Bachelorarbeit

**Untersuchung verschiedener Neuronaler Netze zur Emotionserkennung durch Merkmale im Gesicht**

Examination of Several Neural Networks for Facial Emotion Recognition

Simon Al Nomer

Angewandte Informatik

Matrikel: 64082

21. Februar 2023

**Erklärung**

Hiermit versichere ich, die vorliegende Arbeit ohne Hilfe Dritter und nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus den Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Freiberg, den \_\_.\_\_.20\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Vorname Name)

**Inhalt**

[**1.** **Einleitung** 1](#_Toc124018819)

[**2.** **Grundlagen** 2](#_Toc124018820)

[**2.1. Menschliche Emotionen** 2](#_Toc124018821)

[**2.2. Klassifikation von Emotionen** 3](#_Toc124018822)

[**2.2. Gesichtserkennung** 4](#_Toc124018823)

[**2.1.1. Viola & Jones Algorithmus** 4](#_Toc124018824)

[**2.2.1. Erkennung von Gesichtsmerkmalen** 9](#_Toc124018825)

[**2.3. Künstliche Neuronale Netze** 10](#_Toc124018826)

[**2.3.1. Biologische und künstliche Neuronen** 10](#_Toc124018827)

[**2.3.2. Aktivierungsfunktion** 12](#_Toc124018828)

[**2.3.3. Struktur eines neuronalen Netzes** 13](#_Toc124018829)

[**2.4. Faltungsneuronaler Netze** 14](#_Toc124018830)

[**3.** **Datensatz** 17](#_Toc124018831)

[**4.** **Implementierung** 18](#_Toc124018832)

[**4.1. Vergleich zwischen der Verwendung von Viola & Jones und Gesichtsmerkmalen** 18](#_Toc124018833)

[**4.2. Preprocessing-Datensatz** 18](#_Toc124018834)

[**4.2.1. Ohne Preprocessing** 18](#_Toc124018835)

[**4.2.2. Preprocessing mit Gesichtserkennung** 19](#_Toc124018836)

[**4.2.3. Preprocessing mit Erkennung von Augen und Mund** 20](#_Toc124018837)

[**4.3. CNN-Model** 22](#_Toc124018838)

[**5.** **Ergebnisse** 23](#_Toc124018839)

[**5.1. Ergebnis des Vergleichs zwischen der Verwendung von Viola & Jones und Gesichtsmerkmalen** 24](#_Toc124018840)

[**5.2. Ergebnisse des Trainings ohne Preprocessing** 25](#_Toc124018841)

[**5.3. Ergebnisse des Trainings mit Gesichtserkennung** 25](#_Toc124018842)

[**5.4. Ergebnisse des Trainings mit Augen- Munderkennung** 25](#_Toc124018843)

[**6.** **Code-Erklärung** 25](#_Toc124018844)

[7. Diskussion 26](#_Toc124018845)

[8. Zusammenfassung 27](#_Toc124018846)

[**Abbildungsverzeichnis** 28](#_Toc124018847)

[**Tabbellenverzeichnis** 29](#_Toc124018848)

[References 31](#_Toc124018849)

# **Einleitung**

In den letzten Jahren hat sich die Erkennung von Emotionen zu einem wichtigen Bereich der künstlichen Intelligenz entwickelt, und es wurden zahlreiche Forschungsarbeiten auf diesem Gebiet durchgeführt. Es gibt bereits viele Systeme und Anwendungen Zur Emotionserkennung, die in der Lage sind, Emotionen zu erkennen und zu interpretieren, jedoch gibt es noch viele Herausforderungen und offene Fragen. Daher ist die Emotionserkennung nach wie vor ein aktuelles und wichtiges Thema im Bereich der künstlichen Intelligenz, und es wird erwartet, dass sie in den kommenden Jahren weiter an Bedeutung gewinnt.

Ziel dieser Arbeit ist es, mehrere Systeme mit Hilfe der neuronalen Netze zur Emotionserkennung mit unterschiedlichen Strategien zu entwickeln und diese miteinander zu vergleichen. Der Schwerpunkt liegt darauf, die Systeme auf ihre Funktionalität zu testen, um sie später möglicherweise in der Praxis einsetzen zu können.

Bei den in dieser Arbeit verwendeten neuronalen Netzen handelt es sich um Faltungsneuronale Netze (Convolutional Neural Networks „CNNs“). Faltungsneuronale Netze sind für die Erkennung von Emotionen von besonderem Interesse, auf Grund ihrer besonderen Fähigkeit, Muster in Bilddaten zu erkennen und zu interpretieren. Darüber hinaus eignet sich CNN besonders gut für die Erkennung von Emotionen in Gesichtern, da Emotionen oft durch komplexe Muster von Gesichtsausdrücken dargestellt werden, die in Bildern erfasst werden können. CNNs verwenden Faltungsschichten, um komplexe Muster in Bilddaten zu erkennen und zu analysieren. Diese Schichten können lernen, bestimmte Merkmale im Bild, wie z. B. bestimmte Gesichtsausdrücke oder Körpersprache, mit bestimmten Emotionen zu assoziieren.

In dieser Arbeit wird ein Datensatz für den Trainingsprozess von Faltungsneuronalen Netzen ausgewählt. Er enthält Bilder von Personen, die mit sieben Emotionen aufgelistet sind. Für den Datensatz werden drei Strategien verwendet.

* Bei der ersten Strategie werden die Bilder des Datensatzes nicht verändert, was am Ende zu guten Ergebnissen führte.
* In der zweiten Strategie liegt der Fokus auf dem Gesicht in den Bildern, so dass die Bilder verarbeitet werden, was zu etwas besseren Ergebnissen mit sich bringt.
* Bei der dritten Strategie liegt der Schwerpunkt auf bestimmten Gesichtsmerkmalen, so dass die Bilder gemäß der zweiten Strategie weiterverarbeitet werden, aber dabei gab es ein Problem, das zum Verlust einiger Bilder des Datensatzes führte. Daher blieb die zweite Strategie die beste für diesen Datensatz.

Die Fähigkeit, Emotionen zu erkennen und zu interpretieren, könnte für viele Anwendungsbereiche äußerst wichtig sein. Einige dieser Bereiche sind im Folgenden genannt.

**Gesichtserkennung**: Die Fähigkeit, Emotionen in den Gesichtern von Menschen zu erkennen, spielt bei der Gesichtserkennung eine große Rolle, weil sie die Zuverlässigkeit und Genauigkeit von Gesichtserkennungssystemen verbessern kann.

**Bild- und Videoanalyse**: Im Bereich der Bild- und Videoanalyse hilft die Emotionserkennung dabei, die Stimmung und die Emotionen von Menschen in Bildern oder Videos besser zu verstehen und zu interpretieren. Dies kann beispielsweise bei der Analyse der „social media posts“ oder der Überwachung der Videomaterialien von praktischem Nutzen sein.

**Robotik**: Die Erkennung von Emotionen kann dazu beitragen, die Interaktion zwischen Roboter und Mensch in der Robotik zu verbessern, indem Roboter in die Lage versetzt werden, die Emotionen von Menschen zu erkennen und angemessen darauf zu reagieren.

Alles in allem ist die Erkennung von Emotionen, wie schon erwähnt, ein wichtiges Thema der künstlichen Intelligenz. Sie kann dabei helfen, die Interaktion von Maschinen mit Menschen zu verbessern und sie menschenähnlicher und verständlicher zu machen. Darüber hinaus lässt sich mit Hilfe der Emotionserkennung die Genauigkeit und Zuverlässigkeit von Systemen in vielen Anwendungsbereichen verbessern, indem zusätzliche Informationen für die Interpretation von Mustern und Daten bereitgestellt werden.

# **Grundlagen**

In diesem Kapitel werden zunächst die für diese Arbeit relevanten Grundlagen vorgestellt, um in das Thema der Emotionserkennung einzusteigen. In einem ersten Schritt werden die menschlichen Emotionen in Abschnitt (**Menschliche Emotionen**) und (**Klassifikation von Emotionen**) kurz beschrieben. In Abschnitt (**Gesichtserkennung**) wird dann auf die Gesichtserkennung und deren Bedeutung für diese Arbeit eingegangen. Und in Abschnitt (**Viola & Jones Algorithmus**) und (**Erkennung von Gesichtsmerkmalen**) werden zwei Methoden zur Gesichtserkennung geklärt. Als nächstes werden in Abschnitt (**Künstliche Neuronale Netze**) die künstliche neuronale Netze im Allgemeinen diskutiert. Danach werden in Abschnitt (**Biologische und künstliche Neuronen**) biologische und künstliche Neuronen eingeführt, und in Abschnitt (**Aktivierungsfunktion**) werden einige Aktivierungsfunktionen vorgestellt, außerdem wird in Abschnitt (**Struktur eines neuronalen Netzes**) die Struktur künstlicher neuronaler Netze dargestellt. Schließlich werden in Abschnitt (**Faltungsneuronaler Netze**) Faltungsneuronale Netze klar definiert.

## **2.1. Menschliche Emotionen**

Emotion ist eine komplexe Wahrnehmung von Körperempfindungen und Verhaltensweisen, die die individuelle Wirkung einer Situation oder eines Erlebnisses widerspiegelt. Es wird als Emotion bezeichnet, wenn eine Person das Gefühl einer anderen Person durch sowohl ihren Inneren Einfluss als auch ihre Gesichtsausdrücke erkennt.

Das Wort Emotion leitet sich von dem lateinischen Wort „*emovere*“ ab und bedeutet "*heraus bewegen, in Bewegung setzen, in einen erregten Zustand versetzen*". Diese Umschreibung spiegelt die zentrale Eigenschaft von Emotionen wider. Emotionen berühren uns, erregen uns und bewegen uns in eine bestimmte Richtung.

Bei Emotionen kommen Merkmale vor, die berücksichtigt werden müssen, um die Emotionen eindeutig darzustellen:

1. **Affektiv:** Da wir als Menschen Emotionen, wie Freude, Angst, Wut etc., haben. Ist uns logischerweise dieser affektiven Empfindungen bewusst, weil wir sie wahrnehmen [11].
2. **Objektgerichtetheit (Intentionalität):** Die Emotionen einer Person werden durch einen bestimmten Gegenstand oder einen speziellen Prozess beeinflusst. Dieser Auslöser kann physisch oder psychisch sein.
3. **Unwillkürlichkeit:** Durch bestimmte Situationen treten Emotionen zwangsläufig bei Personen auf. Obwohl wir die Entscheidung treffen, bestimmte Erlebnisse zu erleben, um bestimmte Gefühlzustände zu haben, liegt die Auslösung der Emotion selbst nicht in unserer Kontrolle.
4. **Dauer:** Wie lange der Ausdruck einer Emotion anhält, hängt von der Auslösung dieser Emotion ab. Sie kann logischerweise kurz- oder langfristig sein.

