAX[0],gm))
min=10.0
max=0.0
summe=0
gm=1
dis=0

print("----------Hunda----------

for pika in range(27,36):
    dis=math.sqrt(abs(( (vleratx.item(pika)-float(res[vx])))) * a
    if(dis < min):
        min = dis
    if(dis > max):
        max = dis
    gm *= dis
    vx+=2
    vy+=2
gm=gm**(1/9)
#global MAX2
#MAX2=max
global MAX
MAX[1]=max
print("Max:%s\t GM:%s\n"%(MAX[1],gm))
min=10.0
max=0.0
summe=0
gm=1
dis=0

print("----------Syni Djatht----------
```

```
for pika in range(36,42):
    dis=math.sqrt(abs(( (vleratx.item(pika)-float(res[vx])))) * a
    if(dis < min):
        min = dis
    if(dis > max):
        max = dis
    gm *= dis
    vx+=2
    vy+=2
gm=gm**(1/6)
global MAX
MAX[2]=max
print("Max:%s\t GM:%s\n"%(MAX[2],gm))
min=10.0
max=0.0
summe=0
gm=1
dis=0

print("-------------------------------------Syni Majt-------------------------

for pika in range(42,48):
    dis=math.sqrt(abs(( (vleratx.item(pika)-float(res[vx])))) * a
    if(dis < min):
        min = dis
    if(dis > max):
        max = dis
    gm *= dis
    vx+=2
    vy+=2
gm=gm**(1/6)
global MAX
MAX[3]=max
print("Max:%s\t GM:%s\n"%(MAX[3],gm))
min=10.0
max=0.0
summe=0
gm=1
dis=0

print("-------------------------------------Goja-------------------------

for pika in range(48,68):
    dis=math.sqrt(abs(( (vleratx.item(pika)-float(res[vx])))) * a
    if(dis < min):
        min = dis
```

```
        if(dis > max):
            max = dis
        gm *= dis
        vx+=2
        vy+=2
gm=gm**(1/20)
global MAX
MAX[4]=max
print("Max:%s\t GM:%s\n"%(MAX[4],gm))
min=10.0
max=0.0
summe=0
gm=1
dis=0
print("-----------------------------------------Total-----------------------------
kx=5
ky=6
for pika in range(0,68):
        dis=math.sqrt(abs(( (vleratx.item(pika)-float(res[kx])))) * a
        if(dis < min):
            min = dis
        if(dis > max):
            max = dis
        gm *= dis
        kx+=2
        ky+=2
gmT=gm**(1/68)
global MAX6
MAX6=max
print("Max:%s\t GM:%s\n"%(MAX6,gmT))

#1)  Mindestens 2 von 5 Maximumwerte sind groesser als 0.06:
global count
count = 0
for i in range(0,len(MAX)):
        if MAX[i] > 0.06:
            count+=1
status1=True
status11=True
status2=True
status22=True
#  Wenn JA:     Nur wenn gmT kleiner als 0.01 sind sie richtig, so
if count ==1:
        global status11
        status11=False
if count >= 2:
```

```
        global status22
        status22=False
        if gmT > 0.01:
            global status1
            status1 = False
     #    Wenn NEIN:  gmT muss kleiner als 0.027 sein,um RICHTIG zu se
    else:
        if gmT > 0.027:
            global status2
            status2 = False

# 2)  Eine MAXwert groesser als 0.07:
    count = 0
    for i in range(0,len(MAX)):
        if MAX[i] > 0.07 and MAX[i]< 0.084:
            count+=1
    status3=True
    status33=True
    status4=True
    #         Wenn JA:     Wenn gmT kleiner als 0.027 ist passt, sonst
    if count == 1:
        global status33
        status33=False
        if gmT > 0.027:
            global status3
            status3 = False
     #         Wenn 2 oder mehr: FALSCH
    elif count > 1:
        global status4
        status4 = False


