

**Hochschule für Technik und Wirtschaft Dresden**

**Fakultät Informatik/Mathematik**

# Bachelorarbeit

**im Studiengang Wirtschaftsinformatik**

**Thema: Java-Web-Gesichtserkennungssystem für Studenten mit  
Pytorch**

Autor:

Ngoc Huy Le

Gutachter:

Prof. Dr. Mario Neugebauer

Zweitgutachter:

Dipl.-Medieninf. Alexander Wülfing

Bearbeitungszeit:

07.12.2023 bis 01.02.2024



# Inhaltsverzeichnis

<b>Inhaltsverzeichnis .....</b>	<b>III</b>
<b>Abbildungsverzeichnis.....</b>	<b>VI</b>
<b>Tabellenverzeichnis.....</b>	<b>VIII</b>
<b>Abkürzungsverzeichnis .....</b>	<b>IX</b>
<b>1 Einleitung .....</b>	<b>1</b>
1.1 Überblick über das Thema.....	1
1.2 Begründung des Themas.....	2
1.3 Methoden .....	2
1.4 Gegenstand und Umfang der Forschung .....	2
1.5 Materialbasis.....	3
<b>2 Untersuchungsziel.....</b>	<b>4</b>
2.1 Analyse der Anforderungen.....	4
2.2 Detailziele .....	5
2.2.1 Machine Learning.....	5
2.2.2 Webanwendung: .....	5
2.2.3 Ziele der Effizienz: .....	5
<b>3 Machine Learning.....</b>	<b>6</b>
3.1 Klassifizierung des maschinellen Lernens.....	6
3.1.1 Supervised Learning .....	7
3.1.2 Unsupervised Learning .....	7
3.1.3 Semi-Supervised Learning.....	10
3.1.4 Reinforcement Learning .....	10
3.2 Fazit .....	10
<b>4 Deep Learning.....</b>	<b>12</b>
4.1 Machine Learning vs Deep Learning .....	12
4.2 Artificial Neural Networks (ANN).....	13
4.3 Perceptron .....	14
4.4 Shallow neural network .....	15
4.5 Deep Neural Network .....	16
4.6 Convolution Neural Networks (CNN).....	17
4.6.1 Convolution Layer .....	18
4.6.2 Pooling Layer .....	19
4.6.3 Fully Connected Layer .....	19
4.7 Multi-task Convolutional Neural Network (MTCNN).....	20

4.7.1	P-Net.....	20
4.7.2	R-Net .....	20
4.7.3	O-Net .....	20
4.8	Bibliothek für maschinelles Lernen.....	22
4.8.1	TensorFlow .....	22
4.8.2	Keras.....	23
4.8.3	PyTorch .....	23
4.9	Fazit .....	23
<b>5</b>	<b>Face Recognition.....</b>	<b>24</b>
5.1	Face Detection .....	24
5.1.1	Feature-Based .....	25
5.1.2	Image-Based .....	25
5.1.3	Vergleich: MTCNN vs. Haar-Kaskade.....	25
5.2	Face Extraction .....	30
5.2.1	One-Shot-Learning .....	30
5.2.2	Learning similarity .....	30
5.2.3	Siam network.....	31
5.2.4	Facenet.....	32
5.2.4.1	Triplet Loss.....	32
5.2.4.2	Face alignment.....	33
5.2.4.3	Pre-trained Model.....	35
5.3	Face Classification.....	35
5.4	Fazit .....	36
<b>6</b>	<b>Lösungsvorschlag.....</b>	<b>38</b>
6.1	Analyse und Entwurf von Webanwendungen .....	38
6.1.1	Server-Side-Rendering vs Client-Side-Rendering.....	39
6.1.2	Single-Page-Application.....	40
6.1.3	MVC-Model .....	43
6.2	Client .....	45
6.2.1	Frontends Framework.....	45
6.2.1.1	VueJs .....	45
6.2.1.2	React .....	45
6.2.1.3	Angular .....	46
6.3	Server.....	46
6.3.1	Java .....	47
6.3.1.1	Java Spring Boot.....	47
6.3.1.2	Spring Security .....	48
6.3.2	Bilderordner.....	48
6.3.3	Datenbank.....	48
6.3.4	Face Recognition .....	49

6.3.4.1	Python.....	49
6.4	Implementierung.....	50
6.4.1	Datenbank.....	50
6.4.2	Bilderordner.....	51
6.4.3	Python.....	51
6.4.4	Client .....	53
6.4.5	Java.....	54
6.5	Fazit .....	58
<b>7</b>	<b>Programmablauf.....</b>	<b>59</b>
7.1	Programmablauf .....	59
7.2	Bewertung.....	64
<b>8</b>	<b>Zusammenfassung und Ausblick.....</b>	<b>66</b>
<b>9</b>	<b>Literaturverzeichnis .....</b>	<b>68</b>
<b>10</b>	<b>Selbstständigkeitserklärung.....</b>	<b>69</b>

## Abbildungsverzeichnis

<b>Abbildung 1:</b> Die Meilensteine des Deep Learning, .....	1
Abbildung 2: Verzweigungsstruktur des maschinellen Lernens.....	7
Abbildung 3: clustering in machinelearning. ....	8
Abbildung 4: Machine Learning vs Deep Learning.....	12
Abbildung 5: Machine Learning vs Deep Learning.....	13
Abbildung 6: Structures of a Neuron .....	13
Abbildung 7: The structure of a perceptron. ....	14
Abbildung 8: Shallow neural network .....	15
Abbildung 9: Deep Neural Network .....	16
Abbildung 10: CNN .....	17
Abbildung 11: Pooling Layer CNN / .....	19
Abbildung 12: P-net MTCNN.....	21
Abbildung 13: R-net - MTCNN.....	21
Abbildung 14: O-net MTCNN .....	22
Abbildung 15: Gesichtserkennung Prozess.....	24
Abbildung 16: Haar-Kaskade with OpenCV .....	26
Abbildung 17: MTCNN with facenet_pytorch .....	26
Abbildung 18: Haar-Kaskade Test Ergebnis.....	27
Abbildung 19: MTCNN Test Ergebnis .....	27
Abbildung 20: Haar-Kaskade.....	28
Abbildung 21: MTCNN. ....	28
Abbildung 22: Haar-Kaskade.....	29
Abbildung 23: MTCNN .....	29
Abbildung 24: Learning similaritiv.....	31
Abbildung 25: Siam network .....	31
Abbildung 26: Facenet .....	32
Abbildung 27: Triplet Loss .....	33
Abbildung 28: Face alignment code.....	34
Abbildung 29: Abstand zwischen 2 Vektoren .....	36
Abbildung 30 Use Case Diagram. ....	39
Abbildung 31 Component Diagram. ....	41
Abbildung 32 Class Diagram .....	42
Abbildung 33 MVC-Model.....	43
Abbildung 34 MVC-Diagramm .....	44
Abbildung 35: MySQL .....	50
Abbildung 36 Python .....	52
Abbildung 37: VueJS -Client.....	53
Abbildung 38 Vuex Diagram. ....	54

Abbildung 39: Javastruktur .....	55
Abbildung 40: Kommunikation zwischen Java und Python. ....	56
Abbildung 41: Repository custom .....	56
Abbildung 42 Login Filter .....	57
Abbildung 43 Custom Authentication Manager .....	57
Abbildung 44: Login .....	59
Abbildung 45 Welcome Page.....	60
Abbildung 46: Student Verwaltung .....	60
Abbildung 47 Edit.....	61
Abbildung 48: Testverwaltung.....	61
Abbildung 49 new Test .....	62
Abbildung 50: Test Detail.....	62
Abbildung 51: Gesichtserkennung .....	63
Abbildung 52 erfolgreich verifiziert .....	63
Abbildung 53 Verifizierung fehlgeschlagen .....	64
Abbildung 54: Testbewertung.....	64

## **Tabellenverzeichnis**

Tabelle 1: Body-Mass-Index 1 .....	9
Tabelle 2: Body-Mass-Index 2.....	9
Tabelle 3: Pre-trained Models .....	35
Tabelle 4: Python .....	52
Tabelle 5: API .....	55



## **Abkürzungsverzeichnis**

Artificial Neural Networks - ANN

Convolution Neural Networks - CNN

Multi-task Convolutional Neural Network - MTCNN

TensorFlow - TF

Fully Connected Layer - FCL

Convolutional Layer - CL

Pooling Layer - PL

JSON Web Tokens - JWT

Graphics Processing Unit - GPU

Central Processing Unit - CPU

Residual Neural Networks - ResNet

Dense Convolutional Network - DenseNet

Personal Computer - PC

Hyper Text Markup Language - HTML

Cascading Style Sheets - CSS

JavaScript - JS

Model-View-Controller - MVC

Single – Page- Application - SPA

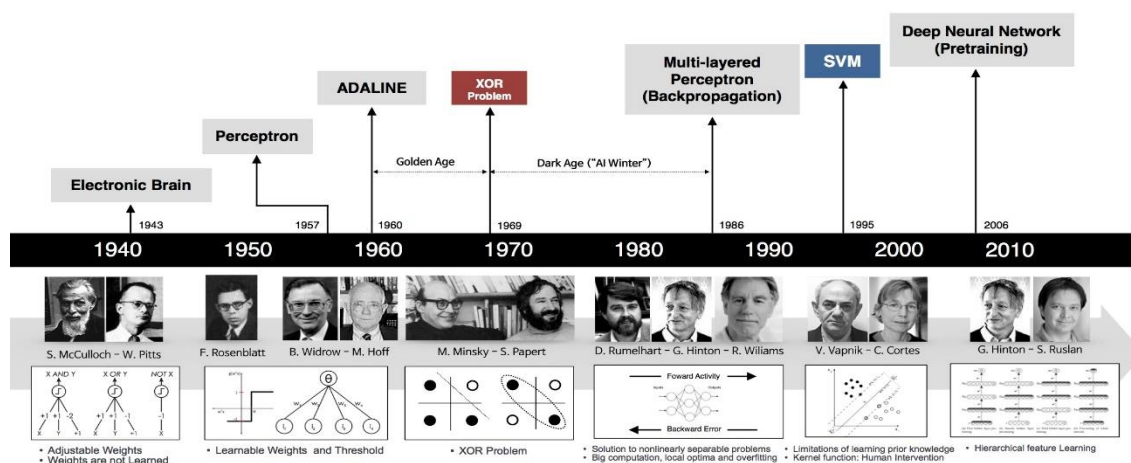
Server-Side-Rendering - SSR

Client-Side-Rendering - CSR

# 1 Einleitung

## 1.1 Überblick über das Thema

Künstliche Intelligenz hat eine lange Geschichte. Abbildung 1 zeigt die wichtigsten Meilensteine des Deep Learning. In den frühen 1940er Jahren wurden mit dem Aufkommen und der starken Entwicklung von Halbleiterbauelementen, elektronischen Komponenten und Computern die ersten Grundlagen für die Entstehung der künstlichen Intelligenz (KI) gelegt. Allerdings hatte die künstliche Intelligenz zu diesem Zeitpunkt noch keine praktischen Anwendungen oder herausragenden Leistungen vorzuweisen. Sogar Experten und Forscher mussten von 1960 bis 2000 eine schwierige Zeit durchmachen, die als *KI-Winter* bezeichnet wird, da es in dieser Zeit zu Stillstand und kaum Durchbrüchen kam.



**Abbildung 1:** Die Meilensteine des Deep Learning,

([http://beamlab.org/deeplearning/2017/02/23/deep\\_learning\\_101\\_part1.html](http://beamlab.org/deeplearning/2017/02/23/deep_learning_101_part1.html))

Im Jahr 2006 gelang Geoffrey E. Hinton der Durchbruch mit der Einführung des Unüberwachten Vortrainings durch Deep Believe Nets (DBN). Der Höhepunkt dieses Artikels ist die Schaffung eines künstlichen neuronalen Netzwerks mit vielen verborgenen Schichten (*hidden layer*) statt nur einer Schicht wie bisher. Von diesem Zeitpunkt an werden neuronale Netze mit vielen verborgenen Schichten als Deep Learning bezeichnet.

Die kontinuierliche Entwicklung in den Bereichen künstliche Intelligenz, maschinelles Lernen und Deep Learning dient nur einem einzigen Zweck, nämlich den Bedürfnissen der Menschen zu dienen. So helfen elektronische Geräte bei der Identifizierung biometrischer Daten (Fingerabdrücke, Netzhautscans, Gesichtserkennung usw.), bei der Wettervorhersage, der Diagnose von Krankheiten oder der Übersetzung verschiedener Sprachen. Um den komplexen und vielfältigen Aufgaben des Menschen gerecht zu werden, ist die künstliche Intelligenz in spezialisierte Bereiche wie die Verarbeitung natürlicher Sprache, Robotik, Computer Vision, usw. unterteilt.

Die Gesichtserkennung ist jedoch nach wie vor kein einfaches Gebiet. Jedes Gesicht hat einzigartige Merkmale, die das Training des Modells erschweren. Daher ist die Anwendung von Deep-Learning-Methoden erforderlich, wobei die Eingabedaten Gesichtsbilder sind, die in ein CNN eingespeist werden, das mit Modellen wie ResNet und DenseNet neu trainiert wurde.

## **1.2 Begründung des Themas**

Das Problem der Anwesenheit der Studenten bei Prüfungen ist oft sehr zeitaufwändig. Bei der klassischen Form der Anwesenheitskontrolle muss sich die verantwortliche Mitarbeiterin oder der verantwortliche Mitarbeiter auf den Studentenausweis mit dem Studentencode und dem Lichtbild verlassen, um zu überprüfen, ob es sich um den richtigen Studenten handelt. Außerdem muss er dann den Namen dieses Studenten in einer sehr langen Liste suchen und die Unterschriften der Studenten einholen. Dies ist mitunter sehr zeitaufwändig und setzt die Studenten manchmal unter Druck, wenn sie sich auf ihre Prüfungen konzentrieren müssen. Nach der Überprüfung muss die zuständige Mitarbeiterin oder der zuständige Mitarbeiter die Daten in das System zur Verwaltung eingeben. Genau aus diesem Grund und wegen der explosionsartigen Entwicklung von Deep Learning in der Gesichtserkennung hoffe ich, ein webbasiertes Programm zu entwickeln, das auf vielen Geräten - vom Handy bis zum PC - eingesetzt werden kann. Es bietet die Möglichkeit, die Mitarbeiter bei der einfachen Verwaltung und Erfassung der Anwesenheit von Studenten durch Gesichtserkennung in kurzer Zeit zu unterstützen und die Daten automatisch mit dem System zu synchronisieren. Dadurch können Personalressourcen und Papierkram reduziert und der Druck vor Prüfungen für die Studenten verringert werden.

## **1.3 Methoden**

Die verwendete Forschungsmethode ist die analytische und zusammenfassende Forschungsmethode. Konkret werden Recherchearbeiten sowie Dokumente zur Pytorch-Bibliothek, Java Spring Boot und VueJs durchgeführt, um die aufgeworfenen Probleme zu lösen.

## **1.4 Gegenstand und Umfang der Forschung**

Gegenstand und Umfang der Forschung sind Studierende der HTW Dresden. Aufgrund der Schwierigkeit, Fotos zu sammeln, und des Gesetzes zur Verwendung von Fotos habe ich jedoch Fotos von Verwandten und Personen verwendet, die zugestimmt haben. Diese Menschen haben unterschiedliche Altersgruppen und Gesichter, daher ist die Realität sehr positiv.

## 1.5 Materialbasis

Das Projekt wurde auf einem PC mit folgender Konfiguration durchgeführt:

- Betriebssystem: Windows 11 Pro 64-Bit-Version
- Prozessor: AMD Ryzen 5 2600 Six-core Processor
- Arbeitsspeicher: 32 GB RAM
- VRAM: Radeon RX 580 Series- 8 GB
- Kamera: Creative Live! Cam Sync HD

Um die Fotofunktion ausführen zu können, muss auf der Client-Seite eine Kamera an den Computer angeschlossen sein. Darüber hinaus gibt es auf der Client-Seite keine weiteren Konfigurationsanforderungen.

Auf der Serverseite sollte der Computer über die oben genannte Mindestkonfiguration verfügen, um effiziente Aufgaben wie Aufsätze durchzuführen. Darüber hinaus verfügt die facenet\_pytorch-Bibliothek über einen Mechanismus zur Nutzung der GPU anstelle der CPU, um die Bildextraktionsaufgabe besser zu erfüllen. Daher wird das Programm effektiver arbeiten, wenn der Server mit einem Nvidia-Grafikprozessor ausgestattet ist (nur auf Nvidia-Grafikprozessoren unterstützt).

## 2 Untersuchungsziel

### 2.1 Analyse der Anforderungen

Aus dem in Abschnitt 1.1 erwähnten Kontext ergibt sich die folgende Problemstellung:

" Wie kann ein effizientes webbasiertes System entwickelt werden, das die Prüfungsaufsicht bei der Anwesenheitskontrolle von Studierenden durch Gesichtserkennung mit PyTorch unterstützt und gleichzeitig effektives Studentenmanagement und Prüfungsverwaltung ermöglicht?".

Ausgehend von der obigen Zielfrage können wir sie in zwei Teilprobleme aufteilen.

Problem 1: Wie kann Pytorch zur Erkennung von Gesichtern eingesetzt werden, um die Anwesenheit von Studenten durch Gesichtserkennung zu erfassen? [1]

Problem 2: Wie man ein stabiles und sicheres Websystem aufbaut, das als Plattform für die Verwaltung von Studenten, die Verwaltung von Prüfungen und die Anwesenheit von Studenten durch Gesichtserkennung dient. [2]

Das erste Problem ist bekanntlich, dass die Gesichtserkennung für die Anwesenheit vor Prüfungen immer eine hohe Genauigkeit und innerhalb der zulässigen Zeit erfordert. Denn wenn die Genauigkeit gering oder die Ausführungszeit zu lang ist, kann die Anwendung keine praktische Lösung sein. Darüber hinaus ist die Gesichtserkennung, wie im vorherigen Abschnitt erwähnt, ein Bereich des maschinellen Lernens und des Deep Learning. Um dieses Problem lösen zu können, müssen daher grundlegende Kenntnisse über Algorithmen des maschinellen Lernens und des Deep Learning erworben werden. Danach muss der Algorithmus ausgewählt werden, der den Anforderungen des Problems entspricht. Dies gewährleistet eine kurze Ausführungszeit und eine hohe Genauigkeit.

Für das zweite Problem müssen wird eine effektive Außerdem braucht die Gesichtserkennungsfunktion eine Website-Plattform, in die sie integriert werden kann. Es wäre sinnlos, wenn das Programm nur über die Anwesenheitsfunktion mit Gesichtserkennung verfügen würde, denn dann wäre es sehr unbequem, wenn sich die Zahl der Studenten erhöht oder ein neuer Test nach den tatsächlichen Bedürfnissen erstellt wird. Daher benötigt das Programm weitere grundlegende Funktionen wie das Hinzufügen, Bearbeiten, Löschen und Suchen, um Studenten und Tests zu verwalten. Dies hilft dem Programm, in der Praxis wirklich nützlich zu sein. Aus diesem Grund muss ein starkes, stabiles Webanwendungssystem entwickelt werden, das den Aufbau der oben genannten Funktionen gut unterstützt. Neben dem Aspekt der Benutzerfreundlichkeit muss auch die Sicherheit an erster Stelle stehen. Da ein schwer zu bedienendes Programm unpraktisch ist, muss

ein Gesichtserkennungsprogramm über eine wirklich gute Sicherheit verfügen, um mögliche Fehler während der Ausführung des Programms zu vermeiden.

Zusammenfassend lässt sich sagen, dass zur Beantwortung von Zielfrage die beiden oben genannten Punkte gründlich geklärt werden müssen. Die Struktur dieser Arbeit konzentriert sich ebenfalls auf die Beantwortung der beiden oben genannten Fragen. Kapitel 3, Kapitel 4 und Kapitel 5 konzentrieren sich auf das Verständnis des Grundwissens, der Algorithmen des maschinellen Lernens und des Deep Learning sowie auf die Experimente, die zur Auswahl des am besten geeigneten Algorithmus zur Lösung von Problem 1 durchgeführt werden.

Kapitel 6 konzentriert sich auf den Aufbau einer Webanwendung, in der Funktionen wie Studentenverwaltung, Testmanagement und Anwesenheit durch Gesichtserkennung implementiert sind.

Kapitel 7 wird praktische Anwendungstests sein.

## **2.2 Detailziele**

### **2.2.1 Machine Learning**

Grundkenntnisse in Geschichte sowie Algorithmen für maschinelles Lernen und Deep Learning, insbesondere Algorithmen, die die Gesichtserkennung unterstützen. Wählen auf dieser Grundlage den geeigneten Algorithmus aus und setzen diese mit Unterstützung der face\_pytorch-Bibliothek ein.

#### **2.2.2 Webanwendung:**

Ziel ist der Aufbau eines leistungsstarken und sicheren Webanwendungssystems zur Durchführung von Studentenverwaltungsfunktionen, zur Verwaltung von Tests und zur Identifizierung von Studenten über Gesichtserkennung..

#### **2.2.3 Ziele der Effizienz:**

1. Die angestrebte Erfolgsquote bei der Anwesenheitserfassung durch Fotos liegt bei über 80 %.
2. Die gewünschte Zeit für die Anwesenheitserfassung durch Bilder beträgt weniger als 5 Sekunden.
3. Erstellen einer Webanwendung und Verfassen einer Arbeit innerhalb von zwei Monaten

### 3 Machine Learning

Wie bereits erwähnt, beginnt dieses Kapitel mit den grundlegendsten Kenntnissen und Algorithmen des maschinellen Lernens. Dies wird einen Überblick über diesen boomenden Bereich geben.

Heutzutage wird das maschinelle Lernen immer beliebter und findet in vielen Bereichen des Lebens Anwendung. Dazu gehören Gesichtserkennung, selbstfahrende Autos, Amazon-Einkaufsempfehlungen und eine Vielzahl anderer Anwendungen. Beim maschinellen Lernen geht es darum, Systeme zu schaffen, die in der Lage sind, selbst zu lernen, ohne dass eine spezielle Programmierung erforderlich ist. Es hat die Fähigkeit, auf der Grundlage der Erfahrungen, die es während des Betriebs sammelt, selbst zu lernen, was die menschlichen Denktivitäten simuliert.

*„The field of study that gives computers the ability to learn without being explicitly programmed.“* (Arthur Samuel, 1959)

*„A computer program is said to learn from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $P$ , if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$ .“* (Tom Mitchell, 1997)

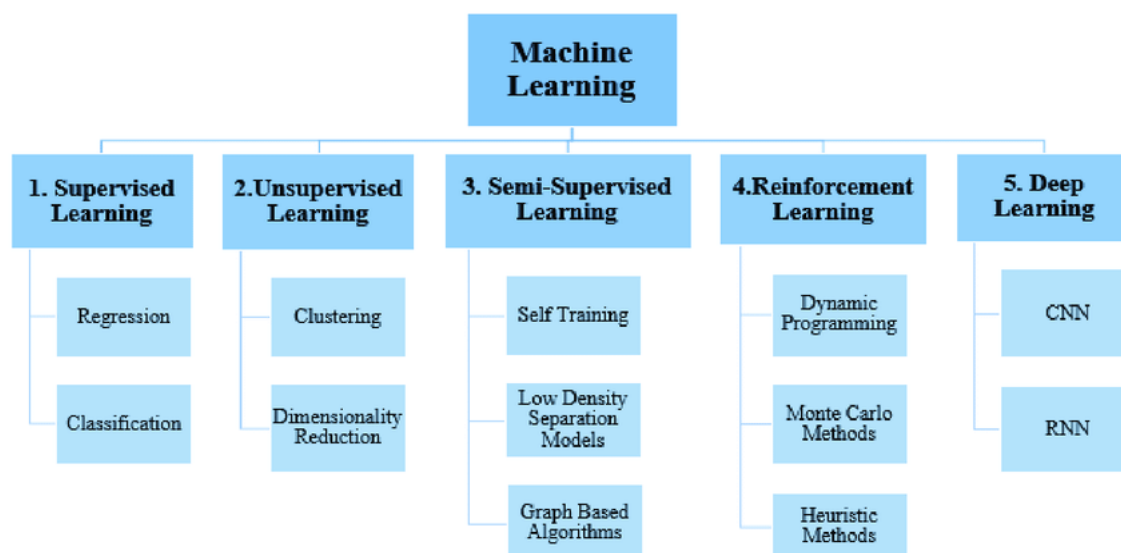
Zum Beispiel: AlphaGo wurde von Google DeepMind entwickelt und ist für die Spielen von Go programmiert. Am 15. März 2016 besiegte AlphaGo den 18-jährigen Go-Weltmeister Lee Seedol. AlphaGo gewann vier von fünf Spielen.

