Bachelorarbeit

# Automatische Gesichtserkennung für Filmcharaktere

Universität Augsburg
Fakultät für angewandte Informatik

Lehrstuhl: Multimedia Computing
Prof. Dr. Rainer Lienhart

Daniel Lercher
Augsburg, 18.09.2006

# Zusammenfassung

Diese Arbeit stellt eine Methode vor, um Gesichter in Bilder/Videos zu erkennen. Dazu werden bereits vorhandene Verfahren und Algorithmen zur Detektion und anschließenden Klassifizierung von Gesichtern angewandt.
Im ersten Teil wird das Training von Detektoren für Gesichtsmerkmale behandelt. Dazu werden das Lernverfahren Boosting in Kombination mit Haar-like-Merkmalen sowie die dazu verwendeten Trainingsdaten und deren Vorverarbeitung vorgestellt. Nach einer experimentiellen Auswertung der gewonnenen Ergebnisse für die Detektoren geht es im zweiten Teil um die Erstellung von Signaturen, die es ermöglichen, einen sinnvollen Vergleich von zwei Gesichtern durchzuführen. Dabei werden die aus Filmen detekterten Gesichtsbilder durch eine Reihe von Vorverarbeitungsschritten in eine einheitliche Form gebracht. Optional kann die Signatur durch eine PCA in ihrer Dimension verringert werden. Das Ergebnis des Vergleichs zweier Signaturen ist die Distanz der beiden Gesichter zueinander. Dadurch ist es möglich Bilder mit vorher erstellten Modellen von Personen zu vergleichen und zuzuordnen. Abschließend wird die Gesichtserkennung experimentiell ausgewertet.

# Inhaltsverzeichnis

**1 Einleitung** **1**

**2 Detektion** **3**

2.1 Klassifizierungsmerkmale . . . . . 4

2.2 Klassifizierung . . . . . 6

2.3 Trainingsdaten . . . . . 8

2.3.1 Vorverarbeitung . . . . . 8

2.4 Training . . . . . 11

2.5 Auswertung . . . . . 12

**3 Gesichtserkennung** **15**

3.1 Signaturen . . . . . 15

3.1.1 Entfernung des Hintergrunds . . . . . 17

3.1.2 Helligkeitsnormalisierung . . . . . 17

3.1.3 PCA, Eigenfaces . . . . . 20

3.2 Distanzberechnung . . . . . 22

**4 Auswertung** **23**

4.1 Rangliste . . . . . 24

4.2 Klassifizierung . . . . . 28

**5 Zusammenfassung** **39**

Besonders danke ich Eva Hörster für die nette und kompetente Betreuung.

# Kapitel 1

# Einleitung

Die inhaltliche Auswertung von multimedialen Daten wie Bilder und Videos nimmt zunehmend an Bedeutung zu. Beispiele dafür sind die Erfassung von Texten sowie die hier behandelte Detektion und Klassifizierung bzw. Erkennung von Personen. Anwendungsgebiete können u.a. Suchmaschinen, Mensch-Computer-Schnittstellen oder Sicherheitssoftware für Zugangskontrolle darstellen.
Die automatische Personenerkennung in Bildern/Videos stellt eine schwierige Aufgabe dar, da eine Reihe von Effekten die Darstellung eines Gesichtes in einem Bild beeinflussen. Dazu zählen z.B. die Orientierung des Charakters, der Gesichtsausdruck, sowie die Verdeckung von Teilen des Gesichts durch Brillen oder andere Gegenstände. Des Weiteren haben die Beleuchtung der Person und die daraus resultierenden Schattierungen und Helligkeitsunterschiede große Auswirkungen auf die Darstellung. In Abb. 1.1 werden unterschiedliche Effekte veranschaulicht.

Abbildung 1.1: Beispiele für Faktoren, die die Darstellung eines Gesichtes in einem Bild beeinflussen: Orientierung, Gesichtsausdruck (a), Verdeckung (b), Beleuchtung (c).

Ziel dieser Arbeit ist es, Charaktere aus Videos automatisch zu erkennen. Abb. 1.2 liefert einen Überblick über das implementierte System. Diesem liegt der Ansatz von [1] zugrunde. Die Aufgabe des Systems besteht darin, ein - von den vorher angesprochenen Einwirkungen - normalisiertes Gesichtsbild (Signatur) zu generieren. Dazu werden zunächst die Einzelbilder eines Films durchlaufen und darin enthaltene Gesichter detektiert. Für jedes detektierte Gesicht wird dann eine Signatur erstellt. Hierfür sind einige Vorverarbeitungsschritte notwendig, die u.a. die Positionen diverser Gesichtsmerkmale (beide Augen und den Mund) benötigen. Die Signaturbilder können dann miteinander verglichen werden.

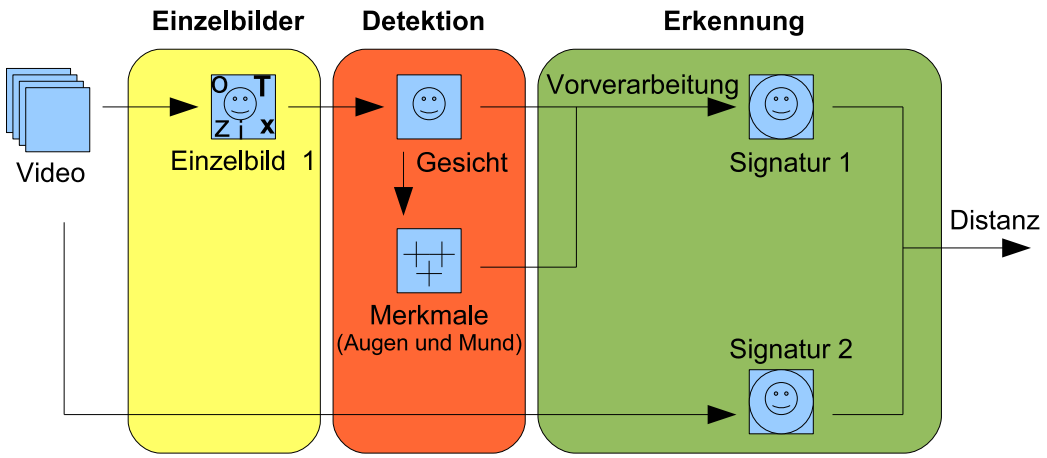


Abbildung 1.2: Überblick über das verwendete System. Signatur 2 wird auf dem gleichen Weg erstellt wie Signatur 1.

Die Arbeit gliedert sich wie folgt. In Kapitel 2 wird die Detektion der Gesichter und ihrer Merkmale behandelt. Neben der Theorie des benutzten Klassifizierungsalgorithmus Boosting wird auch auf dessen Training sowie die dazu verwendeten Trainingsdaten eingegangen. Das dritte Kapitel beschreibt die Gesichtserkennung. Hierzu gehören die Erstellung und der Vergleich von Signaturen. Das gesamte Gesichtserkennungssystem wird in Kapitel 4 experimentell ausgewertet. Kapitel 5 fasst die Erkenntnisse nochmal zusammen.

# Kapitel 2

# Detektion

Als Voraussetzung für die hier behandelte Personenerkennung werden Detektoren für das Gesicht sowie für die darin enthaltenen Merkmale Augen und Mund benötigt (siehe Abb. 2.1). Dies ist notwendig, damit die Gesichtsbilder in der

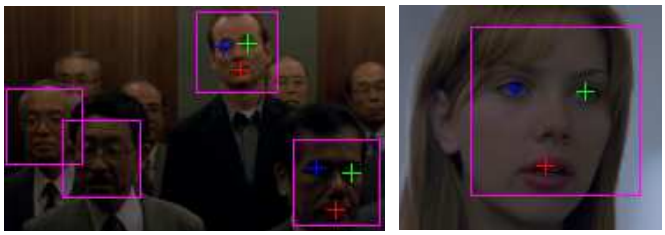

Abbildung 2.1: Beispiele für detektierte Gesichter (rosa). Darin gefundene Merkmale (Augen, Mund) sind blau, grün und rot markiert.

