



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Bianca Niklass

Konzeption und Entwicklung einer Android-App zur Erkennung von
Emotionen auf Fotos

Bianca Niklass

Konzeption und Entwicklung einer Android-App zur Erkennung von
Emotionen auf Fotos

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Stefan Sarstedt
Zweitgutachter : Prof. Dr. Olaf Zukunft

Abgegeben am 14.06.2018

Bianca Niklass

Thema der Arbeit/Ausarbeitung

Konzeption und Entwicklung einer Android-App zur Erkennung von Emotionen auf Fotos

Stichworte

Emotionen, Computer Vision, lernender Algorithmus

Kurzzusammenfassung

In zwischenmenschlichen Beziehungen ist es wichtig einschätzen zu können, wie sich der Gegenüber fühlt. Doch diese Einschätzung ist nicht für jeden Menschen leicht. In dieser Arbeit wurde daher eine Android-App entwickelt, die Emotionen auf Fotos erkennen kann. Dafür wurden die Grundlagen der Bildverarbeitung in der Computer Vision, wie auch Verfahren zur Mustererkennung veranschaulicht.

Bianca Niklass

Title of the paper

Design and development of an android application, which recognizes emotions in photos

Keywords

computer Vision, machine learning algorithms, emotions

Abstract

The ability to recognize how somebody else feels, is an important aspect of a relationship between two people. But there are people who do not recognize emotions in somebody else. Therefore, in this bachelor report, an android application is designed and developed, which recognizes emotions in photos.

Inhaltsverzeichnis

1	Einleitung	6
1.1	Problemstellung	6
1.2	Zielsetzung	6
2	Emotionen.....	7
2.1	Definition.....	7
2.2	Die primären und sekundären Emotionen.....	7
2.3	Abgrenzung des Themas	8
3	Grundlagen	9
3.1	Computer Vision.....	9
3.1.1	Darstellung von Bildern.....	9
3.1.2	Das RGB-Modell	10
3.1.3	Graustufenbilder	11
3.1.4	Umwandlung eines RGB-Bildes in ein Graustufenbild	11
3.2	Gesichtserkennung.....	12
3.2.1	Schwierigkeiten	12
3.2.2	OpenCV	14
3.2.3	Viola-Jones Algorithmus.....	14
3.3	Emotionserkennung	21
3.3.1	Microsoft Emotion API	21
3.3.2	Representational State Transfer	22
3.3.3	Eigenfaces-Algorithmus	23

4	Konzept	28
4.1	Technische Voraussetzungen und Daten	28
4.2	Erster Entwurf	28
4.2.1	Aufbau	28
4.2.2	Datenbank der Trainingsdaten.....	29
4.2.3	Komponenten.....	30
4.2.4	Implementierung.....	31
4.3	Finales Konzept	31
4.3.1	Aufbau	31
4.3.2	Komponenten.....	32
4.3.3	Oberfläche.....	35
5	Fazit	37
5.1	Zusammenfassung	37
5.2	Ausblick	38
6	Literaturverzeichnis	40
7	Abbildungsverzeichnis	41

1 Einleitung

1.1 Problemstellung

Eine wichtige Rolle in der zwischenmenschlichen Beziehung spielt die Fähigkeit einschätzen zu können wie der Gegenüber sich fühlt. Man wählt seine Worte weicher, wenn der Gegenüber verärgert ist, ist euphorischer, wenn sich jemand freut und tröstet, wenn der Andere traurig ist.

Doch es gibt auch Personen, die an Hand der Mimik nicht einschätzen können wie sich jemand anderes fühlt. Dieses erschwert die sozialen Kontakte für diejenigen.

Eine Applikation, die Emotionen des Gegenübers einordnen kann würde für Personen, die damit Probleme haben, eine erhebliche Erleichterung bieten und somit zwischenmenschliche Beziehungen fördern.

1.2 Zielsetzung

Die hier entwickelte Applikation soll an Hand eines Fotos einer Person dessen Emotion zurückgeben. Zuerst soll eine Gesichtserkennung durchgeführt werden, um die Person in dem Foto zu ermitteln. Daraufhin soll ein Verfahren der Emotionserkennung, das primäre Emotionen erkennen kann, eingesetzt werden.

Da Fotos verarbeitet werden, beschäftigt sich diese Arbeit mit den Algorithmen und Verfahren der Computer Vision.

2 Emotionen

2.1 Definition

Emotionen sind psychophysische Reaktionsmuster, die auf der Bewertung einer Reizsituation beruhen. Diese gehen mit einer Reihe psychologischer Veränderungen sowie der Aktivierung zentralnervöser Systeme einher, motivieren zu bestimmten Klassen von Verhalten, drücken sich in spezifischer Mimik und Körperhaltung aus und sind häufig mit einer subjektiven Erlebnisqualität verbunden. (1)



Abbildung 1: Die Emotionsentstehung nach James und Lange

2.2 Die primären und sekundären Emotionen

In der Psychologie spricht man von zwei verschiedenen Klassen von Emotionen, den primären und sekundären. Die primären Emotionen, die sich in sechs Bestandteile gliedern, sind Freude, Trauer, Ärger, Furcht/Angst, Überraschung und Ekel. In den 1970er Jahren hat der Verhaltensforscher Irenäus Eibl-Eibesfeldt eine Studie mit blinden und tauben Kindern durchgeführt, die belegt hat, dass die blinden Kinder, die die Mimik anderer Menschen nie beobachtet haben konnten, die primären

Emotionen mit der gleichen Mimik ausgedrückt haben wie Sehende. Zum Beispiel lächelten sie, wenn sie sich freuten und weinten, wenn sie traurig waren. Somit haben diese Emotionen mit großer Wahrscheinlichkeit einen genetischen Ursprung und werden nicht durch das Beobachten anderer Menschen erlernt. Da die Mimik auf diese Emotionen nicht erlernt werden muss, ist sie kulturübergreifend gleich und unabhängig von sozialen Einflüssen. Intraindividuell, innerhalb eines Individuums, zeigen die primären Emotionen die gleichen Reaktionsmuster. Interindividuell, zwischen einzelnen Individuen, hingegen können auf diese Emotionen unterschiedlich reagiert werden. Die primären Emotionen werden von allen Personen mit den gleichen fünf Muskeln im Gesicht ausgedrückt.

Sekundär werden Emotionen genannt, bei denen die Reaktionen auf bestimmte Emotionen erlernt werden müssen und die durch die soziale Rolle, wie auch die Kultur des Individuums beeinflusst werden. Hinzu werden auch Misch-Emotionen, die sich aus mehreren Basisemotionen zusammensetzen, als sekundär bezeichnet, wie zum Beispiel die Emotion Depression, die sich aus Trauer, Schuld, wie auch Verzweiflung und Angst zusammensetzt. (3) (4)

2.3 Abgrenzung des Themas

Da nur primäre Emotionen in allen Bevölkerungsgruppen identisch ausgedrückt werden, bezieht sich diese Bachelorarbeit nur auf diese.

Um die Emotionen des Gegenübers einzuschätzen, beobachten die Personen nicht nur die Mimik des Gegenübers. Es wird auch die Stimmlage, wie auch die Situation, in der sich der Gegenüber gerade befindet aber auch die Körperhaltung des anderen analysiert.

Da die Bestimmung der Emotion einer anderen Person von vielen Informationen abhängt, welche nicht auf einem Foto vorhanden sind, befasst sich diese Bachelorarbeit auf das Erkennen von Emotionen ausschließlich unter Berücksichtigung der Mimik. (5)

3 Grundlagen

Ziel dieses Kapitels ist es einen Überblick über die relevanten Vorgehensweisen und Algorithmen für die Gesichts- und Emotionserkennung zu vermitteln. Zuerst werden Hintergrundinformationen über das Darstellen von Bildern gegeben. In diesem Zusammenhang werden nicht nur Graustufenbilder, sondern auch das RGB-Modell vorgestellt. Weiterhin werden die wichtigsten Algorithmen und ein Framework diskutiert, welche sich mit dem Thema Gesichtserkennung beschäftigen. Daraufhin werden ein Algorithmus und ein Framework, die man zu dem Identifizieren von Emotionen verwenden kann, vorgestellt.

3.1 Computer Vision

Das Feld der Computer Vision beschäftigt sich mit Aufgabenstellungen, die sich an den visuellen menschlichen Fähigkeiten orientieren und computergestützt gelöst werden. Typische Aufgaben in diesem Bereich sind zum Beispiel Oberflächeninspektionen und Objektlageerkennungen.