- E: Erfahrung aus dem Spielen mit Gegnern und der Selbstverstärkung.
- T: Die Aufgabe besteht darin, Go zu spielen.
- P: Die Wahrscheinlichkeit, dass AlphaGo das nächste Spiel gewinnt.

Die Gesichtserkennung ist ebenfalls ein wichtiges Thema des maschinellen Lernens und steht auch im Mittelpunkt dieser Arbeit. Bevor man jedoch in dieses Gebiet einsteigt, muss man sowohl Grundkenntnisse als auch wichtige theoretische Grundlagen des maschinellen Lernens kennen.

#### 3.1 Klassifizierung des maschinellen Lernens

Maschinelles Lernen wird in viele verschiedene Lernmethoden unterteilt.



**Abbildung 2:** Verzweigungsstruktur des maschinellen Lernens ([https://www.researchgate.net/figure/Different-machine-learning-types-and-algorithms\\_fig6\\_330815113](https://www.researchgate.net/figure/Different-machine-learning-types-and-algorithms_fig6_330815113))

### 3.1.1 Supervised Learning

Überwachtes Lernen ist ein Algorithmus, der die Ausgabe einer neuen Eingabe (new Input) auf der Grundlage von zuvor bekannten (data, label) Paaren vorhersagt. Supervised Learning wird in zwei Typen unterteilt:

- A. Regression: ist eine Art überwachter Algorithmus für maschinelles Lernen, der dabei hilft, eine kontinuierliche Menge vorherzusagen  
Zum Beispiel: Das Alter einer Person muss anhand eines Fotos bestimmt werden.
- B. Classification: ist ein Algorithmus des überwachten maschinellen Lernens, der diskrete Klassenbezeichnungen vorhersagt  
Zum Beispiel: Basierend auf Informationen aus Bluttests, Größe, Gewicht, Blutdruck usw., kann vorhergesagt werden, ob eine Diabeteserkrankung vorliegt oder nicht.

Zum Beispiel hat Microsoft kürzlich eine Anwendung entwickelt, die Geschlecht und Alter anhand von Gesichtern vorhersagt. Der Teil zur Geschlechtshvorhersage kann als Klassifizierungsalgorithmus betrachtet werden, der Teil zur Altersvorhersage kann als Regressionsalgorithmus betrachtet werden.

### 3.1.2 Unsupervised Learning

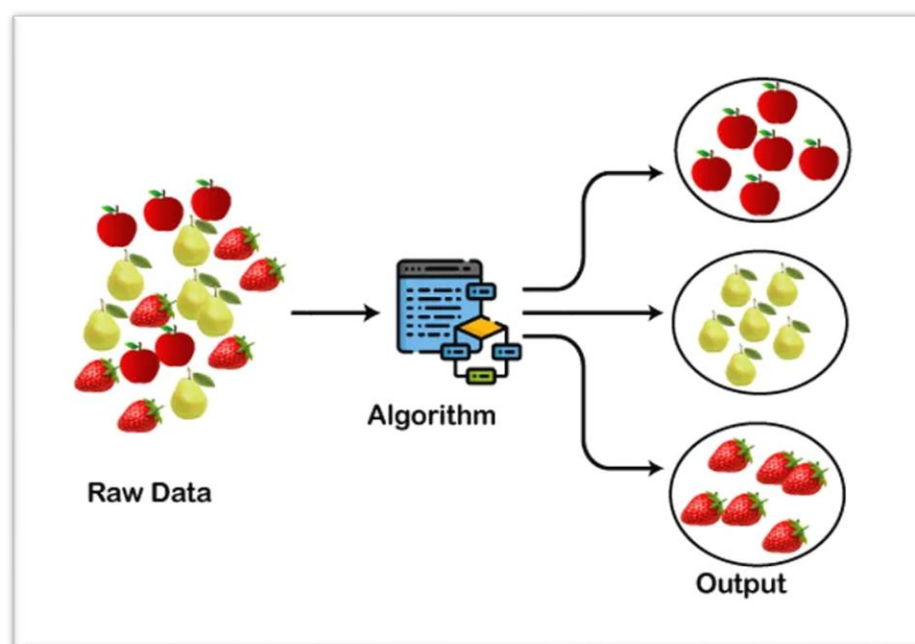
Bei diesem Algorithmus ist weder das Ergebnis noch die Bezeichnung bekannt, sondern nur die Eingabedaten. Der Algorithmus für unüberwachtes Lernen stützt sich auf die



Struktur der Daten, um eine bestimmte Aufgabe zu erfüllen, z. B. Clustering oder Dimensionsreduktion zur bequemen Speicherung und Berechnung.

Supervised Learning wird in zwei Typen unterteilt:

- A. Clustering: Clustering ist eine Art von unbeaufsichtigter maschineller Lerntechnik, die dazu dient, ähnliche Datenpunkte auf der Grundlage ihrer inhärenten Eigenschaften oder Merkmale ohne Kennzeichnungen (Label) zusammenzufassen. Wie in Abbildung 3 zu sehen ist, handelt es sich bei den Eingabedaten um Früchte (Tomaten, Zitronen, Erdbeeren). In diesem Clustering-Algorithmus werden sie jedoch nicht gekennzeichnet (Tomate, Zitrone, Erdbeere). Der Algorithmus klassifiziert sie automatisch anhand der Merkmale, die sie während des Trainingsprozesses entdecken.



**Abbildung 3:** clustering in machinelearning. Quelle: javatpoint, entnommen von <https://static.javatpoint.com/tutorial/machine-learning/images/clustering-in-machine-learning.png>

- B. Die Dimensionalitätsreduktion ist ein Prozess, bei dem die Anzahl der Merkmale oder Variablen in einem Datensatz reduziert wird, während die wichtigen Informationen erhalten bleiben.

Zum Beispiel: Es liegt eine Datentabelle vor, die Geschlecht, Größe und Gewicht umfasst. Basierend auf diesen Daten muss festgestellt werden, ob die betreffende Person fettleibig oder unterernährt ist.

**Tabelle 1: Body-Mass-Index 1**

Geschlecht	Größe in cm	Gewicht in kg
männlich	189	87
männlich	185	110
weiblich	195	104
weiblich	172	67
...	...	...

Diese Angaben reichen nicht aus, um festzustellen, ob die betreffende Person fettleibig ist oder nicht. Der BMI von Adolphe Quetelet kann jedoch angewendet werden:

$$BMI = \frac{w}{h^2}$$

**Tabelle 2: Body-Mass-Index 2**

Geschlecht	BMI	Klassifizierung
männlich	24,35	Normalgewicht
männlich	32,14	Adipositas Grad I
weiblich	27,35	Übergewicht
weiblich	21,87	Normalgewicht
...	...	...

BMI < 18,5	Untergewicht
BMI 18,5 – 24,9	Normalgewicht
BMI 25 – 29,9 kg/m <sup>2</sup>	Übergewicht
BMI 30 – 34,9 kg/m <sup>2</sup>	Adipositas Grad I
BMI 35 – 39,9 kg/m <sup>2</sup>	Adipositas Grad II
BMI ≥ 40 kg/m <sup>2</sup>	Adipositas Grad III

**Abbildung 5:** BMI-Quelle, Adipositas-Gesellschaft, entnommen von <https://adipositas-gesellschaft.de/bmi>

Durch die obige Formel wird die Dimension des Problems von drei (Geschlecht, Gewicht, Größe) auf zwei Dimensionen reduziert: Geschlecht und BMI.

### 3.1.3 Semi-Supervised Learning

Semi-überwachtes Lernen ist eine breite Kategorie von maschinellen Lerntechniken, die sowohl gekennzeichnete als auch nicht gekennzeichnete Daten verwenden; auf diese Weise ist es, wie der Name schon sagt, eine Hybridtechnik zwischen überwachtem und nicht überwachtem Lernen.

Dies ist häufig der Fall, da die Kennzeichnung oft enorme Ressourcen und Kosten erfordert.

### 3.1.4 Reinforcement Learning

Unter den Lernmethoden läuft Supervised Learning auf dem exakten Datensatz (alle Daten sind beschriftet). Unsupervised Learning stützt sich bei der Klassifizierung auf Ähnlichkeiten in den Eingaben. Wenn der Algorithmus jedoch falsch ist, kann sich die Maschine nicht selbst verbessern. Daher geht das verstärkende Lernen von der Situation aus und entwickelt Strategien, um den größten Nutzen zu erzielen (Maximierung der Leistung).

AlphaGo, ein von Google DeepMind entwickeltes Programm, hat im Go-Spiel menschliche Gegner herausgefordert und besiegt. Was die Komplexität betrifft, so hat Go insgesamt  $10^{761}$  Variablen, mehr als Schach, das nur  $10^{120}$  Variablen hat. Das Ziel von AlphaGo ist es, den besten Zug unter allen möglichen Zügen zu finden. Daher enthält AlphaGo Algorithmen, die sowohl zum Supervised Learning als auch zum Reinforcement Learning gehören. Beim Supervised Learning wird AlphaGo mit Spielen geladen, die Menschen bereits durchgeführt haben. Aber um gegen Menschen zu gewinnen, muss AlphaGo Reinforcement Learning durchführen, um seine eigenen Fähigkeiten zu verbessern, indem es mit sich selbst spielt.

## 3.2 Fazit

Traditionelles maschinelles Lernen ist die Fähigkeit von Computern, Aufgaben auszuführen, ohne ausdrücklich programmiert zu werden. Allerdings denken und handeln Computer immer noch mechanisch. Daher sind sie bei Aufgaben, die schwierig sind und menschliches Eingreifen erfordern, um Merkmale für die Klassifizierung zu extrahieren (features extraction), sehr eingeschränkt. Deshalb braucht es nicht nur eine große Menge an Eingabedaten, sondern auch die Unterstützung von Experten, um die Merkmale für das System auszuwählen. Dies führt zu enormen Kosten.

Aufgrund der sehr langsamen und nicht bahnbrechenden Entwicklung der Hardware war der KI-Winter eine Zeit, in der die Branche des maschinellen Lernens fast eingefroren war und es fast keine bedeutenden Fortschritte gab. Der Gesichtserkennungsalgorithmus

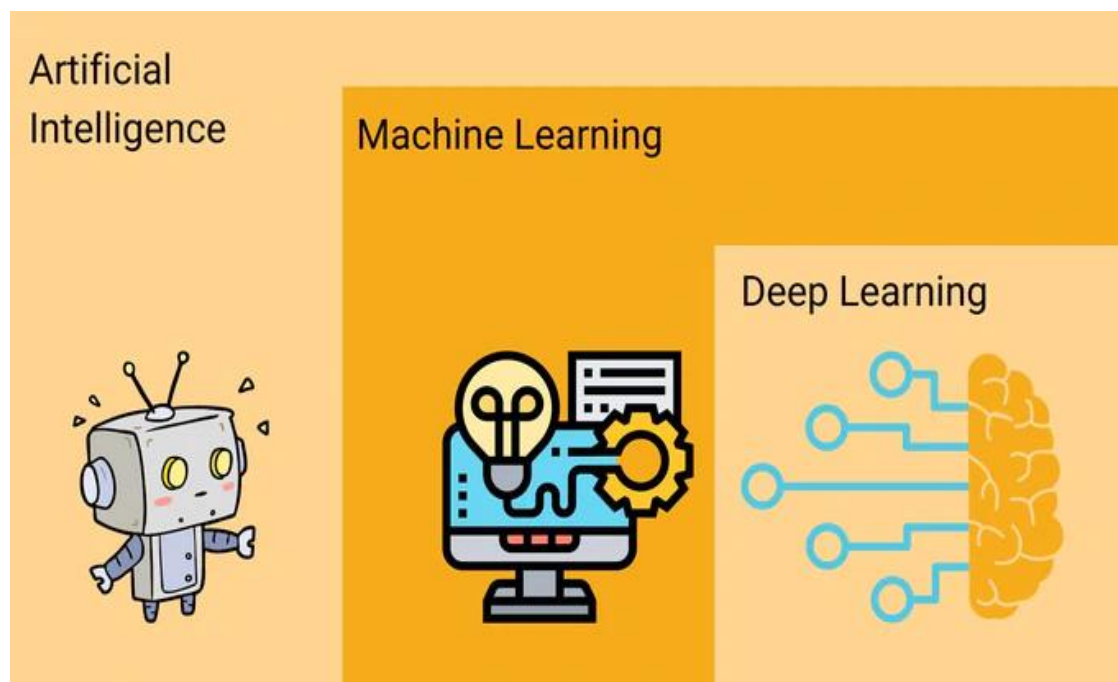
war damals keine Ausnahme. Gesichtserkennungsalgorithmen, das traditionelle maschinelle Lernen anwenden, wie z. B. Haar Kaskade (der Algorithmus wird in den nächsten Kapiteln mit Gesichtserkennungsalgorithmen mit Deep Learning verglichen), können zwar Gesichter erkennen, die Geschwindigkeit ist schnell, aber es gibt immer noch Schwächen der traditionellen maschinellen Lernalgorithmen. Erstens ist es ressourcenaufwändig, wenn man die Merkmale direkt extrahieren und beschriften muss. Zweitens ist die Genauigkeit bei komplexen Bildern nicht gut (es gibt viele Gesichter auf dem Bild, das Bild ist nicht von guter Qualität usw.).

Das Aufkommen der Deep-Learning-Theorie und die starke Hardware-Entwicklung in der letzten Zeit haben jedoch die Entwicklung des maschinellen Lernens gefördert. Außerdem wird sie neue Ansätze und leistungsfähige Algorithmen für die Gesichtserkennung bereitstellen. Kenntnisse über Deep Learning werden im nächsten Kapitel behandelt.

## 4 Deep Learning

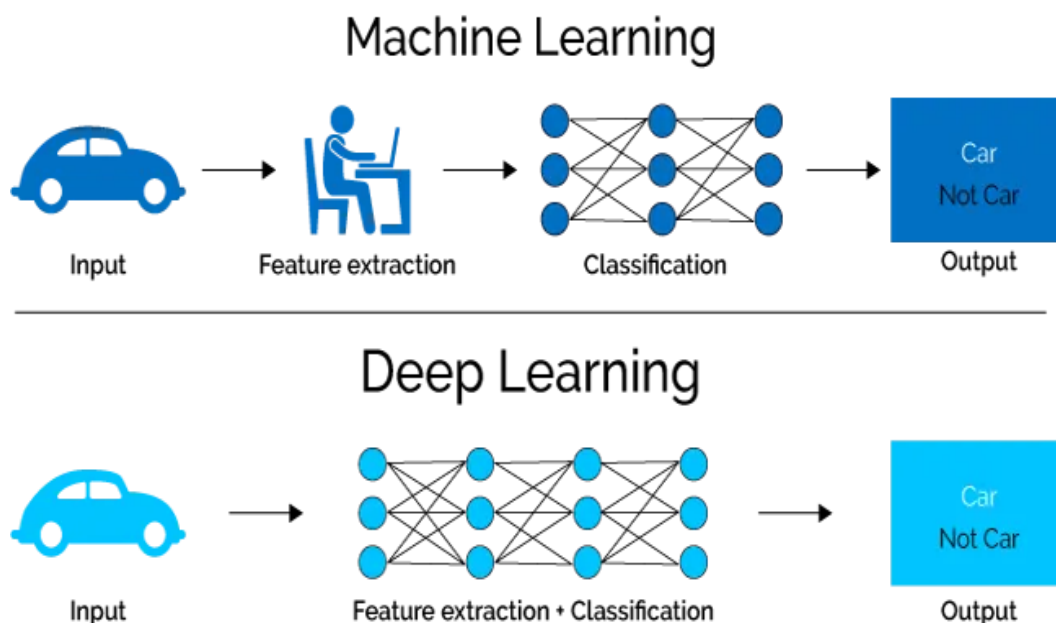
### 4.1 Machine Learning vs Deep Learning

Wie bereits im vorigen Kapitel erwähnt, ist Deep Learning ein Teil des maschinellen Lernens, aber Deep Learning hat bedeutende Durchbrüche erzielt. Bei herkömmlichen Methoden des maschinellen Lernens müssen Menschen Merkmale extrahieren (features extraction). Das System wendet diese Merkmale anschließend auf die Eingabedaten an und verwendet Klassifizierungsmethoden, um eine Ausgabe zu erstellen (Abbildung 4 und 5). Diese Methode ist schnell, aber bei schwierigen Aufgaben nicht sehr genau, und die Merkmalsextraktion erfordert einen hohen Personal- und Kostenaufwand. Deep Learning hat jedoch einen völlig anderen Ansatz. In Deep-Learning-Systemen gibt es künstliche neuronale Netze, die lernen, Merkmale zu extrahieren und selbst zu klassifizieren, um eine Ausgabe zu erzeugen. Früher war dies sehr schwierig, aber mit den aktuellen Hardware-Entwicklungen ist es definitiv möglich. Dies erklärt auch, warum sich das Deep Learning derzeit parallel zur Hardwareexplosion rasant entwickelt. Dies zeigt seine Überlegenheit gegenüber den traditionellen Methoden des maschinellen Lernens.



**Abbildung 4:** Machine Learning vs Deep Learning

(Quelle: <https://cmcglobal.com.vn/digital-transformation/machine-learning-vs-deep-learning-whats-the-difference/>)

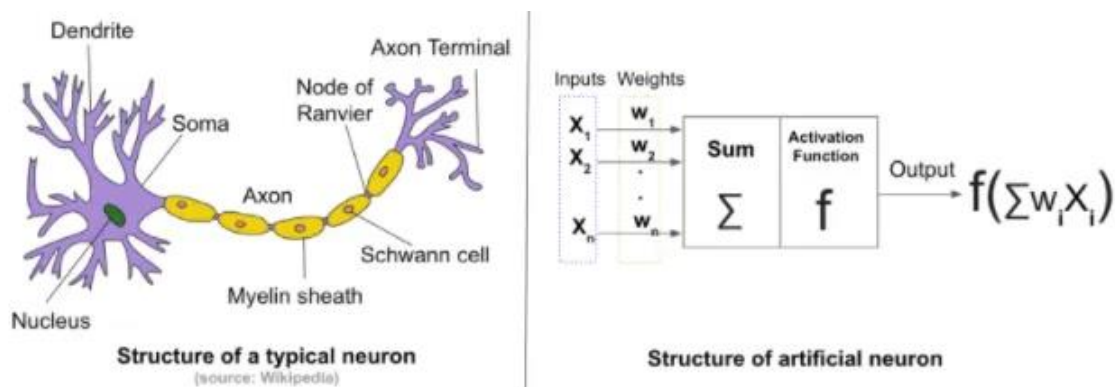


**Abbildung 5:** Machine Learning vs Deep Learning

([https://miro.medium.com/v2/resize:fit:828/format:webp/1\\*oesKtUPU54Xd\\_gP7UWuoCw.png](https://miro.medium.com/v2/resize:fit:828/format:webp/1*oesKtUPU54Xd_gP7UWuoCw.png))

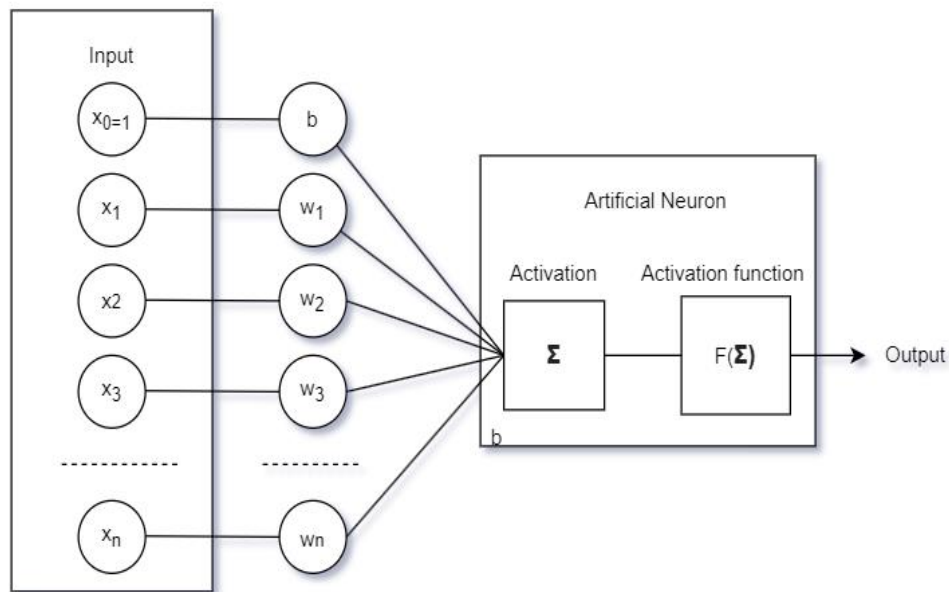
## 4.2 Artificial Neural Networks (ANN)

Deep Learning ist eine Methode, die von der Funktion des menschlichen Gehirns inspiriert ist (Artificial Neural Networks - ANN). Deep-Learning-Algorithmen ähneln der Art und Weise, wie sich Neuronen im Gehirn miteinander verbinden und Informationen übertragen. (Detaillierte Bedeutungen der Begriffe finden sich in Abschnitt 4.3)



**Abbildung 6:** Structures of a Neuron (<https://medium.com/@eraiitk/brain-and-artificial-neural-networks-differences-and-similarities-1d337fe50168>)

### 4.3 Perceptron



**Abbildung 7:** Struktur eines perceptron. (Abbildung vom Autor, erstellt mit draw.io)

Ein Perceptron ist eine Art künstliches neuronales Netz (ANN).

- Die Werte  $x_1, x_2 \dots x_n$  dienen als Eingabe (Input).
- Die Gewichte  $w_1, w_2, \dots w_n$  stellen die Wichtigkeit der einzelnen Eingaben dar. Je höher ihr Wert, desto wichtiger sind sie.
- Der Bias ist eine Abweichung mit einem numerischen Wert. Er wird hinzugefügt, um sicherzustellen, dass nach dem Durchlaufen der Aktivierungsfunktion  $f$  kein Wert von 0 erreicht wird.
- Die Summe der gewichteten Eingaben  $\Sigma$  ist die Summe aller Eingabewerte multipliziert mit ihren jeweiligen Gewichten:

$$z = x_1w_1 + x_2w_2 + x_3w_3 + \dots + x_nw_n + b$$

- Dann durchläuft es die Aktivierungsfunktion  $f$  und gibt die Ausgabe zurück.

Output  $y$ :

$$y = f(x_1w_1 + x_2w_2 + x_3w_3 + \dots + x_nw_n + b)$$

Die Aktivierungsfunktion verarbeitet die Summe der  $z$  Gewichte, um die Ausgabe des Perceptrons zu bestimmen. Aktivierungsfunktionen können viele Arten von Funktionen anwenden, aber für traditionelle Perceptrons wird häufig eine Heaviside-funktion verwendet. Bei der Stufenfunktion hängt der Rückgabewert vom Vergleich zwischen dem  $z$ -Wert und dem Schwellenwert ab.