später folgenden Vorverarbeitung ausgerichtet, d.h. in eine einheitliche Position gebracht werden können. Es gibt verschiedene Ansätze für Detektoren. Beispiele dafür sind die Methode von Cristinacce *et al.* [2] oder das statistische Verfahren von Schneiderman *et al.* [7]. In [1] werden *Support Vector Machines* (kurz: *SVM*) benutzt.
In dieser Arbeit wird für die Detektion von frontalen Gesichtern der in OpenCV [5] enthaltene Detektor benutzt. Dieser implementiert die Methode von Lienhart *et al.* [3], welche eine Weiterentwicklung des von Viola *et al.* [9] vorgestellten Verfahrens ist. Es werden sog. *Haar-like*-Klassifizierungsmerkmale (siehe [3], [4]) in Kombination mit dem Lernverfahren Boosting verwendet. Zur Detektion der drei Gesichtsmerkmale linkes Auge, rechtes Auge und Mund wird jeweils ein eigener Detektor benötigt. Die Berechnung dieser Detektoren erfolgt analog zur Berechnung des Gesichtdetektors. Auch hier wird die Methode aus [3] angewendet. Eine

Implementierung dieser Methode findet sich ebenfalls in OpenCV. Sie wird in den Abschnitten 2.1 und 2.2 näher beschrieben. Die Berechnung der einzelnen Merkmals-Detektoren unterscheidet sich dann ausschließlich in den eingegebenen Trainingsdaten (siehe Abschnitt 2.3).
Im Folgenden werden diese drei Detektoren zusammengefasst und als Merkmalsdetektorsatz bezeichnet. Denkbar wären zudem weitere Detektoren für die Merkmale Nase, Ohren etc. Dazu müssten lediglich die Trainingsdaten angepasst werden.

## 2.1 Klassifizierungsmerkmale

Um einen Klassifizierer, in unserem Fall einen Merkmalsdetektor, berechnen zu können, benötigt der Lernalgorithmus positive und negative Beispiele, die sogenannten Trainingsdaten, für die zu detektierende Eigenschaft. Die positiven und negativen Trainingsdaten werden in unserem Fall aus Bildern, die das Merkmal enthalten bzw. nicht enthalten, mit Hilfe von Klassifizierungsmerkmalen erzeugt. Wie bereits erwähnt, werden hier die in [3] eingeführten Klassifizierungsmerkmalen verwendet (Abb 2.2). Die Merkmale bestehen jeweils aus zwei Rechtecken. Sie

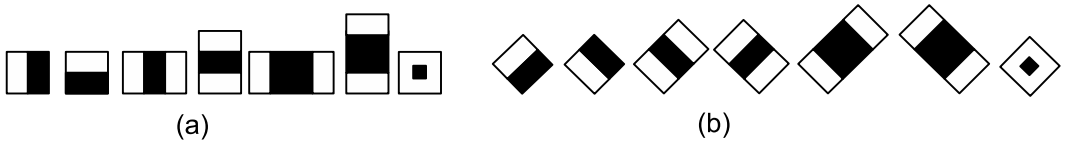


Abbildung 2.2: Klassifizierungsmerkmale: Prototypen für aufrechte (a) und 45°-gedrehte Rechtecke (b)

werden innerhalb der Abmessungen des Bildes beliebig skaliert und auf einen Bildbereich abgebildet. Die Summe der gesamten Pixelwerte innerhalb eines Rechtecks $r$ wird als $RecSum(r)$ bezeichnet. Die beiden Pixelsummen eines Klassifizierungsmerkmals werden je mit einem Gewicht $w_i \in \Re$ multipliziert:

$$feature_I = \sum_{i \in I = \{1,\ldots,N\}} w_i RecSum(r_i) \tag{2.1}$$

mit $N = 2$, $r_i$ sei ein beliebig großes Rechteck mit der Einschränkung, dass es innerhalb des Bildes sein muss und $RecSum(r_i)$ sei die Summe der gesamten Pixelwerte innerhalb von $r_i$.
Die Vorzeichen der beiden Gewichte eines Merkmals müssen unterschiedlich sein,

die Gewichtswerte sollen den Größenunterschied der beiden Rechtecke zueinander kompensieren.
Für ein Bild erhält man so einen - in Verhältnis zur Größe des Bildes stehend - großen Vektor an Summen von gewichteten Pixelwertsummen für diverse Klassifizierungsmerkmale. Auf viele Bilder angewandt erhält man eine große Matrix mit Daten, die als Eingabe für das Lernverfahren dient.
In [3] und [4] wird eine sehr schnelle Methode vorgestellt, um $RecSum(r)$ für ein beliebiges Rechteck $r$ zu ermitteln. Die Matrix $SAT(x, y)$ (*Summed Area Table*) speichert die Summe der Pixel im Bereich von $(0, 0)$ bis $(x, y)$ für die aufrechten Rechtecke ($x$ ist der Pixelwert für die Breite des Bildes, $y$ analog dazu für die Höhe):

$$SAT(x, y) = \sum_{x' \leq x, y' \leq y} I(x', y') \tag{2.2}$$

Die Berechnung erfolgt in einem einzigen Durchgang durch alle Breiten- und Höhenpixel von links nach rechts bzw. oben nach unten durch:

$$SAT(x, y) = SAT(x, y-1) + SAT(x-1, y) + I(x, y) - SAT(x-1, y-1) \tag{2.3}$$

$$\text{mit } SAT(-1, y) = SAT(x, -1) = SAT(-1, -1) = 0$$

Um die Pixelsumme für ein beliebiges Rechteck $r = (x, y, w, h)$, mit $P(x, y)$ als linken oberen Punkt und w, h als Breite bzw. Höhe, innerhalb des Bildes zu erhalten, sind vier Einträge von $SAT(x, y)$ relevant (siehe Abb 2.3):

$$\begin{aligned} RecSum(r) = {} & SAT(x-1, y-1) + SAT(x+w-1, y+h-1) \\ & -SAT(x-1, y+h-1) - SAT(x+w-1, y-1) \end{aligned} \tag{2.4}$$

Das gleiche gilt für die 45°-gedrehten Rechtecke. Die Matrix $RSAT$ (*Rotated Summed Area Table*) der Pixelsummen ist:

$$RSAT(x, y) = \sum_{y' \leq y, y' \leq y - |x - x'|} I(x', y') \tag{2.5}$$

Sie lässt sich analog zu 2.2 errechnen:

$$\begin{aligned} RSAT(x, y) = {} & RSAT(x-1, y-1) + RSAT(x+1, y-1) \\ & -RSAT(x, y-2) + I(x, y) + I(x, y-1) \end{aligned} \tag{2.6}$$

$$\text{mit } RSAT(-1, y) = RSAT(x, -1) = RSAT(x, -2) = 0$$
$$RSAT(-1, -1) = RSAT(-1, -2) = 0.$$

Die Pixelsumme für ein beliebiges 45°-gedrehtes Rechteck innerhalb des Bildes ergibt sich durch:

$$\begin{aligned} RecSum(r) = RSAT(x-h+w, y+w+h-1) + RSAT(x, y-1) \\ -RSAT(x-h, y+h-1) - RSAT(x+w, y+w-1) \end{aligned} \tag{2.7}$$

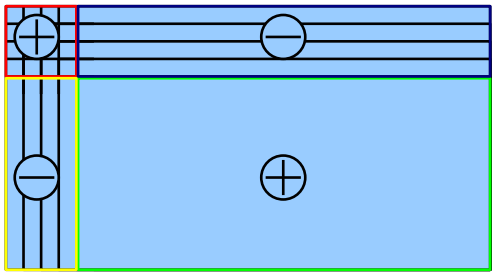

Abbildung 2.3: Berechnung der Pixelsumme für aufrechte Rechtecke anhand der SAT-Matrix. Der Wert für das grüne Rechteck ergibt sich aus dem Gesamtwert mit Abzug der Summe der Werte für das blaue und gelbe Rechteck und Hinzunahme des doppelt gezählten Bereichs (rot).

