

Höhere technische Schule fur Informationstechnologie
Shkolla e mesme profesionale private pér 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, 14. Februar 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 institucionit 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	Projektmanagementmethode	4
3	Gesichtsregistrierung	6
3.1	Umsetzung	6
3.1.1	Allgemein	6
3.1.2	Technische Lösung	7
3.1.3	Herausforderungen	12
3.1.4	Qualitätssicherung und Controlling	14
3.2	Ergebnisse	15
3.2.1	Implementierung	15
4	Umsetzung - Jordi	17
4.1	Allgemeine Beschreibung	17
4.2	Technische Lösungen	20
4.2.1	Datenbank	20
4.3	Herausforderungen	23
4.3.1	Datenbank - MariaDB	23
4.3.2	Python Scripting	23
4.3.3	OpenCV	24
4.4	Qualitätssicherung	24
5	Ergebnisse - Jordi	26
5.1	Datenbank	26

5.2 Tieferkennung	26
6 Bildverarbeitung - Egli	27
6.1 Allgemeines	27
6.1.1 Entwicklungsumgebung und Technologien	27
6.1.2 Frameworks und Bibliotheken	28
6.2 Technische Lösungen	30
6.2.1 Lösungsweg - Structed Software design	31
6.2.2 Gesichtsdetektion	32
6.2.3 Gesichtsdetektion und Zuschneiden	35
6.2.4 Gesichtsschlüsselpunkte Extraktion	35
6.3 Herausforderungen, Probleme und deren Lösung	38
6.4 Qualitätssicherung, Controlling	39
6.5 Ergebnise	39
7 Gesichtserkennung - Rei	41
7.1 Allgemeines	41
7.2 Technische Lösung	42
7.2.1 Hardware und Aufbau	42
7.2.2 Software	45
7.3 Herausforderungen, Probleme, und wie wurden sie gelöst	54
7.4 Projektmanagement und Controlling	54
7.5 Ergebnisse	56
8 Planung vs Realisierung	57
9 Evaluierung und Resümee	60
9.1 Wertschöpfung und Lessons Learned	60

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 Tiefeinschärfe 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 würde 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).

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

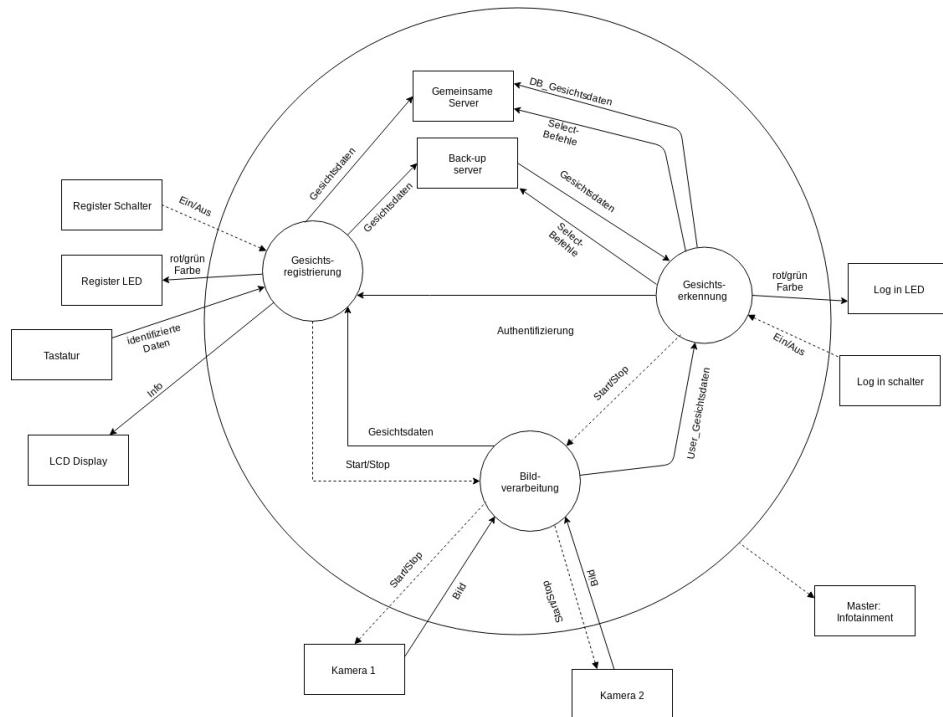


Abbildung 2.1: Big picture

2.3 Projektmanagementmethode

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

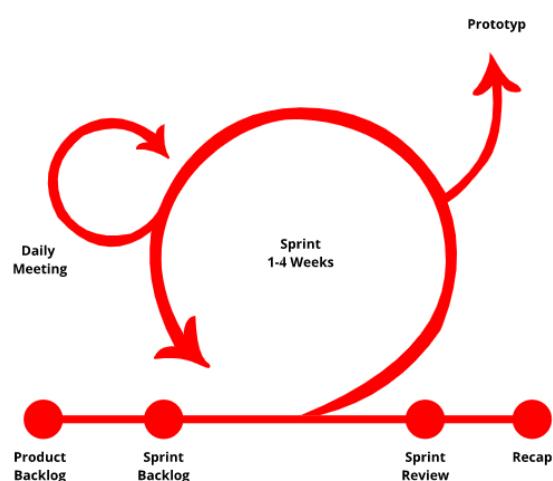


Abbildung 2.2: Scrum

Kapitel 3

Gesichtsregistrierung

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

3.1 Umsetzung

Hier wird erklärt, wie es gedacht ist, den Gesichtsregistrierungsteil zu implementieren.

3.1.1 Allgemein

Sicherheit ist heutzutage hoch interessant und relevant in mehreren technischen und nicht technischen Bereichen. Sicherheit ist eigentlich relativ, niemand weißt, ob er sicher ist oder nicht. Aber es gibt Systeme, bzw. Geräte, Personen usw., die Sicherheit garantieren. 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 identifiziert, weil es einzigartig ist. Das Gesicht hat Daten, die von verschiedenen Algorithmen herausgeholt werden können, und sie für Realisierung der Überprüfung und Identifizierung der Personen verwenden zu können.

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 gemacht.

Das andere Paket namens „git“ ist für das System nicht notwendig, aber git könnte als eine Backup-Strategie verwendet, wenn das System abstürzt. Git ist ein Versionsverwaltungssystem, das verschiedene Versionen bzw. Commits auf einem Github-Server speichert. Auf dem Github-Server gibt dann verschiedene Versionen des Systems und die Daten werden von einem bestimmten Commit dann zurückgeholt. Es wird meistens bei der Implementierung-Phase verwendet, um die Veränderungen der Source-Code, wann geändert hat, wer geändert hat, deutlich zu sehen. Cmake ist ein Paket, das gebraucht wird, wenn das System mit OpenCV-Framework arbeitet wird. 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 es 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.” [18]

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 nötige Paketen, damit das OpenCV-Framework eigentlich verwendet kann.

3.1.2 Technische Lösung

Technologien, die ich für die Implementierung verwendet habe sind:

- Linux als Betriebssystem [32]

Linux ist das weit 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. welches System-Grafiken angezeigt werden, bzw. die ganzen Komponenten der Benutzeroberfläche. Warum ich Linux gewählt habe, gibt es verschiedene Gründe. Linux ist für eingebettete Systeme sehr geeignet. Es ist sicher gegen Schadprogrammen, Viren, Trojanern. Linux ist einfacher. Vorher war ein kompliziertes System, jetzt seit den Bemühungen der Ubuntu-Fundationen und der Ubuntu-Distribution ist es sehr einfach verwendbar.

- Python ”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.” [22] Warum Python gewählt wurde, hat verschiedene Gründe. Python hat weniger Schlüsselwörter, reduziert die Syntax auf das Wesentliche und optimiert die Sprache. Ein Programm, das in Python geschrieben ist, ist vom Betriebssystem unabhängig. Das bedeutet, sie können Plattform unabhängig interpretiert werden. Python hat auch eine gute Lesbarkeit.

Das System besteht aus verschiedenen 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 diese.” [35]

Der wichtigste Terminator ist der ”Register-Schalter”. Er initialisiert das ganze Programm. Schalter in Technik ist nichts anders, 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 und gibt dann einen Output. Das System ist sehr einfach verwendbar.

Das Register-LED dient als einen Anzeiger. Wenn mit dem System etwas nicht stimmt, z.B. nicht richtige Inputdaten, dann wird mit einer bestimmten Farbe geleuchtet, nämlich mit rot. Wenn etwas passt, dann wird mit grün geleuchtet. Eigentlich das

normale LED hat nur eine Farbe, aber es wird ein spezielles LED verwendet, namens RGB LED. RGB LED hat drei Grundfarben, rot, grün, blau, und 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 schreiben müssen. Die andere spezielle Eigenschaft ist die Verwendung eines LCD-Screens. Da werden z.B. Errors gezeichnet, die Daten, die in Log gespeichert sind usw. Es ist leicht auch für den Benutzer zu sehen, dass es z.B. ein Problem mit dem System gibt, damit er nicht vor der Kamera warten muss. Eigentlich die Hauptfunktion des LCD-Screens ist, alles was der Benutzer mit der Tastatur schreibt, da zeigen zu lassen. Warum es so geplant ist? Das Problem steht daran, wenn der Benutzer seine Email schreibt, kann er Fehler machen, weil er nicht sieht, was er schreibt. Um das zu vermeiden, wird das LCD-Screen verwendet, damit der Benutzer sehen kann, was er schreibt.

Um registrierte Personen mit ihren Gesichtsdaten zu speichern, braucht das System einen Server. In diesem Server läuft ein Datenbank Management System, in dem eine Datenbank erstellt ist. Die Datenbank ist so konfiguriert, damit die Person mit ihren Infos gespeichert werden können. Um die Verbindung zwischen System und Server zu ermöglichen, wird das Paket „python-mysqldb“ verwendet.

Das System hat auch einen Backup-Server. Die Daten werden parallel bei dem Hauptserver sowie bei dem Backup-Server gespeichert, damit die Daten noch gesichert sind, wenn der Hauptserver ein Problem hat. Die Verwendung des Backup-Servers ist zustande gekommen, weil das System 24/7 arbeiten muss, und wenn der Hauptserver Maintenance oder Probleme hat, der Backup-Server arbeiten kann. Auf dem Abb. 3.1 können Sie in einem technischen Weg besser sehen, wie der Gesichtsregistrierung-Teil arbeitet.