Aus diesen Merkmalen könnten A. Eder und K. Rothermund die Möglichkeit haben, eine Definition des Begriffs „Emotion“ in ihrem Buch zu bieten [34].

„*Emotionen sind objektgerichtete, unwillkürlich ausgelöste affektive Reaktionen, die mit zeitlich befristeten Veränderungen des Erlebens und Verhaltens einhergehen*“

## **2.2. Klassifikation von Emotionen**

Die Klassifizierung von Emotionen ist ein Verfahren, das in der Psychologie eingesetzt wird, um menschliche Emotionen besser einordnen und bewerten zu können. Dies geschieht durch den Einsatz verschiedener Theorien und Methoden, wie z.B. biologischer Evolutionskonzepte. Darüber hinaus kann die Emotionsanalyse auch mit Hilfe von Techniken des Maschinellen Lernens (machine learning) durchgeführt werden [23, 24].

1971 führte der Psychologe P. Ekman eine eindimensionale Klassifizierung grundlegender Emotionen bei den Mitgliedern eines Stammes aus Neuguinea durch. Er entdeckte, dass ihre Gesichtsausdrücke mit denen anderer Menschen aus der ganzen Welt übereinstimmen. Auf der Grundlage seiner Ergebnisse konnte er das Konzept der Mimik bestätigen, das besagt, dass die Menschen bereits mit den gleichen Reaktionen auf bestimmte emotionale Ereignisse geboren werden. Die von ihm identifizierten Grundemotionen waren Freude, Überraschung, Wut, Angst, Ekel und Trauer [36].

In den folgenden Jahren fügte der Psychologe C. Izard weitere Emotionen wie Schuld, Scham, Widerwillen und Interesse hinzu [7].

## **2.2. Gesichtserkennung**

Im Rahmen dieser Arbeit ist die Erkennung des Gesichtes der betreffenden Person grundsätzlich zur Emotionserkennung dieser Person. Eine Gesichtserkennung muss feststellen, ob ein Bild von beliebiger Größe ein menschliches Gesicht enthält und wenn dies der Fall sein sollte, muss diese Gesichtserkennung herausfinden, wo sich das Gesicht im Bild befindet. Ein geeigneter Ansatz für dieses Problem ist die binäre Klassifikation, um Fehlklassifikation möglichst zu minimieren. Es gibt keine objektive Verteilung, die die Aussage treffen kann, ob das Bild ein menschliches Gesicht enthalten könnte. Daher muss ein Algorithmus sowohl die Falsch-Negativ- als auch Falsch-Positiv- Rate minimieren, um eine angemessene Leistung zu erzielen. Um diese Anforderung zu erfüllen, ist eine genaue numerische Beschreibung der Merkmale erforderlich. Diese Merkmale unterscheiden die menschlichen Gesichter von andren Objekten.

Das Gesicht wird durch seine Merkmale analysiert, um schließlich das Zweck dieser Arbeit zu erfüllen, also den Zustand der Emotionen der betrachteten Person festzustellen. [30]

Zu diesem Zweck wählte ich den Algorithmus von „Paul Viola & Michael J. Jones“, der im Jahr 2011 geschrieben wurde [21]. Mit diesem Algorithmus können effizient bestimmte Objekte in Bild- und Videodaten erkennt werden. Ausßerdem hat dieser Algorithmus drei grundlegende Konzepte, die es ermöglichen, einen zuverlässigen Objektdetektor (in dieser Arbeit Gesischtsdetektor) zu entwickeln. Dieser Gesichtsdetektor kann in Echtzeit laufen. Die Konzepte sind Bildintegral, Klassifikator lernen mit „AdaBoost“ und „attentional cascade“ Struktur [9], die zunächst erklärt werden.

### **2.1.1. Viola & Jones Algorithmus**

**Haar-like features**

Die „Haar-like“ features werden verwendet, um Aussagen zu erstellen, ob ein Sub-Window (Teil-Fenster) entweder „potenziell ein Gesicht“ oder „mit Sicherheit kein Gesicht“ enthält. Diese Merkmale sind in vier Kategorien unterteilt siehe Abbildung 1.

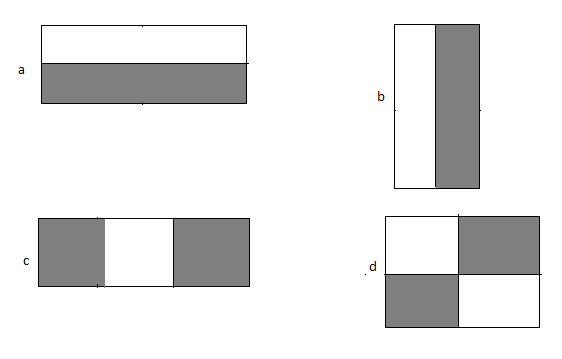


Abbildung 1: Beispiel für haar-features

Aus Abbildung 1 folgt, die Merkmale (a) und (b) sind als zwei rechteckige Merkmale klassifiziert. Das Merkmal (a) würde eher verwendet werden, um den Farbunterschiede im Augen- und Augenbrauenbereich eines menschlichen Gesichtes sowie auch die Stirn zu erkennen. Merkmal (b) kann verwendet werden, um eine Vertikale Kante zwischen dem Gesicht einer Person und dem Hintergrund des Gesichtes zu erkennen. Merkmal (c) ist als Drei-Rechteck klassifiziert und kann zur Erkennung des Farbunterschieds zwischen den Augen und dem Nasenrücken einer Person verwendet werden. Zuletzt ist Merkmal (d) ein Vier-Rechteck und wird zum detektieren diagonaler Unterschiede in verschiedenen Teilen des Gesichtes verwendet.

Unterschiedliche Kombinationen, Orientierungen und Größen von rechteckigen Merkmalen geben dem Algorithmus einen Hinweis darauf, ob ein sub-window ein Gesicht enthält oder nicht. Bei all diesen Merkmalen wird die Summe der Pixel in weißen Rechtecken von der Summe der Pixel in grauen Rechtecken subtrahiert, um die Unterschiede in der Dunkelheit oder Helligkeit in einem bestimmten sub-window zu berechnen.

Aber bei einer großen Anzahl von Pixeln würde die Anzahl der Additionsoperationen sehr groß werden, und der Algorithmus würde sehr langsam laufen. Um dies zu vermeiden, wurde das Konzept des Integralbildes eingeführt. [32]

**Integral Image**

Um die Merkmale schnell zu berechnen, müssen alle Pixel in der rechteckigen Region summiert werden. Dieser Algorithmus verwendet das Konzept des Integralbildes. Der integrale Bild-Wert eines beliebigen Punktes im Bild ist gleich der Summe aller Pixel in der oberen linken Ecke des Punktes. Abbildung 2, es kann durch die Integration des Bildes durch den Graphen die Pixelsumme aller Regionen im Bild ermittelt werden, durch einmaliges Durchlaufen des Bildes. Das verbessert die Berechnungseffizienz des Bildeigenwertes erheblich.

sei der integrale Bild-Wert der Punkte und sei der Grauwert eines beliebigen Pixels im integralen Bild:

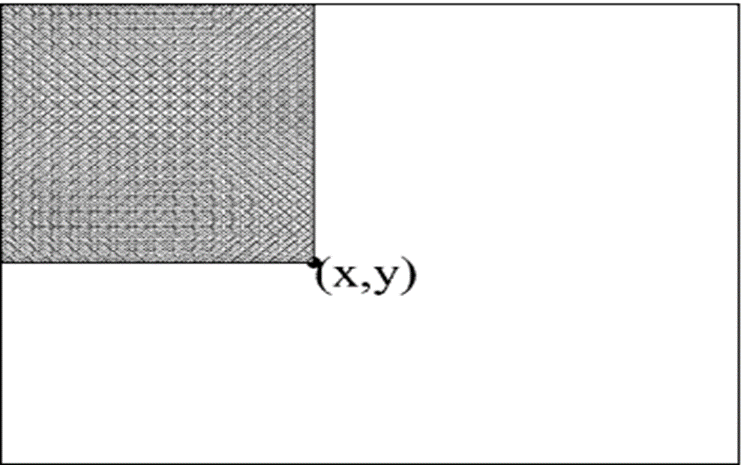


Abbildung 2: punkt (x,y) ist die Summe aller Pixel oberhalb und links davon [30]

Die Folgende Rekursionsformel erhält man, indem man die Reihenfolge von links nach rechts und von oben nach unten durchläuft: [8]

2

Auf die die gleiche Weise lässt sich die Summe der Pixel eines beliebigen rechteckigen Bereich im Bild ermitteln. Abbildung 3 seien die Koordinaten der linken oberen Ecke des zu lösenden Rechteckes und die Breite und Höhe des Rechtecks w,h bezeichnet als ein Rechteck . Die Formel für das Integralbild lautet: [8]

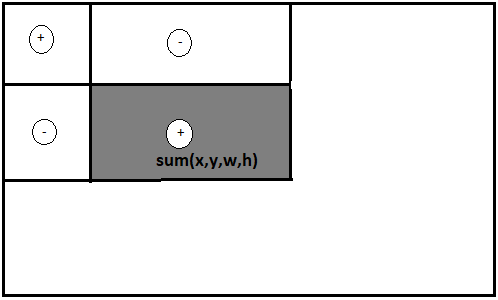


Abbildung 3: Integralbild eines bestimmten recheckigen Bereichs [8]

In Abbildung 4 ist ein Beispiel für ein Bild-Integral, dabei stellt diese Matrix ein 4x4-Pixel-Bild dar. Die Werte skalierte ich nicht auf einen Bereich zwischen 0 und 1 herunter, sondern ließ Sie auf ihrer ursprünglichen Skala von 0 bis 255. Diese Werte repräsentieren die Helligkeit eines Bilds, d.h. bei 0 ist das Bild ganz schwarz und bei 255 ist das Bild ganz weiß. In diesem Beispiel ist das Bild aufgrund des niedrigen Werten schwarz.

Wie vorher besprochen, Beim Bild-Integral wird ein beliebiger Punkt ausgesucht und die Summe aller Pixel in der oberen linken Ecke des Punktes berechnet, sehe die Pfeile in Abbildung 4 in Bild-Integral.