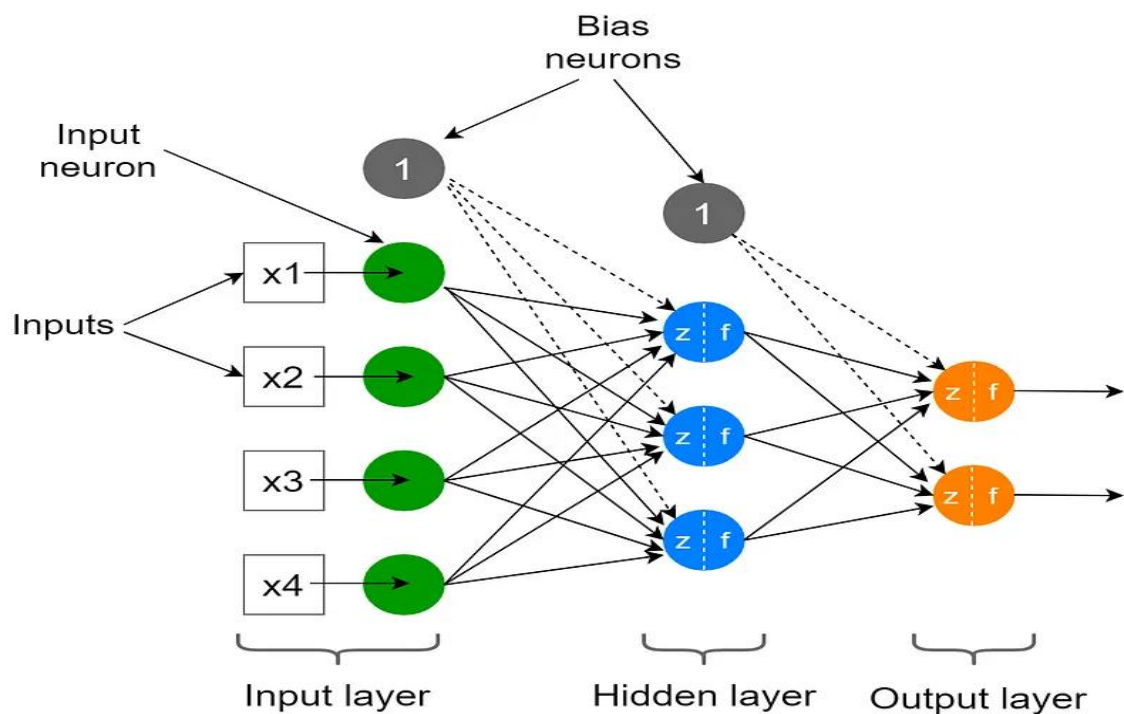
Heaviside-Funktion:  $output = \begin{cases} 1 & \text{wenn } z \geq threshold \\ 0 & \text{wenn } z < threshold \end{cases}$

Das Ziel des neuronalen Netzwerktrainingsprozesses besteht darin, dass das System die optimalen Gewichte  $w$  und die optimale Vorspannung findet.

(Weitere Informationen finden sich auf der Website: Christoph Kempkes, <https://medium.com/@wahlschwabe/das-neuron-und-die-aktivierungsfunktion-grundlagen-und-beispiele-an-logischen-bausteinen-4ef96c13d5d0>)

#### 4.4 Shallow neural network

Die Perceptrons sind sehr nützlich und interessant bei der Modellierung der Aktivität von Neuronen im menschlichen Gehirn. Einzelne Perceptrons reichen jedoch nicht aus, um komplexe Beziehungen und Probleme zu lösen. Daher werden viele Perceptrons zusammengefügt, um eine komplexe Struktur zu schaffen, die als künstliches neuronales Netz (Artificial Neural Network – ANN) bezeichnet wird. Mit dieser Struktur können diese komplexen Probleme angegangen werden.



**Abbildung 8:** Shallow neural network (Rukshan Pramoditha, <https://medium.com/data-science-365/one-hidden-layer-shallow-neural-network-architecture-d45097f649e6>)

- Input Layer: Die Eingabeschicht umfasst Eingabeneuronen und empfängt die Werte  $x_1, x_2 \dots x_n$ . Diese Neuronen führen keine Aktivierungsfunktionen aus. Mit anderen Worten: Sie geben die gleichen Werte aus wie ihre Eingaben.

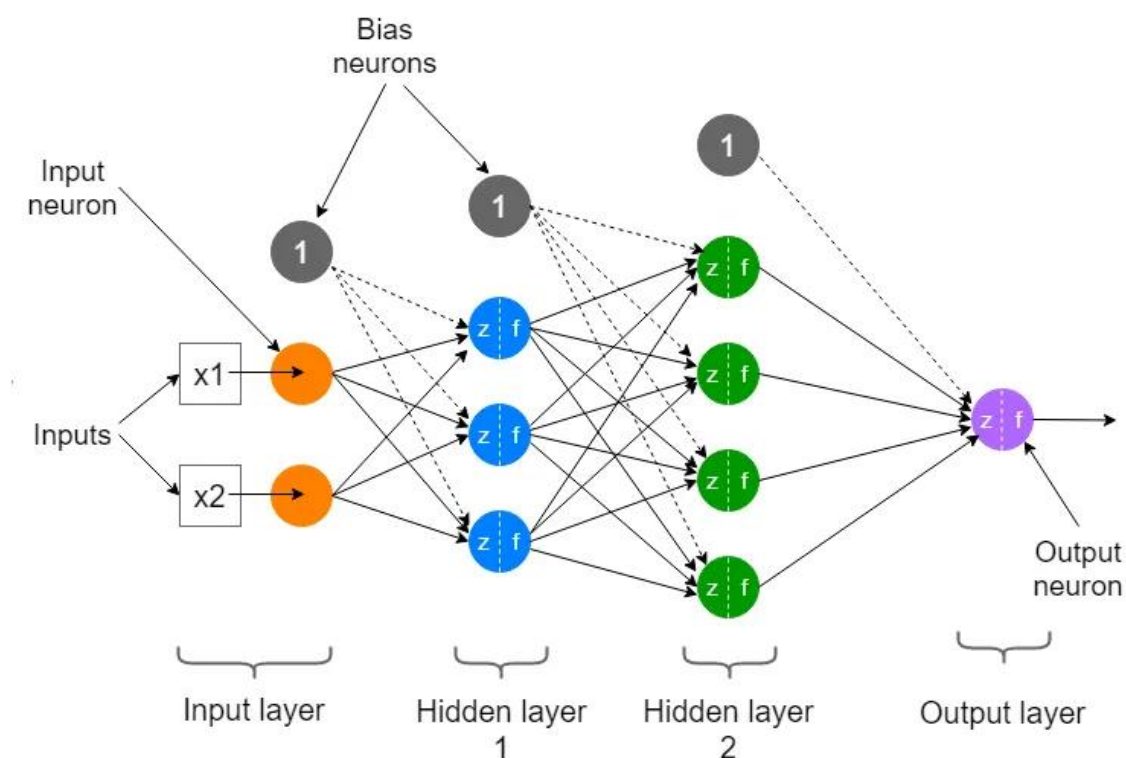


- Hidden Layer: Die verborgene Schicht ist die Schicht zwischen der Eingabeschicht und der Ausgabeschicht. In einem Shallow neural network hat es nur eine versteckte Schicht. Hier werden die Eingabewerte tatsächlich durch die Gewichte in der Funktion  $z$  verarbeitet und dann der Aktivierungsfunktion  $f$  zugeführt.
- Output Layer: Die Ausgabeschicht besteht aus den Ausgangsneuronen, die die endgültige Vorhersage machen. Es ist wichtig, die Anzahl der Ausgangsneuronen zu bestimmen.

Zum Beispiel: Benötigt die Ausgabe bei dem binären Problem, ob es sich bei einem Eingabebild um einen Hund handelt oder nicht, nur ein Neuron, um den vorhergesagten Wert (möglicherweise eine Wahrscheinlichkeit) darzustellen, ob es sich um einen Hund handelt oder nicht.

## 4.5 Deep Neural Network

Wenn ein ANN-Netz zwei oder mehr verborgene Schichten hat, wird es als tiefes neuronales Netz (Deep Neural Network) bezeichnet.



**Abbildung 9: Deep Neural Network** (Rukshan Pramoditha, <https://medium.com/data-science-365/two-or-more-hidden-layers-deep-neural-network-architecture-9824523ab903>)

In tiefen neuronalen Netzen werden die Ausgabewerte durch viele versteckte Schichten berechnet. Der Ausgabewert einer versteckten Schicht ist der Eingabewert für die nächste versteckte Schicht. Durch die Kombination einer Reihe versteckter Schichten können

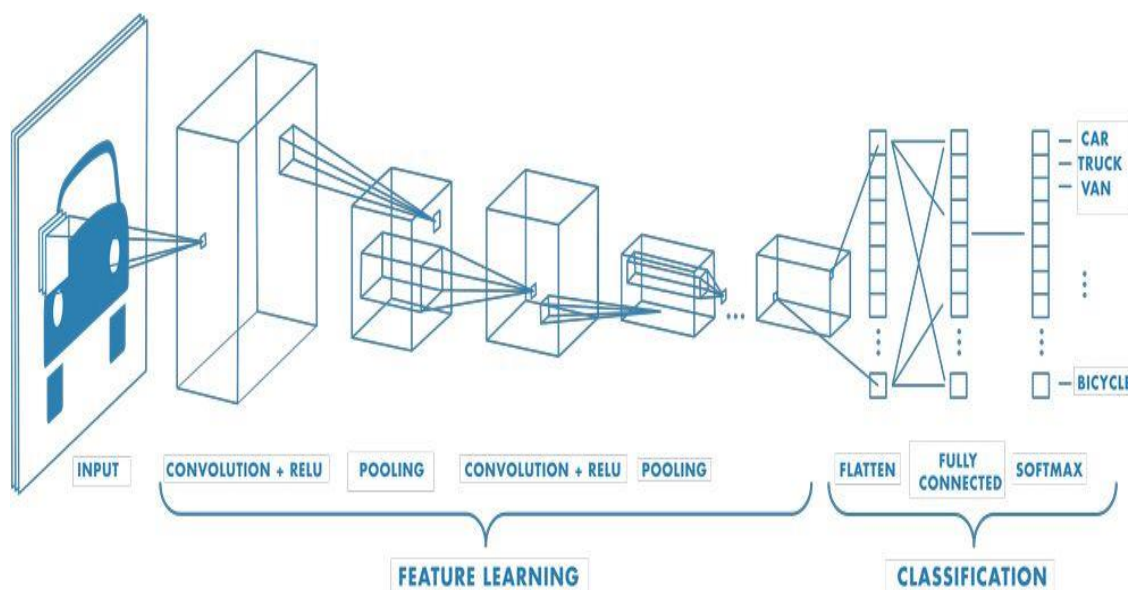
komplexe Probleme bewältigt werden, die während des Trainingsprozesses komplexe Gewichtungs- und Verzerrungsanpassungen erfordern.

Das ANN-Netz hat seine Überlegenheit bewiesen. Das Problem bei unserem Gesichtserkennungsproblem ist jedoch, dass das ANN-Netz nicht in der Lage ist, eigenständig Merkmale zu extrahieren. Stattdessen muss der Mensch immer noch manuell sinnvolle Merkmale extrahieren, bevor er sie in das ANN einspeist. Bei der Bildverarbeitung beispielsweise müssen die Bildpixel in aussagekräftige Merkmale wie Farbe umgewandelt werden, um die Leistung zu verbessern. Dies erfordert enorme menschliche Ressourcen.

Aus diesem Grund wird das CNN-Netzwerk zur Lösung dieses Problems eingesetzt.

## 4.6 Convolution Neural Networks (CNN)

Im Wesentlichen ist CNN ein erweitertes Netzwerk von ANN, das bei der eingehenden Bildverarbeitung hilft. Obwohl ANN in der Lage ist, automatisch Merkmale zu extrahieren, wandelt ANN bei der Bildverarbeitung ein zweidimensionales Bild zunächst in einen eindimensionalen Vektor um und speist ihn dann in das Trainingsmodell ein. Dadurch gehen die räumlichen Merkmale (spatial features) verloren. Dies ist ein Merkmal, das sich auf die Anordnung der Pixel in einem Bild bezieht. Dies führt dazu, dass ANN-Bilder ineffizient verarbeiten. Außerdem führt die Umwandlung eines zweidimensionalen Bildes in einen eindimensionalen Vektor zu einer enormen Anzahl von Eingabeparametern (wenn man um Beispiel ein Bild mit  $120 \times 120$  Pixeln hat, führt die Umwandlung in einen Vektor zu 14 400 Parametern). Inzwischen eignet sich CNN besonders für die Verarbeitung von Daten mit räumlichen Merkmalen, wie z. B. Bilder.



**Abbildung 10:** CNN <https://www.mathworks.com/discovery/convolutional-neural-network.html>

Wenn man das Foto einer Person betrachtet, kann man dank räumlicher Merkmale leicht erkennen, wo sich die Augen, die Nase usw. befinden. CNN hat dies ebenfalls getan. Das Bild wird mit den Filtern  $3 \times 3$ ,  $5 \times 5$ ,  $7 \times 7$  gescannt, so dass die räumliche Beziehung der Pixel im Bild erhalten bleibt. Ein einfacher CNN besteht aus drei Hauptschichten:

- Convolution Layer
- Pooling Layer
- Fully Connected Layer

#### 4.6.1 Convolution Layer

Die Faltungsschicht ist eine wichtige Schicht im CNN. Die Hauptaufgabe ist die Extraktion von Merkmalen des Eingangsbildes durch eine Schicht, die Filter (Kernel) genannt wird.

Das Bild wird zunächst als Pixelmatrix in das CNN eingespeist.

Das CNN verwendet Filter, die  $3 \times 3$ ,  $5 \times 5$  oder  $7 \times 7$ -Matrizen mit anfänglich zufällig generierten Werten sind. Durch den Lernprozess (Backpropagation und Optimierungsalgorithmen wie Gradient Descent) passt sich das CNN selbst an, um diesen Kernel zu optimieren. So kann CNN-Merkmale extrahieren und gleichzeitig die lokalen räumlichen Merkmale des Bildes erhalten.

CNN verschiebt die Filtermatrix über das Bild. Führen Sie eine Matrixmultiplikation von Filter und Bild an dieser Stelle durch und addieren Sie dann die Werte, um die Merkmalskarte zu erstellen.

Ein Beispiel: Ein in Pixeln ausgedrücktes Bild und einen Filterkern, der eine  $3 \times 3$ -Matrix ist, wie folgt:

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

1	0	1
0	1	0
1	0	1

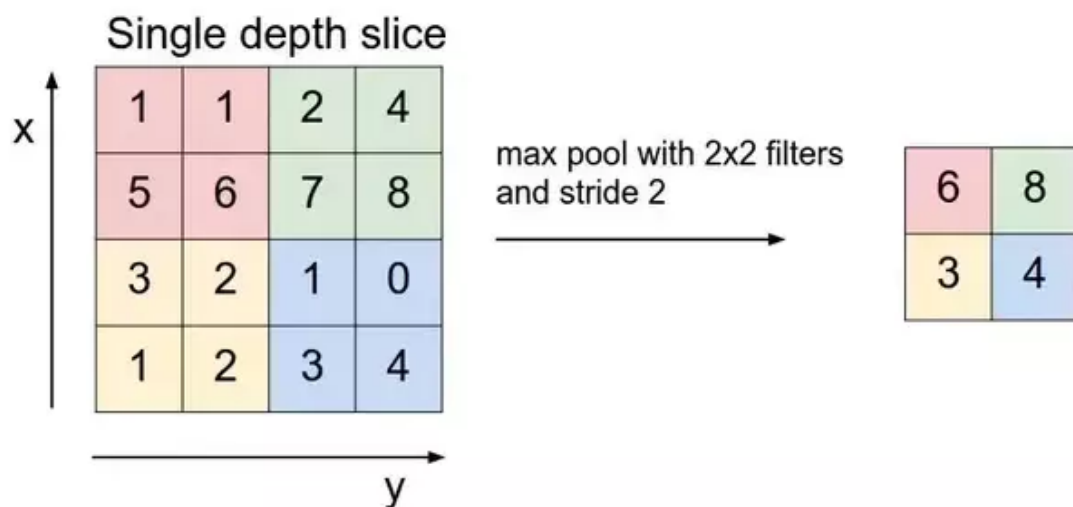
Nach der Multiplikation der Matrix ergibt sich aus der ursprünglichen Bildmatrix eine Merkmalskarte (Feature Map):

4	3	4
2	4	3
2	3	4

### 4.6.2 Pooling Layer

Pooling-Layer verkleinert die räumliche Größe der Merkmalskarte, wodurch die Anzahl der erforderlichen Parameter und Berechnungen verringert wird.

- Max Pooling (nimmt den größten Wert)
- oder Average Pooling (nimmt den Durchschnittswert)



**Abbildung 11:** Pooling Layer CNN <https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>

### 4.6.3 Fully Connected Layer

Die voll verknüpfte Schicht ist ein traditionelles mehrschichtiges Perceptron, das Soft-max-Aktivierung verwendet, um Vorhersagen zu treffen.

Man kann die Faltungsschicht und die Pooling-Schicht viele Male trainieren, bevor man die vollständig verbundene Schicht erreicht. Dank der Merkmalsextraktion ist CNN zu einem leistungsstarken Modell geworden, das die Gesichtserkennung durch den Mechanismus der Gesichtsmerkmalsextraktion unterstützt.

Man beachte, dass in diesem Fully-Connected-Layer ein Flatten-Layer stattfindet, um Daten aus einem mehrdimensionalen Array (als Matrix aus dem Convolution-Layer und dem Pooling-Layer) in einen eindimensionalen Vektor zur Verarbeitung zu bringen. Es ist jedoch zu beachten, dass das System in einem CNN nicht nur lernt, die optimalen Gewichte in der vollständig verknüpften Schicht zu finden, sondern auch lernt, wie man Systemfilter in der Faltungsschicht erstellt. Daher werden die räumlichen Merkmale des CNN durch die Filter erhalten.

## 4.7 Multi-task Convolutional Neural Network (MTCNN)

MTCNN wurde in der Arbeit von Zhang, Kaipeng et al. "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks" beschrieben, die 2016 in den IEEE Signal Processing Letters, Ausgabe 23.10, veröffentlicht wurde. Es besteht aus drei gestapelten CNN-Netzen und arbeitet bei der Gesichtserkennung gleichzeitig. Jedes Netz hat eine andere Struktur und spielt eine andere Rolle bei der Aufgabe. Die Ausgabe von MTCNN ist die Gesichtsposition und Punkte auf dem Gesicht: Augen, Nase, Mund. MTCNN arbeitet in drei Schritten, jeder Schritt hat sein eigenes neuronales Netz: P-Netz, R-Netz und O-Netz.

### 4.7.1 P-Net

Das P-Netz scannt das Foto mit Kernel  $12 \times 12$  schnell, um Bereiche zu finden, in denen wahrscheinlich Gesichter zu finden sind. Nach der drei Convolution Layer teilt sich das Netz in zwei Schichten auf. Die Convolution Layer 4-1 liefert die Wahrscheinlichkeit, dass ein Gesicht in jede Bounding Box fällt, und die Convolution Layer 4-2 liefert die Koordinaten der Bounding Boxen. (Abbildung 12)

### 4.7.2 R-Net

Das R-Netz hat eine ähnliche Struktur wie das P-Netz. Es werden jedoch mehr Schichten verwendet. Hier verwendet das Netz die von P-Net bereitgestellten Begrenzungsrahmen und verfeinert die Koordinaten.

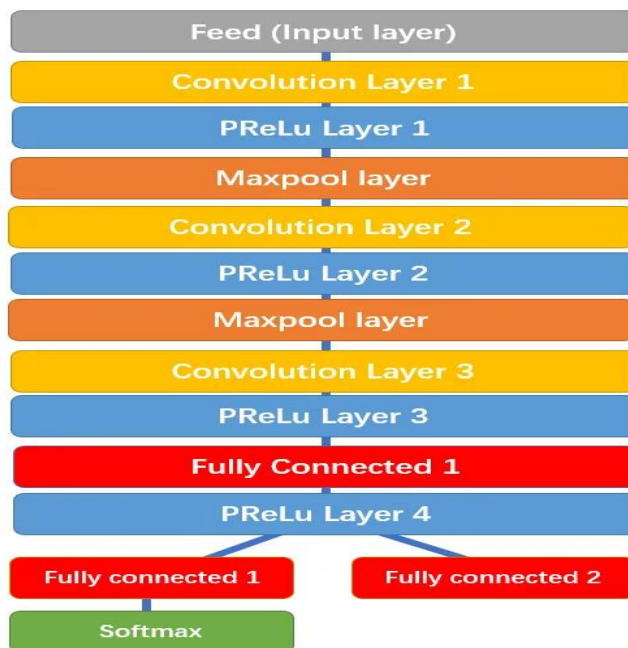
In ähnlicher Weise wird das R-Net im letzten Schritt in zwei Schichten unterteilt, die zwei Ausgaben liefern, nämlich neue, genauere Koordinaten der Bounding Boxes sowie deren Vertrauensgrad. (Abbildung 13)

### 4.7.3 O-Net

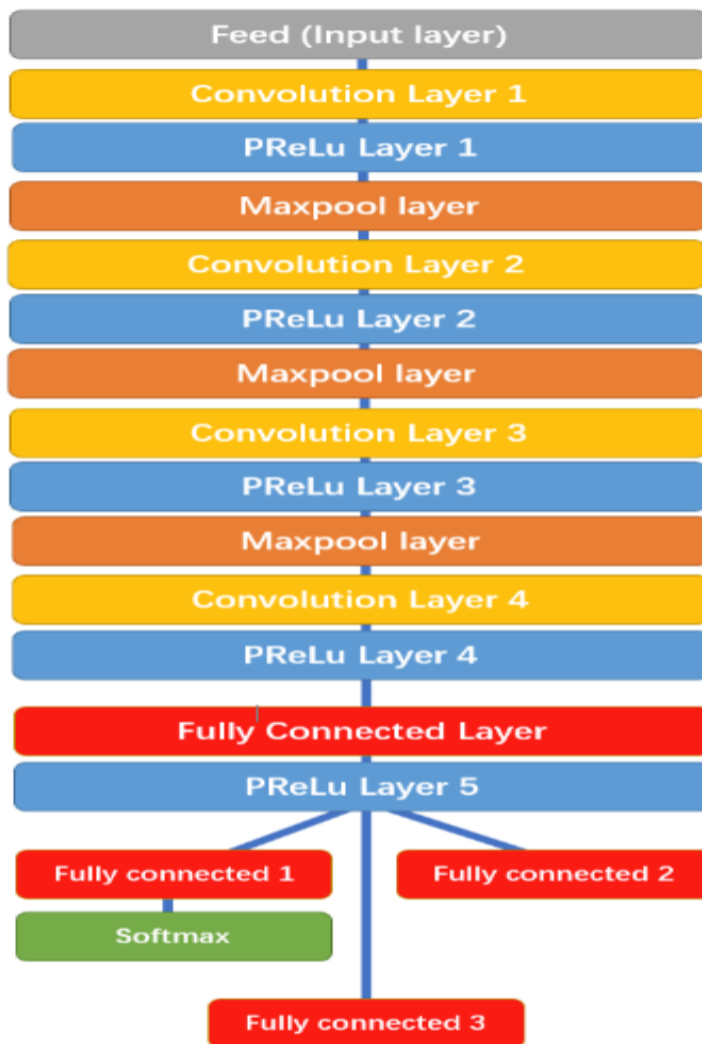
O-Net nimmt Bounding Boxes aus R-Net als Eingabe und markiert die Koordinaten von Gesichtsmerkmalen (Augen, Nase, Mund...) (Abbildung 14)



**Abbildung 12:** P-net MTCNN Chi-Feng Wang, <https://towardsdatascience.com/face-detection-neural-network-structure-257b8f6f85d1>



**Abbildung 13:** R-net - MTCNN Chi-Feng Wang, <https://towardsdatascience.com/face-detection-neural-network-structure-257b8f6f85d1>



**Abbildung 14:** O-net MTCNN Chi-Feng Wang, <https://towardsdatascience.com/face-detection-neural-network-structure-257b8f6f85d1>

## 4.8 Bibliotheken für maschinelles Lernen

### 4.8.1 TensorFlow

TF ist eine Open-Source-Plattform für maschinelles Lernen, die von Google Brain und der KI-Forschungsorganisation von Google entwickelt wurde, um Anwendungen für maschinelles Lernen und Deep Learning auf einfache und optimale Weise bereitzustellen. TF verfügt über ein umfassendes, flexibles Ökosystem von Tools, Bibliotheken und Ressourcen, die die Community bei der Entwicklung und Bereitstellung von Machine Learning-Anwendungen unterstützen.

- Hauptseite: <https://www.tensorflow.org/>
- Quellcode: <https://github.com/tensorflow/tensorflow>

### 4.8.2 Keras

Keras ist eine Open-Source-Bibliothek für Deep Learning in Python. Sie wurde von einem Google-Forscher für künstliche Intelligenz, Francois Chollet, entwickelt. Keras kann auf Open-Source-Bibliotheken wie TensorFlow, Theano, R oder CognitiveToolkit (CNTK) ausgeführt werden. Das Ziel von Keras ist es, das schnelle Testen von Deep Learning-Netzwerken zu ermöglichen. Führende Unternehmen wie Google, Square, Netflix, Huawei und Uber nutzen derzeit Keras.

- Hauptseite: <https://keras.io/>
- Quellcode: <https://github.com/keras-team/keras>

### 4.8.3 PyTorch

PyTorch ist eine Open-Source-Bibliothek für maschinelles Lernen in Python. Sie wird für Anwendungen wie die Verarbeitung natürlicher Sprache verwendet. Ursprünglich wurde sie vom Forschungsteam für künstliche Intelligenz von Facebook und der Pyro-Software von Uber für probabilistische Programmierung entwickelt.