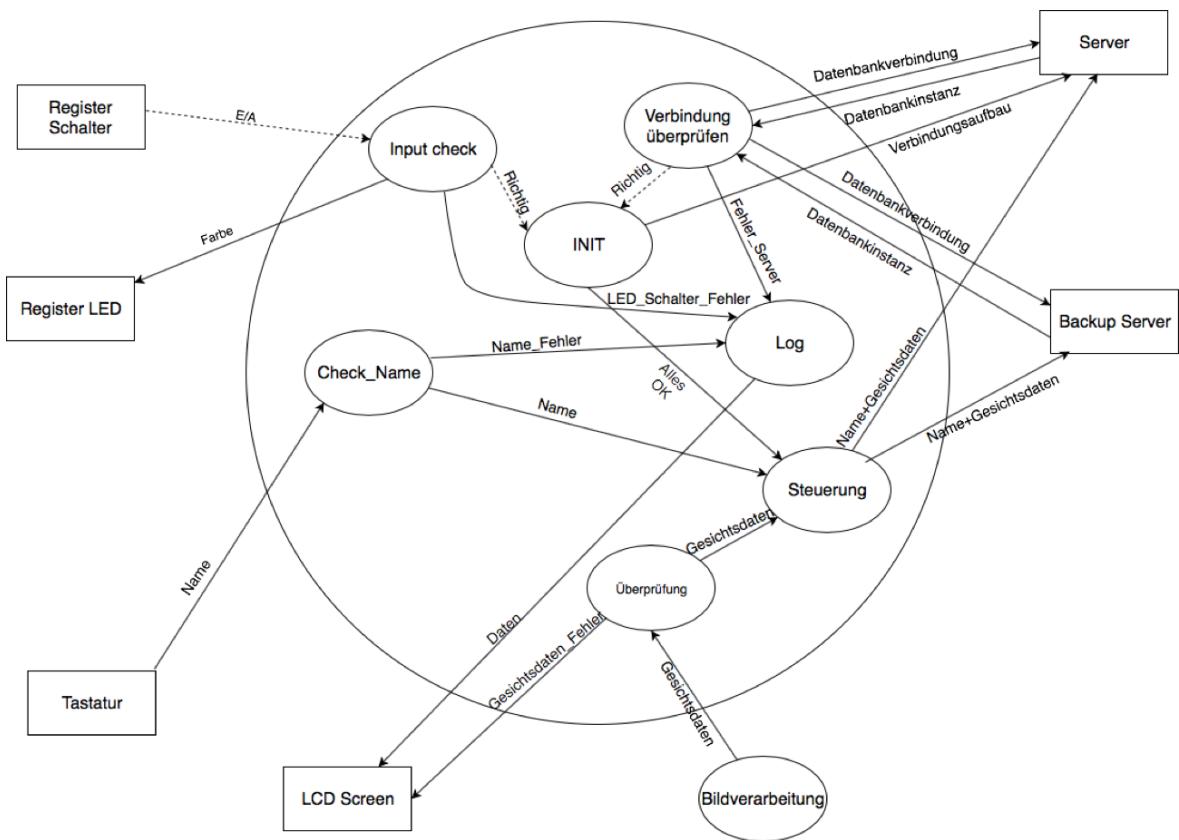


Abbildung 3.1: Structed Software Design bzw. erste Ebene

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 HW (LED und Schalter) verbinden zu können. Dafür werden GPIOs verwendet. Der Schalter hat 3 Beine. Eines wird mit 5V verbunden, das andere mit Ground und das andere ist für Daten. Dies wird dann mit einem GPIO-Port in Raspberry PI verbunden. Das gleiche ist auch für die LED, damit es von Raspberry PI kontrollieren verwendet kann, wird mit einem GPIO-Port verbunden. Mithilfe dieser GPIO-Ports bekommt das System zurück, wenn der Schalter gedrückt wird. Wert „1“ ist der Schalter gedrückt und werden dann die verschiedenen Skripten aufgerufen.

Schritte:

1. Am Beginn des Skripts diese Zeile schreiben: ”#!/usr/bin/python”. Es gibt zwei Gründe, warum diese Zeile geschreibt wird. Der erste Grund ist, dass dieses Programm mit einem Python-interpreter ausgeführt wird, der zweite ist, Verwendung des Programmsuchpfads, um es zu finden.

2. Alle Paketen importieren. Sehen Sie auf Abb. 3.2

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

Abbildung 3.2: Packages zu importieren im Main-Skript

- RPi.GPIO ist ein Paket, das verwendet wird, um Zugriff auf die sogennannten GPIO-Ports zu haben. Vorher habe ich erwähnt, 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. [38]
- time ist ein Paket in Python. Von diesem Paket wird nur die Funktion 'sleep' verwendet. Diese Funktion pausiert das python-Programm. [26]
- os ist das wichtigste Paket in unserem Skript. Es erlaubt mir, dass ich in einem Python-Skript andere Skripten aufrufen kann. Es ist egal, in welcher Programmiersprache diese Skripten geschrieben sind. Es gibt auch verschiedene Methoden, wie man verschiedene Skripten in einem Python-Skript aufrufen kann. Man macht mit dem subprocess Paket, eine Main-Funktion in dem Skript machen und hier die verschiedenen Funktionen des anderen Skripts aufrufen.
- subprocess Paket dient zur Verbindung zwischen verschiedenen Prozessen, in meinem Fall, ein Prozess heit, ein Aufruf eines Skriptes, aber wird nicht im Skript verwendet.
- Das sys Paket wird verwendet, um Console Parameter zu geben. Das bedeutet, wenn der Skript aufgerufen wird, z.B. login.py dann nach dem login.py gebe ich einen Parameter mit login.py <parameter >

3. GPIO-Ports direction einrichten. Direction für LED ist 'out', weil das LED als ein Output für unseres System dient. Direction für Schalter ist 'in', weil der Schalter als ein Input für unseres System dient. Auf Abb. 3.3 ist auch der Python-Code.

```
GPIO.setmode(GPIO.BCM)
GPIO.setup(23, GPIO.OUT) #Ansteuern LED
GPIO.setup(18, GPIO.IN) #Ansteuern Schalter
GPIO.setup(17, GPIO.IN) #Ansteuern LED
GPIO.setup(27, GPIO.OUT) #Ansteuern Schalter
```

Abbildung 3.3: GPIO-Ports Konfiguration

Es gibt verschiedene Betriebsarten für GPIO wie BCM und Board. Ich verwende BCM (Broadcom Pin Number), weil ich das Paket RPi.GPIO verwende. Mit diesem Paket darf nur die Betriebssart 'BCM' verwendet werden. [27] Für die Registrierung der Schüler und Schülerinnen bzw. Lehrer und Lehrerinnen ist es nötig, dass der Admin physisch da ist. 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, indem ich das Skript, das Bild macht, aufrufe, und dann vergleiche ich mit einem anderen Skript diese beide Bilder. Es gibt 'matched' zurück, wenn die Gesichte bei den beiden Bildern gestimmt haben und 'not matched' wenn die Gesichte nicht gestimmt haben.

4. Dann kommt der Teil "Input check". Hier dann verwende ich 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 die Port des Schalters ist IN und für die Port des LEDs ist OUT. Mit der Methode 'input' hole ich das Wert (value) der Schalter-Port. Wenn der Schalter gedrückt wird, wird der Wert '1' herausgekommen und 1 repräsentiert 'true'. Das bedeutet, Input-methode liefert 'true' zurück 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 physisch da ist. Ist der Admin da, können die Personen mit der Registrierung beginnen. Diese Person wird nach ihrem Vornamen, Nachnamen, Email und Rolle gefragt. Die Rolle schreibt der Admin. 1 ist für Admin, 2 für Schüler und 3 für Lehrer und Lehrerinnen. Die E-Mail speichere ich dann in einer Variable und diese Variable übergebe ich dann bei einem anderen Skript. Dieses Skript dann 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 E-Mail einfacher ist.

Abbildung 3.4: Beispiel in Python, wie die Personen registriert werden

```
def insertPath( mycursor ):
    mycursor.execute(" select idP from person where email=%s ;" %( variable3 ))
    myresult=mycursor.fetchall()
    for x in myresult:
        var1=x[0]
        mycursor.execute(" insert into info(imagePath,idP) values ('%s','%s');" %('./','
```

Es wird die E-Mail verwendet, weil es eine performantere Suche in der Datenbank ermöglicht. Die E-Mail ist einzig, nur einmal für jede Person und die SQL-Anweisung (select-Abfrage) eine Selektion durchführt. Von 1000 Datensätze wird nur der Datensatz ausgegeben, in dem die E-Mail mit der eingetippten E-Mail vom Benutzer übereinstimmt.

Anschließend, wenn die Person in der Tabelle 'person' gespeichert ist, hole ich ID dieser Person und füge dann diese ID mit der E-Mail und Path des Bildes in der Tabelle 'info' ein. Sehen Sie auf Abb. ??, wo einen Code in Python dargestellt ist.

3.1.3 Herausforderungen

Eigentlich hatte das Projekt für mich viele Herausforderungen. Die Gründe dafür sind, weil es ein ziemlich großes Projekt ist, haben wir neue Technologien verwendet, die wir vorher nie verwendet haben. Ich habe keine Erfahrung z.B. mit OpenCV, Python und viele verschiedene Dinge, die ich später erwähnen werde. Ich habe von diesen Herausforderungen und Problemen viel gelernt. Einerseits bin ich froh, anderseits bin ich wütend, weil das Datum des Ende des Projekts verzögert ist. Die Herausforderungen waren:

- opencv zu installieren. Das war eigentlich die größte Herausforderung. Es hat mir 3 Woche gedauert, bis ich es installiert habe.
- Beginn des Projektes. Immer der Beginn eines Projektes ist 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 eine richtige große Herausforderung
- Git repository, Einrichtung von git. Manche von den Teammitgliedern 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 committed haben. Ich sollte alle diese lösen, weil ich mehr Erfahrung mit git hatte.
- Python als Programmiersprache. Wir wollten vorher mit C++ es machen, 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 ist nicht gegangen, deshalb sind wir zu Python gewechselt. Wir haben Python gewählt, weil opencv in Python sehr einfach installierbar war. Mit Python hatten wir keine große Erfahrung. Das Maximum, was ich mit Python gemacht habe ist, eine Verbindung mit der Datenbank und Statements schicken (Insert,Select,Update,Delete). Alle andere Wissen sollte ich selbst von Büchern, 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 Datenbank Management System(MySQL) verbindet, 'python-mysqldb' und jetzt heißt es 'python-mariadb'. Ich wusste das nicht und hat mir ein bisschen Zeit gekostet.
- Abhängigkeiten zwischen einzelnen Arbeitsteilen. Die Aufgaben sind so geteilt, dass sie Abhängigkeiten zwischen einander liegen. Das hat dann zu einer Verspätung der Projektabgabe geführt, weil jeder Teammitglieder aufeinander warten mussten.
- Was ich geplant habe, hat nicht gut funktioniert. Die großen Teile meiner Planung haben gepasst, nur wenige Kleinigkeiten musste ich ändern. Sie sind erst in der Implementierungsphase angezeigt.