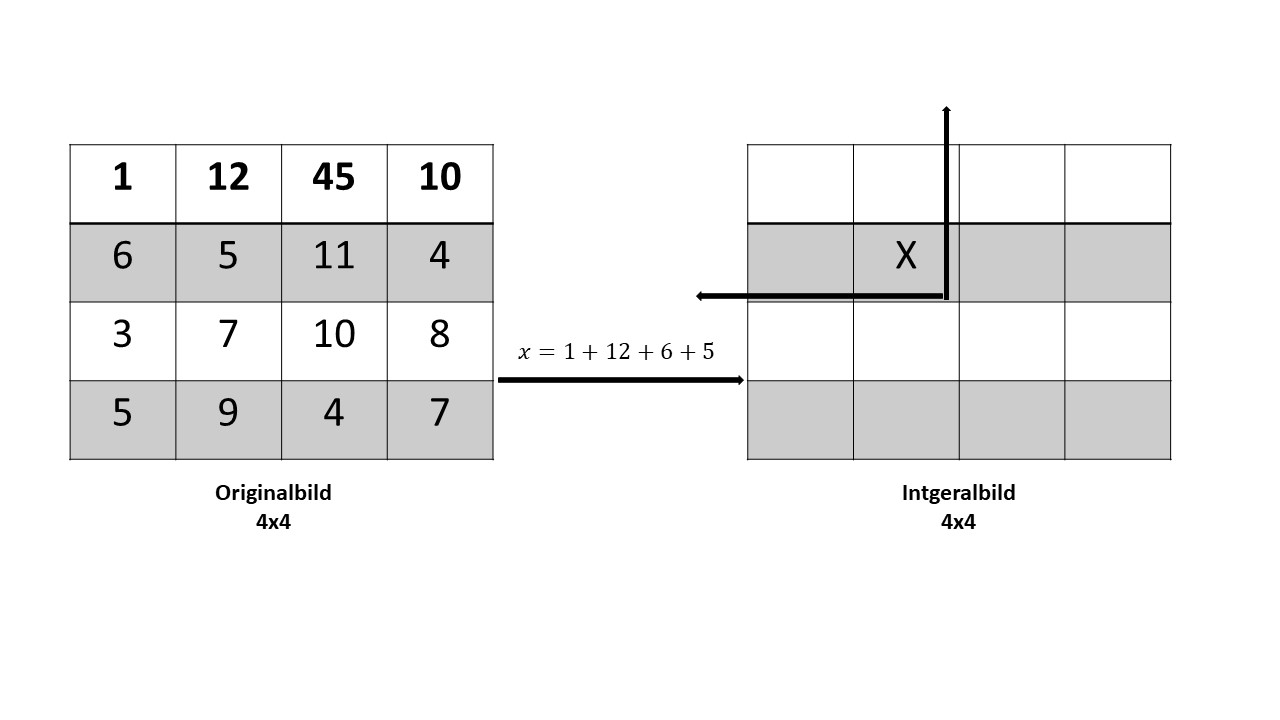


Abbildung 4: Beispiel für Bild- integral

Dieses Verfahren wird so oft iterieret, bis alle Punkte für das Integral-Bild berechnet wurden. Für dieses Beispiel wird das Integral-Bild wie in Abbildung 5 abgebildet.

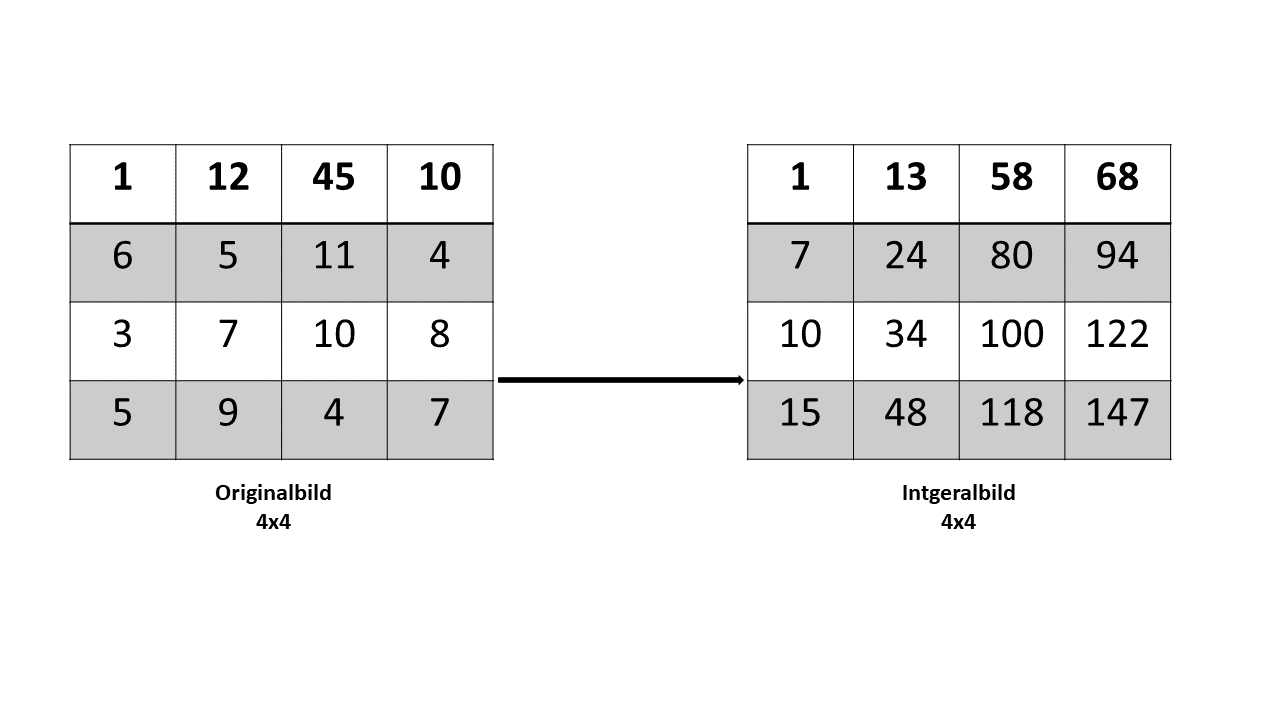


Abbildung 5: Integral-Bild Ergebniss

**AdaBoost**

Der AdaBoost-Algorithmus kann Auswahl der Merkmale und das Training des Klassifikators gleichzeitig durchführen. Es handelt sich um einen iterativen Algorithmus. Dessen Konzept besteht darin, verschiedene Klassifikatoren(schwache Klassifikatoren) für denselben Trainingssatz zu trainieren und dann diese schwachen Klassifikatoren zu einer stärkeren endgültigen Klassifikation(starker Klassifikator) zusammenzufügen. Der Algorithmus selbst wird durch Änderung der Datenverteilung realisiert. Es bestimmt die Gewichtung jeder Probe je nachdem, ob die Klassifizierung jeder Probe in jedem Trainingssatz korrekt ist oder nicht und die Genauigkeit der letzten globalen Klassifizierung. Die neuen Datensätze mit geänderten Gewichten werden zum Training an den unteren Klassifikator gesendet. Schließlich werden die aus jedem Training erhaltenen Klassifikatoren zum endgültigen Entscheidungsklassifikator zusammengelegt. Mit dem AdaBoost-Klassifikator können einige unnötige Merkmale der Trainingsdaten eliminiert und auf die wichtigsten Trainingsdaten gelegt werden.

Wenn der Eigenwert des Eingangsbildes größer als der Grenzwert ist, wird das Gesicht beurteilt, so dass der Prozess des Trainings des optimalen schwachen Klassifikators darin besteht, den geeigneten Grenzwert des Klassifikators zu finden.

In gewöhnlichen Bildern nehmen die Regionen, die menschliche Gesichter enthalten, nur einen kleinen Teil des gesamten Bildes ein. Wenn daher alle lokalen Regionen alle Merkmale durchlaufen müssen, ist der Vorgang sehr schwer und zeitaufwändig. Um Rechenzeit zu sparen, sollten mehr potenzielle Muster getestet werden.

**Cascade-Klassifikatorarchitektur**

Bei der Cascade-Klassifikatorarchitektur enthält jede Stufe einen starken Klassifikator. Alle rechteckigen Merkmale sind in mehrere Gruppen unterteilt, die jeweils einige rechteckige Merkmale enthalten, die in jeder Stufe des kaskadierten Klassifizierers verwendet werden. Jede Stufe des kaskadierten Klassifikators bestimmt, ob es sich bei dem eingegebenen Bereich um ein menschliches Gesicht handelt, und wenn dies nicht der Fall ist, wird der Bereich sofort verworfen. Nur die Bereiche, die als mögliche menschliche Gesichter eingestuft werden, gelangen in die nächste Stufe, die durch komplexere Klassifikatoren weiter unterschieden wird [8]. in Abbildung 6 ist ein Filterverfahren zu sehen, das ein Gesichtklassifikation darstellt.

Ein Bild, das Uhr enthält.

Automatisch generierte Beschreibung

Abbildung 6: Gesichsfilterverfahren bei der Suche nach dem Gesicht im Bild [8]

### **2.2.1. Erkennung von Gesichtsmerkmalen**

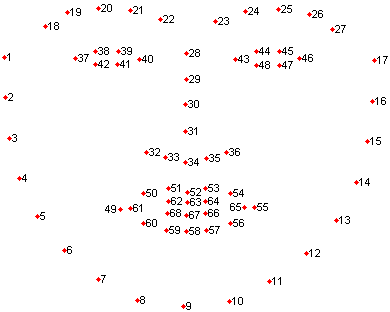
Bei der Verwendung des Algorithmus von Viola & Jones zur Gesichtserkennung sowie der Augen und des Mundes waren die Ergebnisse nicht präzise, daher war es notwendig, das Bild weiter zu analysieren. Eine Möglichkeit, dies zu tun, ist die Verwendung von Landmarks Abbildung 7**.** Dies enthält eine mögliche Zusammenstellung solchen Landmarks. Die 68 abgebildeten Orientierungspunkte beschreiben alle Merkmale des Gesichts, in dieser Arbeit waren die Augen und der Mund das Hauptmerkmale.

Abbildung 7: Die 68 Gesichtsmarkierungspunkte [28]

## **2.3. Künstliche Neuronale Netze**

Künstliche Neuronale Netze bestehen aus einer Reihe einfacher, stark miteinander verbundener Verarbeitungselemente, die Informationen durch ihre dynamische Reaktion auf externe Eingabe verarbeitet [18].

Künstliche Neuronale Netze verarbeiten Informationen auf ähnliche Weise wie das menschliche Gehirn. Das Netzwerk besteht aus einer großen Anzahl stark miteinander verbundener Verarbeitungselemente (Neuronen), die parallel an der Lösung eines bestimmten Problems arbeiten. Neuronale Netze lernen durch Beispiele. Sie können nicht programmiert werden, um eine bestimmte Aufgabe auszuführen. Die Muster müssen sorgfältig ausgewählt werden[37]. Somit sind Künstliche Neuronale Netze dazu fähig, durch Muster und maschinelles Lernen ihre Endergebnisse zu verbessern.