3.1.1 Darstellung von Bildern

Um ein Bild computerbasiert bearbeiten zu können, muss es für den Computer verständlich definiert werden. Dieses geschieht meistens in Form von Matrizen, wobei jeder Eintrag der Matrix ein Pixel repräsentiert. Somit hat die Matrix eines Bildes mit der Höhe x und Breite y schon $x*y$ Einträge. Bei einem Graustufenbild wird jeder Eintrag mit einem einzigen Grauwert belegt. Bei einem RGB-Bild hingegen muss jeder Eintrag durch drei Werte, die für die Farben Rot, Grün und Blau stehen, definiert werden. Dieses kann auf zwei unterschiedliche Weisen dargestellt werden. Entweder wird jeder Eintrag in der Matrix durch drei Einträge ersetzt oder die Einträge werden durch eigene Repräsentationen, also eine Matrix, dargestellt. Eine mögliche Matrix zur Repräsentation eines Graustufenbildes ist folgende:

$$\begin{pmatrix} P_{1,1} & \cdot & \cdot & \cdot & P_{1,x} \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ P_{y,1} & \cdot & \cdot & \cdot & P_{y,x} \end{pmatrix}$$

Im Gegensatz zu einem geometrischen Koordinatensystem liegt der Nullpunkt hier nicht in der Mitte der Matrix, sondern in der linken oberen Ecke (in Abbildung 2 somit bei dem Punkt P1,1).

3.1.2 Das RGB-Modell

Um ein Bild mit unterschiedlichen Farben auf einem Computer darstellen zu können, wird das Dreifarbenmodell angewendet. Dieses geht davon aus, dass man jede Farbe mit den drei Grundfarben Rot, Grün und Blau ausdrücken kann.

Das Verfahren der additiven Farbmischung wird hier eingesetzt. Bei diesem Verfahren startet jede Farbe mit der Farbe Schwarz und wird durch das Addieren mit einer Grundfarbe zu der gewünschten Farbe.

Wie viele unterschiedliche Farben in einem RGB-Bild dargestellt werden können, hängt von der Größe des Speichers, welche man für einen Farbanteil verwendet, ab. Üblicherweise wird jedem Farbanteil 8 Bit zur Verfügung gestellt, resultierend daraus kann ein Bild aus ungefähr 16 Millionen unterschiedlichen Farben bestehen.

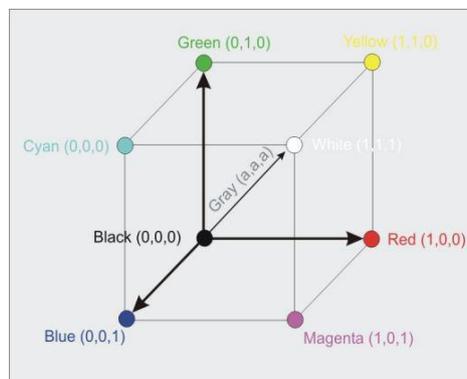


Abbildung 2: Darstellung des RGB Spektrums als dreidimensionaler Würfel

In der Abbildung 3 wird das RGB-Modell an Hand eines dreidimensionalen Würfels repräsentiert. Hier lassen sich die zu erzeugenden Farben anschaulich ablesen. Um zum Beispiel die Farbe Gelb zu erhalten, muss man die Farben Rot und Grün auf eins setzen und die Farbe Blau auf null. (6)

3.1.3 Graustufenbilder

Als Graustufenbilder, die im Druckbereich als Halbtonbild benannt sind, werden Bilder bezeichnet, die Farben enthalten, welche aus dem Mischen von weiß und schwarz resultieren können. Die Anzahl der verschiedenen Farben, die auf solch einem Bild gezeigt werden können, hängt von dem verwendeten Speicher eines Pixels ab. Um nur die Farben schwarz und weiß zu repräsentieren reicht es ein Bit pro Pixel zu verwenden. Heutzutage werden üblicherweise pro Pixel 8 Bit verwendet, wodurch ein Graustufenbild aus 256 verschiedenen Graustufen bestehen kann, wobei für die Farbe Schwarz der Wert 0 und für die Farbe Weiß 255 beträgt. (7)

3.1.4 Umwandlung eines RGB-Bildes in ein Graustufenbild

Die Umwandlung von einem RGB-Bild in ein Graustufenbild beschäftigt sich mit dem Problem der Berechnung eines einzigen Graustufenwertes pro Pixel aus den drei Werten eines RGB-Bildes. Eine logische Schlussfolgerung wäre es den Durchschnitt der drei Werte zu berechnen um einen neuen Wert für das Graustufenbild zu verwenden.

$$\text{Grauwert} = \frac{1}{3} * \text{Rot} + \frac{1}{3} * \text{Grün} + \frac{1}{3} * \text{Blau}$$

Diese Formel wird in der Praxis jedoch nicht angewandt, da das menschliche Auge verschiedene Farben verschieden intensiv wahrnimmt. Die Farbe Blau wird von Menschen stärker wahrgenommen als Rot und Grün. Somit ergibt sich eine neue Formel, die für die Umwandlung von einem RGB-Bild in ein Graustufenbild von der Computer Vision benutzt wird: (6)

$$\text{Grauwert} = 0,299 * \text{Rot} + 0,587 * \text{Grün} + 0,144 * \text{Blau}$$

3.2 Gesichtserkennung

Im vorherigen Abschnitt wurden die Grundlagen der Darstellung von Bildern auf einem Computer erläutert. Nachfolgend werden die Grundlagen für das Erkennen von Gesichtern, die in der entwickelten Anwendung benutzt werden, erklärt.

3.2.1 Schwierigkeiten

Es gibt einige Aspekte, die das maschinelle Erkennen von Gesichtern in einem Foto erschweren können. Diese Probleme können auf der einen Seite durch die Person, die fotografiert wird, hervorgerufen werden, auf der anderen Seite jedoch auch durch die Eigenschaften des zu analysierenden Fotos.

Damit ein Algorithmus Gesichter mit einer hohen Genauigkeit erkennen kann, ist es wichtig, dass das ganze Gesicht deutlich erkennbar ist. Das teilweise Verdecken von Gesichtsmarkmalen kann dazu führen, dass eine computerbasierende Gesichtserkennung fehlschlägt. Dieses kann zum Beispiel durch das Tragen eines Schales, welcher den Mund verdeckt, herbeigeführt werden. Da viele Algorithmen Gesichter an Hand von den hell und dunkel Kontrasten in einem Gesicht analysieren, kann das Tragen einer Sonnenbrille, die die Lichtverhältnisse in der Augenpartie verändert, zudem problematisch werden. Das Ändern der Gesichtsmarkmale, sowie der Gesichtsform, kann außerdem den Algorithmus fehlschlagen lassen. Diese Veränderung kann zum Beispiel durch das Schließen der Augen, jedoch aber auch durch das weite Öffnen des Mundes hervorgerufen werden. Weiterhin ist in manchen Algorithmen die Ausrichtung des Kopfes ein weiterer wichtiger Indikator ob ein Gesicht zu erkennen ist. Manche Algorithmen können nur dann das Gesicht einer Person erkennen, wenn diese frontal in die Kamera blickt. Die unterstehende Abbildung 3 zeigt diese verschiedenen Problemfälle. (8)



Abbildung 3: Beispiele für Fotos, die die Gesichtserkennung erschweren

Es gibt jedoch auch Fotos, in denen das Gesicht einer Person deutlich zu erkennen ist, die Eigenschaften des Fotos jedoch die Erkennung fehlschlagen lässt. Die Eigenschaften, die bewirken, dass ein Gesicht nicht erkannt wird, sind eine schlechte Belichtung und die Orientierung eines Fotos. Wird eine Kamera beim Fotografieren schief gehalten, kann es unter Umständen dazu führen, dass das Gesicht zu weit aus der Vertikalen gedreht ist und dadurch der Algorithmus ein falsches Ergebnis liefert. Durch unzureichende Lichtverhältnisse können einzelne Gesichtsm Merkmale dunkler oder heller erscheinen und nicht mehr als solche wahrgenommen werden. Abbildung vier zeigt unzureichende Lichtverhältnisse auf. (8)