## 2.2 Klassifizierung

Der Lern-Algorithmus Boosting klassifiziert ausschließlich in zwei Klassen (binär). Man kann den Klassifizierer für diverse Lernprobleme verwenden. In dieser Arbeit soll daraus ein Detektor gebildet werden, dessen Aufgabe es ist, zu ermitteln, ob eine bestimmte Eigenschaft (hier: Gesichtsmerkmal) in Daten enthalten ist.
Die Theorie des Boosting besagt, dass mehrere *schwache* Klassifizierer stufenweise hintereinander geschaltet werden. Schwach bedeutet in dem Zusammenhang, dass die einzelnen Klassifizierer in Bezug auf die Trainingsdaten eine gering angesetzte Fehlerrate f (z.B. 0,5) nicht überschreiten, jedoch mindestens eine Trefferrate h (z.B. 0,95) vorweisen können. Nach $N$ Stufen ergeben sich Gesamtraten von $f^N$ bzw $h^N$. Die daraus entstandene Kaskade[1] bildet den entgültigen Klassifizierer (Abb. 2.4).
Dieser muss zunächst einmal trainiert werden. Das bedeutet, dass anhand von

[1] Da in der Arbeit nur eindimensionale Kaskaden verwendet wurden, wird nur darauf eingegangen. Für weiterführende Informationen siehe [3]

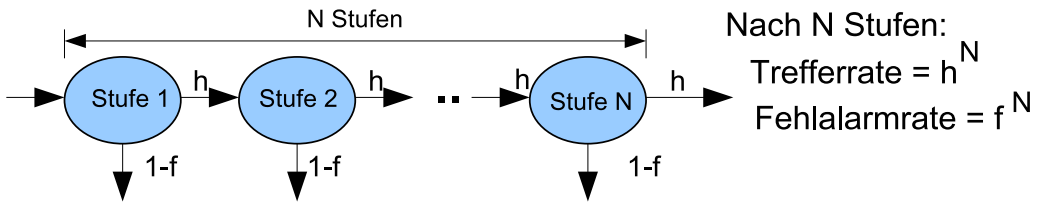


Abbildung 2.4: Kaskade von verstärkenden Klassifizierern

Traininsdaten Werte für die Gewichte der einzelnen Klassifizierungsmerkmale errechnet werden. Wie bereits erwähnt, muss für jede Stufe eine Trefferrate h und eine Rate f der falsch benannten Daten vorgegeben werden.
Für das Training gibt es verschiedene Boosting-Algorithmen. Beispiele sind *Discrete AdaBoost* (*DAB*), *Real AdaBoost* (*RAB*) und *Gentle AdaBoost* (*GAB*). Sie unterscheiden sich hauptsächlich in der Berechnung der Gewichte, die später den Detektor bilden, laufen jedoch nach dem gleichen Prinzip ab. Tabelle 2.1 beschreibt den Algorithmus für *Gentle AdaBoost* Dieser ist standardmäßig für das Training in OpenCV eingestellt und wird auch in dieser Arbeit verwendet.

1. Initialisierung der Gewichte:
   $w_i = 1/N, i = 1, .., N, N = \#_{Samples},\ F(x) = 0$
2. Wiederhole für m $= 1, .., M, M = \#_{Stufen}$
   a) Passe Regressionsfunktion $f_m(x)$ an mit den gewichteten kleinsten Quadraten von $y_i$ bis $x_i$ mit $w_i$
   b) Aktualisiere $F(x) \leftarrow F(x) + f_m(x)$
   c) Aktualisiere $w_i \leftarrow w_i e^{-y_i f_m(x_i)}$
   Renormalisiere $w_i$ so dass $\sum_i w_i = 1$
3. Ergebnis: $sign[F(x)] = sign[\sum_{m=1}^{M} f_m(x)]$

Tabelle 2.1: Gentle AdaBoost Algorithmus

Bei der Klassifizierung von neuen Daten werden alle Bereiche durchlaufen, die die gleiche Größe haben wie die Trainingsdaten. Diese sogenannten *Regions Of Interest* (kurz *ROI*) werden klassifiziert. Heraus kommt dabei ein Wert von $-1$ oder 1, je nachdem ob ein Object detektiert wurde oder nicht. Durch Skalierung des Klassifizierers können Objekte verschiedener Größe ausgemacht werden.

## 2.3 Trainingsdaten

Als Grundlage für die positiven Trainingsdaten, mit denen später das Training stattfindet, dienen die frontalen Gesichtsbilder der CMU-Datenbank[1] (Abb. 2.5), bei denen die Koordinaten für die Merkmale Augen, Nase und Mund bereits in einer Textdatei vorliegen. Diese Graustufen-Bilder variieren in Auflösung und

Abbildung 2.5: Beispiele für positive Trainingsdaten aus der CMU-Datenbank. (Die Zahl in den Bilder stellt den Verkleinerungsfaktor dar.)

Orientierung der Gesichter. Es kommen sowohl Einzelporträts als auch Gruppenfotos vor. Um einen unabhängig von Auflösung arbeitenden Detektor zu generieren, werden alle Bilder zusammengefasst. Diese Menge wird im Folgenden als $T_{CMU}$ bezeichnet. Die Anzahl der verwendeten Gesichter auf den Bildern beträgt 1320 ($|T_{CMU}| = 1320$). Die negativen Daten $T_{neg}$ bestehen aus hochauflösenden Natur- und Textbildern sowie Bildern von Büros und Gegenständen (Abb. 2.6). Dadurch ist eine Vielzahl von Variation innerhalb der Trainingsdaten gegeben. Durch entsprechend große Abmessungen kann die Anzahl der Elemente von $T_{neg}$ geringer sein als $|T_{CMU}|$, da der Trainingsalgorithmus in OpenCV die Bilder von $T_{neg}$ weiter aufteilt in kleinere Bilder gleicher Größe (bestimmt durch die uniforme Abmessung der positiven Daten, anhand denen das Training stattfindet).

### 2.3.1 Vorverarbeitung

Für das Training werden Bilder gleicher Größe benötigt, die das zu detektierende Merkmal enthalten. Um diese aus $T_{CMU}$ zu generieren sind einige Vorverarbei-

[1] http://vasc.ri.cmu.edu/idb/html/face/frontal_images

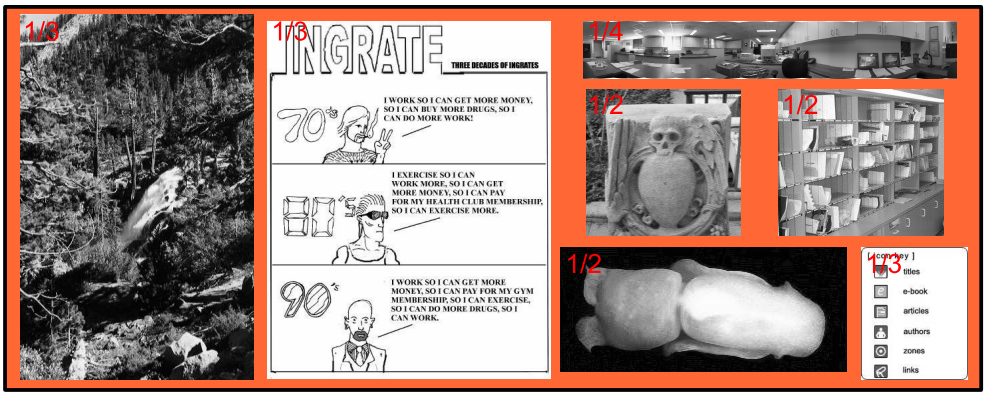


Abbildung 2.6: Beispiele für negative Trainingsdaten. (Die Zahl in den Bilder stellt den Verkleinerungsfaktor dar.)

tungsschritte notwendg. Diese Methode (Tabelle 2.2) ist angelehnt an die Vorverarbeitung von Rowley [6]. Alle Bilder in $T_{CMU}$ werden zunächst in die gleiche Position (Standardposition) gebracht, um die Variation der Merkmale in Ausrichtung, Position und Größe voneinander zu verringern. Das bedeutet, dass die beiden Augen und der Mund auf allen Bildern ungefähr die gleichen Koordinaten besitzen (Abb. 2.7). Die Transformationsmatrizen, anhand denen man die Koor-

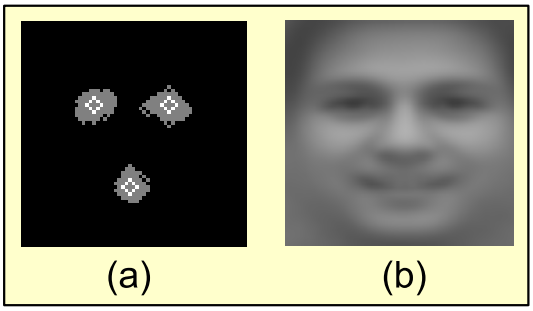


Abbildung 2.7: (a) Die Verteilung der in Standardposition (grau) transformierten Koordinaten (weiss) der Gesichtsmerkmale. (b) Mittelwertsbild aller verwendeten Gesichter aus der CMU-Datenbank.

dinaten $(x_i, y_i)$ der Gesichtsmerkmale $i$ (hier: Augen und Mund) in Standardposition $(x'_i, y'_i)$ bringt, beinhalten die Werte $a$ und $b$ für Rotation und Skalierung sowie $t_x$ und $t_y$ für Verschiebung. Diese berechnet man durch Lösen des Gleichungssystems:

1. Initialisiere leere Menge an Trainingsdaten für jedes Merkmal
2. Wiederhole für jedes Bild:
   a) Transformiere das Bild in die Standardposition
   b) Wiederhole für jedes Merkmal:
      1) Berechne die neuen Koordinaten für das Merkmal
      2) Wiederhole beliebig oft:
         a) Rotiere das Bild in kleinem Winkel (zufällig) um die neuen Koordinaten des Merkmals
         b) Skaliere das Bild mit kleinem Wert (zufällig)
         c) Verschiebe das Bild mit kleinem Wert (zufällig)
         d) Kopiere das Bild und schneide es auf feste Größe zu wobei die neuen Koordinaten den Mittelpunkt darstellen
         e) Füge Kopie zu Trainingsdaten des Merkmals hinzu

Tabelle 2.2: Algorithmus zur Generation von Trainingsdaten für die Merkmalsdetektoren aus markierten Gesichtsbildern.

$$\begin{pmatrix} x_1 & -y_1 & 1 & 0 \\ y_1 & x_1 & 0 & 1 \\ x_2 & -y_2 & 1 & 0 \\ y_2 & x_2 & 0 & 1 \\ \dots & & & \\ \dots & & & \\ x_i & -y_i & 1 & 0 \\ y_i & x_i & 0 & 1 \end{pmatrix} \begin{pmatrix} a \\ b \\ t_x \\ t_y \end{pmatrix} = \begin{pmatrix} x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ \dots \\ \dots \\ x'_i \\ y'_i \end{pmatrix} \tag{2.8}$$

Die einzelnen Bildpunkte $(x_j, y_j)$ eines Bildes werden nun auf ihre neue Position $(x'_j, y'_j)$ transformiert:

$$\begin{pmatrix} a & -b & t_x \\ b & a & t_y \end{pmatrix} \begin{pmatrix} x_j \\ y_j \\ 1 \end{pmatrix} = \begin{pmatrix} x'_j \\ y'_j \end{pmatrix} \tag{2.9}$$

Die Gesichter liegen nun in Standardposition vor.
Die Bilder für ein Merkmal $i, i \in \{linkesAuge, rechtesAuge, Mund\}$, werden als positive Trainingsdaten $T_i$ bezeichnet. Um eine größere und variablere Anzahl an Bildern zu erhalten, wurden die Gesichtsbilder (in Standardposition) erneut

mehrfach transformiert, diesmal mit kleinen zufällig generierten Werte für Rotation, Skalierung und Verschiebung innerhalb vorgegebener Intervalle. Somit kommt es zudem zu einer kontrollierten Varianz innerhalb der positiven Trainingsdaten (Abb. 2.8).

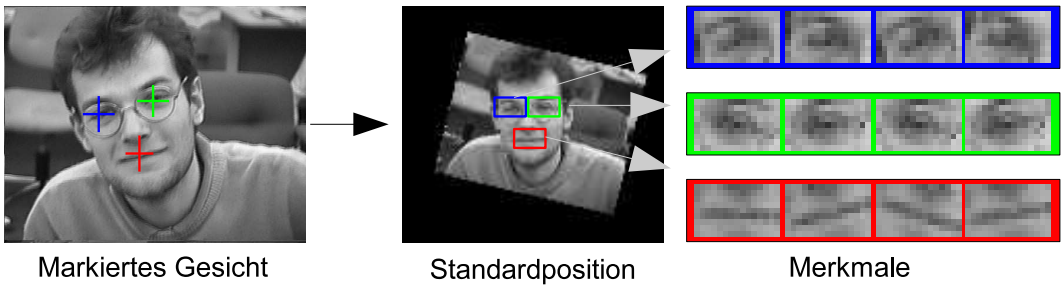


Abbildung 2.8: Das markierte Gesicht (links) wird zunächst in Standardposition gebracht (mitte). Durch mehrmaliges Transformieren erhält man für jedes Merkmal eine Reihe von unterschiedlichen Trainingsdaten (rechts).

Bei den verwendeten Merkmalen kann das Entfernen des Hintergrunds vernachlässigt werden, da sich sowohl Augen als auch Mund stets innerhalb des Gesichts befinden. Die negativen Trainingsdaten müssen nicht bearbeitet werden. Mit den generierten Trainingsdaten $T_i$ und den negativen Trainingsdaten $T_{neg}$ kann nun das Training stattfinden.

## 2.4 Training

Für das erfolgreiche Training eines Detektors sind vor allem die positiven und negativen Trainingsdaten von Bedeutung. Damit der Klassifizierer ein Merkmal richtig benennen kann, wird eine möglichst hohe Zahl an positiven Trainingsdaten benötigt. Ohne entsprechende Anzahl von negativen Daten würde der Detektor jedoch zusätzlich eine große Menge an falschen Merkmalen erkennen. Bei zu vielen wäre der Detektor übertrainert, d.h. zu stark an die Trainingsdaten gebunden. Daher spielt das Verhältnis eine tragende Rolle, welches sich meist nur empirisch ermitteln läßt. Des Weiteren kann man zwischen den o.g. verschiedenen Boosting-Algorithmen wählen sowie den zu benutzenden Boosting-Merkmalen. Tabelle 2.2 fasst die Parameter für das Training in OpenCV zusammen.

| Einstellung | Bedeutung |
|---|---|
| $n_{Stufen}$ | Anzahl der Stufen der Kaskade |
| $n_{pos}$ | Anzahl der benutzten positiven Trainingsdaten pro Stufe |
| $n_{neg}$ | Anzahl der benutzten negativen Trainingsdaten pro Stufe |
| $h_{min}$ | Gewünschte minimale Trefferrate für den Klassifzierer einer Stufe |
| $f_{max}$ | Gewünschte maximale Fehlerrate für den Klassifzierer einer Stufe |
| Modus | Zu verwendende Boosting-Merkmale |
| | (aufrechte, 45°-gedrehte oder alle Rechtecke) |
| Symmetrie | Symmetrie der Trainingsdaten. Ermöglicht Datenspiegelung. |
| Breite | Breite der Trainingsdaten |
| Höhe | Höhe der Trainingsdaten |
| Boost-Typ | Discrete AdaBoost, Real AdaBoost, LogitBoost |
| | oder Gentle AdaBoost |

Tabelle 2.3: Einstellungsmöglichkeiten für das Training des Boosting-Klassifizierers in OpenCV

## 2.5 Auswertung

In diesem Kapitel sollen die Detektoren für Augen und Mund experimentiell ausgewertet sowie der Einfluss einzelner Parameter bestimmt werden. Zunächst möchte ich kurz auf die Terminologie der Evaluierungen eingehen: Daten, die korrekt als positiv klassifiziert werden, werden *True Positives (TP)* genannt. Daten, die zwar positiv benannnt werden, jedoch das Objekt nicht beinhalten, werden als *False Positives (FP)* bezeichnet. Bei Daten, die der Detektor als negativ kennzeichnet, die allerdings das Objekt beinhalten, spricht man von *False Negatives (FN)*. Korrekterweise falsch klassifizierte Daten sind *True Negatives (TN)*. Um eine Aussage darüber zu machen, wie gut ein Detektor arbeitet, verwendet man die beiden Raten *Precision (P)* und *Recall (R)*. Precision sagt etwas über die Korrektheit der Klassifizierung aus und ergibt sich durch:

$$P = \frac{|TP|}{|TP| + |FP|} \tag{2.10}$$

Wie gut der Detektor bei erneutem Klassifizieren wäre, wird durch Recall ausgedrückt. Die Formel dazu lautet:

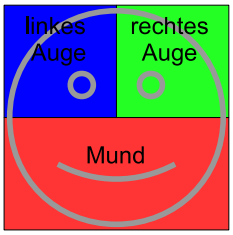


Abbildung 2.9: Klassifizierungskriterium: Es muss pro Merkmal mindestens ein Objekt in dem entsprechenden Bereich erkannt werden.