    #3)  Eine oder mehrere MAXwert groesser als 0.084:
    for i in range(0,len(MAX)):
        if MAX[i] > 0.084:
            count+=1
    status5=True
     #  Wenn JA: FALSCH
    if count >=1:
        global status5
        status5=False
    #4)  Wenn Case 1&2 oder 2&3 eintretten:
     #         Falsch
    if status1==False or status2==False or status3==False or status4=
    #if (status1==False and status2==False) or (status2==False and st
        print("Vergleich nicht erfolgreich, sie sind nicht eingeloggt
```

```
            os.system('./Log_Erkennung_Person_Existiert_Nicht.py')
        else:
            b="existiert"
            print("Vergleich erfolgreich, Sie sind eingeloggt!!!")
            os.system('./Log_Erkennung_Person_Existiert.py')

else:
    b="nicht existiert"
    print("FEHLER!!!!")
```

```
#!/usr/bin/python3
import MySQLdb
import time
import logging
db_server='localhost'
db_user='aronterzeta'
db_password='aronterzeta'
db_dbname='test'
db_tbl_log='log'
log_file_path='/home/pi/diplomarbeit/python_scripts/fehlerlog.txt'
log_error_level='DEBUG'
log_to_db=True

class LogDBHandler(logging.Handler):
    def __init__(self,sql_conn,sql_cursor,db_tbl_log):
        logging.Handler.__init__(self)
        self.sql_cursor=sql_cursor
        self.sql_conn=sql_conn
        self.db_tbl_log=db_tbl_log
    def emit(self,record):
        tm=time.strftime("%Y-%m-%d %H:%M:%S", time.localtime(record.creat
        self.log_msg=record.msg
        self.log_msg=self.log_msg.strip()
        self.log_msg=self.log_msg.replace('\'', '\'\'')
        sql='insert into '+self.db_tbl_log + ' (log_level, ' + \
            'log_levelname,log,created_at,created_by) ' + \
            'values (' + \
            '' + str(record.levelno) + ', ' + \
            '\'' + str(record.levelname) + '\', ' + \
            '\'' + str(self.log_msg) + '\', ' + \
            ' \'' + tm + '\', ' + \
            '\'' + str(record.name) + '\')'
        try:
            self.sql_cursor.execute(sql)
            self.sql_conn.commit()
        except MySQLdb.Error as e:
            print (sql)
```

```
            print("Critical DB Error")
if(log_to_db):
    log_conn=MySQLdb.connect(db_server,db_user,db_password,db_dbname,30)
    log_cursor=log_conn.cursor()
    logdb=LogDBHandler(log_conn,log_cursor,db_tbl_log)

logging.basicConfig(filename=log_file_path)
if(log_to_db):
    logging.getLogger('').addHandler(logdb)
log=logging.getLogger('Gesichtserkennung')
log.setLevel(log_error_level)
test_var='Email existiert nicht!'
log.error('This event occurred: %s' % test_var)
```

```
#!/usr/bin/python3
import MySQLdb
import time
import logging
db_server='localhost'
db_user='aronterzeta'
db_password='aronterzeta'
db_dbname='test'
db_tbl_log='log'
log_file_path='/home/pi/diplomarbeit/python_scripts/fehlerlog.txt'
log_error_level='DEBUG'
log_to_db=True

class LogDBHandler(logging.Handler):
    def __init__(self,sql_conn,sql_cursor,db_tbl_log):
        logging.Handler.__init__(self)
        self.sql_cursor=sql_cursor
        self.sql_conn=sql_conn
        self.db_tbl_log=db_tbl_log
    def emit(self,record):
        tm=time.strftime("%Y-%m-%d %H:%M:%S", time.localtime(record.creat
        self.log_msg=record.msg
        self.log_msg=self.log_msg.strip()
        self.log_msg=self.log_msg.replace('\'', '\'\'')
        sql='insert into '+self.db_tbl_log + ' (log_level, ' + \
            'log_levelname,log,created_at,created_by) ' + \
            'values (' + \
            '' + str(record.levelno) + ', ' + \
            '\'' + str(record.levelname) + '\', ' + \
            '\'' + str(self.log_msg) + '\', ' + \
            ' \'' + tm + '\', ' + \
            '\'' + str(record.name) + '\')'
        try:
```