Abbildung 4: unzureichende Lichtverhältnisse für die Gesichtserkennung

3.2.2 OpenCV

OpenCV ist eine open Source Library, die sich auf die Bildverarbeitung und das maschinelle Lernen im Bereich der Computer Vision spezialisiert hat. Die erste Version ist im Jahre 2006 herausgegeben wurden. Die aktuelle Version der Library ist 3.4.1. Mit über 2500 Algorithmen, mit denen man zum Beispiel in Echtzeit Gesichter erkennen, wie auch wiedererkennen kann, ist die Library auch im kommerziellen Bereich ein beliebtes Hilfsmittel. Unter anderem wird sie von Firmen wie Google und Yahoo benutzt. OpenCV ist in der Programmiersprache C++ implementiert, stellt jedoch auch für Python, Java und MATLAB Interfaces zur Verfügung. Sie unterstützt folgende Betriebssysteme: Windows, Linux, Android und Mac OS. (9)

3.2.3 Viola-Jones Algorithmus

Der Viola-Jones Algorithmus wurde 2001 von Paul Viola und Michael Jones in dem Aufsatz "Rapid Object Detection using a Boosted Cascade of Simple Features" veröffentlicht. Er ist ein lernender Algorithmus, der durch seine Schnelligkeit und seiner hohen Erfolgsrate beim Erkennen von Mustern in Bildern zur Berühmtheit wurde. Die Idee von Viola und Jones besteht aus drei Teilen. Der Erste ist das Umwandeln von Bildern in ein Integralbild (Erklärung siehe unten), welches die Verarbeitung von Bildern effizienter macht. Der zweite Teil ist ein lernender Algorithmus, der sich an AdaBoost anlehnt und eine kleine Anzahl an Features, welche Strukturen eines zu suchenden Objektes repräsentieren, aus einer großen Menge raussucht, um aus diesen effiziente Klassifikatoren zu erzeugen. Die

erfolgsversprechenden Klassifikatoren fasst der Algorithmus im nächsten Teil zu Kaskaden zusammen. Der Algorithmus stellt sicher, dass in diesen Kaskaden nur noch Klassifikatoren vorhanden sind, die Features enthalten, welche auf das gesuchte Objekt hindeuten. Man kann somit davon ausgehen, dass die Klassifikatoren, die nicht in der Kaskade enthalten sind auch nicht zu dem gesuchten Objekt gehören. Im Folgenden werden die unterschiedlichen Teile näher beschrieben. (10)

Features

Die Objektsuche klassifiziert die Bilder an Hand von Features, welche Muster und Strukturen in dem gesuchten Bild darstellen. Die Features werden durch die Anzahl der Rechtecke gekennzeichnet. Ein wichtiger Vorteil, die Bilder an Hand von Features anstelle von Pixel zu untersuchen, ist die Geschwindigkeit. Die in diesem Verfahren verwendeten Features lehnen sich an den Haar-Basis Funktionen an.

Für dieses Verfahren werden folgende drei Features, mit verschieden vielen Rechtecken, verwendet, wobei Feature A und B Repräsentationen eines zwei-Rechteck Features sind:

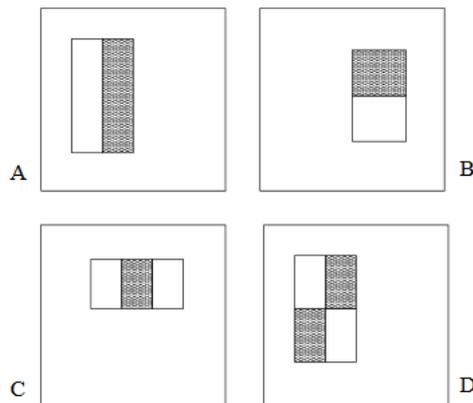


Abbildung 5: Features in dem zu untersuchenden Bild

Die Features werden auf unterschiedliche Positionen des Bildes gelegt. Dabei steigt die Größe der Features, um Muster verschiedener Größen erkennen zu können. Die Anfangsgröße der Features ist 24x24 Pixel. Durch das Subtrahieren und Addieren der

unterschiedlichen Rechtecke miteinander, ergibt sich für jeden Bereich des Bildes einen Wert, welcher die Helligkeitsdifferenz widerspiegelt.

Der Wert der zwei-Rechteck Features(A und B in Abbildung 5) besteht aus der Differenz der Summen der Pixel der beiden Rechtecke. Die beiden Rechtecke sind in diesem Feature gleich groß und sind entweder gegenüberliegend oder untereinander angeordnet. Die Differenz zwischen der Summe von dem Rechteck in der Mitte mit den Summen von den Rechtecken außen, ergeben bei dem drei-Rechteck Feature(C in Abbildung 5) den gesuchten Wert. Der Wert des vier-Rechteck Features setzt sich aus der Differenz der gegenüberliegenden Rechtecke zusammen. Durch die Ergebniswerte können in der Gesichtserkennung die unterschiedlichen Gesichtsmerkmale, wie zum Beispiel die Augen identifiziert werden, da die Augen einer Person im Verhältnis zu ihrer Nase und Wangen dunkler sind. (10)

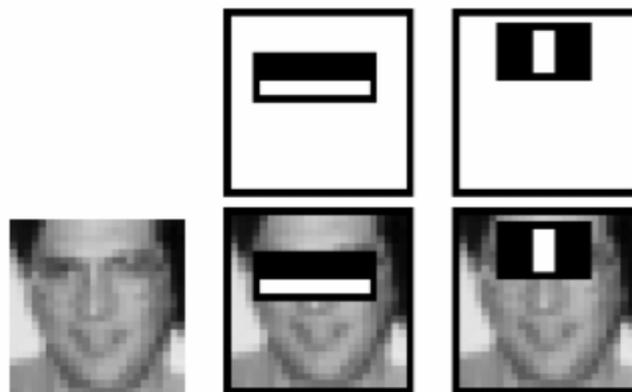


Abbildung 6: Identifizierung der Augen an Hand von Features

Integralbilder

Um die Rechteck Features effizient einzusetzen, wird das zu untersuchende Bild in ein Integralbild umgewandelt. Dadurch müssen die Pixel in einem Feature Suchfenster nicht immer wieder neu berechnet werden, welches Rechenaufwand spart. Um ein Integralbild zu erhalten, werden wenige Operationen pro Pixel benötigt. Diese Operationen werden auf einem Graustufenbild(siehe 3.1.3), welches aus dem Originalbild generiert wurde, angewendet. Die Umwandlung eines Graustufenbildes in ein Integralbild liegt folgender mathematischer Berechnung zugrunde, wobei $ii(x, y)$ das Integralbild darstellt. (10)

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y')$$

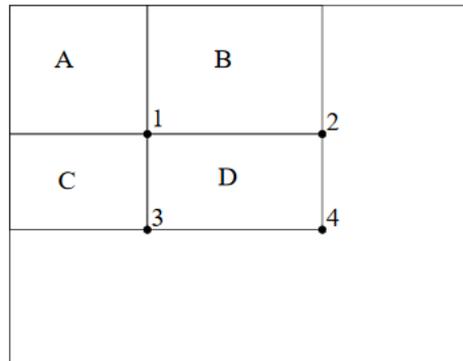


Abbildung 7: Bestimmung von der Pixelsumme innerhalb eines Rechteckes

In der Abbildung sieben kann man erkennen, wie die Berechnung der Pixelsummen für ein positioniertes Rechteck innerhalb eines Features zustande kommt. Hierzu wird eine vier-Array Referenz verwendet. Um die Pixelsumme von dem Array D zu berechnen, werden zuallererst die Werte für die Positionen 1,2,3 und 4 ermittelt. Für die Position 1 ist die Pixelsumme A, für die Position 2 ist sie A + B, für die Position 3 ist sie A + C und für die Position 4 ist sie A + B + C + D. Somit ergibt sich für das Array D die Formel: $1 + 4 - (2 + 3)$.

Um die Pixelsumme eines Rechteckes zu berechnen, werden folgende zwei rekursive Formeln benutzt, wobei $ii(x, y)$ das Integralbild und $i(x, y)$ das Originalbild beschreibt:

$$\begin{aligned} s(x, y) &= s(x, y - 1) + i(x, y) \\ ii(x, y) &= ii(x - 1, y) + s(x, y) \end{aligned}$$

In diesen Formeln, beschreibt $s(x, y)$ die ansteigende Spaltensumme der Pixel. $s(x, -1) = 0$ und $ii(-1, y) = 0$ werden als Rekursionsabbruch definiert.