Ich habe diese Lösungen für die Herausforderungen gefunden:

1. Ich habe viel Tutorials geschaut, Websites gesehen, wie opencv in Raspberry PI installiert werden kann. Ich habe viele verschiedene Methoden probiert, aber mit keinem guten Ergebnis. Nach vielen Proben ist es gegangen. Es ist installiert, und habe ich dann verschiedene Skripte in python gemacht, um es zu testen. Manche der Skripte sind gegangen, manche nicht. Jetzt war eine kleine Herausforderung für mich, dass ich die Skripte, die nicht ausgegangen sind, verbessere. Anschließend habe ich herausgefunden, dass das Problem bei dem Kompilieren von opencv war (cmake). Ich habe es noch einmal vom Beginn kompiliert. Jetzt ist alles in Ordnung, alle Skripte arbeiten, keinen Fehler mehr, der mit opencv Paket zu tun hat.
2. Ein Treffen mit meiner Gruppe vor dem Beginn des Projektes war notwendig. Ich hab es Ihnen gesagt und erklärt, in welchen Verzeichnisse sie arbeiten sollten, die Struktur der Dokumentation, welcher Kommunikationskanal verwenden wir, um Probleme, Herausforderungen usw. zu besprechen usw. Jede Person hatte dann ihre Vorschläge, um das so und so zu lösen, und dieses Treffen hat zu lange gedauert, bis alle verstanden hatten, wie, wo, was, wann machen sollen. Aber auch nach dem Treffen gab es zwischendurch Missverständnisse bzw. Probleme mit der Kommunikation, z.B. wurde nicht im richtigen Verzeichnis gearbeitet usw.

3. Ein Git-Repository erstellen und einzurichten war einfach. Ich hab es online in github.com erstellt. Einen Name eingegeben und dann als Collaborators die anderen Teammitglieder hinzugefügt. Um strukturierter zu werden, habe ich dann verschiedene Branches angelegt. Wie immer, gab es mit dem Befehl 'push' und 'pull' wieder Probleme. Das habe ich gelöst, in dem ich allen gesagt habe, dass, wenn sie in einem Github-Repository arbeiten möchten, dann bevor dem Beginn der Arbeit, müssen sie ein 'pull' machen, damit die änderungen, die von anderen in dem Repository gemacht wurden, mit deiner Version am Computer synchronisiert werden. Sie wissen nie, was die anderen in diesem Repository machen. Sie machen 'push', ohne zu sagen, dass sie ein '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. Damit wir die Abhängigkeiten zu minimieren, habe ich gedacht, dass jeder Teammitglieder andere Aufgaben bekommt, als die, die in der Dokumentation stehen. Ich war gezwungen, diese änderung zu machen, sonst würde das Projekt viel länger dauern.
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 keinen solchen Fehler(Kleinigkeiten) mehr in der Planungsphase machen werde.

3.1.4 Qualitätssicherung und Controlling

Ein Risiko ist meistens nur eine Einschätzung, was kostet es einem Unternehmen, wenn die Projektziele nicht erreicht werden. Ich, als Projektleiter, muss das machen. Eine Risikoanalyse zu planen ist sehr schwierig, weil es mit der Zukunft zu tun hat. Zuerst muss ich an die Zukunft denken, bei welchen Bauteilen z.B. können Fehler auftreten, welche Programme können ausfallen. Das bedeutet, ein überblick an der Zukunft und einschätzen, was für Fehler und Risiken es geben kann. Dann schätze ich die Wahrscheinlichkeit ihres Eintretens und am Ende die Maßnahmen. Dahinter versteckt sich eine grosse Arbeit.[24] Auf Abb. 3.5 können Sie die Risikoanalyse sehen. Wahrscheinlichkeit, Kosten usw. alles sollen beachtet werden.

Risikoanalyse: Gesichtsregistrierung und Gesichtserkennung

Risikotyp	Nr.	Wahlsch.	Auswirk.	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 Hasmeka	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 Informationen) + 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	All	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	All	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	All	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	All	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	All	Ausfall einzelner Projektglieder	Ein anderer Projektglieder diese Arbeit machen	Einem neuen Team das Projekt einrichten

Abbildung 3.5: Risikoanalyse in Excel

3.2 Ergebnisse

Es sind 3 Monaten vergangen, seit ich angefangen habe zu arbeiten. Weil meiner Teil nicht viel Hardware hatte, nur ein LED, einen Schalter, Tastatur, war es nicht schwierig, die mit dem ganzem System zusammenzusetzen. Bis jetzt ist es gedacht, dass das System keine LCD Anzeige haben wird, weil ich nicht viel Zeit habe, um sie zu programmieren. Statt LCD Anzeige wird einen Bildschirm verwendet.

3.2.1 Implementierung

Nachdem eine grosse und gute Arbeit meinerseits, befindet sich das Produkt in der Inbetriebnahme-Phase. Das bedeutet, bis jetzt gibt es einen Prototyp. Ich habe für diesen Prototyp die sogenannte Kernfunktionen implementiert. Kernfunktionen sind Grundfunktionen bzw. wesentliche Funktionen, ohne denen nichts geht. Die Funktionen, die in diesem Prototyp implementiert sind, sind folgende:

- Admin Account. Wenn sich eine Person registriert möchte, dann muss sich der Admin sich einloggen. Für diesen Prototyp gibt es nur ein Passwort, damit der Admin erkannt wird. Auf dem anderen Prototyp wird es kein Passwort geben, sondern die Überprüfung wird durch den Gesichter-Vergleich erfolgt.
- Eine Person wird mit ihrem Vornamen, Nachnamen, Email und Role in der Datenbank gespeichert
- Ein Bild von einer Person wird gemacht, und in der Datenbank speichern mit der ID der Person speichern
- Wenn die Person erfolgreich in Datenbank gespeichert wird, leuchtet die LED.

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 Implementierung, 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.[34]

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.[23]



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.[31] 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.[15]



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 Datenbank

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, während 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 Platz 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 Primär 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 während 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 Primär 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 Primär 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 einer 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 protokolliert 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 Primär 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 bekommt 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 Datenbank - 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 Programmierregeln von MySQL, obwohl MariaDB in Verwendung war. Zwischen diese zwei Datenbankverwaltungssystem gibt es kleine Unterschiede, die zu Problemen führen. 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 erstellt 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 Änderungen 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 Übertragung von den Parametern zwischen den verschiedenen Funktionen.

TypeError: not all arguments converted during string formatting python

Der folgende Code nimmt als Parameter die Daten von dem neuen User, damit der User in der Datenbank registriert wird. Aber wenn man nur die Methode zu probieren möchte, 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. Bei der zweiten Möglichkeit musst diese Parameter geparsst 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 erstellt werden. Aber trotzdem wurde dieser Fehler generiert, weil wenn ein Parameter auf den falschen Datentyp geparsst wird, dann wird dieses Problem dargestellt.

4.3.3 OpenCV

OpenCV Bibliotheken können manchmal schwierig zu installiert werden, weil es so viele Möglichkeit zum Einrichten von OpenCV gibt, die aber von dem Betriebssystem abhängen. Am meisten Fallen wenn die Installation Fehler produziert, musste OpenCV von Beginn noch einmal eingerichtet werden.

Unable to locate package libjasper-dev

Diese Bibliotheken ermöglichen ein paar Funktionen 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

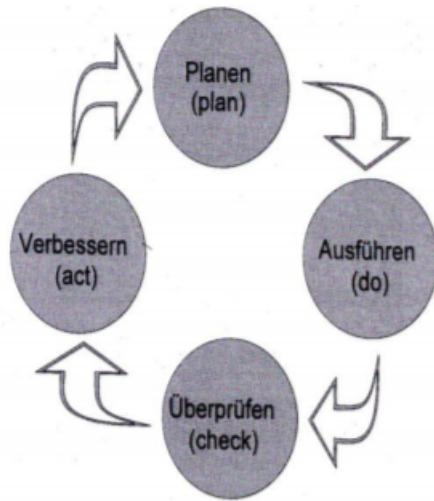


Abbildung 4.5: Plan-Do-Check-Act-Zyklus

[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 gelöst. Durch verschiedene Zyklen beachtet man auch bei den Risiken, weil wenn etwas nicht funktionier, wird schnell das Problem gefunden und später wird eine andere Lösungswegen 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 durchgeföhrt 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 - Egli

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 erledigt werden.

Alles wird leicht und bequem durch Konsole eingegeben.

So wurden die Entwicklung, das Testen von den genutzten Algorithmen und anderen Technologien vielmals erleichtert. [28]

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.

[33]

”Git ist ein freies und Open Source verteiltes Versionskontrollsysteem, das entwickelt wurde, um alles von kleinen bis zu sehr großen Projekten mit Geschwindigkeit und Effizienz abzuwickeln.”[16]

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.

[25]

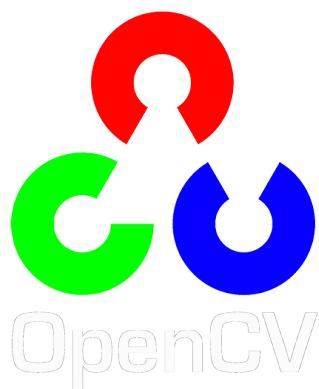


Abbildung 6.1: OpenCV Logo

[6]

SciPy hat sich nützlich, beim Lösen von gewisse mathematische Angaben und bei zusätzliche Installationen von Bibliotheken zu helfen.

SSciPy ist eine kostenlose und Open-Source-Python-Bibliothek für wissenschaftliches und technisches Rechnen.”[40]

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.

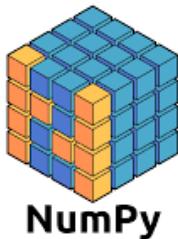


Abbildung 6.2: Numpy logo

[19]

Dlib ist ein so genanntes Toolkit, das Algorithmen für Machine Learning und Tools 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.[30]

Es wurde prinzipiell bei der Extrahierung der so genannten „Facial Landmarks“ gebraucht.



Abbildung 6.3: dlib logo

[30]

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 „Structured Design“ erfolgt.

Structured 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 verschiedenen Ebenen unterteilt, von außen beginnend.