$$R = \frac{|TP|}{|TP| + |FN|} \tag{2.11}$$

Als Grundlage für die Evaluierung dienten die beiden Filme *Und täglich grüsst das Murmeltier* (USA 1993) und *Lost In Translation* (USA 2003) (im Folgenden als M und L bezeichnet). Das zweite Video hat eine schlechtere Qualität. Dies erkennt man daran, dass die Bilder allesamt viel dunkler und kleiner sind.
Es wurden verschiedene Detektorsätze trainiert, die je aus aus drei Detektoren für das linke und rechte Auge sowie den Mund bestanden. Innerhalb eines Satzes waren die Abmessungen der verwendeten Trainingdaten identisch.
Da die Bilder für die Erkennung vorverarbeitet und dazu die Koordinaten von allen drei Merkmalen benötigt werden, wurden nur Gesichter zugelassen, bei denen alle drei Merkmale in einem bestimmten Bereich (beim Mund die Mundregion etc.) gefunden worden. Die Regionen werden in Abb. 2.9 dargestellt. Zudem wurde nach einem detektierten Gesicht ein Sprung von zehn Einzelbildern innerhalb des Videos gemacht, damit die Menge der auszuwertenden Bilder nicht zu groß wird. Für die Auswertung der späteren Erkennung schränkt es zudem die Anzahl von nahezu gleichen Bilder ein.
Die Detektionsrate ist in Film M ziemlich gut, wobei sie in Film L ein eher schlechtes Ergebnis liefert. Das liegt bestimmt u.a. an der vorher angesprochenen schlechteren Qualität des Videos. Die Trefferrate liegt ziemlich hoch mit ungefähr 90%. Zum größten Teil wurde der Mund falsch detektiert.
Durch einen höheren Wert für die Anzahl der negativen Trainingsdaten werden weniger Gesichter detektiert. Wie bereits in Kapitel 3.4 angemerkt, werden die Detektoren durch zu viele negative Trainingsdaten übertrainiert und sind somit weniger gut was neue Daten angeht.
Durch Symmetrie beim Mund wird eine kleine Erhöhung der Erkennungrate be-

wirkt. Dies kommt daher, dass die Varianz innerhalb der Daten gesteigert wird. Der Mund kann ebenso gut spiegelverkehrt sein kann, ohne dass sich was Grundlegendes im Gesicht ändert.
In Tabelle 2.4 werden die Einstellungen und Ergebnisse der erstellten Detektorsätze zusammengefasst.

| Parameter | Satz A | Satz B | Satz C |
|---|---|---|---|
| $n_{Stufen}$ | 20 | 20 | 20 |
| $n_{pos}$ | 5280 | 5280 | 5280 |
| $n_{neg}$ | 5000 | 8000 | 8000 |
| $h_{min}$ | 0.998 | 0.998 | 0.998 |
| $f_{max}$ | 0.5 | 0.5 | 0.5 |
| Modus | alle Merkmale | alle Merkmale | alle Merkmale |
| Symmetrie | * | ** | * |
| Breite | 15 | 13 | 13 |
| Höhe | 9 | 7 | 7 |
| Boost-Typ | GAB | GAB | GAB |
| Film M | | | |
| $n_{Gesichter}$ | 750 | 750 | 750 |
| $n_{detektiert}$ | 529 | 437 | 445 |
| Detektionsrate | 0,705 | 0,583 | 0,593 |
| $\|TP\|$ | 501 | 416 | 426 |
| $\|FP\|$ | 28 | 21 | 19 |
| Trefferrate | 0,947 | 0,952 | 0,957 |
| Film L | | | |
| $n_{Gesichter}$ | 750 | 750 | 750 |
| $n_{detektiert}$ | 282 | 164 | 162 |
| Detektionsrate | 0,376 | 0,219 | 0,216 |
| $\|TP\|$ | 268 | 143 | 154 |
| $\|FP\|$ | 14 | 21 | 8 |
| Trefferrate | 0,950 | 0,872 | 0,951 |

* Augen: keine Symmetrie, Mund: Symmetrie
** Augen und Mund: keine Symmetrie

Tabelle 2.4: Ein Satz beinhaltet Detektoren für die Merkmale Augen und Mund. Die Größe und Anzahl sind für jedes Merkmal gleich.

# Kapitel 3

# Gesichtserkennung

Im vorigen Kapitel wurde die Detektion von Gesichtern bzw. Gesichtsmerkmalen behandelt. Ziel ist es nun, die detektierten Gesichter miteinander zu vergleichen. Zu Beginn der Arbeit wurde eine Reihe von Effekten erwähnt, die die Darstellung eines Gesichtes in einem Bild beeinflussen. Diese Auswirkungen sollen durch geeignete Vorverarbeitungsverfahren verringert werden. Nach dem Durchlaufen der Gesichtsbilder durch die einzelnen Vorverarbeitungsschritte werden aus den Bildern sogenannte Signaturen erstellt. Eine Signatur ist ein Vektor, der Werte enthält, die das Gesicht beschreiben. Daher stellen z.B. die vorverarbeiteten Bilder in Vektorform bereits Signaturen dar. Zusätzlich kann als weiterer Vorverarbeitungsschritt eine *Principal Component Analysis* (kurz PCA) [8] durchgeführt werden. Hierbei wird das vorverarbeitete Bild zunächst in Vektorform gebracht und dann mit Hilfe der PCA in einen niedrig dimensionalen Raum transformiert. Der resultierende Vektor kann dann als Signatur verwendet werden.
Um verschiedene Signaturen miteinander vergleichen zu können, d.h. auf Ähnlichkeit zu überprüfen, wird die Distanz zwischen ihnen berechnet. Das Ergebnis kann zur Erkennung benutzt werden. Eine vergleichende Auswertung der verschiedenen Verfahren der Signaturerstellung erfolgt in Kapitel 4.

## 3.1 Signaturen

Zur Erstellung der Signatur aus einem Gesichtsbild muss dieses eine Reihe von Vorverarbeitungsschritten durchlaufen. Als erstes wird das Gesicht in eine vorher festgelegte Standardpostion und -größe gebracht. Da in Kapitel 2.3 bereits hierauf in einem anderen Zusammenhang eingegangen wurde, wird der Prozess zur

Ausrichtung der Gesichter hier nur kurz zusammengefasst.
Um ein Gesicht geeignet auszurichten und zu skalieren, müssen die Koordinaten für die drei Merkmale linkes Auge, rechtes Auge und Mund bekannt sein. Mittels der Formel (2.8) ermittelt man dann die Matrix für die Transformierung eines Bildes in die vorher angesprochene Standardposition und Größe. Anhand dieser Matrix, die Werte für Rotation, Skalierung und Verschiebung enthält, wird das Bild nun transformiert. Durch eine geeignete Wahl der Standardkoordinaten lässt sich das Gesichtsbild für beliebige Größen anpassen. Da die Transformationsmatrix mittels der *Least Mean Square*-Methode berechnet wird, kann es zu einer Abweichung der transformierten Punkte von der idealen Position der Standardkoordinaten kommen (siehe Abb. 2.7). Da die Abweichung jedoch gering ist, kann dieser Effekt vernachlässigt werden.
Nachdem das Gesicht auf Standardposition und -größe gebracht wurde, wird mittels einer Bildmaske der Hintergrund entfernt (Schritt 2). Die Maske ist ein Binärbild, wobei ein Bereich das Gesicht darstellt und der andere Bereich den Hintergrund. Im dritten und letzten Schritt der Vorverarbeitung wird eine Helligkeitsnormalisierung durchgeführt, um die Auswirkungen auf das Gesicht, die sich durch die Art der Beleuchtung ergeben, zu verringern. Hierfür können zwei unterschiedliche Verfahren angewendet werden.
Die Pixelwerte der vorverarbeiteten Gesichtsbilder können direkt als Signatur verwendet werden. Eine weitere Möglichkeit besteht darin, den Vektor, der die Pixelwerte des vorverarbeiteten Bildes enthält, in der Dimension vorher zu verringern. Dies lässt sich u.a. durch die Verwendung einer PCA realisieren. Verfahren 1 verwendet für die Normalisierung eine lineare Funktion über die Pixelwerte. Bei der zweiten Methode werden dazu Gaussfilter benutzt. Beide Verfahren wurden implementiert und eine vergleichende Auswertung findet sich in Kapitel 4. Abb. 3.1 stellt die Verfahren zur Erstellung von Signaturen für Gesichtsbilder grafisch dar.
In den nächsten Abschnitten werden die beiden Schritte der Hintergrundentfernung und Helligkeitsnormalisierung nochmals detailiert erläutert.
Die Berechnung der PCA sowie die Erzeugung von Signaturen aus dem Pixelvektor eines Bildes über die PCA wird im Folgenden ebenfalls vorgestellt.