Da bei den zwei-Rechteck Features die angeordneten Rechtecke auch nebeneinander liegen können, wird eine sechs-Array Referenz benötigt. Eine acht-Array Referenz wird im Falle der drei-Rechteck Features benötigt und bei einem vier-Rechteck Feature das Bild in neun-Array aufgeteilt. (10)

Die lernende Klassifikationsfunktion

Der Ansatz von Viola und Jones benutzt den AdaBoost Algorithmus, welcher ungenaue(schwache) Klassifikatoren kombiniert um einen genauen(starken) Klassifikator zu bilden. In dem Fall von dem Algorithmus von Viola und Jones sind die schwachen Klassifikatoren die Features, die dazu gebraucht werden Strukturen eines zu suchenden Objektes in einem Bild zu finden. Der Vorteil von dem Benutzen des AdaBoost Algorithmus ist es, dass diese schwachen Klassifikatoren, welche eine zu ungenaue Klassifikation der vorhandenen Trainingsdaten erreicht, selektiert. Würde dieses nicht geschehen wäre der Rechenaufwand um jedes Feature, in unterschiedlicher Position und Skalierung, zu untersuchen, zu hoch. Bei einem Suchfenster von 24x24 Pixel wären es mehr als 180 000 zu untersuchende Features. Bei dem AdaBoost Algorithmus können verschiedene Gewichtungen auf einzelne Trainingsdaten gelegt werden. Hierdurch können Bilder, in denen es schwerer ist Gesichter zu erkennen stärker gewichtet werden, so dass schneller unbrauchbare Features identifiziert werden können.

Ein schwacher Klassifikator kann durch folgende Formel ausgedrückt werden, wobei x für das zu untersuchende Suchfenster des Bildes, f für das Feature, p für die Polarität zur Steuerung des Vergleichsoperators und Θ für den Schwellenwert steht. (10)

$$h_j(x) = \begin{cases} 1, & \text{if } p_j f_j(x) < p_j \Theta_j \\ 0, & \text{otherwise} \end{cases}$$

Anhand des berechneten Schwellenwert entscheidet der AdaBoost Algorithmus welcher Klassifikator der Beste beim Klassifizieren, ob ein Gesicht im Bild vorhanden ist, ist.

Um die Features zu untersuchen, werden zwei verschiedene Klassen von Trainingsdaten verwendet. Positive Trainingsdaten, die ein Gesicht auf einem Foto abbilden, und negative Daten, auf dessen Bilder keine Gesichter vorhanden sind.

Nachfolgend wird der AdaBoost Algorithmus aufgeführt: (10)

- Given example images $(x_1, y_1), \dots, (x_n, y_n)$ where $y_i = 0, 1$ for negative and positive examples respectively.
- Initialize weights $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ for $y_i = 0, 1$ respectively, where m and l are the number of negatives and positives respectively.
- For $t = 1, \dots, T$:
 1. Normalize the weights,

$$w_{t,i} \leftarrow \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}}$$
 so that w_t is a probability distribution.
 2. For each feature, j , train a classifier h_j which is restricted to using a single feature. The error is evaluated with respect to w_t , $\epsilon_j = \sum_i w_i |h_j(x_i) - y_i|$.
 3. Choose the classifier, h_t , with the lowest error ϵ_t .
 4. Update the weights:

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$$
 where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.
- The final strong classifier is:

$$h(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{otherwise} \end{cases}$$
 where $\alpha_t = \log \frac{1}{\beta_t}$

Abbildung 8: Der AdaBoost Algorithmus, welcher je Durchgang einen der 180 000 Features untersucht

Der Adaboost Algorithmus setzt sich aus vier Schritten zusammen.

1. Schritt:
 - Übergeben von Trainingsdaten, die entweder das gesuchte Muster enthalten, oder nicht.
2. Schritt:
 - Versehen der schwachen Klassifikatoren mit Gewichten

3. Schritt:
Die Gewichte werden für die Klassifikatoren neu bestimmt. Das Gewicht, welches die Fehlerrate des Ergebnisses senkt, wird für einen Klassifikator ausgewählt.
4. Schritt:
Ein starker Klassifikator wird aus den schwachen Klassifikatoren gebildet.

Kaskaden Klassifikator

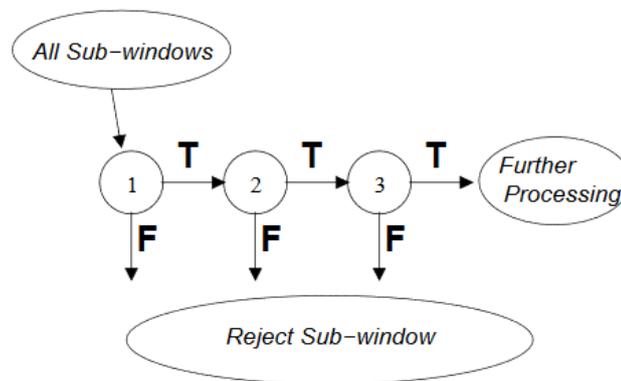


Abbildung 9: Eine Kaskade nach Viola und Jones

Die neunte Abbildung zeigt einen Algorithmus zum Erstellen einer Kaskade von Klassifikatoren, die die Performance zum Erkennen erhöht und gleichzeitig den Rechenaufwand senkt. Das Ziel hierbei ist es, kleinere Klassifikatoren, die die negativen Suchfenster identifizieren und somit fast alle positiven Instanzen finden, zu erzeugen. Nachdem einfachere Klassifikatoren die negativen Suchfenster ausgeschlossen haben, werden komplexere Klassifikatoren eingesetzt um das Resultat zu optimieren. Eine Kaskade ist hierbei ein Entscheidungsbaum. Wie in Abbildung neun aufgezeigt, untersucht der erste Klassifikator das Suchfenster. Ist sein Ergebnis positiv, wird der nächste Klassifikator eingesetzt um das Suchfenster zu analysieren. Wenn ein Klassifikator in dieser Kaskade ein negatives Ergebnis zurückliefert, scheidet das ganze Suchfenster aus und wird nicht weiter untersucht.

Auf der Abbildung steht das „T“ für true, welches ein positives und „F“ für false, welches ein negatives Ergebnis beschreibt.

Trainieren einer Kaskade von Klassifikatoren

Um die Kaskade von Klassifikatoren zu trainieren müssen zwei Aspekte berücksichtigt werden, der Rechenaufwand und die Genauigkeit des Ergebnisses. Um das beste Ergebnis zu erzielen, müssen Klassifikatoren mit möglichst vielen Features untersucht werden. Dieses führt zu einem sehr hohen Rechenaufwand. Deshalb wird in der Praxis ein Kompromiss zwischen den genannten Aspekten geschlossen.

3.3 Emotionserkennung

Im vorherigen Abschnitt wurde die Idee von Viola und Jones beschrieben um ein Gesicht in einem Bild zu identifizieren. Im Folgenden werden eine API und eine Technik zum Auffinden von Emotionen in Gesichtern beschrieben und miteinander verglichen.

3.3.1 Microsoft Emotion API

Microsoft stellt seit Oktober 2017 im Rahmen seiner Face API eine API zur Emotionserkennung auf Fotos zur Verfügung. Die API-Schnittstelle nimmt ein Foto als Eingabe entgegen und interpretiert dieses Foto mit Hilfe eines Cloud-basierten Emotions-Algorithmus. Um diesen Service nutzen zu können benötigt man einen Subscription Key, den man kostenlos nach der Registrierung auf der Microsoft Azure Webseite erhält. In einem REST-Aufruf, welches im folgenden Unterkapitel erläutert wird, übergibt man das zu analysierende Foto, dieses kann man entweder als URL oder als Byte Array übertragen. Weiterhin wird das gewünschte Antwortformat, wie zum Beispiel JSON, und der Subscription Key gesetzt. Befindet sich keine Person auf dem übermittelten Foto, erhält man eine leere Antwort. Wurden Gesichter auf dem Foto identifiziert, wird dieses auf dem Foto gekennzeichnet, zudem wird die Auswertung der Emotionen als Array aller Gesichter, geordnet nach der

Gesichtsgröße, zurückgeliefert. Diese API identifiziert die Primäremotionen und noch zwei weitere auf Fotos, da diese interkulturell identisch sind. Somit ist es nicht relevant aus welchem kulturellen Umfeld die zu analysierenden Personen stammen. (11)

3.3.2 Representational State Transfer

Eine Softwareanwendung verrichtet in den meisten Fällen ihre Aufgabe nicht alleine, sondern benötigt oftmals Informationen anderer Anwendungen. Um einen solchen Informationsaustausch zu gewährleisten, ist eine Kommunikationsmöglichkeit zwischen verschiedenen Anwendungen erforderlich.

Roy T. Fielding untersuchte in seiner Dissertation "Architectural Styles and the Design of Network-based Software Architectures" (12) ob die Architektur des Internets auf Anwendungen übertragbar ist. In seiner Arbeit stellte er den Architekturstil „Representational State Transfer“, kurz „REST“ vor.