PyTorch wurde von Hugh Perkins als Python-Wrapper für LusJIT auf der Grundlage des Torch-Frameworks entwickelt.

- Hauptseite: <https://pytorch.org/>
- Quellcode: <https://github.com/pytorch/pytorch>

Die drei Python-Bibliotheken unterstützen Deep Learning und maschinelles Lernen sehr gut, haben eine große Unterstützergemeinde und ihre eigenen Stärken und Schwächen. Pytorch ist jedoch besonders flexibel und für den Bereich Studium und Forschung geeignet. Daher werden in dieser Arbeit die Pytorch-Bibliotheken in Python verwendet, um die Gesichtserkennungsfunktion zu implementieren.

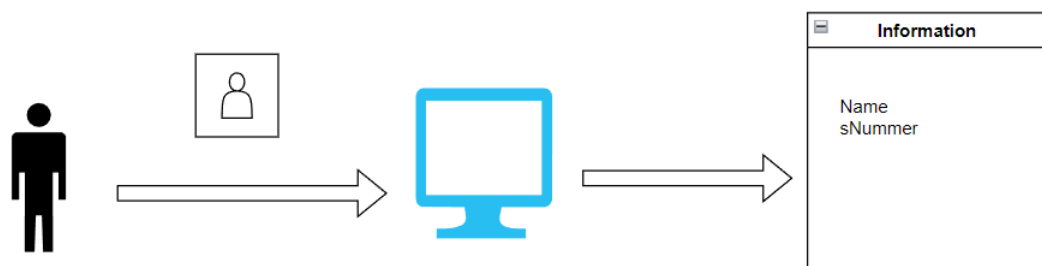
## 4.9 Fazit

In Kapitel 4 wurden die grundlegenden Konzepte des Deep Learning sowie bemerkenswerte neue Aspekte des Deep Learning im Vergleich zum traditionellen maschinellen Lernen vorgestellt. Tatsächlich ist Deep Learning ein sehr breites Feld mit einer Vielzahl nützlicher Erkenntnisse. Im Rahmen dieser Arbeit werden jedoch nur Technologien und Konzepte behandelt, die speziell für die Aufgabenstellung dieser Arbeit relevant sind: Gesichtserkennung. Im nächsten Kapitel sollen die Technologien und Konzepte des maschinellen Lernens und des Deep Learning angewendet werden, um das Problem [1] zu lösen.



## 5 Face Recognition

Nach dem Kennenlernen der grundlegenden Konzepte des maschinellen Lernens und des Deep Learning wird nun zum Hauptthema der Arbeit zurückgekehrt: der Erfassung der Anwesenheit von Studenten durch Bilder. Um dies zu ermöglichen, muss eine Identifikation der Studenten erfolgen. Mit anderen Worten, die Eingabe wird ein Foto des Studenten sein. Sobald das Programm verarbeitet wird, werden Informationen über diesen Studenten als Output ausgegeben.



**Abbildung 15:** Gesichtserkennung Prozess (Draw.io)

Diese Aufgabe lässt sich in drei kleine Probleme unterteilen

- Erkennung von Gesichtern (Face Detection)
- Extraktion von Gesichtern (Face Extraction)
- Klassifizierung von Gesichtern (Face Classification)

### 5.1 Face Detection

Das erste Problem ist die Gesichtserkennung. Dieses Problem spielt eine äußerst wichtige Rolle bei der Gewährleistung der Genauigkeit des Programms.

Das Problem akzeptiert ein Bild als Eingabe.

Die Ausgabe, die das Problem liefert, ist die Koordinate des Bereichs, in dem sich das Gesicht befindet, normalerweise dargestellt durch ein Rechteck, das das Gesicht umgibt (Bounding Box).

Laut dem Artikel *Face Detection: A survey (2001)* lassen sich Probleme der Gesichtserkennung in zwei Hauptgruppen unterteilen:

- Feature-Based (Merkmalsbasiert)
- Image-Based (Bildbasiert)

### 5.1.1 Feature-Based

Verwendung manueller Filter (oder feature extraction, von Menschen für das Training vorausgewählt). Diese Methode ist sehr schnell, aber bei komplexen Fällen wie Fotos mit vielen Personen oder unklaren Fotos ist die Genauigkeit nicht sehr hoch.

Bei den traditionellen Methoden des maschinellen Lernens ist die Haar-Cascade eine typische Methode, die ebenfalls Feature Based ist.

Haar-Cascade bietet eine Reihe von Filtern, die auf Haar-ähnlichen Merkmalen basieren, um Gesichter in Bildern zu erkennen. Bei diesen Filtern handelt es sich um einfache rechteckige Formen, mit denen Gesichtsmerkmale wie Kanten, Konturen und unterschiedliche helle und dunkle Bereiche schnell und effektiv definiert werden können.

### 5.1.2 Image-Based

Verwendung von DL-Methode zum Erlernen und automatischen Auffinden von Gesichtern auf der Grundlage des gesamten Fotos. Der Vorteil dieser Methode ist, dass die Genauigkeit höher ist als bei der merkmalsbasierten Methode, aber die Ausführungsgeschwindigkeit langsamer ist.

MTCNN ist eine image-based Methode

### 5.1.3 Vergleich: MTCNN vs. Haar-Kaskade

Bei der Haar-Kaskade handelt es sich um eine alte Methode, die jedoch nach wie vor sehr beliebt ist und verwendet wird, da sie schnell durchgeführt werden kann. Sie wird häufig in mobilen Anwendungen verwendet oder die Hardwarekonfiguration des Geräts ist nicht gut. Außerdem wird MTCNN zwar wegen seiner Genauigkeit und Fortschrittlichkeit hochgeschätzt, aber bei der praktischen Umsetzung eines Systems setzen sich neue Technologien nicht immer durch. Die Schwäche von MTCNN ist die Ausführungszeit. Obwohl eine schnelle Methode erforderlich ist, um einen reibungslosen Ablauf des Anwesenheitsprozesses zu gewährleisten, muss das Gesichtserkennungssystem vor allem sehr genau sein, um mögliche Fehler zu vermeiden.

Aus diesem Grund werde ich Experimente mit zwei Methoden durchführen: MTCNN und Haar-Kaskade unter den gegebenen Bedingungen. Auf diese Weise ist es möglich eine für das Projekt geeignete Methode zu wählen.

Ich sammelte 20 Fotos mit unterschiedlichen Bedingungen: Fotos von einer Person aus vielen verschiedenen Winkeln, Fotos mit vielen Personen, Fotos mit vielen Personen, aber nur einem Teil des Gesichts. Dann markierte ich die Namen nach einer bestimmten Konvention und speichere sie im Testordner. Nach der Erfassung der Ergebnisse wurden die beiden Methoden anhand der beiden wichtigsten Kriterien verglichen: Ausführungszeit und Genauigkeit.

Während des Experiments verwendete ich die openCV-Bibliothek. Die openCV-Bibliothek ist eine sehr beliebte Open-Source-Bibliothek in Python. Diese Bibliothek unterstützt die Implementierung vieler Funktionen, insbesondere die Gesichtserkennung mit dem Haar-Kaskade-Algorithmus. Für das MTCNN-Modell verwende ich die facenet\_pytorch-Bibliothek, eine Python-Bibliothek, die die Implementierung von Algorithmen für die Gesichtserkennung unterstützt. Außerdem verwende ich die time-Bibliothek, um die Laufzeit des Algorithmus für jedes Bild zu ermitteln.

```
for filename in os.listdir(path):
    if filename.endswith(".jpg") or filename.endswith(".png"):
        image_path = os.path.join(path, filename)
        image = cv2.imread(image_path)
        gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

        start_time = time.time()
        faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=4)
        end_time = time.time()

        print(f"time for {filename}: {end_time - start_time} s. number of detected faces: {len(faces)}")

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

**Abbildung 16:** Haar-Kaskade mit OpenCV

```
for filename in os.listdir(path1):
    if filename.endswith(".jpg") or filename.endswith(".png"):
        image_path = os.path.join(path1, filename)
        image = Image.open(image_path)

        start_time = time.time()
        boxes, _ = mtcnn.detect(image)
        end_time = time.time()

        print(f"time for {filename}: {end_time - start_time} s. Số khuôn mặt phát hiện: {len(boxes) if boxes is not None else 0}")

    # bounding box
    draw = ImageDraw.Draw(image)
    if boxes is not None:
        for box in boxes:
            draw.rectangle(box.tolist(), outline=(255, 0, 0), width=2)
```

**Abbildung 17:** MTCNN with facenet\_pytorch

Nachdem ich den Satz von Fotos getestet hatte, zeichnete ich die Ergebnisse auf.

Haarcascade					
Img	time	number detected face	number face	wrong	
ImgTest0	0,52	4	5	2 F	
ImgTest1	0,26	1	1	0 T	
ImgTest2	0,12	2	1	1 F	
ImgTest3	0,1	1	1	0 T	
ImgTest4	0,14	2	1	1 F	
ImgTest5	0,25	4	6	2 F	
ImgTest6	0,13	2	1	1 F	
ImgTest7	0,51	4	2	2 F	
ImgTest8	0,02	4	4	0 T	
ImgTest9	0,14	2	2	0 T	
ImgTest10	0,15	3	3	0 T	
ImgTest11	0,15	3	3	0 T	
ImgTest12	0,11	1	1	0 T	
ImgTest13	0,15	1	1	0 T	
ImgTest14	0,1	1	1	0 T	
ImgTest15	0,18	1	1	0 T	
ImgTest16	0,03	2	2	0 T	
ImgTest17	0,6	8	3	5 F	
ImgTest18	0,28	4	6	2 F	
ImgTest19	0,36	3	1	2 F	
SUM	4,3			18	0,55
AVERAGE	0,215				

Abbildung 18: Haar-Kaskade Test Ergebnis

MTCNN					
Img	time	number detec	number fa	wrong	
ImgTest0	0,4	5	5	0 T	
ImgTest1	0,4	1	1	0 T	
ImgTest2	0,17	1	1	0 T	
ImgTest3	0,16	1	1	0 T	
ImgTest4	0,22	1	1	0 T	
ImgTest5	0,13	6	6	0 T	
ImgTest6	0,23	1	1	0 T	
ImgTest7	0,9	3	2	1 F	
ImgTest8	0,06	4	4	0 T	
ImgTest9	0,23	3	2	1 F	
ImgTest10	0,29	3	3	0 T	
ImgTest11	0,28	3	3	0 T	
ImgTest12	0,23	1	1	0 T	
ImgTest13	0,26	1	1	0 T	
ImgTest14	0,27	1	1	0 T	
ImgTest15	0,3	1	1	0 T	
ImgTest16	0,13	2	2	0 T	
ImgTest17	0,9	4	3	1 F	
ImgTest18	0,55	6	6	0 T	
ImgTest19	0,93	2	1	1 F	
SUM	7,04			4	0,8
AVERAGE	0,352				

Abbildung 19: MTCNN Test Ergebnis

Aus den oben genannten Ergebnissen geht hervor, dass der Haar-Kaskade-Algorithmus mit durchschnittlich 0,125 Sekunden pro Bild eine schnellere Ausführungszeit hat, während der MTCNN-Algorithmus 0,352 Sekunden benötigt (der Haar-Kaskade-Algorithmus hat eine Ausführungszeit von nur 35 % des MTCNN-Algorithmus). Jedoch liefert der MTCNN-Algorithmus 25 % genauere Ergebnisse als der Haar-Kaskade-Algorithmus (55 % im Vergleich zu 80 %).

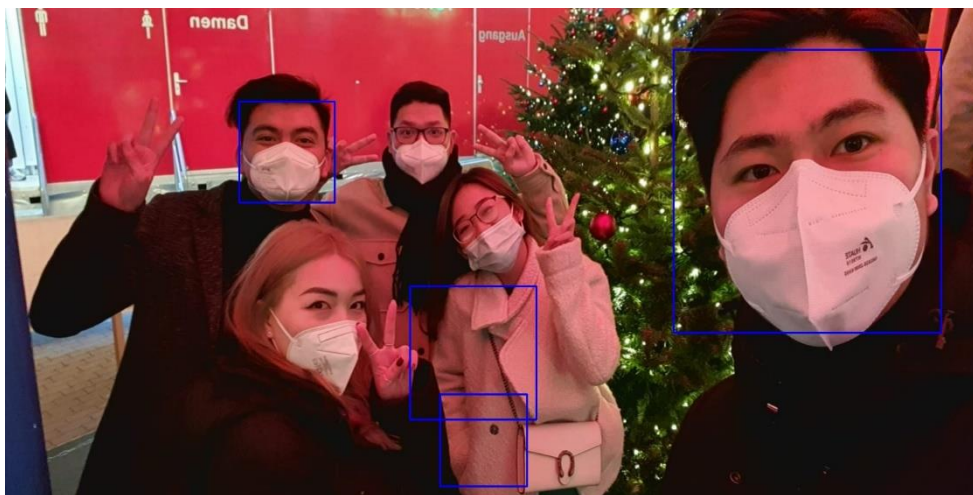
Ich testete nicht nur, ob der Algorithmus Gesichter erkennt, sondern auch, ob der Algorithmus die richtige Anzahl von Gesichtern auf dem Foto erkennt. In der Vergleichstabelle gibt es einen Falsch-index „wrong“, um einen weiteren Aspekt der Genauigkeit zu bewerten. Der Falsch-Index erfasst die Differenz zwischen der Anzahl der vom Algorithmus erkannten Gesichter und der tatsächlichen Anzahl der Gesichter in jedem Foto. Überraschend ist, dass der MTCNN-Algorithmus einen deutlich niedrigeren Falschindex aufweist als der Haar-Kaskade-Algorithmus. Der Haar-Kaskade-Algorithmus identifiziert häufig Nicht-Gesichts-Objekte fälschlicherweise als menschliche Gesichter, oder bei vielen Personen auf einem Bild, werden diese nicht richtig erkannt (wie in den Abbildungen 22, 23, 24, 25 gezeigt: Das blaue Quadrat steht für den Haar-Kaskade-Algorithmus, das rote Quadrat steht für den MTCNN-Algorithmus)



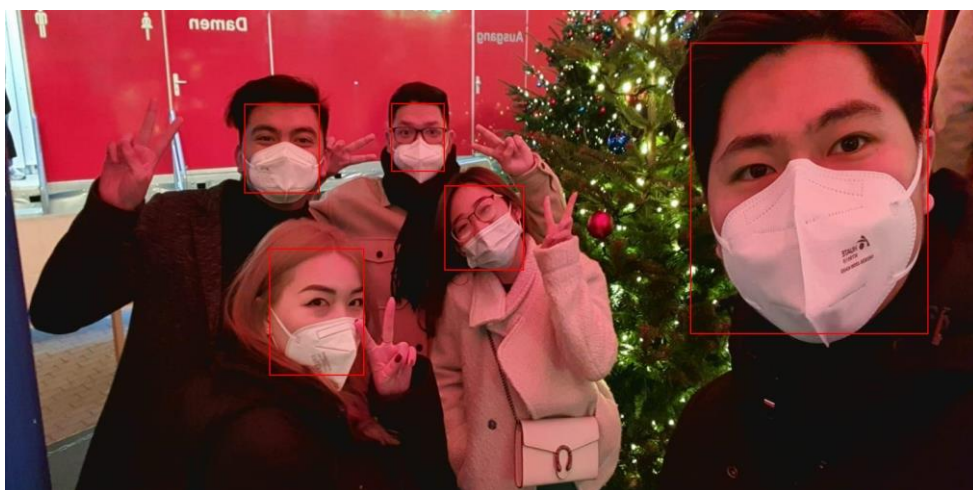
Abbildung 20: Haar-Kaskade



Abbildung 21: MTCNN.



**Abbildung 22:** Haar-Kaskade



**Abbildung 23:** MTCNN

Die Ergebnisse zeigen, dass der MTCNN-Algorithmus Gesichter mit hoher Genauigkeit (80 %) unter einer Vielzahl komplexer Bedingungen erkennen kann. Bei Bildern guter Qualität erreicht er sogar eine Genauigkeit von 95 %. (Es handelt sich um Bilder mit guter Helligkeit, die nicht dunkel oder unscharf sind und von der Vorderseite des Gesichts aufgenommen wurden). Dies gilt auch für Gesichter mit Masken bei schlechten Lichtverhältnissen. Obwohl der MTCNN-Algorithmus langsamer ist als der Haar-Cascade-Algorithmus, ist diese Geschwindigkeit (weniger als 1 Sekunde pro Bild) für den Anwendungsbereich - Erkennung der Anwesenheit von Studenten bei Prüfungen - akzeptabel, da eine hohe Genauigkeit erforderlich ist. Um die Gesichtserkennungsfähigkeit des Systems weiter zu verbessern, können Maßnahmen ergriffen werden, die eine bessere Bildqualität auch unter schwierigen Testbedingungen gewährleisten.

Um die experimentellen Ergebnisse zusammenzufassen, werden in der folgenden Tabelle die Vor- und Nachteile von MTCNN und Haar-Kaskade verglichen.

Vorteile	Nachteile
1.hohe Erfolgsquote: 80%	1.Die Ausführungszeit ist lang. Der Durchschnitt liegt bei 0,35s.
2.Erkennen von Gesichtern unter vielen Bildbedingungen: Die Lichtverhältnisse sind nicht gut, ein Teil des Gesichts ist verdeckt, eine Maske wird getragen.	

**Aufgrund dieser Vorteile von MTCNN setzte ich MTCNN in meinem Projekt ein.**

## 5.2 Face Extraction

Dies ist der zweite Schritt des Gesichtserkennungsproblems. Seine Eingabe ist das Bild des in Schritt 1 gefundenen Gesichts, während die Ausgabe ein mehrdimensionaler Vektor ist, der die Merkmale dieses Gesichts darstellt.

Derzeit gibt es mehrere Algorithmen für die Gesichtserkennung.

### 5.2.1 One-Shot-Learning

One-Shot-Learning ist ein überwachter Lernalgorithmus, bei dem jede Person nur wenige, sehr wenige oder sogar nur ein einziges Foto benötigt (um eine große Variation zu vermeiden). Bei der Eingabe eines Fotos einer Person verwenden eine einfache CNN-Algorithmusarchitektur, um vorherzusagen, wer diese Person ist. Der Nachteil dieser Methode ist jedoch, dass man den Algorithmus regelmäßig neu trainieren muss, wenn eine neue Person auftaucht, da sich die Form der Ausgabe um eins ändert.

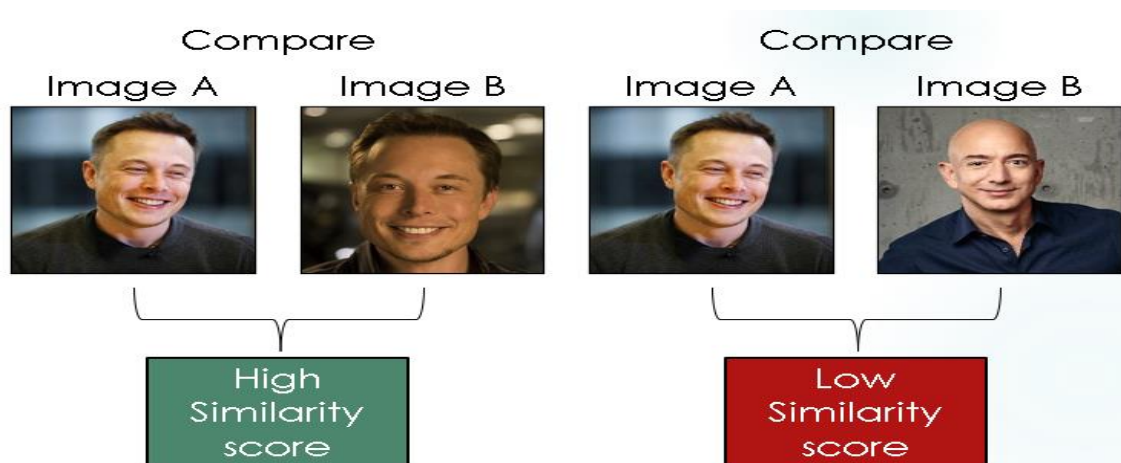
Um dieses Problem zu überwinden, wird die Methode des Ähnlichkeitslernens verwendet.

### 5.2.2 Learning similarity

Diese Methode basiert auf der Messung des Abstands zwischen zwei Fotos. Wenn es sich um dieselbe Person handelt, ist dieser Abstand am kleinsten, andernfalls ist der Abstand größer (der akzeptable Abstand wird durch einen Schwellenwertindex eingestellt). Je nach Schwellenwert kann das Ähnlichkeitslernen mehr als ein Bild zurückgeben, das vom gleichen Typ wie das Eingabebild ist.

Darüber hinaus ist diese Methode nicht von der Anzahl der Klassen abhängig. Daher ist es nicht notwendig, die Methode neu zu trainieren, wenn eine neue Klasse auftaucht.



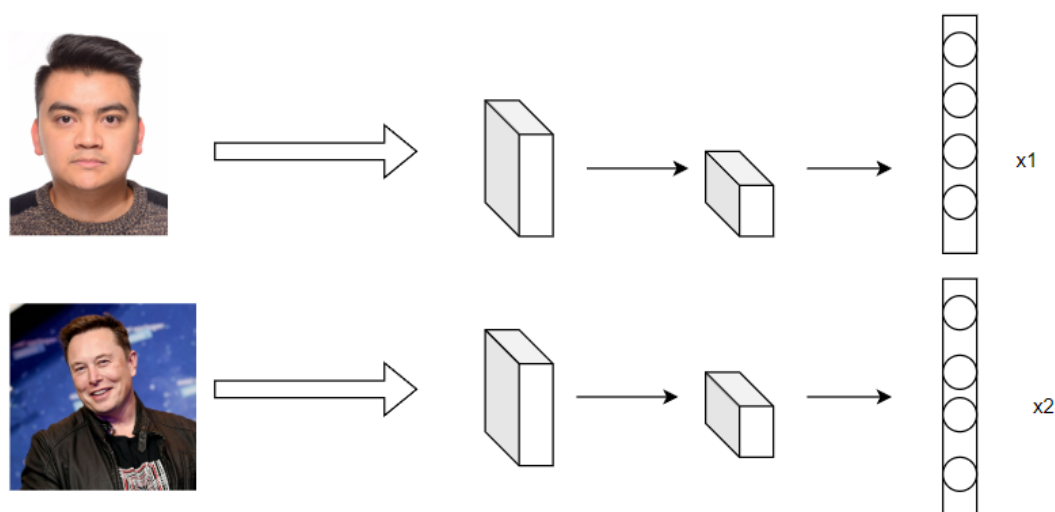


**Abbildung 24:** Learning similaritv (Strahinja Stefanovic, <https://datahacker.rs/019-siamese-network-in-pytorch-with-application-to-face-similarity/>)

### 5.2.3 Siam network

Netzwerkarchitekturen, bei denen zwei Fotos eingegeben werden und das Modell die Frage beantwortet, ob sie zu derselben Person gehören oder nicht, werden als Siam-Netzwerke bezeichnet.

Die Architektur des Siam-Netzes basiert auf dem Basisnetz, einem CNN, bei dem die Ausgabeschicht, die Bilder in Einbettungsvektoren kodiert, entfernt wurde. Die Eingabe des Siam-Netzwerks sind zwei beliebige Bilder, die zufällig aus den Bilddaten ausgewählt werden. Die Ausgabe des Siam-Netzes sind zwei Vektoren, die der Darstellung von zwei Eingabebildern entsprechen. Anschließend werden die beiden Vektoren in die Verlustfunktion eingegeben, um die Differenz zwischen ihnen zu messen.



**Abbildung 25:** Siam network



### 5.2.4 Facenet

Facenet ist eine Form des Siam-Netzwerks, das Bilder in einem n-dimensionalen euklidischen Raum (in der Regel 128) darstellt, so dass die Ähnlichkeit zwischen embedding Vektoren umso größer ist, je kleiner der Abstand zwischen ihnen ist.

Die Schwäche der Algorithmen aus der Zeit vor Facenet besteht darin, dass die Anzahl der Einbettungsdimensionen zu groß ist und die Verwendung von Verlustfunktionsalgorithmen dazu führt, dass das System nur Ähnlichkeiten oder Unterschiede in Abhängigkeit von der Entfernung lernt, nicht aber sowohl Ähnlichkeiten als auch Unterschiede zur gleichen Zeit. Facenet hat diese beiden Probleme gelöst, indem es CNN verwendet hat, um die Embedding-Dimensionen auf 128 zu reduzieren (in der Realität kann sich die Anzahl der Dimensionen je nach Hardwareentwicklung ändern) und den Triplet-Verlustalgorithmus (triplet loss) zum Trainieren verwendet hat. Dadurch kann das System sowohl Ähnlichkeiten als auch Unterschiede zwischen Fotos lernen, um die Effizienz zu erhöhen.

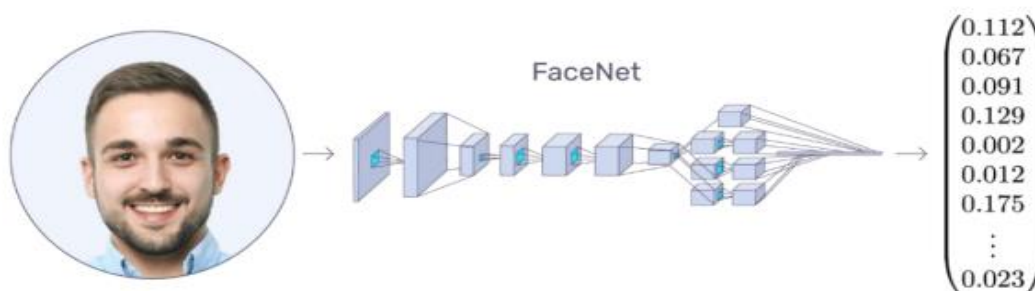


Abbildung 26: Facenet

#### 5.2.4.1 Triplet Loss

Die Triplet-Verlustfunktion ist eine Funktion, die dem System hilft, nicht nur Ähnlichkeiten (Nahdistanz), sondern gleichzeitig auch Unterschiede (Ferndistanz) zu erkennen.

$$\|f(\mathbf{A}) - f(\mathbf{P})\|_2^2 + \alpha \leq \|f(\mathbf{A}) - f(\mathbf{N})\|_2^2$$

Bei der Triplet Loss Funktion gibt es drei Bilder, ein Bild ist der Anker (Anchor), ein positives Bild ist "gleich" wie das Ankerbild und negativ ist das Bild, das sich vom Ankerbild unterscheidet. Das Ziel der Triplet-Verlustfunktion ist in der Formel klar dargestellt. Das System muss lernen, wie es sicherstellen kann, dass der Abstand zwischen Anker und Positivbild plus Alpha immer kleiner sein muss als der Abstand zwischen Anker und Negativbild (Alpha hilft uns, einen gewissen Spielraum zu gewährleisten).