```
            self.sql_cursor.execute(sql)
            self.sql_conn.commit()
        except MySQLdb.Error as e:
            print (sql)
            print("Critical DB Error")
if(log_to_db):
    log_conn=MySQLdb.connect(db_server,db_user,db_password,db_dbname,30)
    log_cursor=log_conn.cursor()
    logdb=LogDBHandler(log_conn,log_cursor,db_tbl_log)

logging.basicConfig(filename=log_file_path)
if(log_to_db):
    logging.getLogger('').addHandler(logdb)
log=logging.getLogger('Gesichtserkennung')
log.setLevel(log_error_level)
test_var='Person wurde nicht gefunden, Vergleichung stimmt nicht'
log.error('This event occurred: %s' % test_var)
```

```
#!/usr/bin/python3
import MySQLdb
import time
import logging
db_server='localhost'
db_user='aronterzeta'
db_password='aronterzeta'
db_dbname='test'
db_tbl_log='log'
log_file_path='/home/pi/diplomarbeit/python_scripts/fehlerlog.txt'
log_error_level='DEBUG'
log_to_db=True

class LogDBHandler(logging.Handler):
    def __init__(self,sql_conn,sql_cursor,db_tbl_log):
        logging.Handler.__init__(self)
        self.sql_cursor=sql_cursor
        self.sql_conn=sql_conn
        self.db_tbl_log=db_tbl_log
    def emit(self,record):
        tm=time.strftime("%Y-%m-%d %H:%M:%S", time.localtime(record.creat
        self.log_msg=record.msg
        self.log_msg=self.log_msg.strip()
        self.log_msg=self.log_msg.replace('\'', '\'\'')
        sql='insert into '+self.db_tbl_log + ' (log_level, ' + \
            'log_levelname,log,created_at,created_by) ' + \
            'values (' + \
            '' + str(record.levelno) + ', ' + \
            '\'' + str(record.levelname) + '\', ' + \
```

```
                '\'' + str(self.log_msg) + '\', ' + \
                ' \'' + tm + '\', ' + \
                '\'' + str(record.name) + '\')'
            try:
                self.sql_cursor.execute(sql)
                self.sql_conn.commit()
            except MySQLdb.Error as e:
                print (sql)
                print("Critical DB Error")
if(log_to_db):
    log_conn=MySQLdb.connect(db_server,db_user,db_password,db_dbname,30)
    log_cursor=log_conn.cursor()
    logdb=LogDBHandler(log_conn,log_cursor,db_tbl_log)

logging.basicConfig(filename=log_file_path)
if(log_to_db):
    logging.getLogger('').addHandler(logdb)
log=logging.getLogger('Gesichtserkennung')
log.setLevel(log_error_level)
test_var='Person existiert schon, Vergleichung stimmt'
log.error('This event occurred: %s' % test_var)
```

```
#!/usr/bin/python3
import MySQLdb
import time
import logging
db_server='localhost'
db_user='aronterzeta'
db_password='aronterzeta'
db_dbname='test'
db_tbl_log='log'
log_file_path='/home/pi/diplomarbeit/python_scripts/fehlerlog.txt'
log_error_level='DEBUG'
log_to_db=True

class LogDBHandler(logging.Handler):
    def __init__(self,sql_conn,sql_cursor,db_tbl_log):
        logging.Handler.__init__(self)
        self.sql_cursor=sql_cursor
        self.sql_conn=sql_conn
        self.db_tbl_log=db_tbl_log
    def emit(self,record):
        tm=time.strftime("%Y-%m-%d %H:%M:%S", time.localtime(record.creat
        self.log_msg=record.msg
        self.log_msg=self.log_msg.strip()
        self.log_msg=self.log_msg.replace('\'', '\'\'')
        sql='insert into '+self.db_tbl_log + ' (log_level, ' + \
```