Dieser neue Architekturstil unterscheidet sich von seinen Vorgängern, zum Beispiel SOAP, in vielerlei Hinsicht. Beispielweise werden bei REST die eigenen Funktionalitäten direkt auf Methoden von HTTP abgebildet, sodass HTTP nicht zum Tunneln einer prozeduralen Datenaustauschverbindung genutzt wird. Dabei ist REST nicht an ein bestimmtes Protokoll gebunden, wird aber überwiegend über HTTP betrieben. (13)

Jeder RESTful Web Service muss, nach Roy T. Fielding, folgende fünf Prinzipien erfüllen:

Adressierbarkeit

Jedes Dokument, im Folgendem als Ressource bezeichnet, muss über eine eindeutige ID (URI) erreichbar sein. Hierbei werden die Ressourcen direkt adressiert, sodass im Gegensatz zu anderen Technologien nicht deren Funktionen, denen Daten als Parameter übermittelt werden, im Vordergrund stehen.

Einheitliches, knappes Interface

In einem RESTful Web Service gibt es keine speziellen Methoden für Aktionen. Ein solcher Service bietet ausschließlich einfache Methoden an, die auf das unterliegende HTTP Protokoll abgebildet werden.

Ressourcen/ Repräsentationsorientierung

Mittels geeigneter Repräsentation, beispielweise JSON oder XML, wird ein Zustandsaustausch der Ressourcen gewährleistet. Hierbei bietet der Web Service Ressourcen zur Änderung, Abfrage etc. an.

Statuslosigkeit

Um die Anwendungen leichter und besser zu skalieren, werden auf Sitzungsinformationen auf der Serverseite verzichtet. Der Status der Anwendung befindet sich somit auf der Clientseite, dieses hat zur Folge, dass der Client alle benötigten Daten übermitteln muss.

HATEOAS

Die Abkürzung HATEOAS steht für „Hypermedia as the Engine of Application State“. Hierbei wird Bezug auf die Hypermedia-Strukturen des Internet genommen, welche dazu beitragen, dass die Clients von Web Services in gewissen Grenzen als endliche Automaten aufgebaut werden können, welche durch die serverseitigen Antworten erlernen, welche weiteren Möglichkeiten der Service anbietet. Um dieses zu erreichen, werden Links in die übertragenden Repräsentationen eingebettet, die von den Clients zur Navigation durch die Anwendung benutzt werden. (13)

3.3.3 Eigenfaces-Algorithmus

Ein Verfahren, welches man für das Erkennen von Emotionen in Gesichtern verwenden kann, ist der Eigenfaces-Algorithmus. Das Verfahren der Eigenfaces wurde im Jahr 1991 von Matthew Turk und Alex Pentland in ihrem Aufsatz „Eigenfaces for Recognition“ entwickelt, um Gesichter in Bildern zu erkennen. (14) Das Ziel der Eigenfaces ist es aus einem Set von Bildern, die das Gleiche abbilden, in unserem Fall eine bestimmte Emotion, ein Durchschnittsbild zu erzeugen. Um zu entscheiden ob ein zu analysierendes Bild das bestimmte Muster aufweist, wird die Differenz der Matrix des Durchschnittsbildes und des zu analysierenden Bildes genutzt um einen Differenzvektor der beiden Matrizen zu erzeugen. Wie das Verfahren im Einzelnen arbeitet, wird im folgendem erläutert.

Als erster Schritt werden die vorhandenen Bilder, die das gleiche Muster enthalten, in Graustufenbilder umgewandelt.

Als zweiter Schritt wird aus jedem Bild mit $K \times R$ Pixel ein K^R -dimensionaler Vektor. Aus dem Set von N Trainingsbildern wird darauffolgend ein Durchschnittsbild mit folgender Formel berechnet: (15)

$$\psi = \frac{1}{N} \sum_{m=1}^N \tau_m$$

ψ : Durchschnittsbild

τ_m : Vektor des Trainingsbildes

N : Anzahl der Trainingsbilder

Weitergehend wird das Durchschnittsbild mit allen Trainingsbildern zu einem Differenzvektor subtrahiert:

$$\phi_m = \tau_m - \psi$$

ϕ_m : Differenzvektor des Trainingsbildes

τ_m : Vektor des Trainingsbildes

ψ : Durchschnittsbild

Im Folgendem werden alle Differenzvektoren zu einer Matrix A konkateniert:

$$A = [\phi_1, \phi_2, \dots, \phi_N]$$

Die Kovarianzmatrix C der Matrix A wird gebildet:

$$C = AA^T$$

Als Nächstes werden die Eigenwerte und Eigenvektoren von der Kovarianzmatrix C gebildet.

Ein Eigenvektor \vec{x} einer Matrix A ist ein Vektor, der seine Richtung, durch das Multiplizieren mit der Matrix, nicht verändert, sondern nur gestreckt wird. Der Streckungsfaktor des Eigenvektors wird als Eigenwert λ bezeichnet.

$$A * \vec{x} = \lambda * \vec{x}$$

Um die Eigenwerte für die Matrix zu ermitteln, wird zuerst das charakteristische Polynom gebildet, woraufhin die Nullstellen dessen die Eigenwerte darstellen.

Das charakteristische Polynom $X_A(\lambda)$ für eine gegebene Matrix A:

$$A = \begin{pmatrix} 1 & 5 \\ 3 & 0 \end{pmatrix}$$

$$X_A(\lambda) = \lambda^2 - \lambda - 15$$

Die Nullstellen des charakteristischen Polynoms lauten:

$$x_1 = -3,4 \quad x_2 = 4,4$$

x_1 und x_2 sind die Eigenwerte der Matrix A. Um den dazugehörigen Eigenvektor \vec{x} zu berechnen, wird der Vektor gesucht, welcher mit der Matrix A multipliziert das Produkt eines Eigenwertes mit dem Vektor ergibt:

$$\vec{x} * A = \lambda * \vec{x}$$

Um diese Gleichung lösen zu können gibt es mehrere Verfahren, zum Beispiel das Additionsverfahren oder auch den Gauß-Algorithmus. Im folgendem wird das Additionsverfahren angewendet.

$$E_A(\lambda) = \begin{pmatrix} (1 - \lambda) & 5 \\ 3 & (0 - \lambda) \end{pmatrix}$$

Für $x_1 = -3,4$ ergibt sich die folgende Matrix:

$$E_A(-3,4) = \begin{pmatrix} (1 - (-3,4)) & 5 \\ 3 & (0 - (-3,4)) \end{pmatrix}$$

Für $x_2 = 4,4$ ergibt sich die folgende Matrix:

$$E_A(4,4) = \begin{pmatrix} (1 - 4,4) & 5 \\ 3 & (0 - 4,4) \end{pmatrix}$$

Darauffolgend werden die zwei Matrizen $E_A(4,4)$ und $E_A(-3,4)$ in ein Gleichungssystem umgewandelt und null gesetzt.

Folgendes Gleichungssystem ergibt sich für die Matrix $E_A(-3,4)$:

$$\begin{aligned} 0 &= ((1 - (-3,4)) * x + 5 * y) \\ 0 &= (3 * x + (0 - (-3,4)) * y) \end{aligned}$$

Für $E_A(4,4)$ ergibt sich das Gleichungssystem:

$$\begin{aligned} 0 &= ((1 - 4,4) * x + 5 * y) \\ 0 &= (3 * x + (0 - 4,4) * y) \end{aligned}$$

Werden diese beiden Formeln nach x und y aufgelöst, bilden diese die Eigenvektoren der Matrix A.

Im nächsten Schritt werden die Eigenvektoren mit den höchsten Eigenwerten ausgewählt, denn diese enthalten die markantesten Informationen des Bildes. Die Vektoren mit niedrigen Eigenwerten können vernachlässigt werden, da diese nur eine geringe Anzahl der Features beschreiben. Um zu entscheiden ob ein übergebenes Testdatum in die Klasse der Trainingsdaten einzuordnen ist und somit ein gesuchtes Muster erkannt wurde, wird der euklidische Abstand berechnet und ein Schwellenwert für die Distanz der Trainingsdaten zu dem Testdatum eingeführt.

$$d_i = \min | |\Omega - \Omega_i| |$$

d_i : Distanz zwischen den Trainingsdaten und dem Testdatum

Ω : Vektor des Testdatums

Ω_i : Vektor des Durchschnittsbildes der Trainingsdaten

Um ein noch sichereres Ergebnis zu erhalten, werden die Eigenvektoren gewichtet um einen Durchschnittsvektor der Trainingsdaten zu erhalten.