### **2.3.1. Biologische und künstliche Neuronen**

Im Gehirn erfolgt ein Informationsfluss in Form von kodierten Informationen von den Synapsen zum Axon. Das Axon eines jeden Neurons übermittelt Informationen an eine Reihe von anderen Neuronen. Das Neuron empfängt an den Synapsen Informationen von einer großen Anzahl anderer Neuronen. Es gibt Vermutungen, dass jedes Neuron Reize von bis zu 10.000 anderen Neuronen erhalten kann. Gruppen von Neuronen sind in Subsystemen organisiert, und die Integration dieser Subsysteme bildet das Gehirn. Es wird geschätzt, dass das menschliche Gehirn etwa 100 Milliarden miteinander verbundene Neuronen hat. In Abbildung 10 ist ein Biologisches Neuron dargestellt [10].

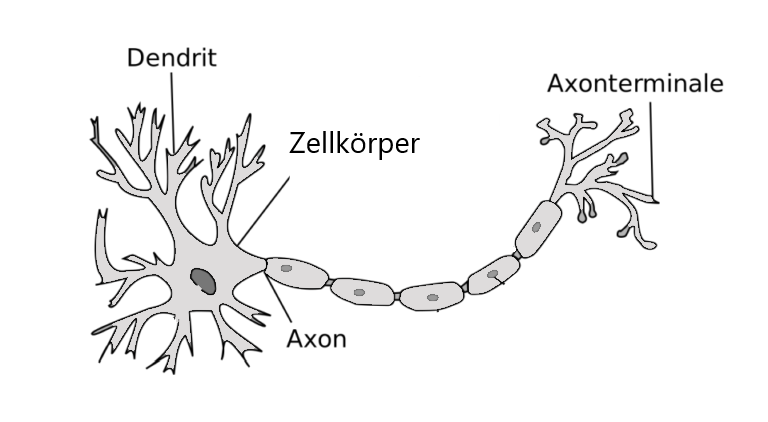


Abbildung 8: Biologisches Neuron [20]

In Abbildung 11 ist ein stark vereinfachtes Modell eines Künstlichen Neurons zu sehen. Bei der Erstellung eines Funktionsmodells des biologischen Neurons werden die Synapsen des Neurons als Gewichte modelliert. Die Stärke der Verbindung zwischen einem Eingang und einem Neuron wird durch den Wert des Gewichts angegeben. Negative Gewichtswerte spiegeln hemmende Verbindungen wider, während positive Werte erregende Verbindungen kennzeichnen. Die nächsten beiden Komponenten modellieren die tatsächliche Aktivität innerhalb der Neuronen-Zelle. Ein Addierer summiert alle Eingänge, die durch ihre jeweiligen Gewichte verändert wurden. Das Ergebnis wird als Parameter an die Akitvierungsfunktion weitergegeben. Diese Aktivität wird als Linearkombination bezeichnet. Schließlich steuert eine Aktivierungsfunktion die Amplitude der Ausgabe des Neurons. Ein akzeptabler Bereich für die Ausgabe liegt in der Regel zwischen 0 und 1 oder -1 und 1. Der zuletzt ermittelte Wert stellt den Anfangswert des Neuron dar und das ähnelt sich den Axon des biologischen Neurons [16].

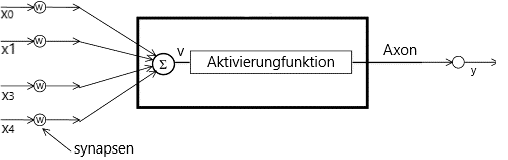


Abbildung 9: Vereinfachtes Modell eines künstlichen Neurons

### **2.3.2. Aktivierungsfunktion**

Das biologische Neuron wird in Neuronalen Netze durch eine Aktivierungsfunktion simuliert [2]. Die Aktivierungsfunktionen sind dabei sehr wichtig, da sie beim Erlernen und Verstehen nichtlinearer und komplizierter Verknüpfungen zwischen den Eingaben und den entsprechenden Ausgaben helfen [18]. Ohne Aktivierungsfunktionen wäre das Ausgangsignal eine einfache lineare Funktion. Lineare Funktionen sind zwar einfacher aber ihre Komplexität ist begrenzt und diese Art von Funktionen haben nicht die Fähigkeit, komplexe Zuordnungen von Daten zu lernen und erkennen. Darüber hinaus ist es wünschenswert, dass ein neuronales netz nicht nur eine lineare Funktion lernt und bearbeitet, sondern auch kompliziertere Aufgaben wie die Modellierung komplizierter Datentypen wie Bilder, Audio usw. [18]

Ein Beispiel für eine Aktivierungsfunktion ist „*Sigmoid Funktion*“. Diese Funktion hat Sigmoidkurve und die Kombinationen dieser Funktion sind nicht linear. In Abbildung 8 ist eine Darstellung dieser Funktion zu sehen [35].

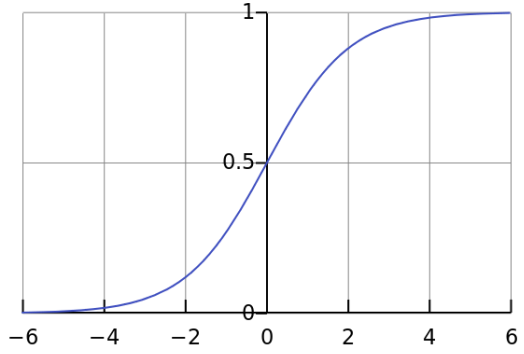


Abbildung 10: Darstellung einer Sigmoidfunktion [35]

Sigmoid-Funktion hat einen stetigen Gradientenwert. Daher eignet sie sich für flache Netze wie die Simulation von Funktionen. Die folgende Gleichung zeigt die Verwendung von Sigmoid-Funktion.[35].

Sigmoid-Funktion wird häufig in der praktischen Anwendung ausgewählt. Die Eingabewert bei dieser Aktivierungsfunktion liegt im Bereich des Intervalls .

Ein anderes Beispiel für Aktivierungsfunktion ist die Schwellenwertfunktion. Bei dieser Funktion erfolgt die Weitergabe erst ab einem Wert, das schon festgelegt wurde:

Ein weiteres Beispiel für eine Aktivierungsfunktion ist die ReLU-Funktion. ReLU steht für rectified linear unit (gleichgerichtete lineare Einheit) und stellt eine nichtlineare Aktivierungsfunktion, welche in neuronalen Netzen weit verbreitet ist.

Der Vorteil der ReLU-Funktion ist, dass nicht alle Neuronen gleichzeitig aktiviert werden. Dies bedeutet, dass ein Neuron nur dann deaktiviert wird, wenn die Ausgabe der linearen Transformation Null ist. ReLU Aktivierungsfunktion kann wie folgt definiert werden [17]:

In Abbildung 9 ist eine Darstellung dieser Funktion zu sehen.

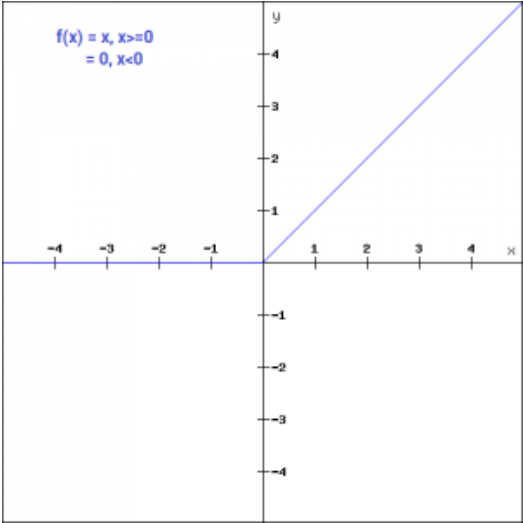


Abbildung 11: Darstellung einer RELU-Funktion [17]

### **2.3.3. Struktur eines neuronalen Netzes**

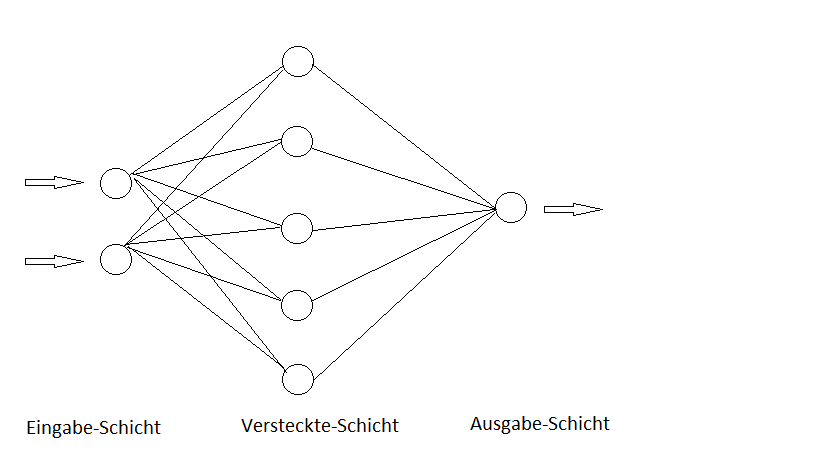
Grundsätzlich haben alle neuronalen Netze eine ähnliche Struktur oder Topologie wie in Abbildung 12 dargestellt. In dieser Struktur haben einige der Neuronen eine Schnittstelle zur realen Welt, um deren Eingaben zu empfangen. Andere Neuronen versorgen die reale Welt mit den Ausgaben des Netzes. Diese Ausgabe könnte das bestimmte Zeichen sein, das das Netz glaubt, gescannt zu haben, oder das bestimmte Bild, von dem es glaubt, dass es betrachtet wird. Alle anderen Neuronen sind nicht sichtbar. [26]

Abbildung 12: Einfaches Beispielstruktur für Neuronale netze

Die miteinander verbundenen Knoten unter die sogenannten Schichten (Layer) bilden ein künstliches neuronales Netz. Die Eingabeschicht (input layer) nimmt die Informationen (Eingaben) aus der Domain auf. In dieser Schicht befinden sich noch keine Berechnungen, darüber hinaus werden die Informationen an die Versteckteschichten (hidden layer) durch die Konten weitergeleitet, die in dieser Schicht sind. Die Versteckte-Schicht führt die nötigen Berechnungen mit den vorherigen Informationen durch und gibt das Ergebnis an die Ausgabe-Schicht (output layer) weiter. die Ausgabe-Schicht bekommt die gelernten Informationen und liefert den endgültigen Wert.