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

6.2.1 Lösungsweg - Structed Software design

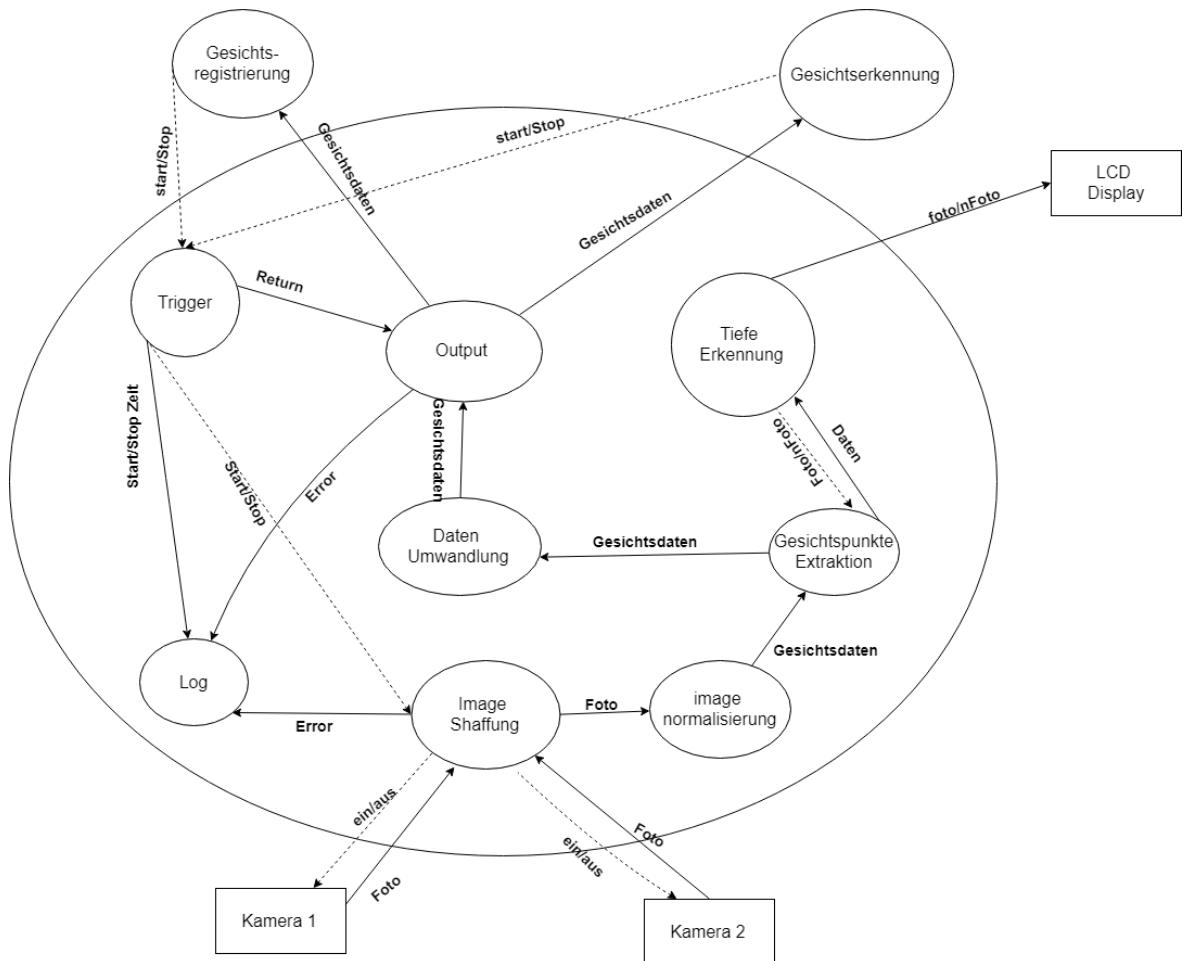


Abbildung 6.4: Bildverarbeitung Structured Design

Wie es in der Abb.6.4 sichtbar ist, ist die Bildverarbeitungsteil in diese Einheiten unterteilt:

1. Image Schaffung
 2. Image Normalisierung
 3. Gesichtsschlüsselpunkten 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 das Image Schaffung Modul initialisiert. Unabhängig von welchen von diesen beiden Schnittstellen die Anfrage kommt, macht die Image Schaffung aus der zwei Kameras ein Foto.

Implementierungsnahe 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 Processing“.

Wenn ein Bild zu klein ist wird die Größe angepasst, wenn ein Gesicht verdreht ist wird es gerade rotiert. Affines und perspektivisches Warping wird falls notwendig geführt. Farben werden nach Bedarf auch angepasst usw.

Danach erfolgt die Erkennung von Gesichtern, das Ausschneiden von ihnen und die Extrahierung der Gesichtsschlüsselpunkte.

Sie werden dann in Vektoren umgewandelt und zur Abgleich oder Registrierung je nach Bedarf, geschickt.

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

6.2.2 Gesichtsdetektion

Das erste Ereignis, dass erfüllt werden muss bei der Gesichtserkennung ist das Finden von Gesichtern, also die Feststellung, wo sich die Gesichter im Bild befinden.

Dafür ist die „Haar Cascade Classifiers“ pre-trained Classifier gebraucht 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“. [39]

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 6.5 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 Sum-

menberechnungen dient.

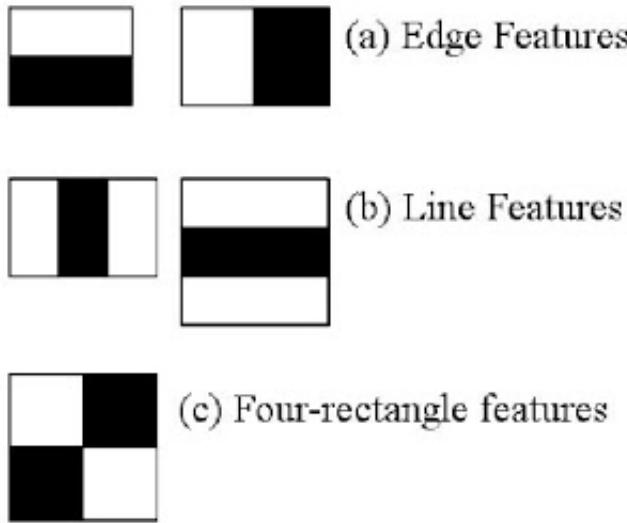


Abbildung 6.5: Haar Features

[39]

Man muss aufpassen, da nicht alle Merkmale von Nutzen sein könnten. Man kann es zum Beispiel deutlich auf Abbildung 6.6 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.

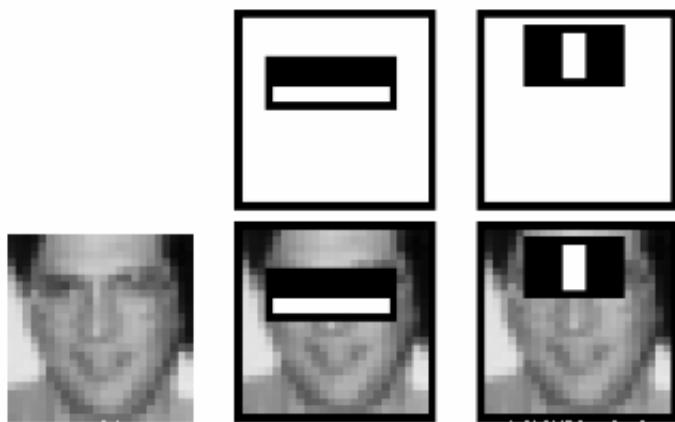


Abbildung 6.6: Haar Einsetzung

[39]

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

Durch die Verwendung von classifiers die OpenCV bietet, setzen wir das um.

```

//haar cascade classifier laden
1      face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_def

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

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

4      for(x, y, w, h) in faces:
//parameter: image, startpunkt, endpoint(w..width, h..height), farbe, D
5          cv2.rectangle(image, (x, y), (x+w, y+h), (255,0,0), 2)

//opencv und numpy importieren
//Bild als input einfügen
//Bild einlesen
//Bild in „gray scale“ umwandeln wodurch die Rechenleistung reduziert wird. //(Da
es keine Farben zuhanden sind).
//Parametern: src(source image), dst(destination image), code(conversion code)
//Hier benutzt man die Cascade Classifier um Gesichter zu finden.
//Parameter:
//image, objects, scaleFactor = 1.1, minNeighbors = 3, flags = 0, minSize = new
cv.Size(0, 0), maxSize = new cv.Size(0, 0))
//Zeichenrahmen für das Gesicht zeichnen.
//Bild zeigen durch ‘image’ Fenster
//Press any key zum schließen vom Fenster

```

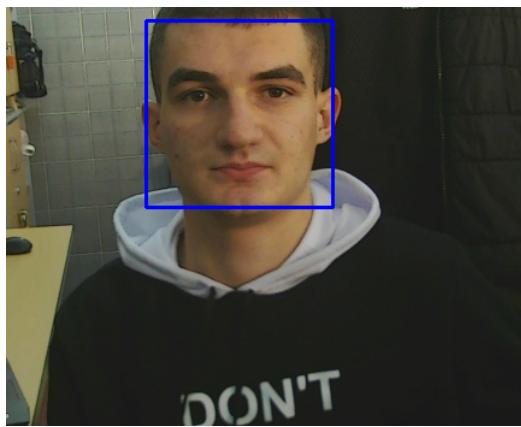


Abbildung 6.7: Box auf Gesicht

Die Funktionalität des Code ist genau auf Abb.6.7 ersichtlich.

6.2.3 Gesichtsdetektion und Zuschneiden

Nachdem Gesichter gefunden werden, muss man daraus eigene Bilder machen, die zur Erleichterung von der Extrahierung der Schlüsselpunkte dienen.

nrFace bestimmt die Anzahl von Gesichtern, die von den cascade classifier gefunden wurden. Nur wenn dieser Wert größer als Null ist soll die Logik von der Crop Funktion weiterlaufen.

```

if nrFace > 0:
    for face in faces:
        for(x, y, w, h) in faces:
            1         r = max(w, h) /2
            2         centerx = x + w /2
            3         centery = y + h /2
            4         nx = int(centerx - r)
            5         ny = int(centery - r)
            6         nr = int(r * 2)
            7         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")

```

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. Auf Abb.6.8 sieht man deutlich nur das Gesicht.



Abbildung 6.8: abgeschnittenes Gesicht

6.2.4 Gesichtsschlüsselpunkte Extraktion

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

Dazu wurden die „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 Landmarken?

Das ist eine Teilmenge des Problems der Formvorhersage.