Das Benutzen des Eigenfaces-Algorithmus führt zu einer hohen Richtigkeit der Mustererkennung mit einem, in Vergleich zu anderen Algorithmen, geringerem

Rechenaufwand. Jedoch können auch hier Schwierigkeiten, auf Grund der Bildqualität und verdeckten Gesichtsmarkmalen entstehen, wie in Kapitel 3.2.1 beschrieben worden. (15)

4 Konzept

Nachdem die Grundlagen für die Emotionserkennung in Fotos erläutert wurden, beschreibt dieses Kapitel den ersten Entwurf und das finale Konzept der Anwendung, die in dem Rahmen dieser Arbeit entwickelt wurde.

4.1 Technische Voraussetzungen und Daten

Um die Anwendung auf einem Smartphone zu benutzen, sind zwei Voraussetzungen zu erfüllen.

Zum einen muss das Handy mit dem Internet verbunden und zum anderen ein Konto bei dem Play Store vorhanden sein.

Da die hier entwickelte Anwendung Schnittstellen von dem OpenCV Framework benutzt, muss auf dem Endgerät die OpenCV App, die kostenlos aus dem Play Store heruntergeladen werden kann, installiert sein. Die Internetverbindung wird zudem, in dem finalen Konzept, für einen REST-Aufruf benötigt.

Die Anwendung wurde mit Hilfe eines Samsung Galaxy S6, mit der Android-Version 7, entwickelt und auf diesem getestet.

4.2 Erster Entwurf

Das nachfolgende Unterkapitel beschreibt die erste Idee für die Umsetzung einer emotionserkennenden Anwendung.

4.2.1 Aufbau

Eine mögliche Umsetzung für eine Anwendung, die Emotionen in Gesichtern erkennen kann, besteht aus den Verfahren, die im Kapitel zwei beschrieben wurden. Als erster Schritt müssen die Gesichter auf einem gegebenen Foto erkannt werden. Hierfür wird der Viola-Jones Algorithmus verwendet, da dieser eine hohe Erfolgsrate und einen geringen Rechenaufwand aufweist.

Wurden die Gesichter identifiziert, wird an Hand des Eigenfaces-Algorithmus eine Emotionserkennung durchgeführt. Dieser Algorithmus benötigt Trainingsdaten um das Erkennen eines Musters, welches für jede Testgruppe bestimmt wird, zu identifizieren. Die Programmiersprache Java wird zur Entwicklung der Anwendung benutzt.

4.2.2 Datenbank der Trainingsdaten

Um ausreichend viele Trainingsdaten für das Erlernen der Identifikation von verschiedenen Emotionen zu erhalten, wurde die Datenbank „Extended Cohn-Kanade Dataset“ benutzt, welche die acht primär Emotionen auf 593 Fotos mit 123 verschiedenen Personen abbildet.

Den einzelnen Fotos wurden die darauf gezeigten Emotionen zugeordnet.

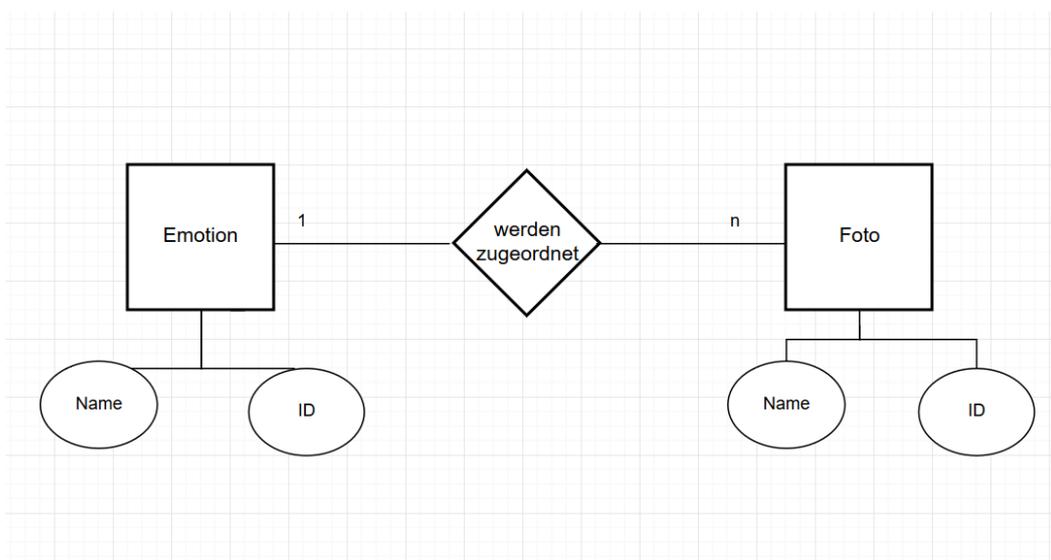


Abbildung 10: Entity-Relationship Diagramm

In diesem Entity-Relationship Diagramm sind jeweils die IDs die primary Keys. Eine Emotion, zum Beispiel die Emotion „Überraschung“, kann vielen Fotos zugeordnet werden. Jedoch kann ein Foto nur genau eine Emotion abbilden.

4.2.3 Komponenten

Gesichtserkennung

Um ein Gesicht in einem Foto zu identifizieren, wird, wie auch in dem finalen Konzept, das Framework „OpenCV“ verwendet. OpenCV stellt eine XML-Datei zur Verfügung, die die Kaskaden zum Erkennen eines Gesichtes bereitstellt. Hierbei ist es nicht entscheidend, ob ein farbiges- oder Graustufenbild übergeben wird, da die Methode des Frameworks die übergebenen Bilder in ein Graustufenbild konvertiert.

Emotionserkennung

Nach dem erfolgreichen Identifizieren eines Gesichtes wird die Emotionserkennung angewandt. Hierfür wird die Datenbank und das Framework „OpenCV“ genutzt. Das Framework stellt Methoden für das Anwenden eines Eigenfaces-Algorithmus zur Verfügung.

Um diese Methoden nutzen zu können, wird eine CSV-Datei mit den Trainingsdaten und zugehörigen Emotionen erwartet. Mit dem Befehl „read_csv(fn_csv, images, labels)“ wird der Pfad zu der CSV-Datei und zwei Vektoren, die Trainingsdaten und Emotionen, zugeordnet. Darauffolgend werden alle Trainingsbilder auf die gleiche Größe formatiert. Im nächsten Schritt wird der Eigenfaces-Algorithmus trainiert:

```
Ptr<FaceRecognizer> model = createEigenFaceRecognizer();  
model->train(images, labels);
```

images: Vektor der Trainingsdaten

labels: Vektor der Emotionen, dargestellt als Integer Werte

Hierbei wird zuerst ein Eigenfaces-Algorithmus erzeugt und diesem dann die Trainingsdaten mit den dazugehörigen Emotionen, welche als Integer Werte repräsentiert werden, übergeben.

Mit der folgenden Methode wird ein Testdatum ausgewertet und die resultierende Emotion zurückgegeben:

```
int predictedLabel = model->predict(testSample);
```

testSample: Testdatum

predictedLabel: Integer Wert der gefundenen Emotion

Das Framework stellt zudem Methoden für das Anzeigen der Eigenwerte und Eigenvektoren zur Verfügung.

4.2.4 Implementierung

Da OpenCV die Methode zum Erzeugen eines Eigenfaces-Algorithmus nur in der Programmiersprache Python zur Verfügung stellt und auch kein anderes geeignetes Framework gefunden wurde, welches eine Schnittstelle mit der Programmiersprache Java bietet, wurde das Framework Jython verwendet.

Jython ist eine Java-Implementierung der Programmiersprache Python, mit Hilfe dessen Python Code aus jeder Java-Umgebung ausgeführt werden kann.

Nach der Implementierung mit Jython, wurde festgestellt, dass die Anwendung nicht stabil läuft, weshalb der erste Entwurf verworfen und ein finales Konzept entwickelt wurde (siehe Kapitel 4.3).

4.3 Finales Konzept

In dem folgenden Kapitel werden der Aufbau und die Komponenten der im Rahmen dieser Bachelorarbeit entwickelten Anwendung erläutert.

4.3.1 Aufbau

Die Anwendung wurde mit der Programmiersprache Java entwickelt.

Der Grundaufbau dieser Anwendung orientiert sich an dem Prinzip des Model View Controllers.

Bei diesem Architekturmuster werden drei Aspekte der Software getrennt:

Das Model

Das Model enthält die Geschäftslogik und die bearbeiteten Daten. Im Fall der hier entwickelten Anwendung besteht das Model aus den Klassen „EmotionRecognition“, „Emotions“, „FaceDetection“, „Photo“, die Methoden zum Identifizieren von Gesichtern und deren Emotionen bereitstellen und diese Informationen in einem geeigneten Format zur Verfügung stellen.