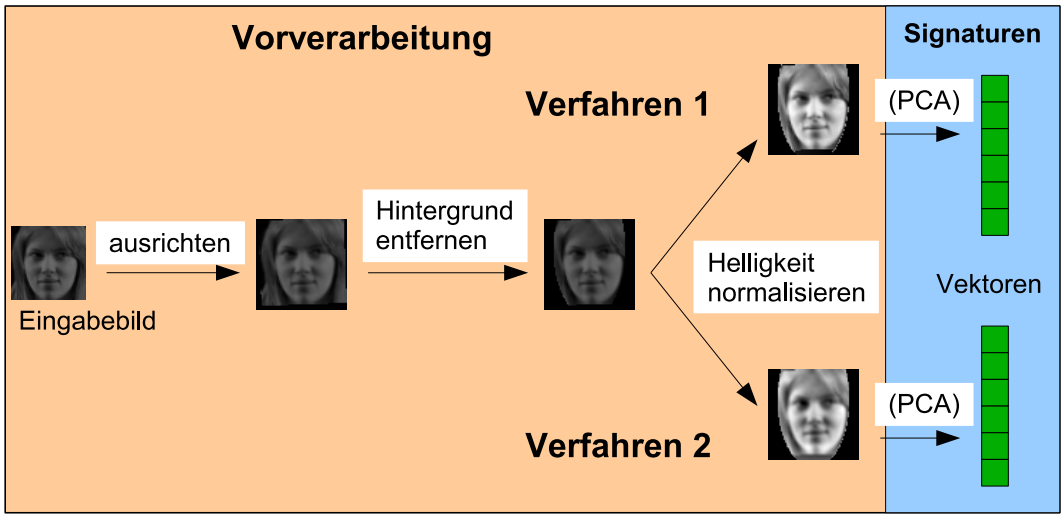


Abbildung 3.1: Überblick über die beiden verwendeten Verfahren zur Erstellung von Signaturen

### 3.1.1 Entfernung des Hintergrunds

Die detektierten und ausgerichteten Gesichter werden mittels einem Rechteck aus dem Bild ausgeschnitten. Dieses Rechteck ist im Idealfall durch die Gesichtsumrandungen begrenzt. Aufgrund der meist ovalen Form eines Gesichts enthält dieser Ausschnitt jedoch noch einen Hintergrundbereich, der nicht zum Gesicht gehört. Da die Signatur nur das eigentliche Gesichtsbild beschreiben soll, ist ein notwendiger Schritt in der Erstellung der Signaturen die Entfernung des Hintergrunds. Die einfachste Methode zur Entfernung des Hintergrunds ist die Anwendung einer Maske. Diese besteht aus einem binären Bild, das in zwei Bereiche unterschiedlicher Farbe aufgeteilt ist (normalerweise schwarz und weiss). Der weisse Bereich stellt den später sichtbaren Teil des Eingabebildes (der hoffentlich das detektierte Gesichtsbild enthält) dar. Bildpunkte, die im anderen Bereich liegen, werden mit einem einheitlichen Farbwert (hier: schwarz) versehen. Abb. 3.2 zeigt drei unterschiedliche Masken, die in der Arbeit verwendeten wurden. Für die weitere Verarbeitung ist nur der Bereich bzw. die Pixelwerte innerhalb des weissen Bereichs wichtig, der restliche Teil wird ignoriert.

### 3.1.2 Helligkeitsnormalisierung

Wie vorher bereits erwähnt, werden im Rahmen dieser Arbeit zwei verschiedene Verfahren zur Helligkeitsnormalisierung vorgestellt. Zunächst wird die von Row-

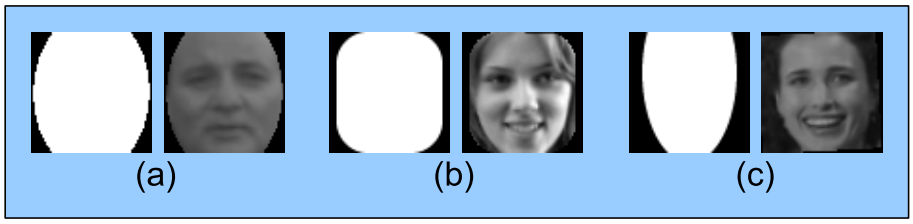


Abbildung 3.2: Beispiele für Masken zur Entfernung des Hintergrunds.

ley in [6] präsentierte Methode (im Folgenden als $H_1$ bezeichnet) besprochen. Diese wird in Abb. 3.3 veranschaulicht.

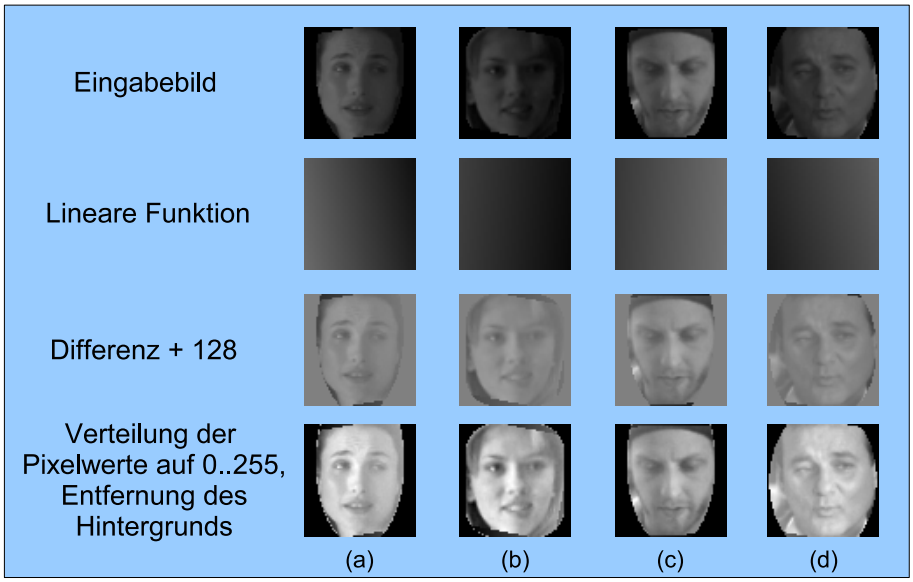


Abbildung 3.3: Helligkeitsnormalisierung über die lineare Funktion

Aus einem ausgerichteten Gesichtsbild wird die zugehörige lineare Funktion über die Pixelwerte $z$ der Bildpunkte $x, y$ ermitteln, die den Farbverlauf im hintergrundentfernten Bild am besten beschreibt:

$$z = a + bx + cy \tag{3.1}$$

Für ein Bild der Größe $w \times h$ bestimmt man $a, b, c$ durch die Lösung eines überdeterminierten Gleichungssystems:

$$(G^T * G)^{-1} * G^T * d = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \tag{3.2}$$

wobei

$$G = \begin{pmatrix} 1 & x_0 & y_0 \\ 1 & x_1 & y_1 \\ ... & & \\ 1 & x_m & y_m \end{pmatrix} \tag{3.3}$$

und

$$d = \begin{pmatrix} z_0 & z_1 & ... & z_m \end{pmatrix} \tag{3.4}$$

Dabei ist $m$ die Anzahl der Bildpunkte anhand denen die Werte für $a, b, c$ berechnet werden. Da nur die im Gesichtsbereich der Maske liegenden Pixel relevant sind, wird die Hintergrundmaske dazu benutzt, um diese auswählen zu können. Nun kann für jeden Bildpunkt der entsprechende Wert bestimmt werden. Das entstandene Bild zeigt den Farbverlauf.
Das generierte Farbbild wird nun vom eigentlichen Gesichtsbild subtrahiert. Das Ergebnis ist ein helligkeitsnormalisiertes Bild. Dieses enthält jedoch nur einen eingeschränkten Bereich an Farbwerten. Daher werden die einzelnen Pixelwerte $I(x, y)$ aus dem benutzten Grauwertbereich $\{I_{min}, ..., I_{max}\}$ abschließend noch auf das komplette Wertspektrum $\{I'_{min}, ..., I'_{max}\}$ (hier: $\{0, ..., 255\}$) ausgedehnt (Histogrammspreizung):

$$I'(x, y) = I'_{max} * \frac{I(x, y) - I_{min}}{I_{max} - I_{min}} \tag{3.5}$$

Das zweite Verfahren ($H_2$) stammt aus [1] und verwendet zwei unterschiedlich große Gauss-Filter für die Normalisierung (Abb. 3.4).
Mittels einem Gauss-Filter kann man Bilder *weicher* darstellen, d.h. das Bild wird geglättet. Zur Annäherung an Gaussfilter werden Binomiafilter benutzt. Ein Filter besteht aus einer $m \times m$-Matrix mit $m \geq 3$ und $m$ ungerade. Dabei wird der Wert jedes Pixels durch gewichtete Umgebungspixel ersetzt, die man anhand der Matrix und des Eingabebildes berechnet. Ein wichtiger Sachverhalt ist die Verteilung der Gauss-Funktion, die mit $\sigma$ ausgedrückt wird. Dieser Wert bestimmt v.a. die Größe $m \times m$ des Filters, die sich berechnen lässt aus:

$$m = floor(\sigma \cdot 3) \cdot 2 + 1 \tag{3.6}$$

Die Funktion $floor(x)$ rundet einen beliebigen Wert auf den nächst kleineren Ganzzahlenwert ab.
Neben einem Gaussfilter mit $\sigma = 0,5$, d.h. einem $3 \times 3$-Filter wird für die Normierung ein Gauss-Filter mit $\sigma = 8$, also ein $49 \times 49$-Filter, auf das Bild angewandt.