```
                'log_levelname,log,created_at,created_by) ' + \
                'values (' + \
                '' + str(record.levelno) + ', ' + \
                '\'' + str(record.levelname) + '\', ' + \
                '\'' + str(self.log_msg) + '\', ' + \
                ' \'' + tm + '\', ' + \
                '\'' + str(record.name) + '\')'
            try:
                self.sql_cursor.execute(sql)
                self.sql_conn.commit()
            except MySQLdb.Error as e:
                print (sql)
                print("Critical DB Error")
if(log_to_db):
    log_conn=MySQLdb.connect(db_server,db_user,db_password,db_dbname,30)
    log_cursor=log_conn.cursor()
    logdb=LogDBHandler(log_conn,log_cursor,db_tbl_log)

logging.basicConfig(filename=log_file_path)
if(log_to_db):
    logging.getLogger('').addHandler(logdb)
log=logging.getLogger('Gesichtserkennung')
log.setLevel(log_error_level)
test_var='Taster gedruckt'
log.error('This event occurred: %s' % test_var)
```

```
#!/usr/bin/python3.5
import RPi.GPIO as GPIO
import time
import os
import subprocess
import sys
from getpass import getpass
#from Person_Registrierung import (variable3)
#os.system("./enter.sh")
GPIO.setmode(GPIO.BCM)

GPIO.setup(23, GPIO.OUT)#AronLed
GPIO.setup(18, GPIO.IN)#AronSchalter
GPIO.setup(17, GPIO.IN)#ReiLED
GPIO.setup(27, GPIO.OUT)#ReiSchalter
a=""
b=""
correctpaswadmin="@r0nt3rz()"
#image2=sys.argv[1]
#foto2=sys.argv[1]
while True:
```

```
if GPIO.input(18):
    os.system('./fehlerlog_Gesichtsregistrierung.py')
    print("Password for admin please :")
    inputpasw=getpass()
    if(inputpasw == correctpaswadmin):
        os.system('./fehlerlog_Gesichtsregistrierung_admin.py')
    #os.system('./Test '+(image2)+".jpg")
    #exec(open('./Vergleich_2_Fotos.py').read())
    #if(c=="matched"):
    #os.system('./connection.py')
        exec(open('Person_Registrierung.py').read())
        print(variable3)
        if (a=="inserted values"):
            os.system('./Test '+(variable3)+".jpg")
            #os.system('./selectID_UpdateBild.py '+(variable3))
            os.system('./68fLandmarks.py %s'%(variable3))
            GPIO.output(23,GPIO.HIGH)
            time.sleep(1.5)
            GPIO.output(23,GPIO.LOW)
        #exit()
        else:
            print("Fehler")
            GPIO.output(23,GPIO.LOW)
    #exit()
    else:
        print("Admin ist nicht da")
else:
    #os.system('./connection.py')
    #GPIO.output(23,GPIO.LOW)
    print("Schalter fuer Gesichtsregistrierung nicht gedrueckt")
    time.sleep(0.5)
    #exit()
if GPIO.input(17):
    os.system('./Log_Erkennung_TasterGed.py')
    exec(open('reitest3.py').read())
    if b=="existiert" and skaftyra=="face":
        GPIO.output(27,GPIO.HIGH)

        time.sleep(1.5)
        GPIO.output(27,GPIO.LOW)
    else:
        print("nicht existiert")
        GPIO.output(27,GPIO.LOW)
else:
    print("Schalter fuer Gesichtserkennung nicht gedrueckt")
    time.sleep(0.5)
```