Die View

Die View dient zur Kommunikation mit dem Benutzer der Anwendung. Diese Komponente nimmt die Benutzereingaben, wie zum Beispiel Eingaben und Kommandos, entgegen, sie ist aber nicht für deren Verarbeitung zuständig. Wenn der Benutzer in Interaktion mit der Anwendung tritt, registriert die Komponente „activity_emption.xml“ diese und ruft eine Methode in der Klasse „MainActivity“ auf.

Der Controller

Der Controller ist für die Kommunikation zwischen dem Model und der View zuständig. Er verknüpft die Benutzereingaben mit der Geschäftslogik in dem Model. Die Klasse „HandleRequests“ ist in dieser Anwendung der Controller, welcher von der View die angeforderten Benutzereingaben an das Model weitergibt, damit diese die Kommandos verarbeitet.

Dieses Architekturmuster wurde erstmals in den 1970er beschrieben und garantiert eine Wiederverwendbarkeit, hohe Flexibilität für Änderungen und Erweiterungen der einzelnen Komponenten. (13)

4.3.2 Komponenten

Die entwickelte Anwendung lässt sich in vier Teile klassifizieren. Als erstes wird die Anwendung initialisiert und alle nötigen Bestandteile geladen. Als nächster Schritt wird das von dem Benutzer hochgeladene Foto auf Gesichter analysiert. Diese Anwendung berücksichtigt nur Fotos mit einem einzigen Gesicht. Als dritter und letzter Schritt wird eine Emotionserkennung mit Hilfe eines REST-Aufrufes durchgeführt.

Für die Kommunikation der Klassen untereinander wird eine weitere Komponente zur Verfügung gestellt.

Initialisierung der App

Als erstes muss die Anwendung initialisiert werden. Die Klasse „InitializeApp“ befasst sich mit dieser Aufgabe. Sie ist dafür zuständig, dass das Framework OpenCV initialisiert und die richtigen Komponenten geladen werden. Damit OpenCV eine Gesichtserkennung durchführen kann, wird in diesem Schritt die Datei „haarcascade_frontalface_alt.xml“, die die Kaskade für das Erkennen von Gesichtern in Fotos beinhaltet, in eine Instanzvariable geladen. Weiterhin wird die Erlaubnis für das Lesen und Schreiben von externem Speicher erteilt, damit die Anwendung auf das Dateisystem des Handys zugreifen darf um Fotos aus dem Fotoalbum des Benutzers benutzen zu können. In dieser Klasse werden auch die benötigten Informationen für den REST-Aufrufs der Microsoft API gespeichert.

Gesichtserkennung

Um eine Emotionserkennung durchführen zu können, benötigt man ein Gesicht einer Person in dem übergebenen Foto. Die Klasse „FaceDetection“ ist zuständig für die Identifikation eines Gesichtes. Mit Hilfe des OpenCV Frameworks werden die Fotos analysiert.

Da das übergebene Foto in dem Format Bitmap übergeben wird, das OpenCV Framework jedoch ein Drawable erwartet, wurde eine zusätzliche Klasse implementiert um zwischen die Fotos zu konvertieren. Weiterhin wird diese Klasse in der Emotionserkennung benötigt, um für den REST-Aufruf aus den Fotos ein Byte Array zu erzeugen.

Emotionserkennung

Die Emotionserkennung wird mit Hilfe eines REST-Aufrufes zu einer Microsoft Anwendung durchgeführt. Die Klasse „EmotionRecognition“ erzeugt einen http-Client um den REST-Aufruf zu realisieren. Die Antwort der Microsoft Anwendung kommt in Form eines JSON zurück. Mit dem Weitergeben eines Fotos an Microsoft, wird automatisch dazu eingewilligt, dass Microsoft dieses benutzen darf.

Da die zurückgelieferte JSON-Datei sehr viele verschiedene Kriterien beinhaltet, wird hier nur der relevante Teil dargestellt:

```
"emotion": {  
    "anger":0,  
    "contempt":0,  
    "disgust":0,  
    "fear":0,  
    "happiness":1,  
    "neutral":0,  
    "sadness":0,  
    "surprise":0  
},
```

Die übermittelten Emotionen werden mit Zahlen von 0 bis 1 belegt. Eine 1 steht hierbei für 100% Übereinstimmung. Somit wird eine JSON-Datei, wie oben abgebildet (Happiness:1) bei einer zufriedenen oder auch lächelnden Person zurückgeliefert. Zu den sechs primären Emotionen wurden bei dieser Schnittstelle noch die Emotionen „neutral“ und „contempt“ (Zufriedenheit) hinzugefügt. Die „EmotionRecognition“ Klasse verarbeitet das zurückgelieferte Ergebnis und stellt es mit Hilfe der Klasse „Emotions“ in passender Form dar.

Kommunikation zwischen den einzelnen Klassen

Die Klasse „MainApplication“ ist die Verbindung von der View „activity_emotion.xml“ zu den Klassen der Anwendung. Wird von dem Benutzer ein Button betätigt, wird eine Methode in dieser Klasse aufgerufen. Mit Hilfe der Klasse „HandleRequest“ werden die Informationen der Benutzereingabe an die passende Klasse übermittelt, die Antworten dieser empfangen und gegebenenfalls Änderungen der View Elemente veranlasst.

4.3.3 Oberfläche

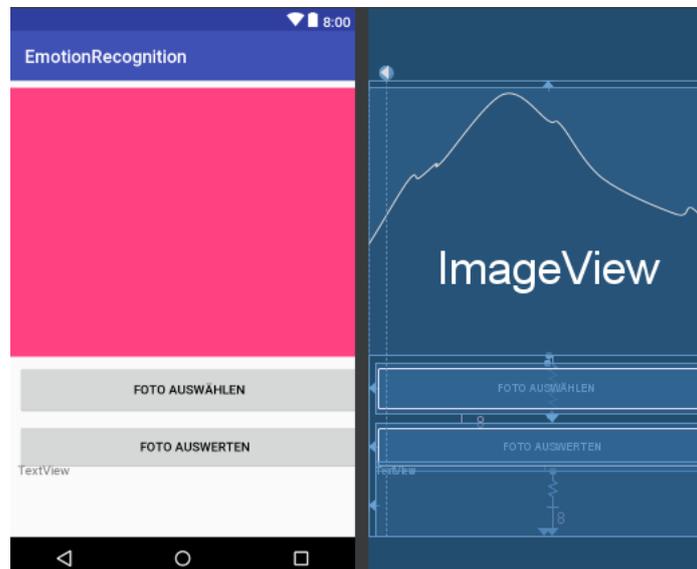


Abbildung 11: Oberfläche der Anwendung

Die Oberfläche der entwickelten Anwendung, siehe Abbildung 11, ist übersichtlich und simpel gehalten. Sie besteht aus einer ImageView, zwei Buttons und einer TextView.

Die ImageView ist nach dem Starten der Anwendung rosafarben gefüllt. Hat der Anwender ein Foto aus seiner Foto-Galerie ausgewählt, erscheint dieses in der ImageView. Nachdem das Foto auf vorhandene Gesichter untersucht wurde, werden die gefundenen Gesichter mit einem farbigen Viereck umrandet.

Die zwei Buttons haben folgende Aufgaben:

Der obere Button wird für das Auswählen eines Fotos genutzt. Nachdem der Anwender diesen Button betätigt hat, öffnet sich seine Foto-Galerie, woraufhin er aus dieser ein zu analysierendes Foto auswählen kann.

Der untere Button ist für das Auswerten des Fotos zuständig. Durch das Klicken auf den Button wird zuerst der Viola-Jones Algorithmus ausgelöst um vorhandene Gesichter zu identifizieren, darauffolgend wird die Emotionserkennung durchgeführt.

Die TextView blendet, nach Ausführung der Analyse, das Ergebnis ein. Dieses kann entweder positiv, also eine gefundene Emotion, oder auch negativ, dass keine passende Emotion gefunden wurde, sein.

5 Fazit

Dieses Kapitel gibt einen Überblick über die Möglichkeiten für eine Weiterentwicklung einer emotionserkennenden Anwendung und fasst diese Arbeit zusammen

5.1 Zusammenfassung

Am Anfang dieser Arbeit wurden die zwei verschiedenen Arten von Emotionen erläutert und es wurde verdeutlicht, weshalb nur die Primäremotionen, die kulturkreisunabhängig sind, in einer Anwendung, untersucht werden können.