## **2.4. Faltungsneuronaler Netze**

Faltungsneuronaler Netze im Englischen „Convolutional Neural Network“ (CNN, ConvNet) hat Feed-Forward-Architektur und verfügt über eine gute Generalisierungsfähigkeit als Netze mit vollständig verbundenen Schichten [19]. Es kann sehr abstrakte Merkmale lernen und Objekte effizient erkennen [1]. Faltungsneuronaler Netzwerke verwendet das Konzept der Gewichtsteilung, wodurch die Anzahl der Parameter, die trainiert werden müssen, erheblich reduziert wird, das führt zu einer stärken Generalisierung [3]. Aufgrund der geringeren Anzahl von Parametern könnte CNN ohne große Probleme trainiert werden und keine Overfitting verursachen [33]. Die Klassifizierungsphase bei CNN ist mit der Phase der Merkmalsextraktion verbunden [13] , beide setzen Lernprozess ein. Außerdem wird CNN aufgrund ihrer hervorragenden Leistung [25] in verschiedene Bereiche eingesetzt z. B. bei der Bildklassifizierung, der Objekterkennung, der Gesichtserkennung usw.

Ein Allgemeines Faltungsneuronaler Netze Modell besteht aus Vier Komponente nämlich Faltungsschicht (Convolutional Layer), Pooling-Schicht (Pooling Layer), Aktivierungsfunktion und Vollständig verbundene Schicht (Fully connected Layer). Das Model eines Faltungsnetzes ist in Abbildung 13 dargestellt.

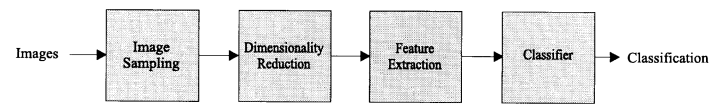


Abbildung 13: Komponenten Von Faltungsnetze [12]

**Faltungsschicht (Convolutional** **Layer):**

Die Eingabeschicht erhält ein zu Klassifizierendes Bild und die Ausgabe ist die vorhergesagte Klassenbezeichnung, die anhand der aus dem Bild extrahierten Merkmale berechnet wird [5]. Ein individuelles Neuron in der nächsten Schicht ist mit einigen Neuronen in der vorherigen Schicht verbunden, diese Lokale Korrelation wird als rezeptives Feld bezeichnet [19]. Die lokalen Merkmale des Eingangsbilds werden mit Hilfe des rezeptives Feldes extrahiert [13]. Das rezeptive Feld eines Neurons, das einer bestimmten Region in der vorherigen Schicht zugeordnet ist, bildet einen Gewichtsvektor, der an allen Punkten der Fläche gleich bleibt, wobei sich die Fläche auf die Neuronen der nächsten Schicht bezieht [12]. Die Neuronen in der Fläche haben die gleichen Gewichte, so dass ähnliche Merkmale, die an verschieden Stellen in den Eingabedaten auftreten, erkannt werden können [22]. In Abbildung 14 wird dieser Prozess veranschaulicht.

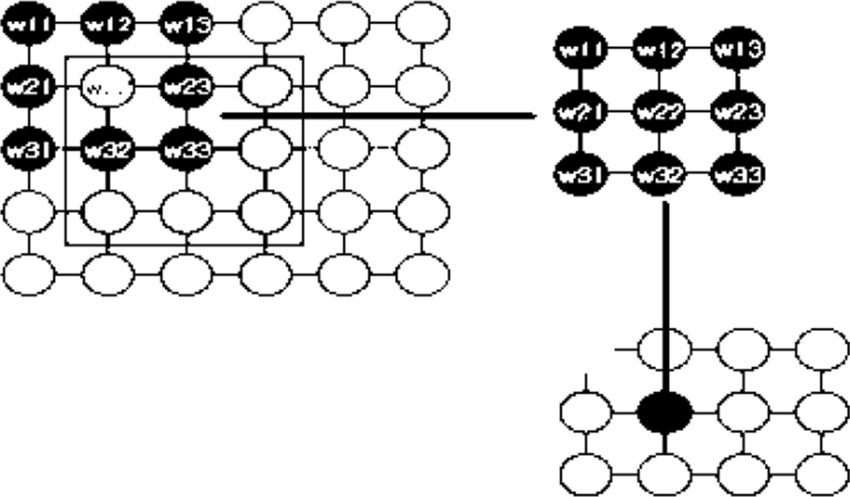


Abbildung 14: Rezeptives Feld eines neurons in der nächsten schicht [29]

Der Gewichtsvektor, der auch als Filter oder Kernel bezeichnet wird, wird über den Eingangsvektor verschoben, um die Feature Map zu erzeugen [14]. Diese Methode, den Filter sowohl horizontal als auch vertikal zu verschieben, wird als Faltungsoperation (convolution operation) bezeichnet. Diese Operation extrahiert eine Anzahl von Merkmalen aus dem Eingabebild in einer einzigen Schicht, die unterschiedliche Merkmale repräsentiert, was zu Filtern und Feature Map führt [22]. Aufgrund des lokalen rezeptiven Feldes wird die Anzahl der trainierbaren Parameter erheblich reduziert [38]. Die Ausgabe in der nächsten Schicht für den Ort wird nach Anwendung der Faltungsoperation mit der folgenden Formel berechnet [15].

Wobei die der Schicht zugeführte Eingabe ist, der Filter oder Kernel ist, der über die Eingabe geleitet wird, b das Bias, die Faltungsoperation darstellt und die in das Netz Nichtlinearität ist.

**Pooling-Schicht (Pooling Layer):**

Die genaue Position eines Merkmals verliert nach seiner Erkennung an Signifikanz [13]. Daher folgt auf die Faltungsschicht eine Pooling- oder Sub-Sampling-Schicht [12]. Der Hauptvorteil der Pooling-Technik besteht darin, dass sie die Anzahl der trainierbaren Parameter deutlich reduziert und eine Transformationsinvarianz einführt [38]. Zur Durchführung der Pooling-Operation wird ein Fenster ausgewählt, und die Eingabeelement, die in diesem Fenster liegen, werden durch eine Pooling-Funktion geleitet, in Abbildung 15 ein Beispiel für das Auswählen eines Fensters durch Max-Pooling-Funktion.

Ein Bild, das Tisch enthält.

Automatisch generierte Beschreibung

Abbildung 15: Pooling-Operation (2x2)- max-Pooling [27]

Die Pooling-Funktion erzeugt einen weiteren Ausgangsvektor. Es gibt nur wenige Pooling-Funktion wie (average pooling) und (max-pooling), von denen wird am häufigsten max-pooling verwendet, die die Größe des Feature-Maps stark reduziert [14].

**Vollständig verbundene Schicht (Fully Connected Layer):**

Der Ausgang der ersten Phase (einschließlich Faltung und wiederholtes Pooling) wird in die Voll vermaschte Schicht eingespeist, und das Punktproduckt aus Gewichtsvektor und Eingangsvektor wird berechnet, um den endgültigen Ausgang zu erhalten [38]. Fer Gradientenabstieg, auch bekannt als Batch-Modus-Lernen, reduziert die Kostenfunktion durch Schätzung der Kosten über einen gesamten Trainingsdatensatz und aktualisiert die Parameter nur nach einer Epoche, wobei eine Epoche dem Durchlaufen des gesamten Datensatzes entspricht. Dies führt zu globalen Minimumwerten, aber wenn der Trainingsdatensatz groß ist, erhöht sich die Zeit, die für das Training des Netzes erfordert. Dieser Ansatz zur Reduzierung der Kostenfunktion wurde durch den stochastischen Gradientenabstieg ersetzt.

# **Datensatz**

Für diese Arbeit habe ich den Datensatz "Facial Expression Recognition 2013 (FER 2013)" verwendet. Abbildung 16 enthält einige Beispielbilder mit ihren Emotionsbezeichnungen für diesen Datensatz.

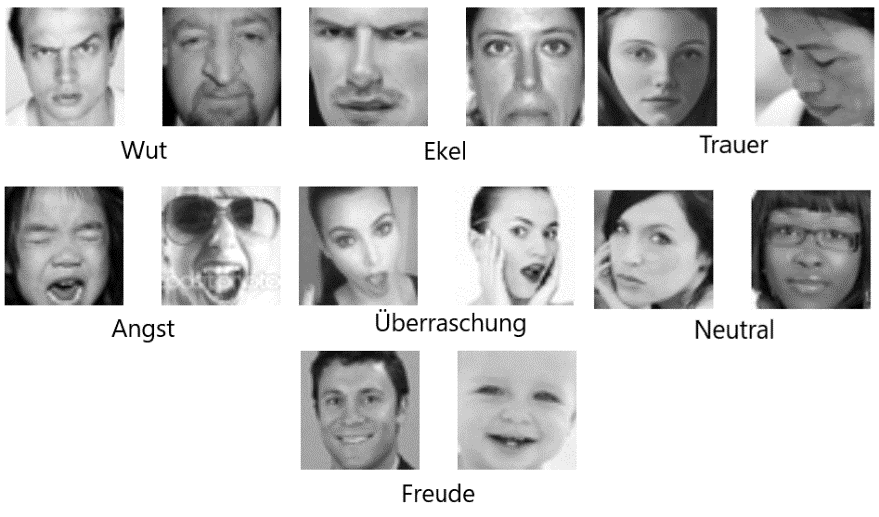


Abbildung 16: Beispiele für den Datensatz mit Bezeichnungen

Der FER-2013 ist ein von Kaggle bereitgestellter Datensatz, der auf der International Conference on Machine Learning (ICML) 2013 von Pierre-Luc Carrier und Aaron Courvill vorgestellt wurde [6]. In diesem Datensatz wurde jedes Gesicht auf der Grundlage von Emotionskategorien klassifiziert, dabei ist der FER-2013-Datensatz ein Graustufenbild mit 48 x 48 Pixel für jedes Bild. Der gesamte FER-2013-Datensatz umfasst 35 887 Bilder, die aus sieben verschiedenen Arten von Mikroausdrücken bestehen und mit Labels gekennzeichnet sind, die auf sieben verschiedenen Klassifizierungen basieren, beginnend mit dem Index-Label 0 bis 6 [31]. Eine ausführliche Tabelle über die Bilddatenverteilung findet sich in (Tabelle 1).

|  |  |  |  |
| --- | --- | --- | --- |
| **Kategorie** | **Trainingsdaten** | **Validierungsdaten** | **Gesamte Daten** |
| Freude | 7215 | 1774 | 8989 |
| Überraschung | 3171 | 831 | 4002 |
| Wut | 3995 | 958 | 4953 |
| Angst | 4097 | 1024 | 5121 |
| Ekel | 436 | 111 | 547 |
| Trauer | 4830 | 1247 | 6077 |
| Neutral | 4965 | 1233 | 6198 |

Tabelle 1: Übersicht über die Aufteilung der Bild Daten des FER-2013 Datensatzes [31]

# **Implementierung**

In diesem Kapitel wird die Durchführung dieser Arbeit erläutert. Zuerst wird ein Vergleich zwischen der Verwendung von Viola & Jones und Gesichtsmerkmalen in dem Abschnitt (**Vergleich zwischen der Verwendung von Viola & Jones und Gesichtsmerkmalen**) erläutert. Des Weiteren wird die Vorverarbeitung des Datensatzes in Abschnitt (**Preprocessing-Datensatz**) diskutiert, woraufhin die drei verschiedenen auf den Datensatz angewandten Strategien in den Abschnitten (**Ohne Preprocessing**), (**Preprocessing mit Gesichtserkennung**) und (**Preprocessing mit Erkennung von Augen und Mund**) beschrieben werden. In Abschnitt (**CNN-Model**) wird schließlich das in dieser Arbeit implementierte Modell der Faltungsneuronalen Netze analysiert.