Bei einem vorgegebenen Eingabebild (und normalerweise einer ROI¹, 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 ihrem Paper: One Millisecond Face Alignment with an Ensemble of Regression Trees entwickelt.[29]

Dieser Methode werden viele Trainingsdaten hinzugefügt womit es ein Kombination von Regressionsbäumen trainiert wird um die Positionen der „Facial Landmarks“ direkt aus den Pixelintensitäten selbst zu beurteilen.[29]

Dadurch werden unsere gewünschten 68 Gesichtsmarkierungen mit hohen Vorhersagbarkeit extrahiert und als x,y Koordinaten weitergegeben. Die Indizes der 68 Koordinaten sind in der Abb.6.9 dargestellt:

¹Region of interest

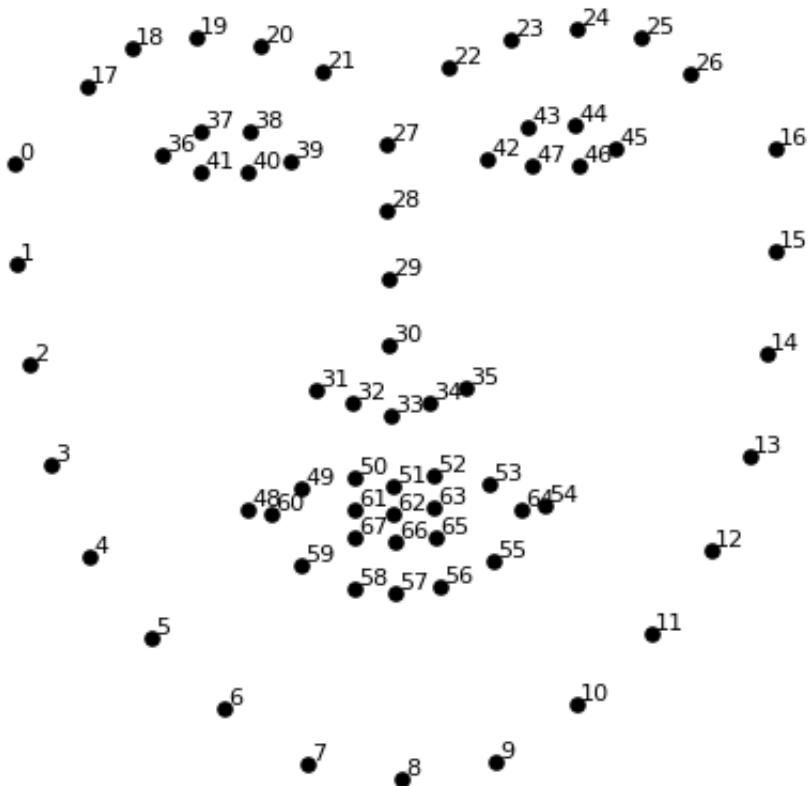


Abbildung 6.9: Facial Landmarks

[29]

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.

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

Nur die Positionen von diesen Merkmalen alleine reichen bei Gesichtserkennung nicht. Deswegen wurden die relative Positionen bzw. Abstände zwischen Gesichtsmerkmalen als Vektoren genommen und betrachtet um die Qualität und die Präzision der Erkennung zu verbessern.

Die Zahlen beziehen sich auf die 6.9 genannten Indizes.

1. Rechtes Auge Höhe: $V1 = 41 - 37$
2. Rechtes Auge Breite: $V2 = 39 - 36$
3. Linkes Auge Höhe: $V3 = 46-44$

4. Linkes Auge Breite: V4 = 45-42
5. Rechte Augenbraue Breite: V5 = 21 – 17
6. Linke Augenbraue Breite: V6 = 26 – 22
7. Rechte augenecke mit mittlerer rechter Augenbraue: V7 = 36 – 19
8. Linkes augenecke mit mittlerer linker Augenbraue: V8 = 45 – 24
9. Nasespitze mit Mundzentrum: V9 = 45 -24
10. Linke Augenecke mit linker Mund Ecke: V10 = 54 – 45
11. Rechte Augenecke mit rechter Mund ecke: V11 = 48 – 36
12. Nase Höhe: V12 = 33 – 27
13. Nase Breite: V13 = 35 – 31

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 waren am Beginn auch unklar wegen den Kommunikationslücken und auch die sehr grobe Planung war problematisch.

Lösungen:

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

Die Kommunikation hat sich mit der Zeit stark verbessert und dadurch wurden auch die Unklarheiten abgeklärt.

6.4 Qualitätssicherung, Controlling

Die Qualität wurde durch verschiedene Methoden gesichert. Eine von denen war die Methode 5xWarum. „5x Warum ist eine iterative Fragetechnik, mit der die Ursache-Wirkungs-Beziehungen untersucht werden, die einem bestimmten Problem zugrunde liegen.“ [21] 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 FAST².“

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

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.

6.5 Ergebnise

Ich habe ungefähr 100 Stunden Zeit bis jetzt für die Diplomarbeit investiert und folgendes erreicht.

Face Detection wurde fertig implementiert.
Bilder sind halbwegs normalisiert worden.

²Features from Accelerated Segment Test

Gesichtsschlüsselpunkte wurden extrahiert und die Merkmalvektoren wurden angegeben.

Kapitel 7

Gesichtserkennung - Rei

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

7.1 Allgemeines

Die Gesichtserkennung ist der Teil des Systems, mit dem ich mir beschäftige. Es wird durch einen Vergleich überprüft, ob die Person vorher schon registriert worden ist oder nicht, und ob sein Gesichtsdaten schon am Server existieren oder nicht. Nur wenn die Vergleichsergebnisse positiv sind, darf die Person rein kommen. Der Person werden die Ergebnisse durch Anzeigern kommuniziert. Dieses Teil des Projektes erfordert eine Arbeit mit Datenbanken, Gesichtsvergleichsalgorithmen und mit vielen System Tests. Teil meiner Aufgabe ist auch der Aufbau des Systems und alles was mit Hardware zu tun hat. Alle Hardware Komponenten werden in den folgenden Kapiteln klar, verständlich und deutlich erklärt. Eine grobe Skizze des Systems ist in Abb.7.1 zu sehen.

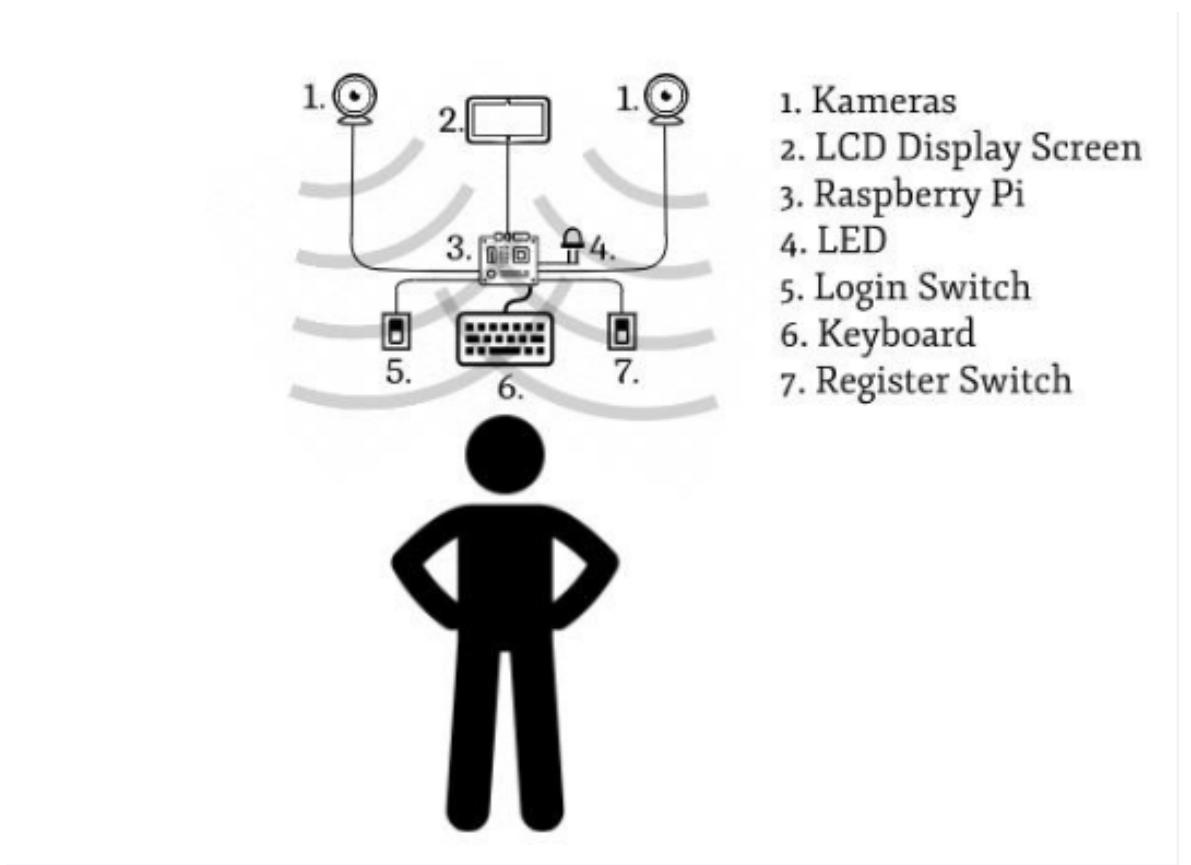


Abbildung 7.1: Grobe Skizze des Systems

7.2 Technische Lösung

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

7.2.1 Hardware und Aufbau

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

Bauteile und HW-Komponenten

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

1. *Steckboard bzw. Steckplatine*

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

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

2. Kabel bzw. Leiter

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

3. LEDs³

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

4. Widerstand

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

5. Taster

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

6. Raspberry Pi

Raspberry Pi ist ein Minicomputer, der in diesem Projekt den normalen Computer ersetzt. Der verwendete Raspberry, Version 3, hat 4 USB-Anschlüsse, einem Netzteil, eine SD-Karte, 16 GPIO⁴ Pins und einem VGA Schnittstelle. Die 32-Bit SD-Karte ist das wichtigste Element, weil dort alle Daten und Informationen gespeichert sind.

7. Bildschirm

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

³Light Emitting Diode

⁴Generated Input Output

8. Tastatur

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

9. Kamera

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

Schaltplan und Erklärung des Aufbaus

Die Hardware Komponenten werden in einer Steckplatine platziert. Dort werden die Verbindungen mit den anderen Komponenten sowie mit dem Raspberry Pi erstellt. Die elektrische Schaltung wird durch einen Schaltplan beschrieben. Dieser Schaltplan wurde mit Hilfe eines Programms, das Fritzing heißt, erstellt und spielt eine sehr wichtige Rolle bei der Organisation und Planung des Schaltkreises. Der Schaltplan ist auf Abb.7.2 zu sehen.