Des Weiteren wurden die Grundlagen, die benötigt werden um Bilder in der Computer Vision darzustellen und zu verarbeiten, erläutert. In diesem Schritt wurden die Vorgehensweisen des Graustufenbildes und des RGB-Modells und die Umwandlung von dem einen in das andere Format beschrieben.

Weiterhin wurden die Probleme beim automatisierten Erkennen von Gesichtern in Fotos dargestellt. Hierbei können nicht nur die Umgebung, in der ein Foto aufgenommen wird, sondern auch das Verhalten der aufgenommenen Person zu Schwierigkeiten führen.

Ein möglicher Lösungsansatz für das Erkennen von Gesichtern mit Hilfe eines Computers ist der Viola-Jones Algorithmus, der das Erkennen von Mustern in Bildern realisiert. In diesem Schritt(Kapitel 3.2.3) wurden Features und Kaskaden Klassifikatoren veranschaulicht und ihre Aufgabe in dem Erkennen von Mustern verdeutlicht.

In diesem Zusammenhang wurde das OpenCV-Framework vorgestellt, welches sich mit den Problemstellungen in der Computer Vision beschäftigt.

In dem nächsten Kapitel wurde ein Prinzip zum Erkennen von Emotionen in Gesichtern beschrieben. Um eine Emotionserkennung zu implementieren ist ein möglicher Ansatz der lernende Eigenfaces-Algorithmus. Dieser erkennt, wie auch der

Viola-Jones Algorithmus Muster in unbekanntem Bildern wieder. Der Algorithmus kann das Erkennen von Emotionen erlernen. Hierfür werden dem Algorithmus Trainingsdaten, in Form von Fotos mit Gesichtern, die dieselbe Emotion enthalten, zur Verfügung gestellt. Der Algorithmus bildet aus den gegebenen Gesichtern jeder Emotion eine Matrix eines Durchschnitts-Gesichts der gegebenen Emotion, welches er mit einem zu untersuchenden Foto vergleicht. Ein anderer möglicher Ansatz für das Lösen der Problemstellung ist ein Framework von Microsoft. Dieses Framework stellt eine REST-Schnittstelle zur Verfügung, welche die Emotionserkennung durchführt. Sowohl das Framework als auch das Prinzip von REST wurden in diesem Kapitel vorgestellt.

Als nächstes wurde der erste Entwurf für die Realisierung einer Anwendung, die Emotionen auf Fotos erkennt, erläutert. Hierbei wurde der Viola-Jones- und Eigenfaces-Algorithmus angewandt. Weiterhin wurde eine Datenbank, die Fotos mit Gesichtern unterschiedlicher Primäremotionen bereitstellt, verwendet.

Das finale Konzept der in dem Rahmen dieser Arbeit entwickelten Anwendung wurde als letztes aufgeführt. In diesem Konzept wurde der Viola-Jones Algorithmus für das Erkennen von Gesichtern auf Fotos verwendet. Mit Hilfe der Microsoft Schnittstelle wurde das Erkennen von Emotionen realisiert.

5.2 Ausblick

Die im Rahmen dieser Arbeit entwickelte Anwendung hat ein hohes Potential für die Weiterentwicklung.

Zum einen kann sie für den kommerziellen Gebrauch für Marketing Abteilungen zum Einsatz kommen. Wird, anstatt von Fotos, zum Beispiel ein Videostream ausgewertet, welcher die Mimik von Personen, die auf eine gezeigte Werbung reagieren, aufnimmt, kann dieses für Unternehmen wichtige Informationen bereitstellen.

Beispielsweise kann eine Tierschutzorganisation mit Hilfe einer solchen Anwendung sicherstellen, dass ihre gezeigten Werbungen, welche oftmals Bilder von gequälten Tieren zeigen, bei den Zuschauern die gewünschten Emotionen, wie Trauer, Ärger, Furcht/Angst, Überraschung oder auch Ekel, auslösen.

Zum anderen kann sie jedoch auch durch eine Weiterentwicklung zu einem Lernspiel für Menschen, die an einer Krankheit wie Autismus leiden, werden.

Autisten fällt es schwer ihre eigenen Gefühle nach außen zu zeigen, resultierend daraus können sie die Mimik anderer Personen oftmals nicht eindeutig einordnen. Psychologen sind jedoch der Ansicht, dass Autisten das Erkennen von Gefühlen bei anderen Personen erlernen können. (16)

Für diesen Zweck kann ein Foto von dem Anwender übergeben werden, woraufhin die Anwendung mehrere Vorschläge, wie sich die Person auf dem Foto fühlt, aufzeigt. Der Anwender kann aus diesen Vorschlägen jene, welche seiner Meinung nach die Emotion auf dem Foto am besten widerspiegelt, auswählen. Nach der getroffenen Auswahl des Anwenders, wird die richtige Antwort angezeigt, woraufhin der Anwender seine Kompetenz der Emotionserkennung verbessern kann. Damit schon Kinder, die noch nicht lesen und schreiben können, diese Anwendung benutzen können, können die Vorschläge und auch die richtige Antwort von der Anwendung vorgelesen werden. Für diese Anwendung bietet sich beispielsweise die Java Speech API von Sun Microsystems an, die Sprachausgaben aus Java Anwendungen ermöglicht. (17)

6 Literaturverzeichnis

1. **Wittchen, Hans-Ulrich, Hoyer, Jürgen.** *Klinische Psychologie & Psychotherapie (Lehrbuch mit Online-Materialien)*. s.l. : Springer, 2011. 978-3-642-13018-2.
3. **Kessler, Henrik.** *Kurzlehrbuch Medizinische Psychologie und Soziologie*. s.l. : Thieme, 2015. 978-31-3-136423-4.
4. **Prof. Dr. Friauf, Eckhard.** *Die Wurzeln der Gefühle*. 2018.
5. **Schmidt-Atzert, Lothar, Peper, Martin and Stemmler, Gerhard.** *Emotionspsychologie*. s.l. : Kohlhammer, 2014. 978-3-17-020595-6.
6. **Burger, Wilhelm and Burge, Mark J.** *Principles of Digital Image Processing*. s.l. : Springer, 2009. 978-1-84800-191-6.
7. **Greve, Kai.** *Farbmodelle in der Praxis*. 2005.
8. **Rowley, Henry A., Baluja, Shumeet and Kanade, Takeo.** *Neural Network-Based Face Detection*. 1996.
9. **Team, OpenCV.** *OpenCV About*. 2018.
10. **Viola, Paul and Jones, Michael.** *Rapid Object Detection using a Boosted Cascade of Simple Features*. 2001.
11. **Corporation, Microsoft.** Microsoft Azures . [Online] <https://docs.microsoft.com/de-de/azure/cognitive-services/emotion/home>.
12. **Fielding, Roy T.** [Online] https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf.
13. **Weil, Dirk.** *Java EE 7*. s.l. : entwickler.press, 2015.
14. **Matthew, Turk and Pentland, Alex.** [Online] <http://www.cs.ucsb.edu/~mturk/Papers/jcn.pdf>.
15. **Thuseethan, S. and Kuhanesan, S.** Eigenface Based Recognition of Emotion Variant Faces. [Online] https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2752808.
16. **e.V., Autismus Westerwald - Mittelrhein.** Sozialverhalten. [Online] <https://www.autismus-wemi.de/erkennungszeichen/articles/sozialverhalten.html>.
17. **Inc., Sun Microsystems.** jsapi-doc. [Online] https://docs.oracle.com/cd/E17802_01/products/products/java-media/speech/forDevelopers/jsapi-doc/.

7 Abbildungsverzeichnis

Abbildung 1: Die Emotionsentstehung nach James und Lange	7
Abbildung 2: Darstellung des RGB Spektrums als dreidimensionaler Würfel	10
Abbildung 3: Beispiele für Fotos, die die Gesichtserkennung erschweren.....	13
Abbildung 4: unzureichende Lichtverhältnisse für die Gesichtserkennung	14
Abbildung 5: Features in dem zu untersuchenden Bild	15
Abbildung 6: Identifizierung der Augen an Hand von Features.....	16
Abbildung 7: Bestimmung von der Pixelsumme innerhalb eines Rechteckes	17
Abbildung 8: Der AdaBoost Algorithmus, welcher je Durchgang einen der 180 000 Features untersucht.....	19
Abbildung 9: Eine Kaskade nach Viola und Jones	20
Abbildung 10: Entity-Relationship Diagramm.....	29
Abbildung 11: Oberfläche der Anwendung.....	35

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, den _____