```

def calculate_rotation_angle(left_eye, right_eye):
    x1, y1 = left_eye
    x2, y2 = right_eye
    angle = math.atan2(y2 - y1, x2 - x1)
    angle = math.degrees(angle)
    return angle

path_to_image = 'E:/HTW/Student/test/donald-trump.jpg'
save_directory = 'E:/HTW/Student/MTCNN'

mtcnn = MTCNN(keep_all=True, post_process=False, device='cpu')
img = Image.open(path_to_image)

boxes, probs, landmarks = mtcnn.detect(img, landmarks=True)

if boxes is not None:
    for i, (box, landmark) in enumerate(zip(boxes, landmarks)):
        left_eye = landmark[0]
        right_eye = landmark[1]

        angle = calculate_rotation_angle(left_eye, right_eye)

        rotated_img = img.rotate(angle)

        rotated_img.save(f'{save_directory}/rotated_image_{i}.png')

```

Abbildung 28: Face alignment code



Abbildung 30: Face alignment

### 5.2.4.3 Pre-trained Model

Ein bereits trainiertes Modell ist ein Modell, das von einer anderen Person zur Lösung eines ähnlichen Problems trainiert wurde. Anstatt ein Modell von Grund auf neu zu erstellen, um ein ähnliches Problem zu lösen, wird ein Modell verwendet, das für ein anderes Problem trainiert wurde als Ausgangspunkt. Ein vorab trainiertes Modell ist vielleicht nicht 100 % genau, aber es spart Zeit und Mühe.

Derzeit gibt es 2 FaceNet-Pretrain-Modelle, die anhand des CASIA-WebFace-Datensatzes und des VGGFace2-Datensatzes trainiert wurden. Der Datensatz CASIA-WebFace (2014) enthält 494 414 Bilder mit 10 575 Identitäten (durchschnittlich 46,8 Bilder pro Person). Der Datensatz VGGFace2 (2018) enthält dagegen 3,31 Millionen Bilder mit 9131 Identitäten (durchschnittlich 362,6 Bilder pro Person) aus vielen verschiedenen Kontinenten.

Die Genauigkeit der beiden Modelle ist wie folgt:

**Tabelle 3: Pre-trained Models**

Model name	LFW accuracy	LFW accuracy
20180408-102900	0,9905	CASIA-WebFace
20180402-114759	0,9965	VGGFace2

Nach den soeben gelernten Theorien ist Facenet eine ziemlich überlegene Technologie und hat eine sehr hohe Genauigkeit bei der Gesichtsextraktion. Darüber hinaus hat das Facenet-Modell den Verlust von Triplets und die Gesichtsausrichtung kombiniert, um die Genauigkeit zu erhöhen. Wie in dem Kapitel über Facenet beschrieben, extrahieren auf diese Weise häufig Einbettungen mit 128 Dimensionen. Mit der Entwicklung der heutigen Hardware wurden jedoch auch die DL-Bibliotheken schrittweise aufgerüstet, um 512-dimensionale Einbettungen zu extrahieren. Mit Hilfe der Bibliotheksunterstützung in Pytorch wird das Modell von Facenet zur Extraktion einer 512-dimensionalen Einbettung verwendet.

## 5.3 Face Classification

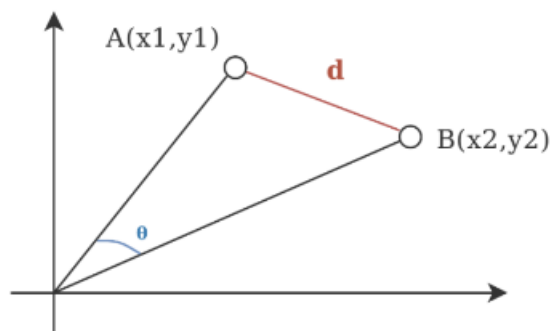
Nach der Extraktion von Gesichtsmerkmalen, die durch 128- oder 512-dimensionale Einbettungsvektoren ausgedrückt werden, können die Merkmale der Gesichter der Studenten in Form von Einbettungen extrahiert und auf dem Server gespeichert werden. Wenn dann ein Gesicht erkannt werden soll, werden auch die Merkmale des Gesichts im neuen Bild extrahiert, um sie mit den gespeicherten Einbettungsvektoren zu vergleichen. Es gibt zwei gängige Methoden zur Erkennung von Gesichtern:

Die Klassifizierungsmethode ist weniger zeitaufwändig, da nur bestimmt werden muss, in welchem Cluster sich der Einbettungsvektor befindet, aber sie erfordert auch viel Zeit und die Qualität des Eingabebildes, um die Cluster richtig zu erstellen. Die Methode des Abstandsvergleichs ist zeitaufwändiger, da der Einbettungsvektor mit allen verfügbaren Vektoren auf dem Server verglichen werden muss, um das Gesicht zu erkennen. Im Rahmen der Betreuung der Studenten an der HTW Dresden ist die Anzahl der Studenten jedoch nicht sehr groß. Vor allem, wenn man bessere Unterstützungsmöglichkeiten hat. In der Regel werden die Prüfungen zu einem bestimmten Modul gehören. Anstatt das Foto des betreuungspflichtigen Studenten mit allen studentischen Einbettungsvektoren der HTW Dresden zu vergleichen, können nur die Studenten, die diese Prüfung (oder dieses Modul) abgelegt haben, mit diesem Foto verglichen werden. Dies ist durch ein vollständiges Datenbanksystem leicht zu erreichen.

Daher werde im Folgenden die Methode des Abstandsvergleichs, genauer gesagt die euklidische Gleichung, verwendet, um den Abstand zwischen zwei Vektoren zu berechnen.

Euklidische Gleichung:

$$\sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$



**Abbildung 29:** Abstand zwischen 2 Vektoren

Nach der Berechnung des Abstands zwischen zwei Vektoren ergibt sich ein Wert für die Differenz zwischen diesen beiden Vektoren. Liegt die Differenz innerhalb eines akzeptablen Bereichs, können die beiden Vektoren als dieselbe Person betrachtet werden, und das Programm gibt Informationen über diesen Studenten zurück (z. B. sNummer). Im Rahmen des Experiments wurde der akzeptable Wert auf 0,6 festgelegt.

## 5.4 Fazit

Am Ende von Kapitel 5 wurden die Grundkenntnisse und Algorithmen des maschinellen Lernens und des Deep Learning vorgestellt. Der Schwerpunkt dieses Wissens liegt auf 3 Hauptbereichen: Face Detection, Face Extraction und Face Classification. Durch Experimente und Vergleiche, sowie die Analyse der Vor- und Nachteile von Methoden und Algorithmen, werden folgende Algorithmen zur Lösung des Problems verwendet [1]: "Wie

kann Pytorch zur Erkennung von Gesichtern eingesetzt werden, um die Anwesenheit von Studenten durch Gesichtserkennung zu erfassen? ".

1. Gesichtserkennung: MTCNN.
2. Gesichtsextraktion: Facenet.
3. Klassifizierung von Gesichtern: Euklidische Gleichung zur Berechnung des Abstands zwischen Vektoren.

Die Kombination der drei oben genannten Algorithmen bietet eine umfassende und genaue Lösung für das erste Problem

In den ersten drei Kapiteln wurde das erste Problem gelöst. Im nächsten Kapitel wird das zweite Problem der Zielfrage analysiert und eine Lösung gefunden. Das zweite Problem ist:

Wie man ein stabiles und sicheres Websystem aufbaut, das als Plattform für die Verwaltung von Studenten, die Verwaltung von Prüfungen und die Anwesenheit von Studenten durch Gesichtserkennung dient. [2]

## 6 Lösungsvorschlag

Zunächst muss das zweite Problem analysiert und spezifiziert werden.

Wie man ein stabiles und sicheres Websystem aufbaut, das als Plattform für die Verwaltung von Studenten, die Verwaltung von Prüfungen und die Anwesenheit von Studenten durch Gesichtserkennung dient. [2]

Zuerst einmal konzentrieren sich die Funktionen, die hier entwickelt werden müssen, auf zwei Hauptfunktionen, nämlich die Studentenverwaltung und die Testverwaltung.

In der Studentenverwaltung gibt es grundlegende Funktionen wie das Hinzufügen, Bearbeiten, Löschen und Suchen von Studenten. Darüber hinaus müssen auch etwas komplexere Funktionen wie die An- und Abmeldung von Studenten zu Kursen und die An- und Abmeldung von Studenten zu Prüfungen entwickelt werden. Der Grund dafür ist, die grundlegendsten Funktionen zu schaffen, die eine effektive Verwaltung der Studierenden ermöglichen.

Die zweite Hauptfunktion ist die Prüfungsverwaltung. Zusätzlich zu den grundlegenden Studentenfunktionen wie Hinzufügen, Bearbeiten, Löschen und Suchen muss die Funktion zur Verwaltung der einzelnen Prüfungen entwickelt werden. Bei dieser Funktion werden auch zwei weitere wichtige Funktionen entwickelt: Die Eingabe von Noten für Tests und die Funktion zur Erfassung der Anwesenheit von Studenten durch Gesichtserkennung. Um das Programm zu optimieren, muss außerdem die traditionelle Anwesenheitsfunktion entwickelt werden (im Wesentlichen die Suche nach Studenten auf der Grundlage der bereitgestellten Informationen).

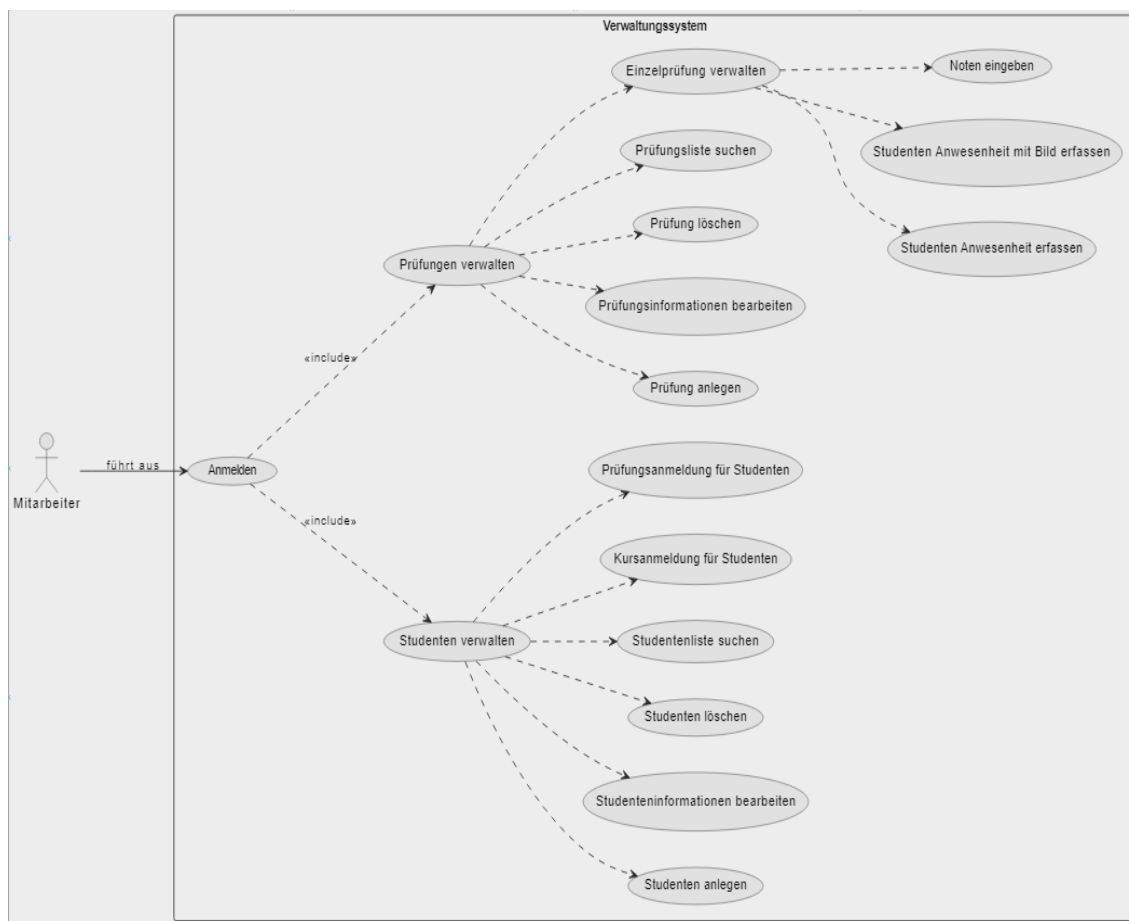
### 6.1 Analyse und Entwurf von Webanwendungen

Die Funktionen, die entwickelt werden müssen, sind im folgenden Anwendungsfalldiagramm dargestellt.

Der Mitarbeiter ist der einzige Akteur in diesem Anwendungsfalldiagramm.

Die Details der Anwendungsfälle sind wie folgt:

1. Anmelden: Der Mitarbeiter meldet sich an. Wenn er sich erfolgreich angemeldet hat, kann er auf andere Anwendungsfälle zugreifen. Hier müssen wir einen Anmeldemechanismus erstellen und diesen Anmeldestatus für die folgenden Anwendungsfälle speichern.
2. Student verwalten: Der Mitarbeiter wählt den Punkt Studentenverwaltung aus und das System navigiert den Mitarbeiter auf die Seite StudentPage.
3. Test verwalten: Der Mitarbeiter wählt den Punkt Testverwaltung aus und das System leitet ihn auf die Seite TestPage weiter.



**Abbildung 30** Use Case Diagramm.

4. Andere Anwendungsfälle werden im Folgenden beschrieben: Wenn ein Mitarbeiter ein bestimmtes Ereignis "auslöst", leitet das System den Mitarbeiter zu den entsprechenden Seiten oder Abschnitten, um die Funktion auszuführen. Das System wird auch über einen Mechanismus verfügen, um Fehler zu überprüfen und zurückzugeben, wenn die Durchführung nicht erfolgreich ist.

### 6.1.1 Server-Side-Rendering vs Client-Side-Rendering

Da es sich um eine Webanwendung handelt, kann sie in zwei Teile unterteilt werden: Client und Server. Von hier aus wurden zwei Konzepte der Webprogrammierung geboren:

1. Server-Side-Rendering (SSR): Man kann davon ausgehen, dass die Webseiten auf dem Server generiert werden. Der Server gibt die HTML-, CSS- und JS-Struktur an den Client zurück. Der Client muss sie nur noch anzeigen, um mit dem Benutzer zu interagieren.
2. Client-Side-Rendering (CSR): Der Client ruft nur die APIs des Servers auf, um die benötigten Daten zu erhalten. Dann bauen sie ihre eigene HTML-, CSS- und JS-Struktur auf und zeigen sie an, um mit den Benutzern zu interagieren.



CSR hat einen Vorteil gegenüber SSR: Wenn ein Anwendungsfall bearbeitet wird, sendet SSR normalerweise eine neue HTML-, CSS- und JS-Struktur an den Client und ignoriert dabei die Duplizierung von Komponenten in der Struktur. Dies vergeudet Ressourcen und macht das Laden der Seite zeitaufwändig, was sich auf das Benutzererlebnis auswirkt und auch den Faktor Ausführungszeit beeinträchtigt (Ziele der Effizienz). In der Zwischenzeit lädt CSR nur die Informationen, die zum Aufbau eines Teils der Webseite benötigt werden, was bedeutet, dass nur ein Teil der Webseite neu geladen wird. In Frontends bezieht sich der Begriff "Komponente" auf diesen "Teil". Daher trägt CSR nicht nur zu einem besseren Benutzererlebnis bei, sondern spart auch Ausführungszeit und Ressourcen, da nur das geladen wird, was für den Aufbau der HTML-, CSS- und JS-Struktur erforderlich ist, und nicht die gesamte Website wie bei SSR.

Dies ist auch im obigen Anwendungsfalldiagramm deutlich zu erkennen. Es passiert wenn viele untergeordnete Funktionen einen Teil der Schnittstelle der übergeordneten Funktion haben. Zum Beispiel haben Funktionen wie die Studentenverwaltung Unterfunktionen wie: Hinzufügen, Bearbeiten, Löschen usw. Diese Funktionen zum Hinzufügen, Bearbeiten und Löschen sind nur ein Teil der Schnittstelle in einer größeren Schnittstelle, der übergeordneten Funktion - der Studentenverwaltung. Genau dieses Problem muss berücksichtigt werden, um Lösungen zu finden, die die Benutzerfreundlichkeit verbessern und die Leistung sicherstellen.

Wenn von Client-Side-Rendering die Rede ist und die Website nur einen Teil der Komponente ändert, wird Konzept Single-Page-Application genannt.

### **6.1.2 Single-Page-Application**

Vor dem Aufkommen der Single-Page-Application wurde die Webprogrammierung hauptsächlich nach dem MVC-Modell durchgeführt. Das Funktionsprinzip dieser Art von Struktur konzentriert sich jedoch nur auf die Serververarbeitung und wird der Benutzererfahrung nicht gerecht. Daher besteht ein großer Bedarf an einem neuen Modell, das die Anforderungen von Server und Client erfüllen kann. Und das ist auch der Grund, warum die Single-Page-Application geboren wurde.

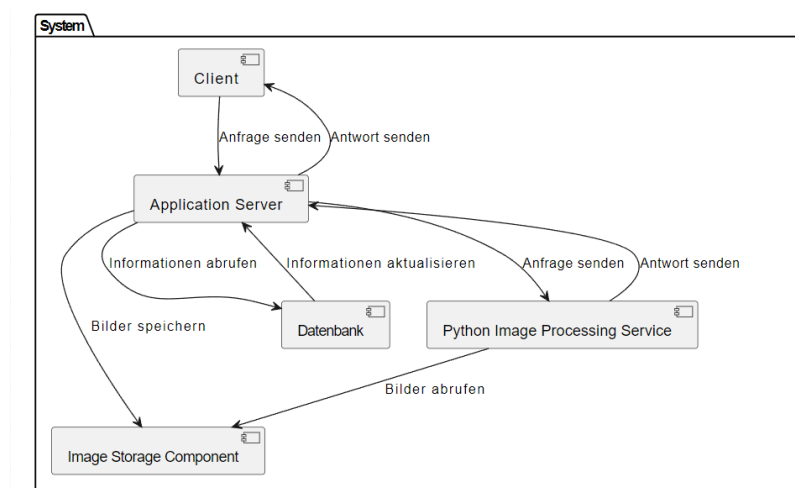
SPA ist eine Art der Webprogrammierung, bei der die Benutzer auf viele verschiedene Unterwebsites zugreifen können, ohne dass die ursprüngliche Website beeinträchtigt wird. Wenn ein Benutzer auf eine beliebige Komponente auf der Seite zugreift, wird bei SPA nur der Inhalt dieser Komponente ausgeführt, ohne dass die gesamte Seite wie bei herkömmlichen Websites neu geladen wird. Gemeinsame Elemente wie Kopfzeile, Fußzeile und Menüleiste bleiben unverändert.

SPA wird sich auf die Client-Verarbeitung konzentrieren und die Rolle des Frontends stärken. Das Frontend ist der Teil, in dem der Benutzer die Anfrage des Benutzers sieht und empfängt, wodurch die erforderlichen Funktionen und Daten bestimmt werden, um

dann die Anfrage an das Backend zu senden. Das Backend empfängt die Anfragen und gibt die Daten an die Außenwelt der Website zurück. Dies erhöht die Benutzerfreundlichkeit und gibt den Benutzern das Gefühl, ein mobiles Gerät und nicht eine Website zu benutzen.

Diese Methode löst auch das Problem des Anwendungsfalls, das zuvor erwähnt wurde. Daher werden in dieser Arbeit die Erstellung von Websites nach der CSR-Methode und mit SPA angegangen.

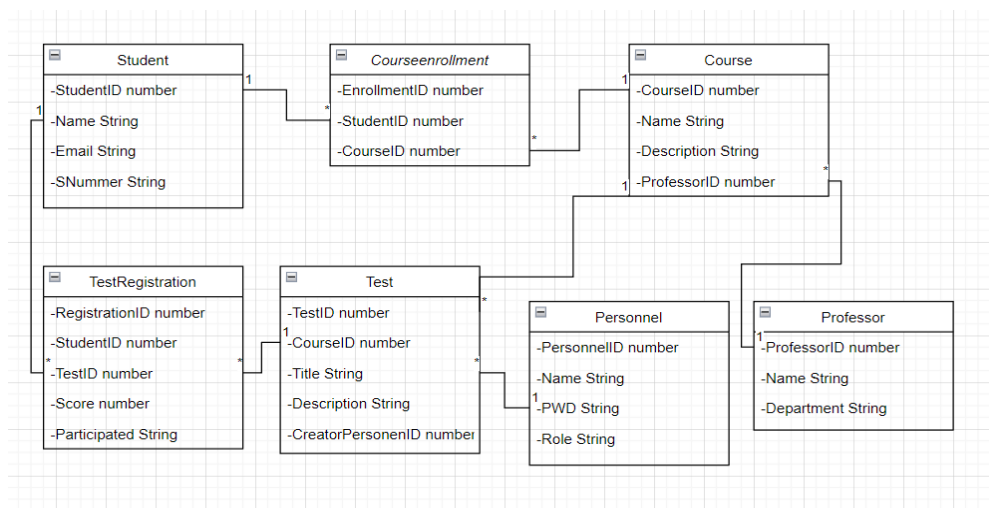
Zu diesem Zeitpunkt ist die Struktur des Clients festgelegt, die Struktur des Servers muss ebenfalls analysiert und aufgebaut werden. Das System kann als Komponentendiagramm wie folgt dargestellt werden:



**Abbildung 31** Component Diagram.

Im obigen Diagramm wurde der Client analysiert. Die übrigen Komponenten bilden den Server. Er umfasst einen Bildverarbeitungsdienst in Python, eine Datenbank zum Speichern der erforderlichen Informationen in der Software, eine Bildspeicherkomponente zum Speichern von Studentebildern auf dem Server und eine Serveranwendung, die alle diese Komponenten auf dem Server miteinander verbindet und APIs für den Client erstellt.

Um das Projekt detaillierter umsetzen zu können, müssen ausgehend vom obigen Anwendungsfalldiagramm die Klassen im Detail beschrieben, die Parameter dargestellt und die Beziehungen zwischen ihnen aufgezeigt werden. Zu diesem Zweck wird ein Klassendiagramm entwickelt.



**Abbildung 32** Class Diagram

Auf der Grundlage des obigen Anwendungsfalldiagramms sind die beiden grundlegendsten Klassen, die erstellt werden müssen, die Klasse Student und die Klasse Test. Grundlegende Studenten- und Testinformationen müssen ebenfalls hinzugefügt werden. Im Rahmen dieser Arbeit werden der Klasse Student einige Felder hinzugefügt, z. B. id, name, E-Mail. Andere Felder werden der Klasse Test hinzugefügt, z. B. id, Titel.

Es ist leicht zu erkennen, dass die Beziehung zwischen den beiden Klassen Test und Student eine Beziehung von „many-to-many“ ist. Um die Struktur zu vereinfachen und andere notwendige Informationen zu speichern, wurde daher eine zusätzliche Klasse mit dem Namen TestRegistration erstellt, die die Klassen Student und Test verbindet. Diese Klasse enthält Felder wie StudentID, TestID, Partizipation, um die Anwesenheit zu markieren, oder Score, um das Testergebnis des Studenten zu speichern.

Die Klasse Personnel ist ebenfalls implementiert, sie stellt die Mitarbeiter dar. In dieser Arbeit wird jedoch der Name Personnel gewählt, damit er in Zukunft auf die Abteilungen der Hochschule erweitert werden kann. Diese Klasse enthält ebenfalls grundlegende Informationen über die Mitarbeiter. Darüber hinaus ist es notwendig, Benutzernamen und Passwörter zu speichern, um den oben erwähnten Anwendungsfall Anmelden zu bedienen.

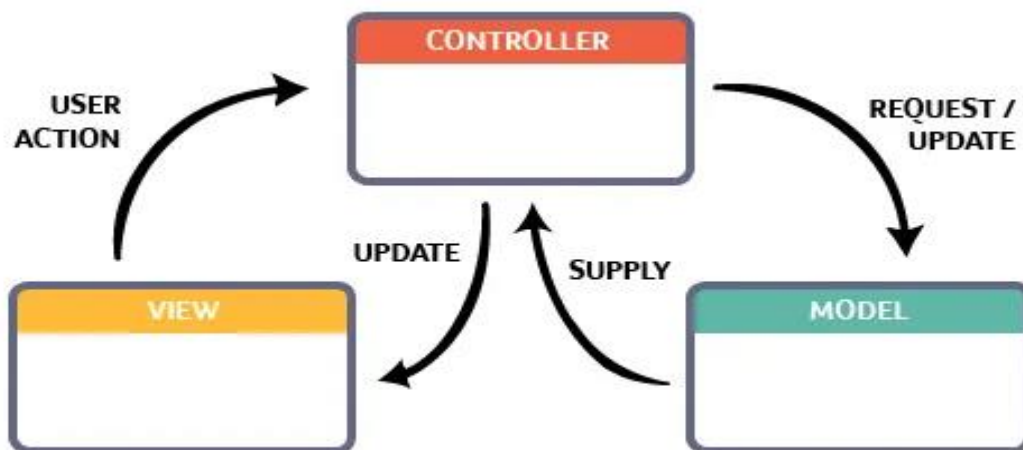
Um die notwendigen Informationen zu konkretisieren, werden außerdem Klassen wie Course und Professor initialisiert. In der Untertabelle Courseenrollment werden "many-to-many" mit der Klasse Student verknüpft, um den Anwendungsfall Kursmeldung zu repräsentieren.

Es sei darauf hingewiesen, dass dies wichtige Klassen sind, um ein Programm erstellen zu können. Damit eine Server-Application-Komponente jedoch mit anderen Komponenten im Komponenten-Diagramm interagieren kann, müssen gleichzeitig auch Utils-Klassen entwickelt werden, also Klassen, die Methoden für die Arbeit mit Python und Ordnern

bereitstellen. In diesem Projekt werden AIUtils und FileUtils die beiden Klassen sein, die diese Rolle übernehmen.

### 6.1.3 MVC-Model

Ein Problem, das hier auftritt, ist: Wenn die Anzahl der Klassen immer größer wird, nehmen auch die in der Klasse verwendeten Methoden zu, und es wird schwierig, sie zu verwalten und für zukünftige Software zu erweitern. Das ist auch einer der Hauptgründe, der zur Entstehung des MVC-Modells geführt haben. Das MVC-Modell ist in separate Teile (Model - Controller - View) und separate Funktionen der Teile unterteilt. Dies trägt zur Verbesserung der Projektstruktur bei und macht das Projekt einfacher und besser wartbar. Auf das aktuelle Problem angewandt, fungiert der Client als View und die Server-Anwendung als Controller und Model-Instanz.

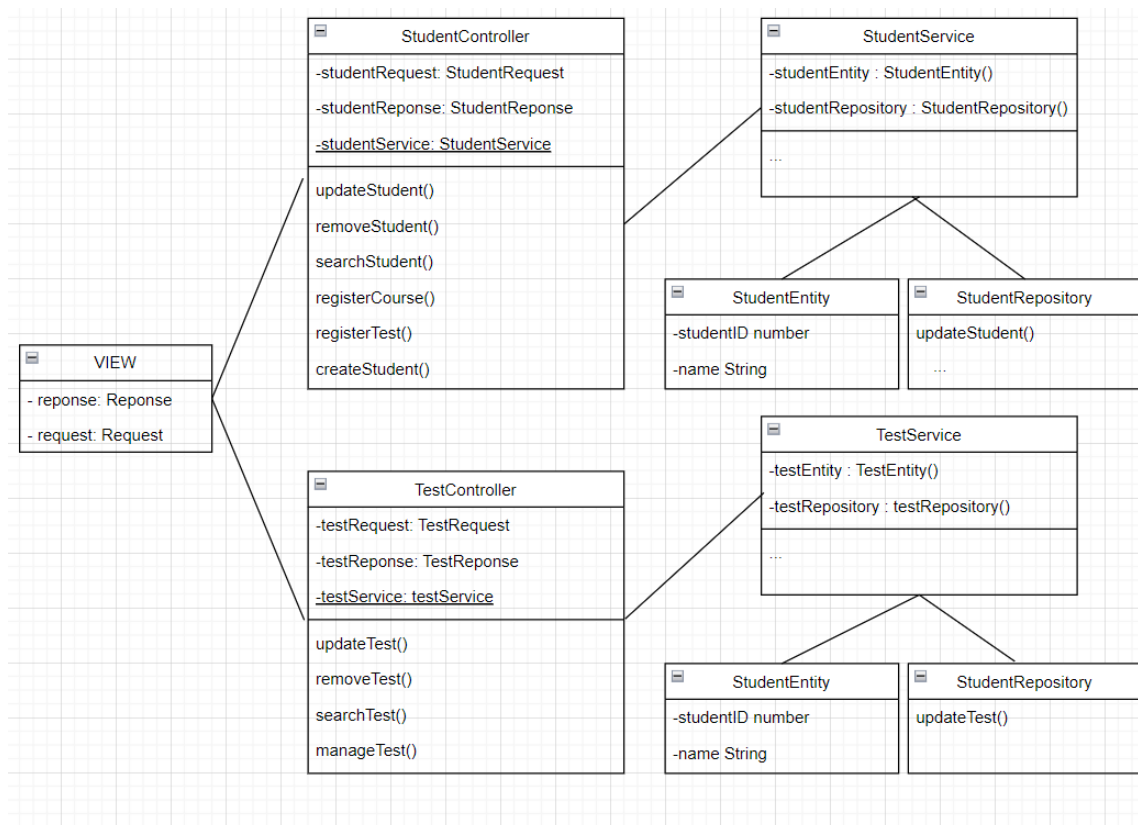


**Abbildung 33** MVC-Model (<https://vietnix.vn/tim-hieu-mo-hinh-mvc-la-gi/>)

Weitere Informationen über das MVC-Muster finden sich unter <https://developer.mozilla.org/en-US/docs/Glossary/MVC>.

Diese Trennung schafft Klarheit, vereinfacht die Wartung und erleichtert die zukünftige Erweiterung und Entwicklung der Software.

Zu diesem Zeitpunkt hat die MVC-Klasse des Projekts die Form:



**Abbildung 34** MVC-Diagramm

Durch dieses Diagramm versteht man auch klar, wie MVC funktioniert.

Jede Komponente übernimmt, wie oben erwähnt, verschiedene Rollen. Vor allem die Modellschicht kann in andere Komponenten unterteilt werden, um verschiedene Rollen zu übernehmen. Zum Beispiel, Repository, um mit der Datenbank zu arbeiten, Entity, um eine Studentenklasse zu repräsentieren. Sie werden dann kombiniert und durch die Logik der Serviceschicht verarbeitet und geben dann den Wert an den Controller zurück.

Eine MVC-Architektur kann komplex sein, ist aber bei komplexen Projekten sehr übersichtlich und wertvoll. Darüber hinaus gibt es heute viele Bibliotheken, die die Erstellung von MVC-Modellen auf einfache Weise unterstützen. Hierauf wird in den folgenden Kapiteln noch näher eingegangen. Im Rahmen dieses Projekts wird das MVC-Modell jedoch für die Erstellung der Website verwendet.

Nach der Analyse und dem Entwurf der Struktur der Webanwendung können folgende Schlussfolgerungen gezogen werden:

- Das Programm wird nach Client-Side-Rendering-Ansätzen aufgebaut, um die Benutzerfreundlichkeit und vor allem die Geschwindigkeit der Programmausführung zu gewährleisten.
- Die Client-Seite wird in Richtung SPA entwickelt (ist View in MVC).

- Die Serverseite wird nach dem MVC-Modell aufgebaut. Zu den Komponenten gehören: Backend-Sprache, Python für die Bildverarbeitung, Ordner zum Speichern der Bilder und ein Datenbanksystem zum Speichern der notwendigen Daten des Programms.

Das Design ist vollständig, aber es fehlt die Technologie, um es in der Praxis anzuwenden. Daher werden im nächsten Abschnitt Technologien untersucht, die auf diesen leeren Entwurf angewendet werden können.