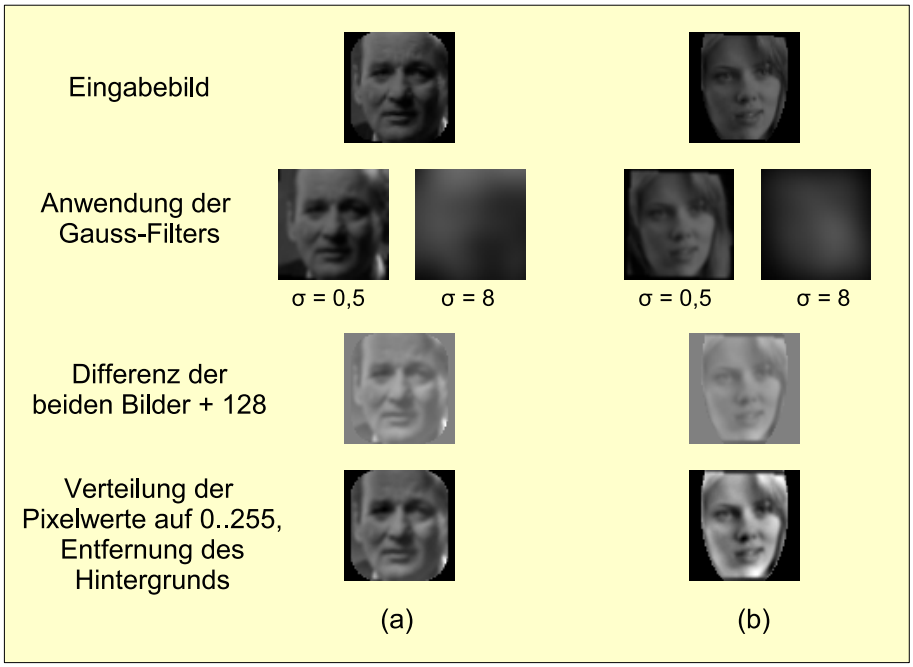


Abbildung 3.4: Helligkeitsnormalisierung durch Anwendung von Gaussfiltern

Die Differenz aus beiden so gefilterten Bildern ergibt das normalisierte Bild. Die Wirkung entspricht dem eines Breitbandfilters. Die Formel dazu sieht folgendermaßen aus:

$$I_{norm} = I * G_{\sigma=0,5} - I * G_{\sigma=8} \tag{3.7}$$

Das Eingabebild ist mit $I$ bezeichnet, $I_{norm}$ ist das normalisierte Bild. Abschließend wird das neue Bild durch eine Histogrammspreizung auf den kompletten Farbbereich verteilt.

### 3.1.3 PCA, Eigenfaces

Die *Principal Component Analysis* (kurz: *PCA*, wird auch als *Karhunen-Loeve Expansion* bezeichnet) ist eine Methode um die Dimension von Vektoren zu verringern. Bei einer solchen Reduzierung ist stets zu beachten, dass es zu keinem Verlust der Charakteristika, d.h. der wichtigsten beschreibenden Merkmale, kommt. Zunächst einmal wird auf die Berechnung der Hauptkomponenten eingegangen. Als Eingabebilder dienen beliebige Gesichtsbilder $\Gamma_1, \Gamma_2, ..., \Gamma_M$, von denen jedes die gleiche Größe $N \times N$ haben muss. Das Mittelwertsbild $\Psi$ der

Bilder ist definiert durch

$$\Psi = \frac{1}{M} \sum_{n=1}^{M} \Gamma_n \tag{3.8}$$

wobei $m$ die Anzahl der Bilder beschreibt.
Die Differenz eines Gesichtsbildes $\Gamma_i$ zu $\Psi$ lässt sich durch

$$\Phi_i = \Gamma_i - \Psi \tag{3.9}$$

bestimmen. Man erhält somit aus den Eingabebilern $\Gamma_i$ einen Satz Vektoren $\Phi_i$, $i = 1, ..., M$, anhand dem dann die PCA vorgenommen wird. Hierfür müssen die $M$ Eigenvektoren $\mathbf{u}_n$ und die dazugehörigen Eigenwerte $\lambda_n$ der Kovarianzmatrix $\mathbf{C}$ ermittelt werden:

$$\mathbf{C} = \frac{1}{M} \sum_{n=1}^{M} \Phi_n \Phi_n^T \tag{3.10}$$

Anhand der $K$ Eigenvektoren $\mathbf{u}_K$ mit den höchsten Eigenwerten $\lambda_K$ kann die Transformation von Daten aus dem ursprünglich $P$-dimensionalen Raum in einen $K$-dimensionalen Unterraum stattfinden. Die Eigenvektoren $\mathbf{u}_K$ werden auch als *Eigenfaces* bezeichnet. Man ermittelt einen geeigneten Wert von $K$ in der Regel durch Betrachten der nach Größe sortierten Eigenwerte (und Eigenvektoren). Je höher ihr zugehöriger Eigenwert ist, desto besser beschreibt ein Eigenface die Variation in den Gesichtern. Ein Beipiel für die PCA und die daraus generierten Eigenfaces stellt Abb. 3.5 dar.

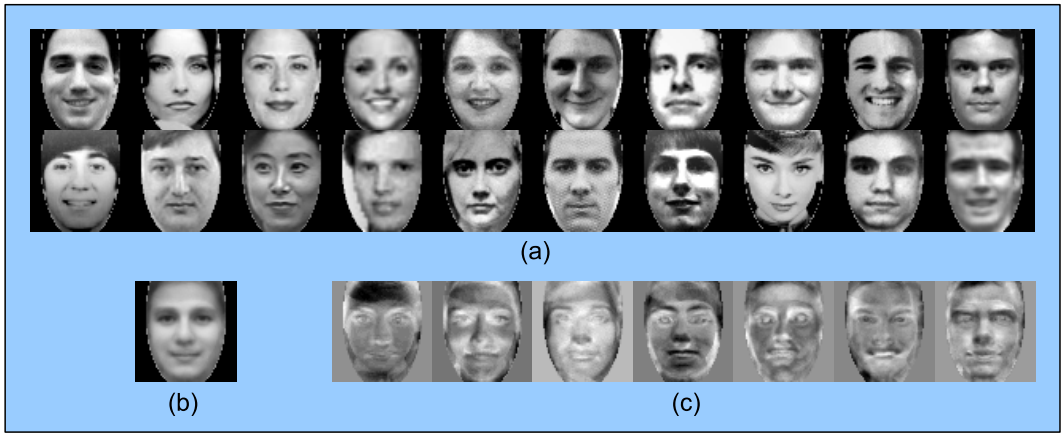


Abbildung 3.5: (a) Eingabebilder für die PCA. (b) Mittelwertsbild. (c) Eigenfaces.

Um nun die Distanz von neuen Eingabebildern $F_j$ zueinander zu bestimmen,

werden diese zunächst in den vorher ermittelten Unterraum transformiert. Dies geschieht durch Anwendung der Formel

$$w_k = \mathbf{u}_k^T (F_j - \Psi) \tag{3.11}$$

für $k = 1, ..., K$.
Für jedes Bild erhält man also einen Vektor $S = [w_1, ..., w_K]$, der das abgebildete Gesicht repräsentiert. Die Einträge $w_k$ in dem $K$-dimensionalen Vektor beschreiben die Gewichtung die jedes Eigenface erhält, um das abgebildete Gesicht bestmöglich zu repräsentieren.