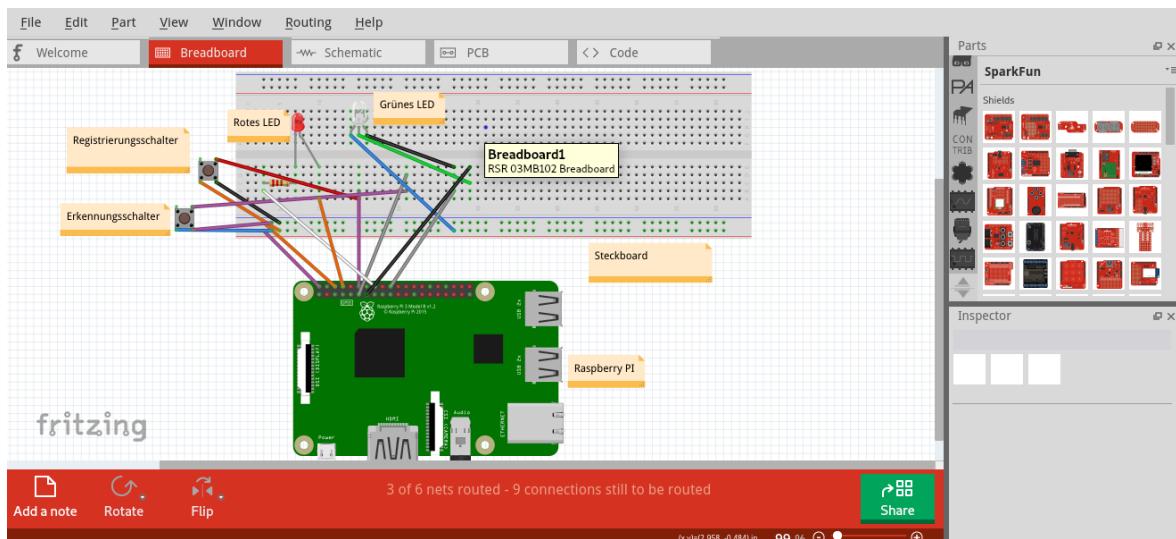


Abbildung 7.2: Schaltplan des Systems

Wie der Schaltplan zeigt, besteht das System aus zwei LEDs, zwei Tastern, einem Widerstand, einer 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 Erkennungstaster⁵ 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

⁵Auch als Login Taster genannt

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

7.2.2 Software

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

Verwendete Betriebssysteme, Programmiersprachen, Frameworks, Technologien und Planungsmethoden

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

1. *Betriebssystem und Programmiersprache*

- Linux(Debian)

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

- Open Source und freie Software

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

- Hohe Sicherheit

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

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

- Vielfältigkeit an Möglichkeiten und Benutzerfreiheit

Linux wurde auch aufgrund der angebotenen Benutzerfreiheit und Benutzerfreundlichkeit verwendet. Beliebige Systemkonfigurationen, verschiedene Gesichtserkennungspakete(OpenCV⁶) und andere Programme oder Technologien(Fritzing) sind von Linux leichter und günstiger erhältlich als bei anderen Betriebssystemen. Endlich ist der Raspberry PI auch kompatibler mit Linux. Als Linux Distribution wurde Debian gewählt. Gründe dafür sind die große Menge der angebotenen Pakete, die hohe Geschwindigkeit und die schnelle Fehlerkorrektur. [17]



Abbildung 7.3: Debian[7]

- Python

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



Abbildung 7.4: Python[8]

2. Frameworks

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

- Open CV

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

⁶Open Computer Vision

verwendet wird. Die Bibliothek verfügt über mehr als 2500 optimierte Algorithmen, die sowohl klassische als auch moderne Computer Vision- und maschinelle Lernalgorithmen umfassen. Diese Algorithmen können verwendet werden, um Gesichter zu registrieren und zu erkennen, um Objekte zu identifizieren, menschliche Handlungen in Videos zu klassifizieren und Kamerasbewegungen 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. [20]



Abbildung 7.5: Open CV[9]

- **Git**

Git ist ein Versionsverwaltungssystem, das für das Back-Up des Systems verwendet wurde. Dieses Framework ist ein verteiltes Versionsverwaltungssystem, das heißtt, die unterschiedlichen Versionen der Dateien werden effizient gespeichert, in dem die ganze Repository vom Server kopiert wird. Dieses erlaubt dem Benutzer lokal zu arbeiten, und alle Änderungen nachträglich am Server zu aktualisieren. Nicht nur das, sondern auch die hohe Geschwindigkeit der Fehlerfindung und die perfekte Unterstützung von großen Projekten, in dem alle Teammitgliedern auf alle Daten Zugriff haben, waren die Gründe, warum Git für das Back-Up des Systems gewählt wurde. [37]



Abbildung 7.6: Git[10]

3. Technologien

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

- Raspberry Pi

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

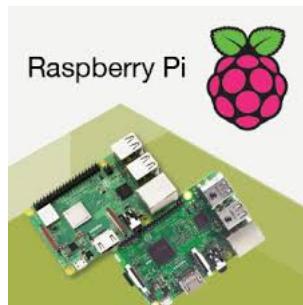


Abbildung 7.7: Raspberry Pi[11]

- 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 im Linux angeboten wird, was für Windows nicht der Fall ist.

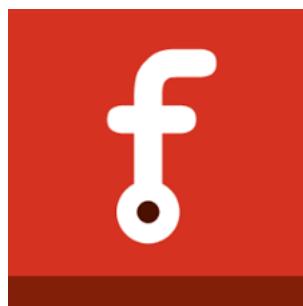


Abbildung 7.8: Fritzing[12]

- MySQL und MariaDB

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



Abbildung 7.9: Mysql[13]



Abbildung 7.10: Mariadb[14]

4. Planungsmethode

- Structed Software Design

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

- Bubbel

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

- Schnittstellen

Sind die Interfaces, die die Verbindung zwischen die Bestandteile machen

- Nachrichten

Sind die geschickten Signale innerhalb des Systems.

- Flüsse
Zeigen die Daten die innerhalb des Systems fließen.
- Modulen
Bestandteil des Systems, das innerhalb liegt.
- Terminatoren
Ähnlich mit den Modulen, aber befinden sich außerhalb des Systems und beeinflussen es von außen.
- definierte Abgrenzungen
Mithilfe der Abgrenzungen wird vom Systemhersteller genau definiert, was er nicht machen wird, sei es auf Zeit-, Schwierigkeits-, oder Komplexitätsverbundene Gründe. Diese Abgrenzungen dienen dazu, dem Auftraggeber genau zu zeigen, was er erwarten kann, damit es keine Missverständnisse gibt. [36]

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

Lösungsweg- Beschreibung und Erklärung

Der Lösungsweg für die Umsetzung der Aufgaben der Gesichtserkennung ist streng mit einer guten vorherigen Planung verbunden. Deshalb wurde nicht nur das Big Picture(Großes Sicht des Systems nach Außen), die in die vorherigen Kapiteln beschrieben wurde erklärt, sondern auch die Erste Ebene des Gesichtserkennungsteils. Die Big Picture sieht man unter Abb.7.11

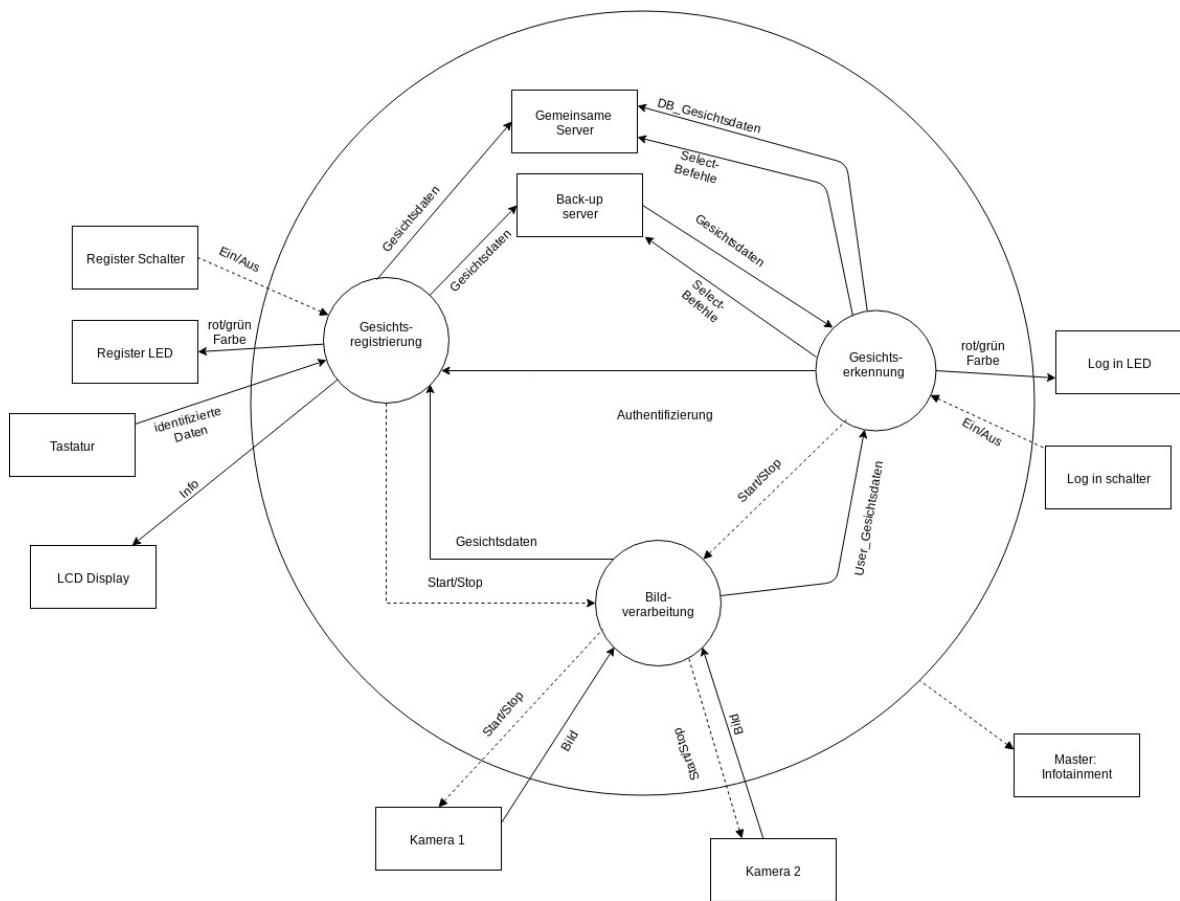


Abbildung 7.11: Big Picture

Die erste Ebene sieht man unter Abb.7.12. Sie ist ein detaillierter Version des Big Picture, das sich nur auf dem Erkennungsteil konzentriert. Man spricht von einer Iteration, die hier passiert ist. Folgend wird es die Vorgehensweise und den Lösungsweg dieser Aufgabe mit Hilfe der 1.Ebene erklären und beschreiben.