## **6.2 Client**

Entsprechend dem erwähnten Ansatz, Client-Side-Rendering und SPA einzusetzen, werden eine Reihe von Frontends Frameworks analysiert und eingesetzt, um die Arbeit der Frontend-Programmierung einfacher und effektiver zu gestalten.

### **6.2.1 Frontends Framework**

Es gibt viele bekannte Frameworks wie die Folgenden

#### **6.2.1.1 VueJs**

Vue kam 2014 auf den Markt und wurde von Evan You entwickelt.

VueJs ist ein flexibles Framework, das zur Erstellung von Benutzeroberflächen verwendet wird. Im Gegensatz zu monolithischen Frameworks ist Vue von Grund auf so konzipiert, dass es eine schrittweise Anwendungsentwicklung ermöglicht und fördert. Bei der Entwicklung der View-Schicht müssen die Benutzer nur die Kernbibliothek von Vue verwenden, die sehr einfach zu erlernen und in bestehende Bibliotheken oder Projekte zu integrieren ist. Gleichzeitig kann Vue in Kombination mit modernen Techniken wie SFC (Single-File-Components) und Support-Bibliotheken auch problemlos die Anforderungen für die Entwicklung von Single-Page-Anwendungen erfüllen. ) mit viel höherer Komplexität erfüllen.

Hauptseite: <https://vuejs.org/>

#### **6.2.1.2 React**

React ist eine leistungsstarke und flexible JavaScript-Bibliothek zur Erstellung von Benutzeroberflächen (UI), insbesondere für einseitige Webanwendungen (SPAs). Sie wurde von Facebook entwickelt und ist derzeit eine der beliebtesten Front-End-Technologien. Hier sind einige Highlights über React:

Hauptseite: <https://react.dev/>

### 6.2.1.3 Angular

Angular ist eine leistungsstarke Plattform und ein Framework für die Entwicklung dynamischer Webanwendungen. Angular wird von Google entwickelt und ist Teil der JavaScript-Frontend-Technologiegruppe. Es bietet einen umfassenden Ansatz für die Erstellung von Anwendungen, der von der Gestaltung der Benutzeroberfläche bis zur Verwaltung des Anwendungsstatus und der Interaktion mit dem Back-End reicht.

Hauptseite: <https://angular.io/>

**Aufgrund der leicht erlernbaren und zugänglichen Eigenschaften von VueJS, aber auch der Unterstützung von mächtigen Features für den Aufbau von SPA, wird im Rahmen dieser Arbeit VueJS für die Frontend-Programmierung verwendet werden.**

Darüber hinaus wird das Projekt VueJs Technologie zwei, Vuex und Vue Router, verwenden, um den Entwicklungsprozess einfacher und effektiver zu gestalten.

1. Vuex: ist eine Bibliothek in VueJs, die den Zustand von Vue-Objekten verwaltet, indem sie diese in "Store" speichert. Dies ist notwendig, da die Website auf einer SPA-Basis aufgebaut ist. Vuex hilft beim Speichern und Verarbeiten von Vue-Elementen, wenn diese Elemente in vielen verschiedenen Komponenten verwendet werden. Dadurch wird vermieden, dass eine API mehrfach aufgerufen werden muss, und API-Aufrufe werden in Stores und nicht in Komponenten aufgeteilt. Das macht die Codeverwaltung und Wartung einfacher. (mehr Information auf <https://vuex.vuejs.org/>)
2. Vue Router: ist eine wichtige Bibliothek im Vue.js Ökosystem, die verwendet wird, um die Navigation in Single Page Application (SPA) Anwendungen zu verwalten, die mit Vue.js erstellt wurden. (mehr Information auf <https://router.vuejs.org/>)

## 6.3 Server

Im vorangegangenen Abschnitt wurde erwähnt, dass das SPA-Modell mit VueJs verwendet wurde, um die Anforderungen an die Erstellung benutzerfreundlicher Websites zu erfüllen. Das bedeutet, dass das Rendering auf der Client-Seite verstärkt (Client-Side-Rendering). Der Client muss nur noch APIs empfangen, die die notwendigen Werte zurückgeben, und nicht mehr eine serverseitig generierte Website (Server-Side-Rendering). Das MVC-Modell ist jedoch nach wie vor ein leistungsfähiges Modell für die Verwaltung und Optimierung auf der Serverseite. Daher muss der Controller-Teil geändert werden, dass er APIs zurückgibt, die den Anforderungen der Client-Seite entsprechen.

Um diesen Bedarf zu decken, muss eine Technologie für die Serverkonstruktion gewählt werden. Derzeit gibt es viele Sprachen und leistungsstarke Technologien, die diese Anforderungen erfüllen können. Im Rahmen dieser Arbeit jedoch Java und die dazugehörigen Technologien wie Java Spring Boot und Spring Security aus Gründen verwendet, die im nächsten Abschnitt erläutert werden.

### **6.3.1 Java**

Java ist eine der objektorientierten Programmiersprachen. Sie wird in der Softwareentwicklung, für Websites, Spiele oder Anwendungen auf mobilen Geräten verwendet. Java wurde 1991 von James Gosling und seinen Kollegen bei Sun Microsystems ins Leben gerufen.

Java wurde unter dem Kriterium "Write Once, Run Anywhere - WORA" entwickelt. In Java geschriebene Softwareprogramme können über eine Ausführungsumgebung auf jeder beliebigen Plattform ausgeführt werden, sofern es eine geeignete Ausführungsumgebung gibt, die diese Plattform unterstützt.

Obwohl in der heutigen Webprogrammierung viele neue Technologien aufgetaucht sind, ist Java immer noch ein Riese in dieser Branche. Der Grund dafür ist, dass Java Web schon seit langer Zeit entwickelt wird und sich bewährt hat. Für viele Projekte, insbesondere große Projekte oder Projekte, die eine hohe Stabilität und Sicherheit erfordern, ist Java immer die erste Wahl.

Da es sich um eine langfristige Entwicklungsplattform handelt, haben sich die Java-Community und die Frameworks zudem stark entwickelt.

Im Rahmen des Projekts ist die Gesichtserfassung ein Projekt, das sehr hohe Stabilität und Sicherheit erfordert. Denn die Anwesenheit per Gesicht hat direkten Einfluss auf die Prüfungsergebnisse der Studenten, aber auch auf den Arbeitsprozess der Mitarbeiter der HTW Dresden. Daher erfordert das Projekt eine Programmiersprache und Technologien, die stark, stabil und in der Praxis bewährt sind. Darüber hinaus wird ein weiteres Java-Framework, Spring Boot, zum Einsatz kommen.

#### **6.3.1.1 Java Spring Boot**

Spring Boot ist ein von JAV (Java-Sprache) entwickeltes Projekt im Spring-Framework-Ökosystem. Es hilft uns Programmierern, den Prozess der Programmierung einer Anwendung mit Spring zu vereinfachen und sich nur auf die Geschäftsentwicklung der Anwendung zu konzentrieren.

Hauptseite: <https://spring.io/projects/spring-boot/>



### 6.3.1.2 Spring Security

Spring Security ist ein leistungsfähiges Werkzeug, das bei der Erstellung und Gewährleistung der Programmsicherheit hilft. Im Rahmen dieser Arbeit werden Spring Security und einige Anpassungen angewendet, um die Ziele der Arbeit zu erreichen. Eine davon ist die Erstellung von Anmeldefiltern und die Überprüfung des Anmeldestatus auf der Grundlage des JWT-Mechanismus. Dies wird im nächsten Abschnitt im Detail gezeigt.

Weitere Information über JWT finden sich unter

*[https://en.wikipedia.org/wiki/JSON\\_Web\\_Token](https://en.wikipedia.org/wiki/JSON_Web_Token)*

Man kann feststellen, dass Java Spring Boot ein leistungsfähiges Java-Framework ist, das bei der Erstellung geeigneter Webanwendungen nach dem MVC-Modell hilft und durch Spring Security auch gute Sicherheitsmethoden bietet. Daher wird in dieser Arbeit Java Spring Boot verwendet.

### 6.3.2 Bilderordner

Bei der Gesichtserkennung müssen sowohl die Bilder, die zum Training gehören, als auch die Bilder, die zur Gesichtserkennung gehören, gespeichert werden. Hier gibt es zwei Hauptmethoden: die Speicherung in der Cloud oder die Speicherung auf dem Server selbst.

Die Speicherung in der Cloud bietet Vorteile wie flexibler Zugriff, große Kapazität usw., hat aber auch Nachteile in Bezug auf die Wartungskosten, die von Dritten abhängige Sicherheit und die vom Anbieter abhängige Bereitstellung.

Daher werden im Rahmen dieses Projekts die Bilder direkt auf dem Server gespeichert. Außerdem dienen die Fotos in diesem Projekt nur den Aufgaben des Servers, so dass die flexiblen Zugriffsmöglichkeiten von Cloud-Speichern nicht benötigt werden.

### 6.3.3 Datenbank

Auch eine Datenbank muss ausgewählt werden. Im Rahmen dieser Arbeit wird MySQL verwendet.

MySQL ist ein quelloffenes Datenbankmanagementsystem (Relational Database Management System, abgekürzt RDBMS), das nach dem Client-Server-Modell arbeitet. RDBMS ist eine Software oder ein Dienst, der zur Erstellung und Verwaltung von Datenbanken verwendet wird, indem er die Beziehungen zwischen ihnen verwaltet.

MySQL kann auf den meisten Betriebssystemen eingesetzt werden, ist leistungsstark, kostenlos und vor allem einfach zu erlernen und anzuwenden. Eine Schwäche von MySQL ist, dass es Probleme bei der Erweiterung des Systems mit einem großen Daten-

block gibt, da MySQL eine zentralisierte Architektur verwendet. Bei einer geringen Anzahl von Tabellen (weniger als 10 Tabellen) und einer geringen Anzahl von Studenten, die gespeichert werden müssen (für die HTW Dresden ca. 4800 Studenten), ist MySQL jedoch ein vollkommen geeignetes Datenbanksystem für die Verwendung in diesem Projekt.

### **6.3.4 Face Recognition**

Nachdem die grundlegenden Komponenten für die Erstellung des Projekts festgelegt wurden, muss die zentrale Funktion des Projekts in Angriff genommen werden: die Gesichtserkennung. Obwohl Java, wie im vorherigen Kapitel erwähnt, eine leistungsstarke Programmiersprache ist. Aber in diesem Projekt ist die Gesichtserkennung nicht die Stärke von Java. Es ist eine Tatsache, dass die Bibliotheken, die die Bildverarbeitung, die Gesichtserkennung im Besonderen, das maschinelle Lernen und das Deep Learning im Allgemeinen unterstützen, alle sehr gut entwickelt und in Sprachen wie Python implementiert sind. Es wäre aber auch eine Verschwendung, wenn wir die Stärken von Java aufgeben würden, weil eine einzige Funktion eine gut aufgebaute Struktur ändern müsste. Daher besteht die vorgeschlagene Lösung darin, Java zu verwenden, um die für die Bildverarbeitung erforderlichen Python-Dateien "aufzurufen". Java erhält dann die Werte zurück, die diese Python-Dateien ausgeben, und setzt den Verarbeitungsprozess des Programms fort. (Dies wird im nächsten Kapitel im Detail beschrieben).

#### **6.3.4.1 Python**

Python ist eine Programmiersprache auf hohem Niveau, die 1991 von Guido van Rossum entwickelt und erstmals veröffentlicht wurde. Die Stärke von Python liegt in seiner einfachen, lesbaren Syntax, die viele Bereiche von der Webentwicklung bis zur Wissenschaft unterstützt. Im Bereich des maschinellen Lernens und des Deep Learning zeichnet sich Python durch umfangreiche Bibliotheken wie TensorFlow, PyTorch und Scikit-Learn aus, die die Konstruktion und das Training komplexer Modelle vereinfachen. Im Bereich der Gesichtserkennung unterstützt Python Bibliotheken wie OpenCV, die eine effektive Verarbeitung und Analyse von Bildern ermöglichen. Allerdings hat Python im Vergleich zu kompilierten Sprachen wie C++ oder Java auch Leistungs Nachteile, vor allem bei schweren Rechenaufgaben.

Wie bereits erläutert, verfügt Python über leistungsfähige Bibliotheken für die Gesichtserkennung. In dieser Arbeit wurde zum Beispiel die Bibliothek `face_pytorch` verwendet.

## 6.4 Implementierung

Durch die Theorien und Techniken, die in den vorherigen Kapiteln gelernt wurden, gibt es ein paar Punkte, die vor der detaillierten Programmierung zusammengefasst werden müssen:

- Die Web-Anwendungssoftware wird mit dem MVC-Modell programmiert.
- Die View, die als Client fungiert, wird mit Hilfe der VueJs-Technologie und den zugehörigen Bibliotheken wie View Router und Vuex implementiert.
- Die Serverseite wird die Aufgaben von Controller und Model übernehmen. Die Java Spring Boot-Technologie wird eingesetzt, um APIs zu erstellen, Dateien zu verarbeiten, mit der Datenbank zu kommunizieren und Python-Dateien zur Ausführung aufzurufen.

### 6.4.1 Datenbank

Ich habe eine Datenbank erstellt, die sich auf zwei Hauptobjekte konzentriert: Studenten und Prüfungen. Weiterhin gibt es andere Tabellen, die die Informationen für die beiden anderen Haupttabellen ergänzen. Die Struktur und die Beziehungen sind in der folgenden Abbildung dargestellt

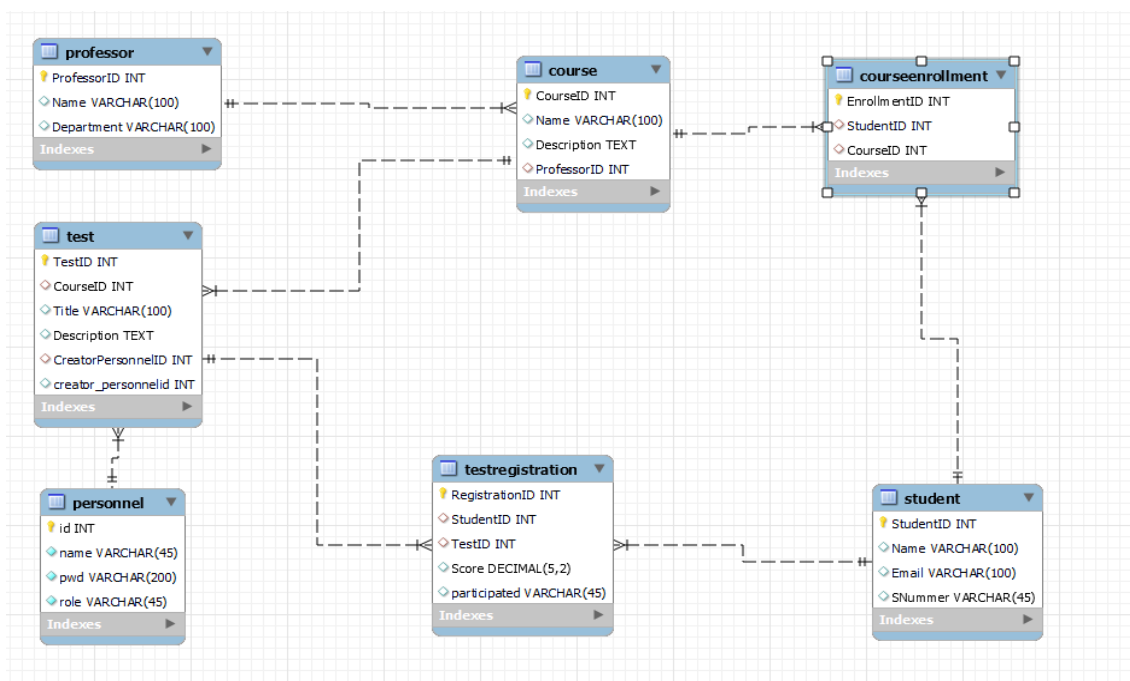


Abbildung 35: MySQL

In dieser Struktur erstelle ich eine Personaltabelle, die eine Tabelle für die Mitarbeiter der HTW ist (in der Tat simuliert diese den Prüfungsamtsmitarbeiter). In dieser Tabelle werden auch die Konten und Passwörter für die Verwaltung der Website sowie für die Verwaltung der Studenten und die Erstellung von Tests gespeichert. Die Informationen der

Tabellen beschränken sich derzeit auf die Basisinformationen. Während des Erweiterungsprozesses können die erforderlichen Informationen hinzugefügt werden. In der angehängten USB-Datei befindet sich eine dump.sql-Datei, die die Initialisierung der Datenbank mit einigen vorab erstellten Studenteninformationen zu Testzwecken unterstützt. (Hinweis: Bitte ein MySQL-Root-Konto mit dem Passwort 12345 erstellen.)

Diese Struktur wird wie folgt beschrieben: Die Tabelle "personnel" speichert Informationen über den Mitarbeiter der HTW Dresden, der die Tests direkt durchführt. Zur Vereinfachung des Modells wurde eine 1: n-Beziehung mit der Tabelle "test" hergestellt. In der Tabelle "Test" werden wichtige Informationen über einen Test gespeichert, u.a.: Course (zu welchem Kurs gehört dieser Test, dies ist eine n:1 Beziehung, viele Tests können zu einem Kurs gehören). Die Tabelle "Test" steht in einer n: n-Beziehung zur Tabelle "Student" über die Untertabelle "testregistration". In dieser Untertabelle werden die Testergebnisse der Studenten gespeichert. Die Tabelle "Kurs" speichert ebenfalls grundlegende Informationen über den Kurs und ist n:1 mit der Tabelle "Professor" verknüpft, da ein Professor mehrere Kurse unterrichten kann. Die Tabelle "Course" hat auch eine n: n-Beziehung mit der Tabelle "Student" über die Untertabelle "courseenrollment", die anzeigt, welche Kurse dieser Student belegt.

Diese Struktur beschreibt die Beziehungen zwischen den Tabellen in der Datenbank genau. Von hier aus können Funktionen erstellt und Daten einfach gespeichert werden

#### **6.4.2 Bilderordner**

In dem Projekt werden beim Anlegen eines neuen Studenten drei seiner Fotos benötigt, um das Gesichtserkennungsmodell zu trainieren und um die Fotoinformationen anzuzeigen. Diese Bilder werden auf dem Server in einem Ordner gespeichert (auf dem USB-Stick ist das der Ordner Student). Diese Ordner werden unter dem Namen Student sNummer gespeichert, um die Dateiverarbeitung auf dem Server zu erleichtern.

#### **6.4.3 Python**

Python empfängt die in den Ordnern gespeicherten Bilder, führt Aufgaben wie Gesichtserkennung und Modelltraining durch und gibt die Daten zur Verarbeitung an die Java-Seite zurück. Der USB-Quellcode von Python befindet sich im Ordner Python. Es ist auch eine requierment.txt-Datei verfügbar, die Informationen über die verwendeten Bibliotheken enthält.

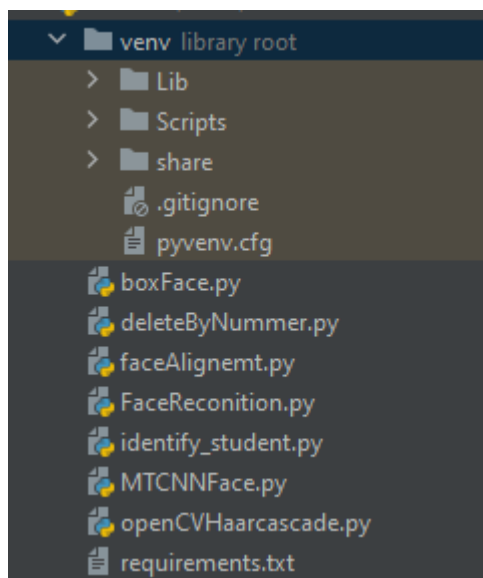


Abbildung 36 Python

Die Python-Dateien führen die Funktionen aus, die ihr Name beschreibt:

Tabelle 4: Python

Name	Input	Output
boxFace.py	Direkt in File	Bilder mit Face Box Bouding
<b>deleteByNummer.py</b>	sNummer	Embedding von Studenten mit sNummer in embedding.pkl löschen
faceAlignemt.py	Direkt in File	Richten Sie das Foto nach dem Gesicht aus
<b>Identify_student.py</b>	void	sNummer für Student mit Bildern in Studenten\Recognizer
MTCNNFace.py	void	Bilder mit Face-Box-Bouding durch MTCNN
OpenCVHaarcascade.py	void	Bilder mit Face-Box-Bouding durch Haar-Kaskade
<b>FaceReconition.py</b>	sNummer	Trainieren die Bilder aus dem erstellten Ordner und speichern sie in der Datei embedding.pkl.

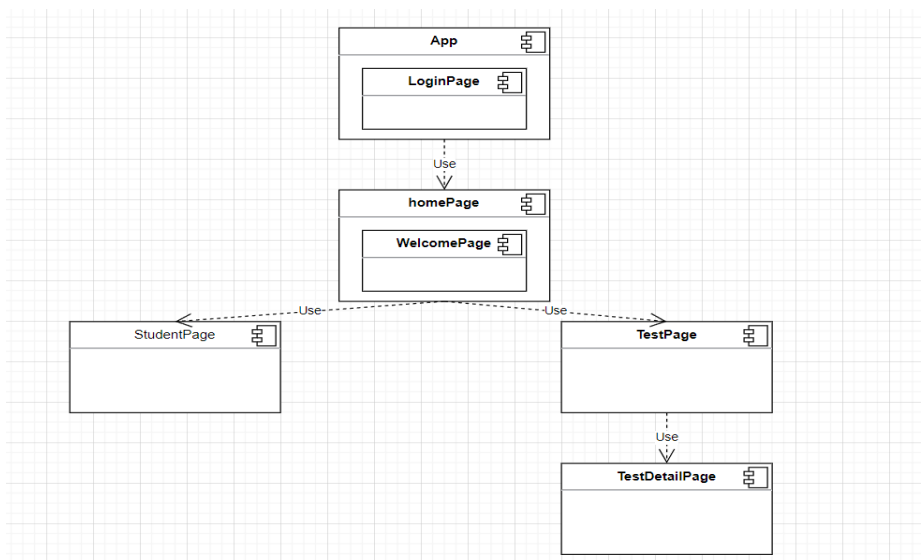
Die obige Tabelle beschreibt die Funktionen der Dateien. Dateien mit fettgedruckten Namen sind Dateien, die so aufgebaut sind, dass Java die notwendigen Werte aufrufen und zurückgeben kann:

- deleteByNum-mer.py: Java sendet sNummer und Python führt die Löschung der Einbettung in der Datei embedding.pkl durch.
- FaceReconition.py: Java sendet sNummer und Python trainiert das Modell aus der angegebenen Bildquelle und speichert es in der Datei embedding.pkl.
- Identify\_student.py: Java sendet einen Bildpfad-Link an Python, und Python identifiziert die sNummer des Studenten und gibt sie zurück; wenn dies nicht gelingt, ist es None.

Die übrigen Dateien werden nicht für die Ausführung in Java unterstützt, sondern für die Durchführung von Tests, die direkt aus den gemessenen Python-Dateien ausgeführt werden.

#### 6.4.4 Client

Die Struktur des Vuejs-Frontends sieht wie folgt aus:



**Abbildung 37:** VueJS -Client

Beschreiben die Struktur des Client-Teils entsprechend dem obigen Komponentendiagramm wie folgt:

Standardmäßig wechselt das Programm zur LoginPage. Nach erfolgreicher Anmeldung wechseln Sie zur Seite HomePage, die standardmäßig die Seite WelcomePage ist. Diese Seite hat gemeinsame Komponenten für StudentPage und TestPage, d. h. die Komponente Header. Hier können die Benutzer wahlweise zur StudentPage oder TestPage wechseln. Für jede Anfrage navigiert VueJs mit der Vue Router Bibliothek den Benutzer automatisch, basierend auf seiner Anfrage. In der TestPage gibt es einen Navigationsbutton, so dass der Benutzer zu der Verwaltungsseite für jeden einzelnen Test gehen kann. Das ist die TestDetailPage Seite.

Store werden die Module Student, Test und Course in Vuex importiert. Der Zweck von Vuex ist es, den Zustand von Vue-Entitäten zu überwachen und zu aktualisieren. Dies ist sehr nützlich, wenn der Wert einiger Variablen über viele verschiedene Komponenten der Anwendung verwendet wird. Darüber hinaus trennt die Verwendung von Vuex auch Funktionen, die zum Aufrufen von APIs verwendet werden (diese Funktionen werden in der action.js jedes Moduls sein). Das macht die Wartung und Entwicklung einfacher. Wenn Module klar und einfach zeigen, auf welches Objekt sie abzielen. Zum Beispiel rufen alle APIs des Student-Moduls APIs auf, die sich auf Studenten beziehen. Das ist fast so, als würde der StudentController auf der Server-Seite nur die APIs verarbeiten, die vom Student-Modul auf der Vuex-Seite kommen.

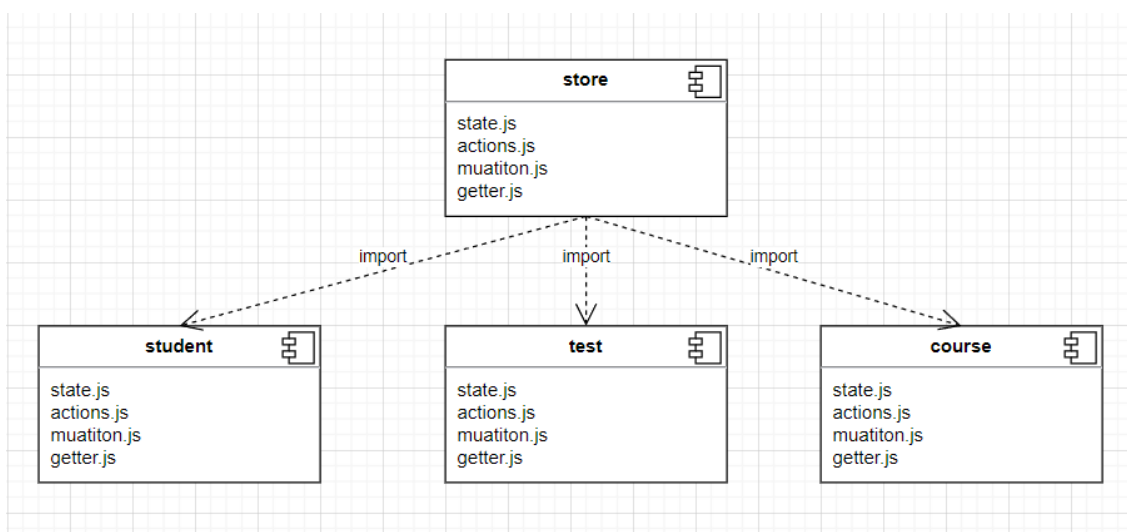
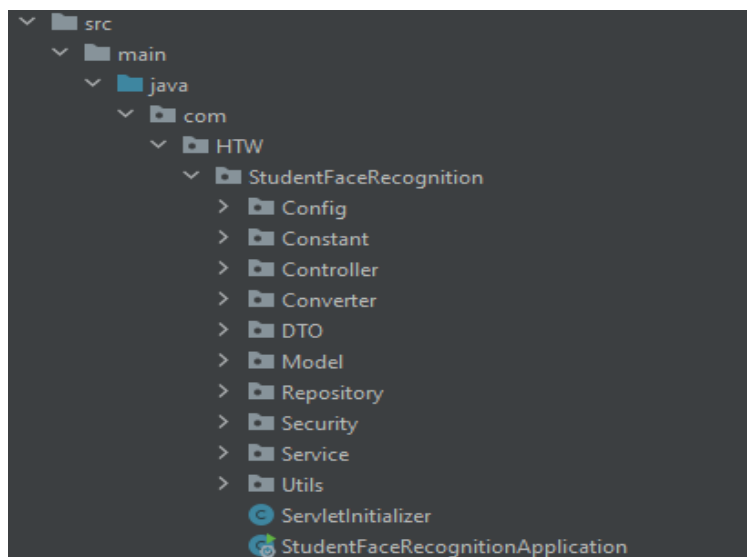


Abbildung 38 Vuex Diagram.

### 6.4.5 Java

Java ist die Seele des gesamten Systems. Es ist für die Verbindung aller Komponenten und die Handhabung der oben genannten APIs verantwortlich. Java hat in dem Projekt die folgende Struktur:



**Abbildung 39:** Javastruktur

Ausgehend von dem Diagramm in Abbildung 34 können weitere, weniger wichtige Klassen hinzugefügt werden, um das MVC-Modell mit dem Java Spring Boot Framework zu implementieren.

Java wird insbesondere in den folgenden vier Punkten eingesetzt:

1. Erstellen von CRUD-APIs für Studenten und Tests. Ich wende das MVC-Modell im gesamten Programm an. Diese Pfade sind Pfade, die der Client zum Aufrufen von APIs verwenden kann.

**Tabelle 5: API**

Path	Bedeutung
.../test/	API für Prüfungen
.../student/	API für Student
.../signin	API für Anmelden
.../course	API für Kurs

2. Mit Python "kommunizieren". Abhängig von der aufgerufenen API kommuniziert Java mit Python und ruft es auf, um Python-Dateien über die ProcessBuilder-Klasse auszuführen (zu finden in Utils/AIUtils).