## **4.1. Vergleich zwischen der Verwendung von Viola & Jones und Gesichtsmerkmalen**

Für den Vergleich zwischen der Verwendung von Gesichtsmerkmalen und der Methode von Viola & Jones zur Gesichtserkennung wurden 100 Bilder aus dem zugänglichen Datensatz entnommen und die Funktionalität beider Methoden an diesen Bildern getestet.

## **4.2. Preprocessing-Datensatz**

Für diese Arbeit wurde beschlossen, den Datensatz im Abschnitt (**Datensatz**) auf drei Möglichkeiten zu trainieren. Es wurde ohne Preprocessing trainiert, d. h. alle Bilder des Datensatzes werden in diesem Fall unverändert belassen. Die nächste Möglichkeit bestand darin, das Gesicht im Bild zu erkennen und eine Maske um es herum zu erstellen. Die letzte Option bestand darin, die Bilder nach der Gesichtserkennung weiter zu verarbeiten und sich beim Training auf die Augen und den Mund des erkannten Gesichts zu fokussieren und ebenfalls wieder eine Maske um die Augen und den Mund zu erstellen.

Um Zeit zu sparen, wurde das Preprocessing vor dem Trainingsprozess separat durchgeführt. Bei der Implementierung wurden zwei Funktionen in der Datei "preprocessing.py" implementiert, die es ermöglichen, die Bilder aus dem Ursprungspfad zu nehmen, sie zu verarbeiten und in einem neuen Ordner zu speichern.

## **4.2.1. Ohne Preprocessing**

Die Bilder des Datensatzes werden unverändert belassen. Abbildung 17 zeigt einige Beispiele für diesen Fall.



Abbildung 17: Unveränderte Bilder des Datensatzes

## **4.2.2. Preprocessing mit Gesichtserkennung**

Um weitestgehend zu verhindern, dass die Trainingsdaten aufgrund von anderen Objekten im Bild vom Gesicht abweichen, wurde diese Idee verwendet, das Gesicht zu erkennen und um es herum eine Maske erstellen. Nach der Bearbeitung sehen die meisten Bilder wie in Abbildung 18 dargestellt aus.

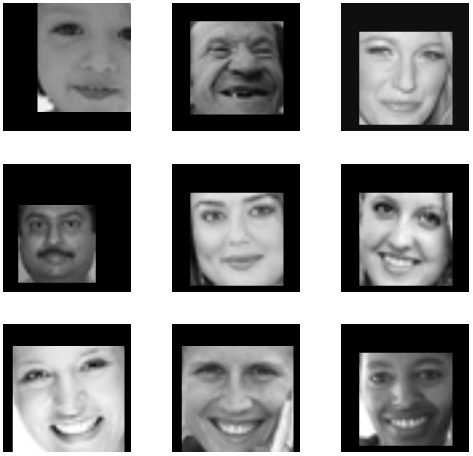


Abbildung 18: Gesichtserkennung und darum eine schwarze Maske

Hier trat das Problem aber auf, dass das Gesicht nicht in allen Bildern erkannt wurde, sehe Abbildung 19. Der Grund dafür könnte die kleine Größe der Bilder sein, die es schwierig macht, das Gesicht mit dem sowhol landmarks-Algorithmus als auch dem viola & jones-Algorithmus gut zu identifizieren.

****

Abbildung 19: Beispiel für das Zusammenfassen des Datensatzes mit allen Bildern

Der Trainingsprozess wurde jedoch trotz dieses Problems mit allen Bildern (denjenigen, die weiterverarbeitet wurden und denjenigen, die nicht weiterverarbeitet werden konnten) durchgeführt. Wie in Abbildung 19 dargestellt, ist bei den nicht bearbeiteten Bildern nur das Gesicht im Bild zu sehen, so dass eine Abweichung vom Gesicht während des Trainings nahezu unmöglich ist. Aus diesem Grund wurden in diesem Fall alle Bilder des Datensatzes aufgenommen.

von 28709 Trainingsdaten wurde das Gesicht in 20349 Bildern erkannt und in 8360 Bildern wurde das Gesicht nicht erkannt.

von 7178 Trainingsdaten wurde das Gesicht in 5074 Bildern erkannt und in 2104 Bildern wurde das Gesicht nicht erkannt.

## **4.2.3. Preprocessing mit Erkennung von Augen und Mund**

Wie bereits erwähnt, können Menschen in der Regel die Emotionen einer Person anhand ihres Gesichts erkennen. Insbesondere zeigen Experimente, dass die Augen und der Mund eine wichtige Rolle bei der Erkennung von Emotionen im Gesicht spielen [4].

Die Autoren in [4] stellen fest, dass sich bei ihren Experimenten die Mimik der Augen und des Mundes stark verändert hat. Die Abbildung 20 zeigt mögliche Darstellungen für wechselnde Gesichtsausdrücke.

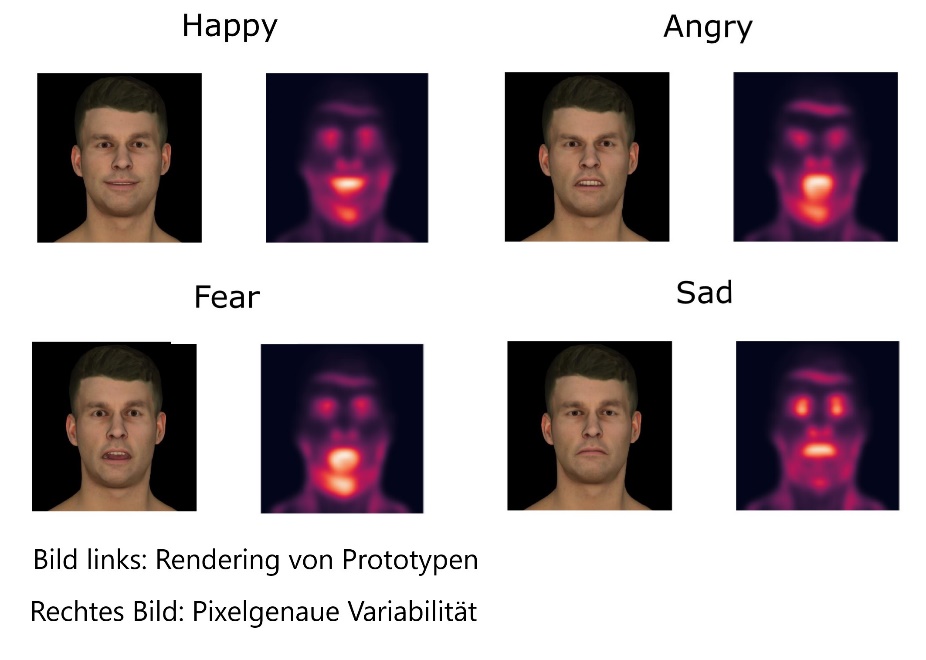


Abbildung 20: Merkmale zur Unterscheidung von Emotionskategorien. Prototypische Ausdrucksdarstellungen und Pixelweise Variabilität [4]

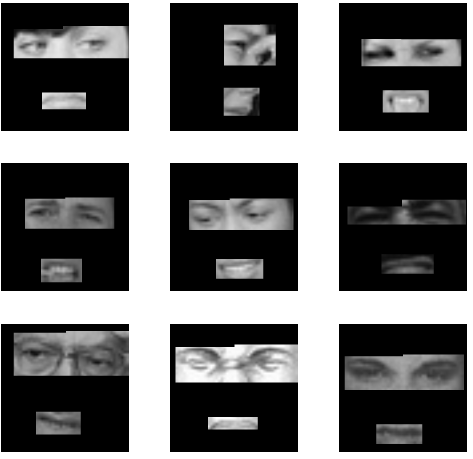
So entstand die Idee, die Bilder nach der Gesichtserkennung weiter zu verarbeiten und eine Maske um seine Augen und seinen Mund zu bilden, sehe Abbildung 21.

Abbildung 21: Augen-/Munderkennung und Maske um sie herum

Aber auch hier trat das gleiche Problem auf wie im letzten Abschnitt (es konnten nicht alle Bilder verarbeitet werden). In diesem Fall wurde der Trainingsprozess nur mit den Bildern durchgeführt, die verarbeitet werden konnten. Nach dieser Verarbeitung wurde die Anzahl der Bilder verringert. Die Tabelle 2 gibt detaillierte Informationen über die neue Bilddatenverteilung.

|  |  |  |  |
| --- | --- | --- | --- |
| **Kategorie** | **Trainingsdaten** | **Validierungsdaten** | **Gesamte Daten** |
| Freude | 5689 | 1401 | 7090 |
| Überraschung | 2402 | 616 | 3018 |
| Wut | 2817 | 665 | 3482 |
| Angst | 2583 | 658 | 3241 |
| Ekel | 341 | 85 | 426 |
| Trauer | 2769 | 705 | 3474 |
| Neutral | 3748 | 944 | 4692 |

Tabelle 2:Übersicht über die Aufteilung der Bild Daten des Datensatzes nach der Bearbeitung

## **4.3. CNN-Model**

„

Bild zum Modell hinzufügen, nachdem das Modell beschrieben wurde.

Und auch das File.txt mit allen Parametern.