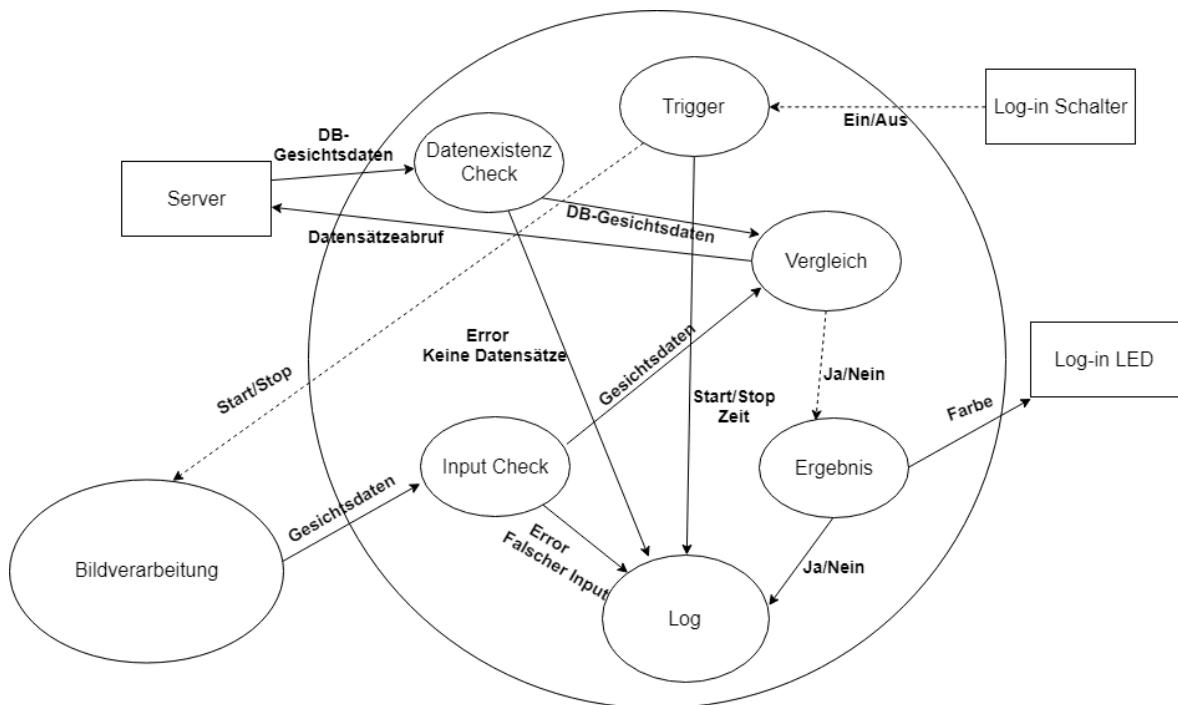


Abbildung 7.12: Erste Ebene

Einzelne Arbeitsschritte:

1. Erkennungstaster wird gedrückt

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

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

2. Benutzer gibt seine Emailadresse ein

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

3. Bild wird gemacht und temporär gespeichert

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

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

4. *Die Existenz derselben Emailadresse des Benutzers in der Datenbank wird geprüft*
Dieser Schritt ist deswegen wichtig, weil falls die eingegebene Email nicht in der Datenbank existiert, muss das System die Bildvergleichung nicht machen.

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

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

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

6. *Die Bildinformationen der beiden Bildern werden verglichen*

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

7. *Passen die beiden Bildern zusammen, wird der Benutzer eingeloggt.*

Wenn das Vergleichsergebnis positiv ist, wird die grüne LED leuchten und der Benutzer darf weitermachen. Sonst wird die rote LED eingeschalten.

7.3 Herausforderungen, Probleme, und deren Lösung

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

- *OpenCV Installation*

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

- *Fritzing*

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

- *Kurzschluss beim Raspberry Pi*

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

- *Pakete FaceRecognition Installation*

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

- *Probleme bei der Aufruf der Skripten*

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

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 der Problem in kleinen Teilen geschnitten damit es klarer wird. Es wurde auch eine detaillierte Soll-Ist Analyse gemacht, die bei

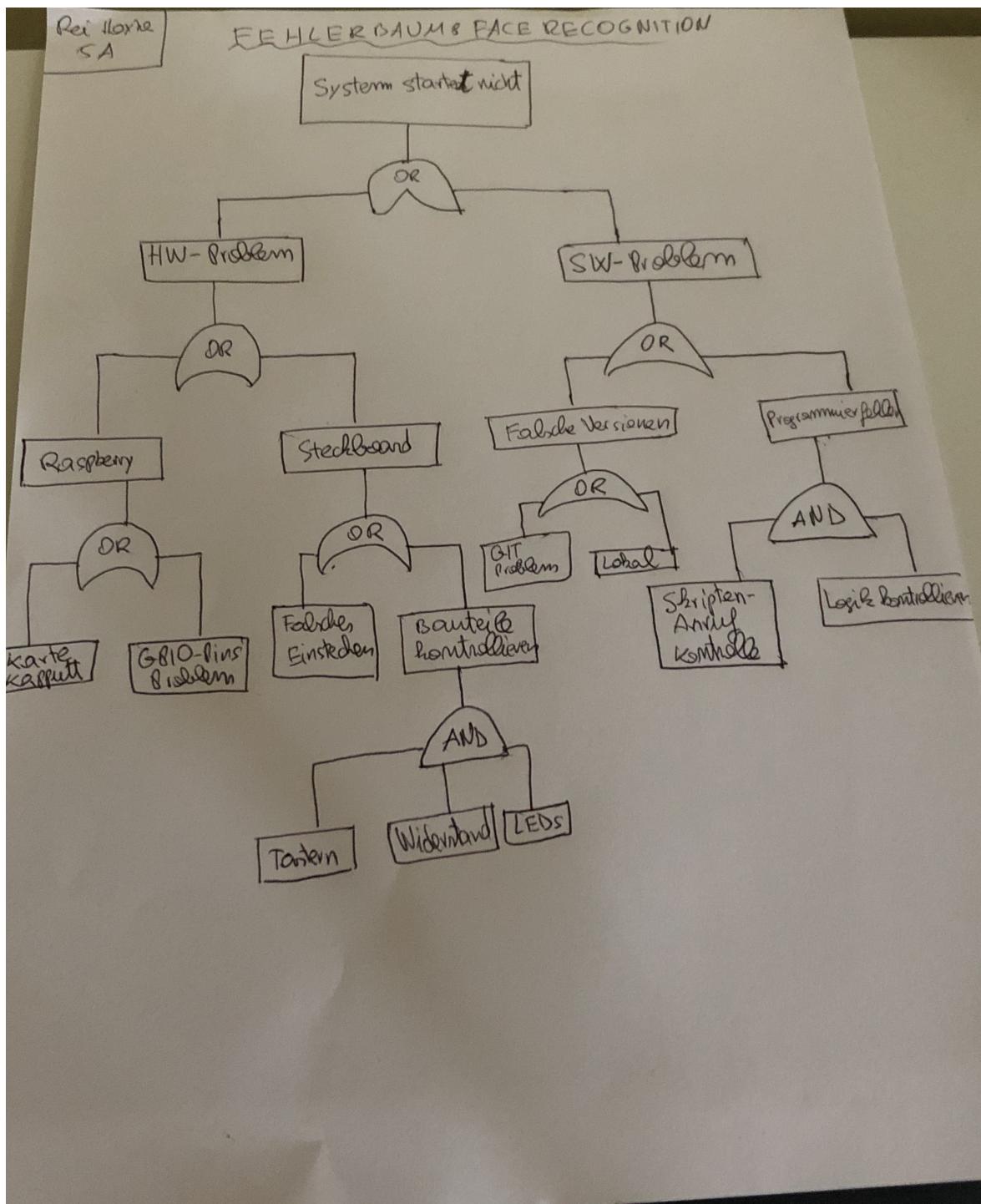


Abbildung 7.13: Fehlerbaum

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

7.5 Ergebnisse

Die bisherige Ergebnisse sind die folgenden:

1. Systemaufbau

Das ganze System wurde zusammen mit der ganzen Hardware aufgebaut.

2. Digitale Darstellung des Systemaufbaus

Der Schaltplan des Systems wurde durch Fritzing digital dargestellt.

3. Erkennung des Admins

Der Admin wurde sofort vom System erkannt, mithilfe eines Skripts, dass OpenCV verwendet um das Adminbild mit den anderen Bildern zu vergleichen.

4. Aufnahme der Benutzer Gesichtsdaten mithilfe der Emailadresse

Gibt der Benutzer seine Emailadresse ein, werden die zugehörigen Gesichtsdaten von der Datenbank geladen, um den Vergleich zu machen.

|||||| HEAD

=====

Kapitel 8

Planung vs Realisierung

Planung vs Realisierung Viele Sachen sind anders umgesetzt worden als sie geplant waren. Die Aufgabenteilung wurde anders geplant aber bei der Implementierung wurde es als vernünftiger angesehen, dass die verändert werden. Es war die Wichtigkeit und die Größe der Arbeitspakete beim Beginn nicht richtig abgeschätzt weil wir uns nicht gut ausgekannt haben. Konkretes Beispiel ist das Datenbankdesign. Zuerst war diese Aufgabe an Aron zugeteilt, und erst später wurde festgestellt, dass Jordi besser für das geeignet war und so hat er sie übernommen. Gewisse Ziele wurden auch entsprechend angepasst und sogar weggenommen, weil sie als unnötig galten. Zum Beispiel: „Licht neben der Kamera“ wird nicht mehr berücksichtigt. Wir wollten auch eine minimale Verarbeitungszeit erreichen, aber dadurch wir fertig trainierte Modelle und Formvorhersager verwenden, auf denen wir nicht zugreifen können, ist es also schwierig so was zu machen. Es war früher auch geplant nur Gesichtsdaten in der Datenbank zu speichern aber erst wenn es in der Implementierungsphase gekommen ist, haben wir gemerkt, dass wir auch Fotos von den Personen(ins Besondere die von den Administratoren) in der Datenbank speichern sollen. Das wurde aber noch nicht fest entschieden. Auf Abb. 8.2 sind die erreichten und nicht erreichten Ziele.

Ziel	Status
Datenbankdesign wurde erstellt.	✓
Datenbank wurde eingerichtet.	✓
Zugriffsberechtigungen wurden implementiert.	✓
Gesichts-Schlüsselpunkte wurden extrahiert.	✓
Position/Verhältnisse der Hauptmerkmale wurden relativ zueinander herausgeholt.	✓
Daten werden für Gesichtserkennung aufbereitet.	✓
Gesichtserkennung wurde implementiert. Data streams von den extrahierten Bildern werden mit erheblichen Daten die auf der Datenbank gespeichert sind.	✓
Gesichter von zwei Administratoren wurden auf der Datenbank gespeichert für den Zweck der Admin Account.	✓
Admin Account (Register-Rechte nur für Schüler und Lehrer eingeben)	✓
Ein Maximum von 500 Personen ist gedacht, in der Datenbank gespeichert zu werden.	
10 Tests wurden gemacht, jeder Test in einer anderen Raumkondition. Alle Betriebskonditionen wurden getestet.	

Abbildung 8.1: Planung vs Realisierung

"Gefälschte" Gesichter wurden gegenüber "echte / legitimate" Gesichtern erkannt. Unterschied zwischen einer reellen Person und einem Bild bzw. einer Visuellen Darstellung von ihnen gefunden.	
Error checking ausgeführt. Es wird signalisiert wenn ein Gesicht nicht 100% identifiziert werden kann.	