```

public static void trainModel(String imagePath, String.pklPath, String studentID, String pythonPath) throws IOException, InterruptedException {
    String[] cmd = {
        FileConstant.pythonInterpreter,
        pythonPath,
        imagePath,
       .pklPath,
        studentID
    };

    ProcessBuilder pb = new ProcessBuilder(cmd);
    Process process = pb.start();

    BufferedReader bfr = new BufferedReader(new InputStreamReader(process.getInputStream()));
    String line;
    while ((line = bfr.readLine()) != null) {
        System.out.println(line);
    }

    int exitCode = process.waitFor();
    System.out.println("Exited with code: " + exitCode);
}

```

**Abbildung 40:** Kommunikation zwischen Java und Python.

3. Anpassung von Repositories: Zusätzlich zu den grundlegenden Repositories, die JPA bietet, gibt es manchmal komplexe Funktionen, die die grundlegenden Repositories von JPA nicht erfüllen können. Daher habe ich einige der Repositories im Programm angepasst, damit es effizienter läuft, vor allem für die Suchfunktion mit mehreren Parametern in verschiedenen Tabellen je nach Studenten und Test. (zu finden im Ordner 'Repository/custom').

```

@Repository
public class StudentRepositoryImpl implements StudentRepositoryCustom {
    1 usage
    @PersistenceContext
    private EntityManager entityManager;
    1 usage
    @Override
    public List<StudentEntity> searchStudent(StudentRequest studentRequest) {...}
}

```

**Abbildung 41:** Repository custom

### 3. Implementierung einer benutzerdefinierten Java Spring Security

Das Bild (Abbildung 42) zeigt die Code-Implementierung eines benutzerdefinierten LoginFilters. Dieser benutzerdefinierte Filter erbt den OncePerRequestFilter von Spring Security, der Requests als Parameter akzeptiert und für die Rückgabe von Responses konfiguriert werden kann. Beim Erben von OncePerRequestFilter muss LoginFilter diesen durch Überschreiben von zwei Methoden implementieren: doFilterInternal und shouldNotFilter.

- doFilterInternal ist der Ort, an dem die Logik implementiert wird. Bei LoginFilter werden Informationen wie Benutzername und Kennwort durch eine Authentication-Klasse überprüft. In Spring Security wird standardmäßig automatisch eine Authentication in der Datenbank mit Spalteninformationen gefunden, die als Standard konfiguriert, werden müssen. Weil aber in diesem Projekt die Spaltennamen an den Kontext angepasst wurden, ist es notwendig, eine Authentication-Klasse weiter anzupassen, um die Authentifizierungsaufgabe durchzuführen (Abbildung 38). Diese Klasse prüft den Benutzernamen und das Passwort in der Datenbank mit Hilfe von JPA. Schlägt dies fehl, wird der Client benachrichtigt. Im Erfolgsfall

wird ein Token erstellt, das Informationen über den Benutzer speichert und an den Client zurückgeschickt wird. Ab diesem Zeitpunkt muss jede aufgerufene API dieses Token enthalten. Der nächste benutzerdefinierte Filter ist für die Überprüfung dieses Tokens verantwortlich.

-shouldNotFilter ist eine Klasse, die es Programmierern ermöglicht, Pfade zu definieren, auf die dieser Filter nicht angewendet werden muss.

```
@Log4j2
@Component
public class CustomRequestHeaderTokenFilter extends OncePerRequestFilter {
    @Autowired
    private PersonnelUserDetails personnelUserDetails;

    no usages
    @Override
    protected void doFilterInternal(HttpServletRequest request, HttpServletResponse response, FilterChain filterChain) throws ServletException, IOException {
        try {
            String token = request.getHeader(SecurityConstant.AUTH_HEADER_TOKEN);
            if (token != null && !token.isEmpty() && TokenUtils.isJWTTokenValid(token)) {
                String uname = TokenUtils.getUsernameFromJWTToken(token);
                UserDetails userDetails = personnelUserDetails.loadUserByUsername(uname);
                if (userDetails != null) {
                    UsernamePasswordAuthenticationToken authentication = new UsernamePasswordAuthenticationToken(uname, credentials: null, userDetails.getAuthorities());
                    SecurityContext context = SecurityContextHolder.createEmptyContext();
                    context.setAuthentication(authentication);
                    SecurityContextHolder.setContext(context);
                    response.setStatus(HttpServletResponse.SC_ACCEPTED);
                    filterChain.doFilter(request, response);
                } else {
                    log.info("user not found");
                    setUnsuccessfulResponse(response, "message: "user not found");
                    return;
                }
            } else {
                log.info("token is null or empty or not valid");
                setUnsuccessfulResponse(response, "message: "token is null or empty or not valid");
                return;
            }
        } catch (Exception ex) {
            log.error("Failed to process authentication request: " + ex.getMessage());
            setUnsuccessfulResponse(response, "message: "Failed to process authentication request:");
            return;
        }
    }
}
```

Abbildung 42 Login Filter