“

Die erste Schicht des Modells in dieser Arbeit ist die Faltungsschicht. Sie verfügt über 32 Filter mit einer Kernelgröße von jeweils 5x5. Die verwendete Aktivierungsfunktion ist die ReLU-Funktion. Die Größe der Schicht wird als Tupel angegeben, d. h. die Eingabedaten haben die Größe (48, 48, 1), wobei 48 die Höhe und Breite des Eingabebildes in Pixeln angibt und 1 bedeutet, dass es sich um ein Graustufenbild handelt (im Gegensatz zu 3 für ein Vollfarbbild). Der Zweck einer Faltungsschicht in einem neuronalen Netz besteht darin, eine Reihe von Filtern auf die Eingabedaten anzuwenden, die es dem Modell ermöglichen, Merkmale aus den Daten zu lernen. Jeder Filter erzeugt eine „Feature Map“, indem er über die Eingabedaten gleitet, elementweise Multiplikationen mit den Eingabedaten durchführt und die Ergebnisse addiert. Die resultierende Feature Map wird dann durch eine Aktivierungsfunktion geleitet, bevor sie an die nächste Schicht des Modells weitergegeben wird.

Die nächste Schicht ist wieder eine Faltungsschicht mit 64 Filter, einer Kernelgröße von (5, 5) und ReLU-Aktivierung. Diese Schicht erwartet, dass sie Eingaben von der vorhergehenden Schicht erhält, daher sollte sie die gleiche Eingabeform wie die Ausgabeform der vorhergehenden Schicht haben. Die Ausgabeform dieser Schicht wird durch die Größe der Filter, die verwendete Schrittweite und die Eingabeform bestimmt.

die Kernelgröße ist (5,5), was bedeutet, dass die Filter 5x5-Matrizen sind. Es ist möglich, dass diese Kernelgröße gewählt wurde, weil sie ein gutes Gleichgewicht zwischen der Fähigkeit des Modells, Muster zu erkennen, und seiner Berechnungseffizienz herstellt. Ohne weitere Informationen über das Problem und das Modell ist es jedoch schwierig, mit Sicherheit zu sagen, warum diese Kernelgröße gewählt wurde.

Dem Modell wurde eine Pooling-Schicht hinzugefügt. Pooling ist ein Prozess des Downsamplings der räumlichen Dimensionen der Daten, der die Anzahl der Parameter im Modell reduzieren und seine Fähigkeit zur Generalisierung auf neue Daten verbessern kann. Die spezifische Art des Pooling, die in dieser Schicht implementiert ist, ist das Max-Pooling, das den Maximalwert aus einem Satz benachbarter Werte in den Eingabedaten auswählt und ihn als das entsprechende Element in der Ausgabe ausgibt. Die Größe des Pooling-Fensters wird auf (2, 2) festgelegt, was bedeutet, dass das Pooling-Fenster eine Höhe und Breite von 2 hat. Die Eingabeform dieser Schicht wird nicht explizit angegeben, da sie automatisch aus der Ausgabe der vorherigen Schicht abgeleitet wird. Die Ausgabe der vorherigen Schicht wird als Eingabe für diese Schicht verwendet.

Die nächste Schicht ist Dropout-Schicht. Dropout ist eine Regularisierungstechnik zur Reduzierung von Overfitting in neuronalen Netzen, indem ein Teil der Eingabeeinheiten während des Trainings zufällig auf null gesetzt wird. Der Anteil der auf null zu setzenden Eingabeeinheiten wird auf 0,25 festgelegt. Das bedeutet, dass während des Trainings 25 % der Eingabeeinheiten bei jeder Aktualisierung der Modellgewichte auf null gesetzt werden. Die verbleibenden Einheiten werden um den Faktor 1 – 0,25 verkleinert, in diesem Fall also um den Faktor 0,75. Die Ausgabe dieser Schicht hat die gleiche Form wie die Eingabe, aber mit einer zufälligen Teilmenge der Eingabeeinheiten, die entsprechend der Dropout-Rate auf null gesetzt wird. Diese Zufallsauswahl wird bei jeder Iteration des Trainings anders angewandt, was dem Modell helfen kann, robustere Merkmale zu lernen und die Überanpassung zu reduzieren. Während der Auswertung und Inferenz leitet die Dropout-Schicht die Eingaben einfach an die nächste Schicht weiter, ohne sie zu verwerfen, so dass das Modell mit allen Einheiten Vorhersagen machen kann.

# **Ergebnisse**

In diesem Kapitel werden die Ergebnisse der Versuche für die drei in Abschnitt (**Preprocessing-Datensatz**) genannten Optionen beschrieben. Für diese Ergebnisse innerhalb des Rahmens dieser Arbeit steht ein Laptop mit folgenden Spezifikationen zur Verfügung:

* Prozessor: Intel Core i5-7200U CPU @ 2.50GHz x 4
* Arbeitsspeicher: 24 GB DDR4
* Bestriebssystem: Linux Ubuntu 22.04.1 LTS
* Python 3.10.6
* OpenCV 4.6.0
* Tensorflow Version 2.10.0

## **5.1. Ergebnis des Vergleichs zwischen der Verwendung von Viola & Jones und Gesichtsmerkmalen**

Der Vergleich zwischen den beiden Gesichtserkennungsoptionen führte zu den in Abbildung 22 dargestellten Ergebnissen.

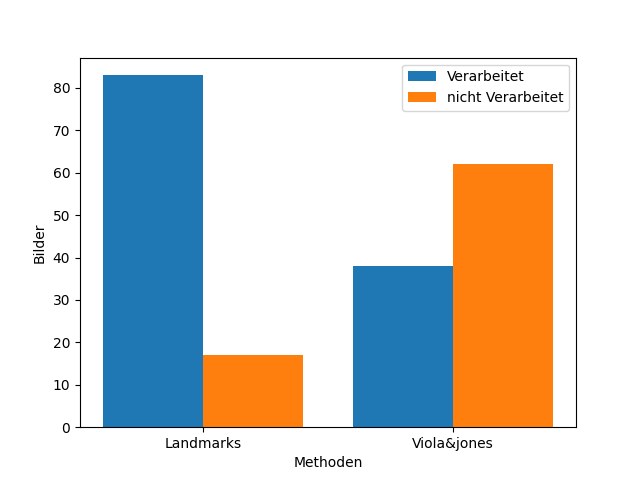


Abbildung 22: Vergleich zwischen der Viola&Jones-Methode und der Verwendung von Landmarken mit 100 Bildern

Die Methode von Viola & Jones konnte 62 von 100 Bildern nicht verarbeiten, d. h. es wurde kein Gesicht erkannt. Bei der Verwendung von Gesichtsmerkmalen konnten nur 17 von 100 Bildern nicht verarbeitet werden, also das Gesicht wurde auf den meisten Bildern erkannt.

Vor allem bei den Bildern, die mit Viola & Jones verarbeitet wurden, waren die Ergebnisse für die Augen- und Munderkennung in den meisten Fällen nicht ganz korrekt. Abbildung 23 zeigt ein Beispiel für die Ergebnisse der Augen- und Munderkennung.



Abbildung 23: Fehlgeschlagene Erkennung für Mund und Augen

Das rote Rechteck soll das Auge repräsentieren und das blaue Rechteck soll den Mund darstellen. In dieser Abbildung 23 ist die Verwechslung zwischen Augen und Mund deutlich zu erkennen. Außerdem wurde an einigen Stellen des Bildes ein Auge erkannt, was ebenfalls nicht korrekt ist. Daher ist es schwierig, Masken um die richtigen Stellen herum zu erstellen.

„

In den Ergebnisse sollen die Diagrame hinzugefügt (die sind 6 )

Die ausführlich beschreiben.

Die accuracy und loss mit Zahlen erläutern, wie genau war das Modell und so weiter

“

## **5.2. Ergebnisse des Trainings ohne Preprocessing**

## **5.3. Ergebnisse des Trainings mit Gesichtserkennung**

## **5.4. Ergebnisse des Trainings mit Augen- Munderkennung**

# **Code-Erklärung**

In diesem Abschnitt sollen die Dateien des Python-Skripts kurz erläutert werden.

* **main.py**

In *main.py* wird das Programm in Abhängigkeit von den Eingabeparametern auf drei Arten ausgeführt.

1. Mit dem Parameter „-t testCam“ werden die Ergebnisse mit der ausgewählten Kamera nach dem Trainingsprozess getestet.
2. Mit dem Parameter „-i 'Bildpfad'“ werden die Ergebnisse mit dem ausgewählten Bild nach dem Trainingsprozess ausprobiert.
3. Mit dem Parameter „-l train“ wird der Trainingsprozess durchgeführt.

* **preprocessing.py**

In *preprocessing.py* werden die Bilder aus dem Datensatz verarbeitet und gespeichert. Diese Datei wurde separat für die gesamte Arbeit mit zwei Möglichkeiten ausgeführt.

1. Parameter „-f face“ führt eine Funktion aus, die eine schwarze Maske um das gesamte Gesicht erzeugt
2. Parameter „-em eyeMouth“ führt eine Funktion aus, die eine schwarze Maske um die Augen und den Mund erzeugt

* **camera.py**

In dieser Datei wurde eine Klasse namens „Camera" erstellt. Mit dieser Klasse wird der Testprozess ausgeführt, die Kamera ausgewählt und die gespeicherten Daten für den Test verwendet.

* **model.py**

In dieser Datei werden die Faltungsneuronale Netze aufgebaut. Auch hier wurde eine Klasse mit dem Namen „Model" erstellt. Mit dieser Klasse können auch die Parameter des Netzes bzw. Datensatz geändert werden.

* **helpFunctions.py**

In *helpFunctions.py* sind zwei Funktionen verfügbar. Eine davon dient dazu, die Bilder aus dem Datensatz abzurufen und sie gegebenenfalls zu plotten. Mit der anderen lässt sich die Größe der Bilder des Datensatzes ändern. Weiterhin kann diese Datei in einem späteren Schritt erweitert werden.

* **compare.py**

wie der Name dieser Datei schon sagt, wurde diese Datei nur zum Vergleichen oder Testen erstellt.

# Diskussion

Die Ergebnisse wieder hier erläutern

Mit anderen Datensatz könnten die Ergebnissen (von den zwei letzten Versuche) besser sein.

Großere Bilder wären auch besser gewesen.

Wenige Bilder bei mensche Emotionen war auch ein Problem, z.B bilder von Happy sind viel Mehr als Bilder von disgusting.

Mit datensatz. die auch die Objekte in dem Bild nimmt, wäre auch gut.

Die Ideen, was ich gemacht habe, sind gut. Für weitere Arbeit würde ich die einsetzen aber mit besseren Datensatz (bessere Bilder).

Erkennung von Augen und Mund würde ich es verbessern, um später bessere Ergebnisse zu bekommen.

# Zusammenfassung

Später würde ich mit dieser Arbeit weiter arbeiten, um eine Applikation zu machen, die im Hintergrund anderer Applikation läuft, und die Emotionen zeigt, z.B bei whatsApp viedocall oder bei Zoom-Konferenz usw. die Applikation sollte für alle Betriebsysteme laufen (Laptopn(Windows, Linux …), Handy(Android, IOS))

Ich würde mehrere Datensatzen hinzufügen, nicht nur Bilder sondern auch audios, um auch durch die Stimme emotionen zuerkennen.