Safe Mode erstellt. Backup Server wurde erstellt und eingerichtet.	
Min. Verarbeitungszeit (min. Zeit für Gesichtsdetektion) eingestellt. Es dauert min x Sekunden um ein Gesicht erfolgreich zu erkennen.	✓
Aufbau des Systems wurde erledigt.	✓
Es ist ein Benutzer-freundliches System mittels einer minimalen Anzahl von Schalter und einem Display erstellt.	✓

Abbildung 8.2: Planung vs Realisierung

Kapitel 9

Evaluierung und Resümee

9.1 Wertschöpfung und Lessons Learned

Also, nach einige Wochen sind wir zu Idee gekommen, dass die Planung eine große Rolle in diesem Projekt spielt, weil das Team must genau spezifizieren was jeder macht, und auch die wichtigsten Punkte, wie z.B der Lösungsweg. Bei den nächsten Treffungen soll mehr über den Problemen diskutiert werden, damit klar für jeden was genau zu tun ist. Eine chaotische Arbeit wird keine guten Ergebnisse bringen, weil auch eine Software bzw. Programm, dass ohne Struktur geschrieben wird, wird schwierig zum Lesen und Verstehen. *l.l.l.l.l ad5d4a4503bd63c93088f9dff686e29b86128b49*

Abbildungsverzeichnis

2.1	Big picture	4
2.2	Scrum	5
3.1	Structed Software Design bzw. erste Ebene	9
3.2	Packages zu importieren im Main-Skript	10
3.3	GPIO-Ports Konfiguration	11
3.4	Beispiel in Python, wie die Personen registriert werden	12
3.5	Risikoanalyse in Excel	15
4.1	MySQL Logo	17
4.2	MariaDB Logo	18
4.3	Python Logo	19
4.4	OpenCV Logo	19
4.5	Plan-Do-Check-Act-Zyklus	25
6.1	OpenCV Logo	28
6.2	Numpy logo	29
6.3	dlib logo	29
6.4	Bildverarbeitung Structed Design	31
6.5	Haar Features	33
6.6	Haar Einsetzung	33
6.7	Box auf Gesicht	34
6.8	abgeschnittenes Gesicht	35
6.9	Facial Landmarks	37
7.1	Grobe Skizze des Systems	42
7.2	Schaltplan des Systems	44
7.3	Debian	46
7.4	Python	46
7.5	Open CV	47
7.6	Git	47

7.7	Raspberry Pi	48
7.8	Fritzing	49
7.9	Mysql	49
7.10	Mariadb	49
7.11	Big Picture	51
7.12	Erste Ebene	52
7.13	Fehlerbaum	55
8.1	Planung vs Realisierung	58
8.2	Planung vs Realisierung	59

Tabellenverzeichnis

4.1 Technologien	17
----------------------------	----

Literatur

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/>.
- [6] URL: <https://opencv.org/>.
- [7] URL: https://www.google.com/search?rlz=1C1GGRV_enAL753AL753&biw=1366&bih=576&tbs=isch&sxsrf=ACYBGNQ9s-s0UP2K5YJ_TPdkR9G7feLepw%3A1576022986105&sa=1&ei=yjPwXZz7BaXgkgWTv4eICw&q=debian&oq=debia&gs_l=img.3.0.0i6716j014.10804.12283..14300...1.0..0.166.825.0j6...0....1..gws-wiz-img.....35i39.fbB2uG6SnFE#
- [8] URL: https://www.google.com/imgres?imgurl=https%3A%2F%2Ffictslab.com%2Fwp-content%2Fuploads%2F2019%2F03%2Fd1326ca6cca8038cd115a061b4e2b3bc-840x430.png&imgrefurl=https%3A%2F%2Ffictslab.com%2Fen%2F6-reasons-why-python-is-the-future-programming-language%2F&docid=UKwQ11G91v7G0M&tbnid=0aCNC7utSq-s0M%3A&vet=10ahUKEwjz_T5pqzmAhXQsKQKHVOPDN4QMwh7KAkwCQ..i&w=840&h=430&bih=576&biw=1366&q=python%20programming&ved=0ahUKEwjz_T5pqzmAhXQsKQKHVOPDN4QMwh7KAkwCQ&iact=mrc&uact=8
- [9] URL: https://www.google.com/imgres?imgurl=https%3A%2F%2Fopencv.org%2Fwp-content%2Fuploads%2F2019%2F02%2Fopencv-logo-1.png&imgrefurl=https%3A%2F%2Fopencv.org%2F&docid=bk2cK6wdUxgMYM&tbnid=BHGjOQUCoS4CjM%3A&vet=10ahUKEwiT_IPsqKzmAhVM-aQKHQNcAjgQMwh3KAAwAA..i&w=512&h=512&itg=1&bih=576&biw=1366&q=opencv&ved=0ahUKEwiT_IPsqKzmAhVM-aQKHQNcAjgQMwh3KAAwAA&iact=mrc&uact=8

- [10] URL: https://www.google.com/search?rlz=1C1GGRV_enAL753AL753&biw=1366&bih=576&tbs=isch&sxsrf=ACYBGNTout4ebIbjJD3KkDdfub0V5SFv1w%3A1576022753156&sa=1&ei=4TLwXbyLCcTWkwX-7pvQBQ&q=git&oq=git&gs_l=img.3..0110.56666.56955..57700...0.0..0.132.490.0j4.....0...1..gws-wiz-img.....0i67j35i39j0i3.Qugf0_VouHo&ved=0ahUKEwj8_--ipqzmAhVE66QKHX73BloQ4dUDCAc&uact=5#.
- [11] URL: <https://www.google.com/imgres?imgurl=https%3A%2F%2Fwww.element14.com%2Fcommunity%2Fcommunity%2Fimage%2F2475%2F2.png&imgrefurl=https%3A%2F%2Fwww.element14.com%2Fcommunity%2Fcommunity%2Fraspberry-pi&docid=SexjLuPxEC3YHM&tbnid=h-hlStmFtIy0qM%3A&vet=10ahUKEwiWwYbLpqzmAhUMyaQKHfamDZAQMwiTAihQMFA..i&w=320&h=320&bih=576&biw=1366&q=raspberry%20pi&ved=0ahUKEwiWwYbLpqzmAhUMyaQKHfamDZAQMwiTAihQMFA&iact=mrc&uact=8>.
- [12] URL: https://www.google.com/imgres?imgurl=https%3A%2F%2Fpbs.twimg.com%2Fprofile_images%2F378800000547620716%2Ff4a72078c6bdb3d379894da3ebd09270_400x400.png&imgrefurl=https%3A%2F%2Ftwitter.com%2Ffritzingorg&docid=QHWNrnPTdZnKkM&tbnid=_LdE_o8j5JTS_M%3A&vet=10ahUKEwj8_--ipqzmAhVE66QKHX73BloQMwimASgjMCM..i&w=400&h=400&bih=576&biw=1366&q=fritzing&ved=0ahUKEwj8_--ipqzmAhVE66QKHX73BloQMwimASgjMCM&iact=mrc&uact=8.
- [13] URL: https://www.google.com/imgres?imgurl=https%3A%2F%2Fcdn.worldvectorlogo.com%2Flogos%2Fmysql.svg&imgrefurl=https%3A%2F%2Fworldvectorlogo.com%2Flogo%2Fmysql&docid=tIz-HXUbbG9S5M&tbnid=Bgugh_pvo8X-JM%3A&vet=10ahUKEwiDtYDsqazmAhVNKuwKHRzPBGEQMwhuKAAwAA..i&w=2500&h=1733&bih=576&biw=1366&q=mysql%20logo&ved=0ahUKEwiDtYDsqazmAhVNKuwK&iact=mrc&uact=8.
- [14] URL: <https://www.google.com/imgres?imgurl=https%3A%2F%2Fmariadb.org%2Fwp-content%2Fuploads%2F2015%2F10%2Fmariadb-usa-inc.png&imgrefurl=https%3A%2F%2Fmariadb.org%2F&docid=E6-SRGvYz0UkdM&tbnid=F1j2ERTV48QWWM%3A&vet=10ahUKEwjv-NmDqqzmAhWEzqQKHTiWB8IQMwhkKAIwAg..i&w=340&h=175&bih=576&biw=1366&q=mariadb%20logo&ved=0ahUKEwjv-NmDqqzmAhWEzqQKHTiWB8IQMwhkKAIwAg&iact=mrc&uact=8>.
- [15] *Einsteigerguide: Was ist OpenCV?* URL: <https://viscircle.de/einsteigerguide-was-ist-opencv/>.
- [16] *Git.* URL: <https://git-scm.com/docs>.
- [17] *Gründe für die Wahl von Debian.* URL: <https://www.debian.org/intro/why-debian.de.html>.
- [18] Kitware inc. *CMake.* 2000. URL: [url](#).
- [19] *Numpy.* URL: <https://numpy.org/>.
- [20] *OpenCV.* URL: <https://opencv.org/about/>.

Der ganze Rest

- [21] 5x Warum. <https://www.quality.de/lexikon/5xwarum/>. [Online; accessed 2019]. 2017.
- [22] Stephan Augsten. Was ist Python? <https://www.dev-insider.de/was-ist-python-a-843060/>. [Online; accessed 2019]. 12/7/2019.
- [23] Martin H. Badicke. "MariaDB-Monitor?" In: (7. März 2017).
- [24] MA BSc Bekim Alibali. "Projektmanagement Teil5".
- [25] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000).
- [26] Jackson Cooper. *Python's time.sleep() – Pause, Stop, Wait or Sleep your Python Code*. <https://www.pythontutorial.net/python-basics/time-sleep-pause-wait-sleep-stop-your-code/>. [Online; accessed Tuesday 23rd July 2013]. 2013.
- [27] Ben Croston. *RPi.GPIO Python Module*. <https://sourceforge.net/p/raspberry-gpio-python/wiki/Home/>. [Online; accessed 2014]. 2014.
- [28] EDUCBA. *Differences Between Linux vs Windows*. <https://www.educba.com/linux-vs-windows/>. [Online; accessed 2019]. 2017.
- [29] 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.
- [30] Davis E. King. "Dlib-ml: A Machine Learning Toolkit". In: *Journal of Machine Learning Research* 10 (2009), S. 1755–1758.
- [31] Stefan Luber. "Was is Python?" In: *Nico Litzel* (4. Juli 2018).
- [32] 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](https://www(chip.de/artikel/Linux-Umstieg-So-einfach-gelingt-der-Windows-Wechsel-2_140047889.html). [Online; accessed 2007]. 2007.
- [33] 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.
- [34] Guido Stepken. "MySQL Datenbankhandbuch". In: Abruf: <http://www.littleidiot.de/mysql> (am 23.04. 2008) (1999).
- [35] Dominik Stocklasser. "Architektur".
- [36] Dominik Stocklasser. "Systemtechnikscherpunkt 4.KlasseSystementwicklungVersion 01.10Autor: Dominik Stocklasser".
- [37] Dominik Stocklasser. "Systemtechnikscherpunkt 4.KlasseVersionskontrolleVersion".
- [38] Jeff Tranter. *Control Raspberry Pi GPIO Pins from Python*. <https://ics.com/blog/control-raspberry-pi-gpio-pins-python>. [Online; accessed Wednesday, July 31, 2019]. 2019.

- [39] Paul Viola und Michael Jones. "Robust Real-time Object Detection". In: *International Journal of Computer Vision*. 2001.
- [40] 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].