```
@Component
@Log4j2
public class CustomAuthenticationManager implements AuthenticationManager {
    @Autowired
    PersonnelUserDetails personnelUserDetails;
    3 usages
    UsernamePasswordAuthenticationToken authToken = null;
    @Override
    public Authentication authenticate(Authentication authentication) throws AuthenticationException {
        String uname = String.valueOf(authentication.getName());
        String upassw = String.valueOf(authentication.getCredentials());
        if (authentication == null) {
            return null;
        }
        UserDetails userDetails = personnelUserDetails.loadUserByUsername(uname);
        log.info("userDetail : "+userDetails.toString());
        if (userDetails == null) {
            log.info(new UsernameNotFoundException(String.Format("Username not found").toString()));
            return null;
        }
        if (userDetails.getPassword().equals(upassw)) {
            authToken = new UsernamePasswordAuthenticationToken(userDetails.getUsername(), credentials: null, userDetails.getAuthorities());
            log.info("authToken : "+authToken);
        } else {
            log.info("authToken false:");
            return null;
        }
        return authToken;
    }
}
```

Abbildung 43 Custom Authentication Manager

(Weitere Informationen über Filter und Spring Security finden sich auf folgender Website <https://medium.com/@zzpzaf.se> oder <https://spring.io/projects/spring-security/>)

## 6.5 Fazit

In diesem Kapitel habe ich eine Aufgabenanalyse durchgeführt, um ein System zu entwickeln. Dieses System ist eine Kombination aus vielen verschiedenen Techniken. Erstens ist die Client-Seite der Ansatz zur Erstellung von Websites nach dem Client-Side-Rendering sowie der SPA-Struktur. Bei der Serverseite handelt es sich um das von Java implementierte MVC-Modell. Darüber hinaus spielt Java auch eine wichtige Rolle bei der Anbindung anderer Komponenten auf der Serverseite. Das heißt, der Aufruf einer Python-Datei zur Durchführung von Aufgaben im Zusammenhang mit der Gesichtserkennung, die Kommunikation mit der MySQL-Datenbank und die Speicherung und Verwaltung von Dateien in Ordnern.

An diesem Punkt haben in der Theorie ein leistungsfähiges und hochsicheres Webanwendungssystem aufgebaut, um das zweite Problem [2] zu lösen. Damit ist die Zielfrage in der Theorie gelöst. Um jedoch einen genauen Überblick und eine Bewertung der Wirksamkeit zu erhalten, muss das Programm in der Praxis getestet werden.

Im nächsten Kapitel wird die Software in der Praxis getestet, wobei eine Reihe von Tests durchgeführt wird, um die Leistung zu überprüfen und festzustellen, ob das Programm die ursprünglichen Anforderungen erfüllt oder nicht.

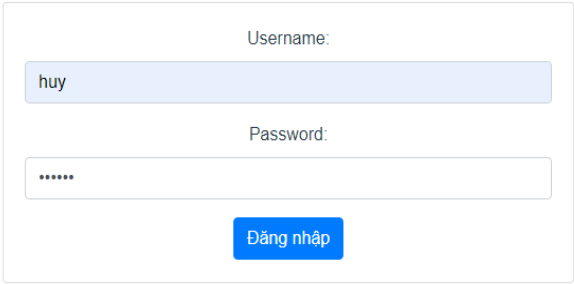
## 7 Programmablauf

Tatsächlichen Ablauf getestet werden, und es werden Tests durchgeführt, um zu prüfen, ob das Programm die ursprünglich festgelegten Ziele erreicht oder nicht. Dieses Kapitel ist in zwei Teile gegliedert: Teil 1 ist ein Abriss darüber, wie ein Programm in der Praxis funktionieren würde. Teil 2 ist den Tests und der Programmbewertung gewidmet.

### 7.1 Programmablauf

Im Folgenden findet man die Schnittstelle für die Verwaltung von Studenten und Prüfungen sowie Funktionen wie die Anwesenheit von Studenten. Wenn man eine Funktion verwendet, sendet sie die erforderlichen Anfragen an den Server. Der Server sendet daraufhin Antworten an den Client zurück. Informationen und Bilder werden von Client und Server synchronisiert.

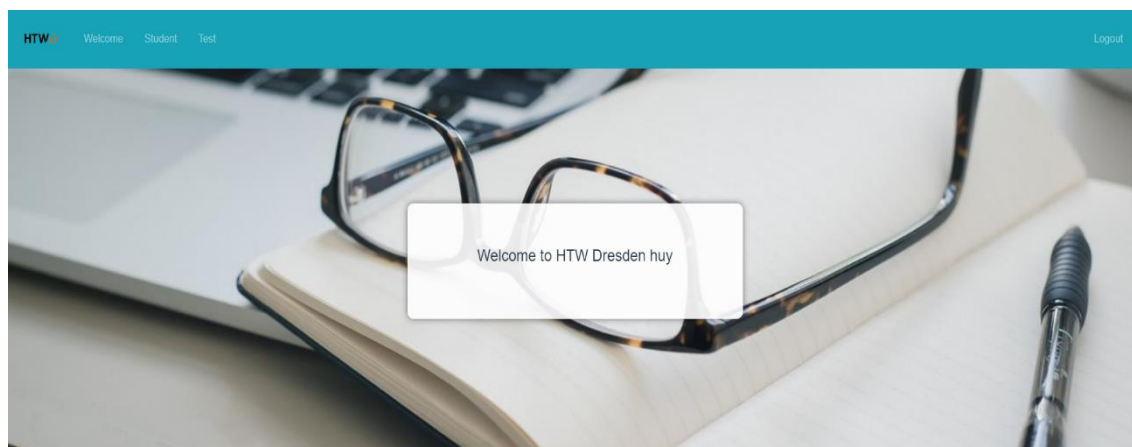
**HTW**



The image shows a login form for HTW. It has a light gray border and a white background. At the top, the text 'Username:' is followed by a text input field containing the name 'huy'. Below this, the text 'Password:' is followed by a password input field with masked characters '\*\*\*\*\*'. At the bottom center, there is a blue button with the white text 'Đăng nhập'.

**Abbildung 44:** Login

Die Anmeldeseite ist die Standardseite beim Zugriff auf die Website. Hier müssen sich die Benutzer anmelden und authentifizieren. Wenn der Benutzer den Benutzernamen und das Passwort eingibt, wird die Login-API vom Server aufgerufen. Wenn der Server diese Anfrage erhält, wird er nur den Filter 'LogInFilter.java' ausführen. Dieser benutzerdefinierte Filter ist so programmiert, dass er die Benutzerinformationen überprüft. Wenn die Informationen gültig sind, wird das Token zurückgegeben, die Kontrolle wird gewährt und der Benutzer wird zur Startseite weitergeleitet. Von hier aus muss jede API, die von der Client-Seite aufgerufen wird, dieses Token zur Authentifizierung über den Filter 'CustomRequestHeaderTokenFilter.java' bereitstellen. Dieser Filter ist für die Verifizierung des Tokens verantwortlich. Wenn es korrekt ist, wird die API bereitgestellt, wenn es nicht korrekt ist, wird die API nicht bereitgestellt.



**Abbildung 45** Welcome Page

Auf der Willkommenseite können die Benutzer zu zwei Seiten navigieren: Student und Test. Jede Seite wird für die Verwaltung der jeweiligen Zielgruppe verwendet

**Student Verwaltung**

Name  sNummer

Course: ☐ Grundlagen der Informatik I ☐ Programmierung I ☐ Betriebssysteme I ☐ Allgemeine Betriebswirtschaftslehre

Nr	Name	sNummer	Funktion
1	Markus Schnepfer	s79548	<input type="button" value="Edit"/> <input type="button" value="Remove"/> <input type="button" value="Course"/> <input type="button" value="Test"/>
2	Lutz Groissaa	s79547	<input type="button" value="Edit"/> <input type="button" value="Remove"/> <input type="button" value="Course"/> <input type="button" value="Test"/>
3	Tim Rommel	s79549	<input type="button" value="Edit"/> <input type="button" value="Remove"/> <input type="button" value="Course"/> <input type="button" value="Test"/>
4	Alena Geier	s79550	<input type="button" value="Edit"/> <input type="button" value="Remove"/> <input type="button" value="Course"/> <input type="button" value="Test"/>

**Abbildung 46:** Studenten Verwaltung

Auf der Seite Studentenverwaltung können Benutzer grundlegende CRUD-Funktionen über die Schaltflächen ausführen: New, Edit, Remove. Führen die Suchfunktion durch, indem Daten in die leeren Zellen eingegeben und die Schaltfläche Suchen gedrückt wird. Die APIs, die den Funktionen entsprechen, werden Server aufgerufen und ausgeführt, und die Ergebnisse werden an den Client gesendet. Darüber hinaus gibt es zwei Schaltflächen: Course und Test dienen dazu, Kurse oder Tests für Studenten zu registrieren. Beachten bei der Anmeldung eines Studenten für einen Kurs mit Prüfungen, dass der Student automatisch für alle Prüfungen in diesem Kurs angemeldet wird. Über die Schaltfläche Test können Anmeldungen abbestellt oder bearbeitet werden.

**Abbildung 47 Edit**

Das obige Bild ist eine Bearbeitungsfunktion (Edit), mit der Benutzer Informationen ändern oder Bilder von Studenten betrachten können. Der Client ruft automatisch die entsprechenden APIs auf, so dass der Server Folgendes ausführen kann.

Nr	Name	Course	Description	Professor	Funktionen
1	90m(MP)	Betriebssysteme I	Z304 HTW Dresden	Fynn Dehnhardt	Edit Remove Test

**Abbildung 48: Testverwaltung**

**Add New Test**

Name:

Description:

Select Course:

Nr	Name	Course	Description	Professor	Funktionen
1	90m(MP)	Betriebssysteme I	Z304 HTW Dresden	Fynn Dehnhardt	<input type="button" value="Edit"/> <input type="button" value="Remove"/> <input type="button" value="Test"/>
2	90m(SP)	Programmierung I	Z804 HTW Dresden	Karl Heitjan	<input type="button" value="Edit"/> <input type="button" value="Remove"/> <input type="button" value="Test"/>

**Abbildung 49** new Test

Der Benutzer kann Testverwaltung wählen. Auch diese Seite hat die gleichen Funktionen wie die Studentenverwaltung. Beachten bei der neuen Funktion "Neu", dass Benutzer sich nur für Tests pro Klasse anmelden können. Das bedeutet, dass alle Studenten einer Klasse registriert werden, wenn ein neuer Test für diese Klasse hinzugefügt wird.

Darüber hinaus gibt es hier für jeden Test eine Schaltfläche "Test". Diese Schaltfläche ist dafür verantwortlich, dass die Benutzer zu dieser Testverwaltungsseite navigiert werden.

**Test Detail**

Title: 90m(MP) Course Name: Betriebssysteme I

Description: Z304 HTW Dresden Professor: Fynn Dehnhardt

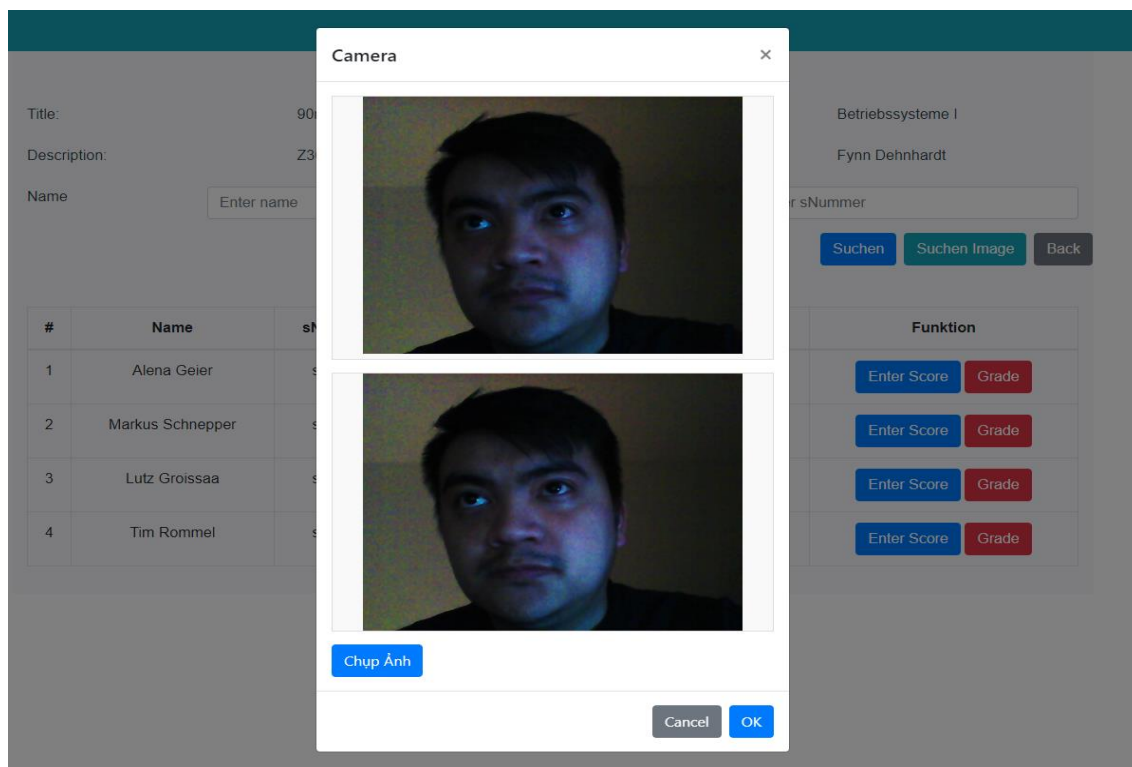
Name:  sNummer:

#	Name	sNumber	Score	Participated	Funktion
1	Alena Geier	s79550	<input type="text"/>		<input type="button" value="Enter Score"/> <input type="button" value="Grade"/>
2	Markus Schnepfer	s79548	<input type="text"/>		<input type="button" value="Enter Score"/> <input type="button" value="Grade"/>
3	Lutz Groissaa	s79547	6 <input type="text"/>	T	<input type="button" value="Enter Score"/> <input type="button" value="Grade"/>
4	Tim Rommel	s79549	<input type="text"/>		<input type="button" value="Enter Score"/> <input type="button" value="Grade"/>

**Abbildung 50:** Test Detail

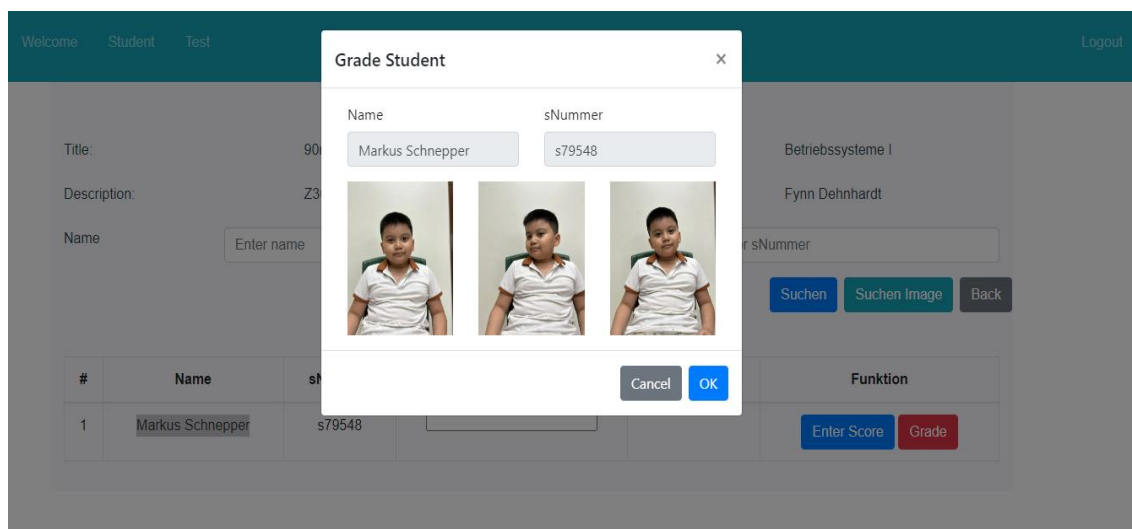
Auf dieser Seite werden grundlegende Informationen über den Test und alle Studenten, die an dem Test teilnehmen, angezeigt. Die Benutzer können die Ergebnisse für die Studenten direkt eingeben, indem sie die Ergebnisse in das leere Feld eingeben und dann die

Schaltfläche "Score" eingeben auswählen. Darüber hinaus können die Benutzer die Studentensuchfunktion nutzen, indem sie die Werte eingeben, nach denen sie suchen möchten. Wählen "Grade" und überprüfen die Anwesenheit. Außerdem können die Benutzer die Bildsuchfunktion verwenden, um nach Studenten wie im Beispiel auf dem Bild zu suchen. Wenn das System den Studenten erkennt, wird in der Liste nur dieser Student angezeigt. Um die Anwesenheit zu überprüfen, können den Button "Note" wählen.



**Abbildung 51:** Gesichtserkennung

Wenn die Authentifizierung erfolgreich ist, werden die Studenteninformationen angezeigt. Wenn die Authentifizierung fehlschlägt, erscheint die folgende Meldung (Abbildung 45+46)



**Abbildung 52** erfolgreich verifiziert



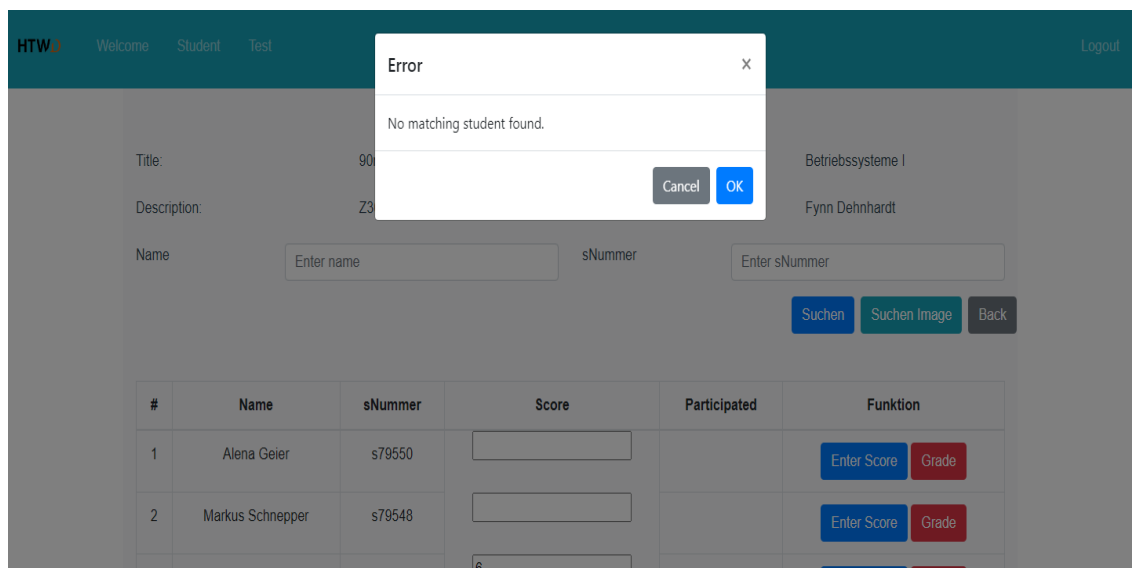


Abbildung 53 Verifizierung fehlgeschlagen

## 7.2 Bewertung

Der Programmtest läuft folgendermaßen ab:

Er basiert auf zwei Faktoren: Genauigkeit und Ausführungszeit. Andere APIs (die nicht mit der Bildverarbeitung zusammenhängen) werden in der Regel sehr schnell ausgeführt (unter 1 Sekunde), so dass das ursprüngliche Ziel von unter 5 Sekunden erreicht wird. APIs, die sich auf die Bildverarbeitung beziehen, benötigen mehr Zeit. Die Experimente haben gezeigt, dass die Zeit, die das Programm zum Extrahieren des Embedding Vektors von Bildern benötigt, am längsten ist. Aus diesem Grund werden zwei APIs getestet: Eine API zur bildbasierten Studentensuche und eine API zur Erstellung neuer Studenten (mit drei Fotos darin).

Search API						Student Create API	
sNummer	sNummer	predicts	time(ms)	result	note		time
79547	79547	3529	T			test1	4111
79511	79511	3557	T			test2	4500
79548	79548	3530	T			test3	4241
Unknow	Unknow	3517	T			test4	4170
79550	79550	3473	T			test5	4310
79548	79548	3566	T		many face	test6	4539
79552	79552	3540	T		student but not in course: not show	test7	4465
79551	Unknow	3470	F		picture train younger than now	test8	4210
79548	Unknow	3371	F		quality not good	test9	4241
79548	79548	3357	T		2 Student, but only 1 "more clearly"	test10	4451
AVERAGE		3491					

Abbildung 54: Testbewertung

Ich habe den Test für zehn Fälle von Bildsuche durchgeführt. Jeder Fall weist unterschiedliche Bildqualitätsbedingungen auf (viele Gesichter im Bild, Bilder mit schlechter Qualität usw.). Die Ergebnisse zeigen, dass das Programm bei Fotos von guter Qualität

sehr gut funktioniert, aber bei Fotos von schlechter Qualität oder unter schwierigen Bedingungen versagt das Programm immer noch. Die Erfolgsquote ist gerade genug 8/10. Mit einer durchschnittlichen Ausführungszeit von 3,491 Sekunden wurde das ursprüngliche Ziel (weniger als 5 Sekunden) erreicht.

Bei API zur Erstellung neuer Studenten benötigt das System mehr Zeit für die Ausführung, da es drei Bilder extrahieren muss, anstatt eines Fotos bei der API zur bildbasierten Studentensuche. Die Ergebnisse zeigen, dass die Ausführungszeit weniger als 4,6 Sekunden beträgt und das Ziel erreicht wurde.

Grundsätzlich sind die Anforderungen an Zeit und Genauigkeit erfüllt. Dies beweist die Wirksamkeit des Programms und löst die beiden von Anfang an aufgeworfenen Problemen [1] und [2] vollständig und die Zielfrage erfolgreich beantworten.

Allerdings ist das Programm sowohl im Backend als auch im Frontend ausfolgenden Gründen noch unvollständig:

Der Code wurde nicht optimiert, ist noch recht lang und kompliziert.

Ausnahmen können nicht behandelt werden, weil der Testprozess keine effektive Methode hat, die Ausführungszeit kurz ist und es keinen ausreichend großen Datensatz zum Testen gibt.

Das Frontend ist immer noch nicht benutzerfreundlich. Wenn eine Ausnahme auftritt, ist die Art und Weise, wie sie behandelt wird, immer noch nicht gut (freundliche und höfliche Benachrichtigung des Benutzers).

Der Mechanismus zur Überprüfung der Eingabewerte ist noch nicht vollständig ausgereift.

## 8 Zusammenfassung und Ausblick

Das Programm hat erfolgreich die Anwesenheit von Studenten mit Hilfe von Gesichtserkennung mit einer erwarteten Erfolgsrate (über 80%) und einer Ausführungszeit von weniger als 5 Sekunden angewandt, sowie die Sicherstellung des Arbeitsfortschritts in zwei Monaten. Darüber hinaus hat das Programm fortschrittliche Technologien wie Java Spring Boot, VueJS angewandt, um ein Studentenverwaltungssystem, eine Notenverwaltung und grundlegende Tests auf der Webplattform zu erstellen, und das System erfolgreich aufgebaut. So wurde auch die Zielfrage erfolgreich beantwortet "Wie kann ein effizientes webbasiertes System entwickelt werden, das die Prüfungsaufsicht bei der Anwesenheitskontrolle von Studierenden durch Gesichtserkennung mit PyTorch unterstützt und gleichzeitig effektives Studentenmanagement und Prüfungsverwaltung ermöglicht?"

Allerdings hat das Programm noch viele Schwächen:

Die Erste ist die mangelnde Optimierung des Frontends. Das Frontend-Programm hat viele Punkte, die den Benutzer stören:

- Es gibt keinen Warte-Mechanismus, um auf die erfolgreiche Ausführung der Funktion zu warten. Die Funktion zum Hinzufügen neuer Studenten erfordert die Übermittlung von Studentinformationen und drei Fotos an den Server. Die geschätzte Zeit bis zur erfolgreichen Ausführung der Funktion beträgt etwa 4,5 Sekunden. Die Benutzer wissen aber immer noch nicht, wann diese Funktion abgeschlossen sein wird. Dies ist derzeit nur bekannt, wenn man die Entwicklerkonsole des Browsers benutzt.
- Einige unbearbeitete Benachrichtigungen verwenden immer noch die Standardmeldung (Alert ()) des Browsers, um den Benutzer zu benachrichtigen.

Die zweite Schwäche ist, dass obwohl das Gesichtserkennungsmodell erfolgreich mit einer hohen Erkennungsrate angewendet wurde, gibt es immer noch viele Fehler. Da die Eingabedaten klein sind, ist in manchen Fällen eine erfolgreiche Erkennung nicht möglich, wenn die Bildqualität nicht gut ist. Man kann dies verbessern, indem man das Eingangsbild erweitert oder Bildverarbeitungsalgorithmen einsetzt, um die Effizienz zu erhöhen.

Die dritte Schwäche ist, dass bei der Anwendung der Gesichtserkennung noch viele Einschränkungen bestehen.

- Das neue Programm kann die Gesichter der Studenten nur einzeln erfassen. Das bedeutet, dass das Programm bei einem Foto mit vielen Studenten nur das deutlichste Gesicht verarbeiten kann. Dies kann in Zukunft ohne allzu große Schwierigkeiten erweitert werden.

- Die Identifizierung durch das Fotografieren kann zu Unannehmlichkeiten führen. Man kann die Identifizierung über die Live-Kamera verbessern und erweitern. Im Wesentlichen handelt es sich dabei um eine Aufgabe, bei der die Live-Kamera eine bestimmte Anzahl von Fotos aufnimmt und diese zur Verarbeitung an den Server sendet. Es ist jedoch zu beachten, dass diese Option eine Verarbeitung erfordert, um sicherzustellen, dass die empfangenen Bilder den Anforderungen entsprechen.

- Der Anwendungsbereich der Gesichtserkennung ist begrenzt. Neben der Anwesenheit von Studenten mit Hilfe der Gesichtserkennung kann man die Gesichtserkennung auch für viele andere Anwendungen einsetzen, wie z. B. die Suche nach Studenten anhand eines Bildes oder die Anmeldung anhand eines Bildes. Dies kann durch die Anpassung eines Filters in Java Spring Security erfolgen, der die Gesichtserkennung anwendet.

Der vierte Schwachpunkt ist, dass es keinen Mechanismus zur Behandlung von Ausnahmen und zur Überprüfung der Eingabedaten gibt. Dieser Prozess erfordert eine lange und sorgfältige Testphase. Dies kann mit der weiteren Entwicklung des Projekts verbessert werden.

Grundsätzlich ist das Programm jedoch erfolgreich aufgebaut und kann unter bestimmten Bedingungen (Sicherstellung der Bildqualität, bestimmte Anzahl von Studenten) praktisch angewendet werden. Noch wichtiger ist, dass das Programm mit einer sorgfältig aufgebauten Struktur leicht erweitert werden kann, wenn es sich nach den oben genannten Ideen weiterentwickelt. Das Programm isoliert auch Funktionen, so dass sie leicht ersetzt werden können, wenn man zum Beispiel eine Verbindung zu einer anderen Datenbank oder zu einem Cloud-Speicherdienst statt zu einem Ordner auf dem Server benötigt.

## 9 Literaturverzeichnis

- [1] BBCNews, “Artificial intelligence: Google’s AlphaGo beats Go master Lee Sedol,” 2016. <https://www.bbc.com/news/technology-35785875>.
- [2] Chi-Feng Wang “What Does A Face Detection Neural Network Look Like?”, <https://towardsdatascience.com/face-detection-neural-network-structure-257b8f6f85d1>
- [3] D.Sandberg <https://github.com/davidsandberg/facenet>
- [4] G. Hinton, “Deep Belief Nets,” IEEE Trans. Neural Networks, 2006
- [5] <https://docs.spring.io/spring-framework/reference/index.html>
- [6] <https://vuejs.org/guide/introduction.html>
- [7] J. Brownlee „A Tour of Machine Learning Algorithms “, <https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/>
- [8] JASON BROWNLEE “Deep Learning for Computer Vision Image Classification, Object Detection and Face Recognition in Python”, 2019
- [9] K. Zhang, Z. Zhang, Z. Li, Y. Qiao “Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks”, 2016
- [10] L. Shi, Z. Li, and D. Song, “A Flower Auto-Recognition System Based on Deep Learning,”, 2019
- [11] Panos Zafeiropoulos, „Spring Boot Security Configuration, practically explained “, <https://medium.com/@zzpzaf.se>
- [12] Pham Dinh Khanh, <https://phamdinhkhanh.github.io/2020/03/12/faceNetAlgorithm.html#33-siam-network>
- [13] R. Pramoditha, <https://medium.com/@rukshanpramoditha>
- [14] T. Esler <https://github.com/timesler/facenet-pytorch>
- [15] T. Tiay, P. Benyaphaichit, and P. Riyamongkol, “Flower recognition system-based on image processing,”, 2014
- [16] Omkar.M.Parkhi, <https://www.robots.ox.ac.uk/~vgg/publications/2015/Parkhi15/poster.pdf>

## 10 Selbstständigkeitserklärung

Ich versichere hiermit, dass ich die Bachelorarbeit mit dem Titel  
„Java-Web-Gesichtserkennungssystem für Studenten mit Pytorch“  
selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel  
benutzt habe.

Dresden, 25. Januar 2024

Ngoc Huy Le