# **Abbildungsverzeichnis**

[Abbildung 1: Beispiel für haar-features 4](#_Toc124010555)

[Abbildung 2: punkt (x,y) ist die summe aller pixel oberhalb und links davon [29] 6](file:///C:\Users\Simon\Desktop\BA-Simon%5b37%5d.docx#_Toc124010556)

[Abbildung 3: integralbild eines bestimmten recheckigen Bereichs 7](#_Toc124010557)

[Abbildung 4: Beispiel für Bild- integral 8](#_Toc124010558)

[Abbildung 5: Integral-Bild ergebniss 8](#_Toc124010559)

[Abbildung 6: Gesichsfilterverfahren bei der Suche nach dem Gesicht im Bild [8] 9](#_Toc124010560)

[Abbildung 7: Die 68 Gesichtsmarkierungspunkte [26] 10](file:///C:\Users\Simon\Desktop\BA-Simon%5b37%5d.docx#_Toc124010561)

[Abbildung 8: Darstellung einer Sigmoidfunktion [34] 11](#_Toc124010562)

[Abbildung 9: darstellung einer RELU-funktion [18] 12](#_Toc124010563)

[Abbildung 10: Biologisches Neuron [20] 13](#_Toc124010564)

[Abbildung 11: Vereinfachtes Modell eines künstlichen Neurons. 14](#_Toc124010565)

[Abbildung 12: Einfaches Beispielstruktur für Neuronale netze 14](#_Toc124010566)

[Abbildung 13: Komponenten Von Faltungsnetze [12] 15](#_Toc124010567)

[Abbildung 14: Rezeptives Feld eines neurons in der nächsten schicht [28] 16](#_Toc124010568)

[Abbildung 15: Pooling-Operation (2x2)- max-Pooling [27] 17](#_Toc124010569)

[Abbildung 16: Beispiele für den Datensatz mit Bezeichnungen 18](#_Toc124010570)

[Abbildung 17: Unveränderte Bilder des Datensatzes 19](#_Toc124010571)

[Abbildung 18: Gesichtserkennung und darum eine schwarze Maske 20](#_Toc124010572)

[Abbildung 19: Beispiel für das Zusammenfassen des Datensatzes mit allen Bildern 20](#_Toc124010573)

[Abbildung 20: Merkmale zur Unterscheidung von Emotionskategorien. Prototypische Ausdrucksdarstellungen und Pixelweise Variabilität [4] 21](file:///C:\Users\Simon\Desktop\BA-Simon%5b37%5d.docx#_Toc124010574)

[Abbildung 21: Augen-/Munderkennung und Maske um sie herum 21](file:///C:\Users\Simon\Desktop\BA-Simon%5b37%5d.docx#_Toc124010575)

[Abbildung 22: Vergleich zwischen der Viola&Jones-Methode und der Verwendung von Landmarken mit 100 Bildern 24](#_Toc124010576)

[Abbildung 23: fehlgeschlagene Erkennung für Mund und Augen 25](#_Toc124010577)

# **Tabbellenverzeichnis**

[Tabelle 1: Übersicht über die Aufteilung der Bild Daten des ger-2013 Datensates [31] 18](#_Toc124113972)

[Tabelle 2:Übersicht über die Aufteilung der Bild Daten des Datensatzes nach der Bearbeitung 23](#_Toc124113973)

References

[1] 2016.10 - Derivation of Backpropagation in Convolutional Neural Network (CNN).

[2] Andy. Microsoft Word - A beginners introduction to neural networks V2.

[3] Arel, I., Rose, D. C., and Karnowski, T. P. 2010. Deep Machine Learning - A New Frontier in Artificial Intelligence Research [Research Frontier]. *IEEE Computational Intelligence Magazine* 5, 4, 13–18.

[4] Binetti, N., Roubtsova, N., Carlisi, C., Cosker, D., Viding, E., and Mareschal, I. 2022. Genetic algorithms reveal profound individual differences in emotion recognition. *Proceedings of the National Academy of Sciences of the United States of America* 119, 45, e2201380119.

[5] Fang, J., Zhou, Y., Yu, Y., and Du, S. 2017. Fine-Grained Vehicle Model Recognition Using A Coarse-to-Fine Convolutional Neural Network Architecture. *IEEE Transactions on Intelligent Transportation Systems* 18, 7, 1782–1792.

[6] Goodfellow, I. J., Erhan, D., Carrier, P. L., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.-H., Zhou, Y., Ramaiah, C., Feng, F., Li, R., Wang, X., Athanasakis, D., Shawe-Taylor, J., Milakov, M., Park, J., Ionescu, R., Popescu, M., Grozea, C., Bergstra, J., Xie, J., Romaszko, L., Xu, B., Chuang, Z., and Bengio, Y. 2013. *Challenges in Representation Learning: A report on three machine learning contests*.

[7] Google Books. 2022. *Human Emotions*. https://​books.google.de​/​books?hl=en&lr=&id=DYoHCAAAQBAJ&oi=fnd&pg=PA1&dq=izard+c.+e.+(1977).+human+emotions.+new+york+plenum+press&ots=HyTh4XBO7x&sig=Dh1fstJfrOD-YemBkHwYHxOlt-jE. Accessed 21 November 2022.

[8] Huang, J., Shang, Y., and Chen, H. 2019. Improved Viola-Jones face detection algorithm based on HoloLens. *J Image Video Proc.* 2019, 1, 1–11.

[9] Kaiqi Cen. 2016. Study of Viola-Jones Real Time Face Detector.

[10] Kalogirou, S. A. 2001. Artificial neural networks in renewable energy systems applications: a review. *Renewable and Sustainable Energy Reviews* 5, 4, 373–401.

[11] Lambie, J. A. and Marcel, A. J. 2002. Consciousness and the varieties of emotion experience: a theoretical framework. *Psychological Review* 109, 2, 219–259.

[12] Lawrence, S., Giles, C. L., Tsoi, A. C., and Back, A. D. 1997. Face recognition: a convolutional neural-network approach. *IEEE Transactions on Neural Networks* 8, 1, 98–113.

[13] Lecun, Y., Bottou, L., Bengio, Y., and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 11, 2278–2324.

[14] Lee, K. B., Cheon, S., and Kim, C. O. 2017. A Convolutional Neural Network for Fault Classification and Diagnosis in Semiconductor Manufacturing Processes. *IEEE Transactions on Semiconductor Manufacturing* 30, 2, 135–142.

[15] Maggiori, E., Tarabalka, Y., Charpiat, G., and Alliez, P. 2017. Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification. *IEEE Transactions on Geoscience and Remote Sensing* 55, 2, 645–657.

[16] Microsoft Word - 54621145.docx.

[17] misra, p. .

[18] misra, p. .

[19] Nebauer, C. 1998. Evaluation of convolutional neural networks for visual recognition. *IEEE Transactions on Neural Networks* 9, 4, 685–696.

[20] Oppermann, A. 2021. *Künstliche Neuronale Netze - KI Tutorials*. https://​artemoppermann.com​/​de/​kuenstliche-neuronale-netze/​. Accessed 21 November 2022.

[21] P. Viola and M. Jones. 2001. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001* 1, I. DOI=10.1109/CVPR.2001.990517.

[22] Palsson, F., Sveinsson, J. R., and Ulfarsson, M. O. 2017. Multispectral and Hyperspectral Image Fusion Using a 3-D-Convolutional Neural Network. *IEEE Geoscience and Remote Sensing Letters* 14, 5, 639–643.

[23] Plutchik, R. 19]84. Chapter 1 - A GENERAL PSYCHOEVOLUTIONARY THEORY OF EMOTION. In *Emotion*. *Theory, research, and experience*, R. Plutchik, Ed. Acad. Press, New York [u.a.], 3–33. DOI=10.1016/B978-0-12-558701-3.50007-7.

[24] Plutchik, R., Ed. 19]84. *Emotion*. *Theory, research, and experience*. Acad. Press, New York [u.a.].

[25] R. Xu, B. Yuan, X. You, and C. Zhang. 2017. Efficient fast convolution architecture based on stochastic computing. In *2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, 1–6. DOI=10.1109/WCSP.2017.8171031.

[26] Research\_Paper\_on\_Basic\_of\_Artificial\_Neural\_Network-with-cover-page-v2.

[27] ResearchGate. 2022. *Fig. 2. Illustration of Max Pooling function*. https://​www.researchgate.net​/​figure/​Illustration-of-Max-Pooling-function\_fig2\_339906803. Accessed 22 November 2022.

[28] ResearchGate. 2022. *Figure 4.12: The ibug 68 facial landmark points mark-up*. https://​www.researchgate.net​/​figure/​The-ibug-68-facial-landmark-points-mark-up\_fig9\_327500528. Accessed 16 December 2022.

[29] ResearchGate. 2022. *Receptive field of particular neuron in next layer*. https://​www.researchgate.net​/​figure/​Receptive-field-of-particular-neuron-in-next-layer\_fig1\_357991037. Accessed 21 November 2022.

[30] ResearchGate. 2022. *(PDF) An Analysis of the Viola-Jones Face Detection Algorithm*. https://​www.researchgate.net​/​publication/​272643562\_An\_Analysis\_of\_the\_Viola-Jones\_Face\_Detection\_Algorithm. Accessed 27 October 2022.

[31] Sambare, M. 2020. *FER-2013*. https://​www.kaggle.com​/​datasets/​msambare/​fer2013. Accessed 16 December 2022.

[32] Skelly, J. ECG703\_SKELLY\_Final\_Report.

[33] Smirnov, E. A., Timoshenko, D. M., and Andrianov, S. N. 2014. Comparison of Regularization Methods for ImageNet Classification with Deep Convolutional Neural Networks. *AASRI Procedia* 6, 89–94.

[34] SpringerLink. 2022. *Motivation und Emotion*. https://​link.springer.com​/​book/​10.1007/​978-3-531-93420-4. Accessed 21 November 2022.

[35] Tomasz Szandała. 2020. *Review and Comparison of Commonly Used Activation Functions for Deep Neural Networks*.

[36] Universal-Facial-Expressions-of-Emotions1.

[37] van Veen, F. 2016. The Neural Network Zoo. *The Asimov Institute* (Sep. 2016).

[38] Zhou, Y., Wang, H., Xu, F., and Jin, Y.-Q. 2016. Polarimetric SAR Image Classification Using Deep Convolutional Neural Networks. *IEEE Geoscience and Remote Sensing Letters* 13, 12, 1935–1939.