## 3.2 Distanzberechnung

Nachdem die Gesichter in Form von Signaturvektoren vorliegen, kann nun der Vergleich erfolgen.
Ein sehr einfacher Weg zum Vergleichen zweier Vektoren $x$ und $y$ (beide der Dimension $n$) ist die $L_1$-Distanz:

$$L_1(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i| \tag{3.12}$$

Eine weitere Möglichkeit ist die Berechnung der $L_2$-Distanz (auch *Euklidischer Abstand* genannt):

$$L_2(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \tag{3.13}$$

Beide Methoden setzen vorraus, dass die zu vergleichenden Vektoren die gleiche Größe haben. Dies wird in der Vorverarbeitung bereits sichergestellt (Skalierung im Ausrichtungsschritt etc.).

# Kapitel 4

# Auswertung

Dieses Kapitel wertet das vorgestellte Verfahren zur Personenerkennung aus. Als Grundlage dienen die korrekt detektierten Gesichtsbilder aus den beiden Filmen *Und täglich grüsst das Murmeltier* (M) und *Lost in Translation* (L). Zur Auswertung wurden die Gesichter von vier Schauspielern genommen: Bill Murray (*Bill*), Andy MacDowell (*Andy*), Scarlett Johansson (*Scarlett*) und Chris Elliot (*Chris*). Während die letzten drei genannten Personen nur in je einem dieser Filme mitwirken, ist Bill Murray der Hauptdarsteller in beiden Produktionen.
Die Anzahl der korrekt detektierten Gesichtsbilder, anhand denen die Auswertungen durchgeführt wurden, unterscheidet sich bei den Personen zum Teil erheblich. Da diese Zahl in der Auswertung durchaus eine Rolle spielt, sind in Tabelle 4.1 die einzelnen Werte zusammengefasst.

| Charakter | Bezeichner | Film | Anzahl der Bilder |
|---|---|---|---|
| Chris Elliot | *Chris* | M | 54 |
| Scarlett Johansson | *Scarlett* | L | 424 |
| Andie MacDowell | *Andy* | M | 712 |
| Bill Murray I | *Bill* | M | 1497 |
| Bill Murray II | $Bill_2$ | L | 417 |

Tabelle 4.1: Zusammenfassung der verwendeten Daten für die Auswertung

Durch die entsprechende Vorgabe der Standardposition und Größe der Gesichter in der Vorverarbeitung (siehe Kapitel 3.1) wurden alle Gesichtsbilder in einer Größe von $64 \times 64$ Bildpunkten ausgeschnitten. Auf dieser Basis wurden Signaturen mit verschiendenen Masken und Helligkeitsnormalisierungen erstellt (siehe Kapitel 3).

Die Anwendung der in dieser Arbeit beschriebenen Gesichtserkennung auf Filme/Filmcharaktere kann auf zwei unterschiedliche Arten erfolgen. Zum Einen ist es denkbar, eine Abfrage für ein bekanntes Gesicht durchzuführen und eine Liste an ähnlichen Bildern zu erhalten. Dies könnte für eine Benennung der einzelnen Gesichtsbilder mit einem geringeren Aufwand im Vergleich zu einer manuellen Identifizierung der gesamten Gesichtsmenge benutzt werden. Dieser Ansatz wird in Abschnitt 4.1 empirisch ausgewertet. Eine zusätzliche Anwendung dazu stellt die Untersuchung von neuen Filmen/Videos auf bestimmte enthaltene Personen dar. Hierzu wurden zunächst Modelle (z.B. ) für bekannte Personen erstellt. Dann sollten unbekannte Gesichter korrekt diesen Modellen zugeordnet werden. Die Person, deren Modell die geringste Distanz zum bestimmenden Bild hat, wird mit diesem assoziiert. In Abschnitt 4.2 findet eine Auswertung dieses Verfahrens statt.

## 4.1 Rangliste

Um die Anwendung unseres Ansatzes zur Gesichtserkennung auf die oben genannte Abfrage eines bekannten Gesichts zu untersuchen, wurden zunächst für die in Tabelle 4.1 genannten Gesichtsbilder Signaturen erstellt. Es wurden sechs verschiedene Signaturen generiert, die sich in der verwendeten Maske (die drei unterschiedlichen Masken sind in Abb. 3.2 gezeigt) und der Methode zur Helligkeitsnormalisierung (siehe Anschnitt 3.1.2) unterscheiden. Die Dimension wurde dabei nicht durch die Anwendung einer PCA verringert. Um zu ermitteln, wie gut die erstellten Signaturen für die genannte Anwendung der Abfrage geeignet sind, wurden alle anderen Signaturen in der Testmenge mit der abgefragten Signatur verglichen und anschließend nach der Distanz zu einer Rangliste angeordnet. Die geringste Distanz steht dabei an oberster Stelle in der Rangliste. Zur Distanzberechnung kommt entweder die $L_1$- oder die $L_2$-Distanz (siehe Abschnitt 3.2) zum Einsatz.

Für das erste Experiment wurde eine Teilmenge verwendet, die nur Bilder eines Charakters aus einem Film enthält, d.h. für *Andy*, *Bill* und *Chris* aus Film M und für *Scarlett* aus Film L. Aus Tabelle 4.1 kann die Anzahl der verwendeten verwendeten Bilder pro Person entnommen werden. Die Gesamtzahl der Bilder in der Testmenge beträgt somit 2687. Es wurde für jeden Abfrageversuch jeweils nur eine Parametereinstellung verwendet, d.h. eine Signatur pro Bild und

eine Distanzberechnung. In Abb. 4.1 sind einige Ergebnisse mit der zugehörigen Parametereinstellung gezeigt. Das abgefragte Bild ist gelb umrandet. Zusätzlich werden die 29 Bilder mit der kürzesten Distanz gezeigt, wobei die Distanz zum Eingabebild von links oben nach rechts unten hin wächst. Man erkennt direkt, dass die Ergebnisse hier sehr gut sind. Für einen geringen Teil der Signaturen waren die Ergebnisse deutlich schlechter. In Abb. 4.2 gibt es dazu ein paar Beispiele.

Insgesamt konnten folgende Beobachtungen gemacht werden: Die ersten 2-3 Bilder in der Rangliste zeigen sogar bei schlechten Ergebnissen die abgefragte Person. Signaturen von einer Person, die aus Bildern einer Szene stammen, sind durch ihre Ähnlichkeit häufig direkt hintereinander in der Rangliste vertreten. Aus diesem Grund kommt es bei einer Abfrage mit einem Bild aus einer Szene mit vielen Bildern zu der Person meist zu sehr guten Ergebnissen, wogegen man bei Gesichtern, die sozusagen einzigartig in ihrer Darstellung vorkommen, oft sehr viele Bilder von verschiedenen Personen zurückbekommt, d.h. das Ergebnis wesentlich schlechter ist.

Die Signaturen von *Bill* und *Andy* liefern sehr gute Ergebnisse. Dagegen sind die von *Chris* deutlich schlechter. Bei *Scarlett* stellen die Ergebnisse ein Mittelmass dar. Das lässt die Vermutung aufkommen, dass die Anzahl der Bilder für eine Person, anhand deren die Vergleiche zum Abfragebild durchgeführt werden, Einfluss auf das Ergebnis hat.

Der Darsteller Bill Murray spielt in beiden untersuchten Filmen die Hauptrolle. Dieser Sachverhalt ermöglicht es, zu untersuchen, in wie weit die Anwendung auch filmübergreifend funktioniert. Hierfür wird ein Abfragebild des Schauspielers $Bill_2$ aus dem Film L genommen. Die Testmenge bestand aus Gesichtsbildern von *Bill* (Film M) und Bildern der diversen anderen Personen aus beiden Filmen. Erneut wurde ein Vergleich der Signatur mit allen Testbildern durchgeführt und sie anschließend der ermittelten Distanz nach angeordnet. Ein Ergebnis ist in Abb. 4.5 gezeigt. Wie man sofort sieht, ist die Untersuchung auf bekannte Gesichtsbilder in dem anderen Video erfolgreich, es sind von 29 Bildern nur 3 falsche Gesichter gefunden worden und diese recht weit hinten in der Rangliste. Ein Problem kann allerdings bei dieser Anwendung auftreten, falls sich Schauspieler durch starke äußerliche Veränderungen an die Rolle anpassen.

Eine Verbesserung der Abfrageergebnisse kann dadurch erreicht werden, dass, anstatt nur ein Bild für die Abfrage der ähnlichen Gesichter zu nehmen, mehrere Eingabebilder verwendet werden. Es wird dann z.B. die geringste Distanz oder

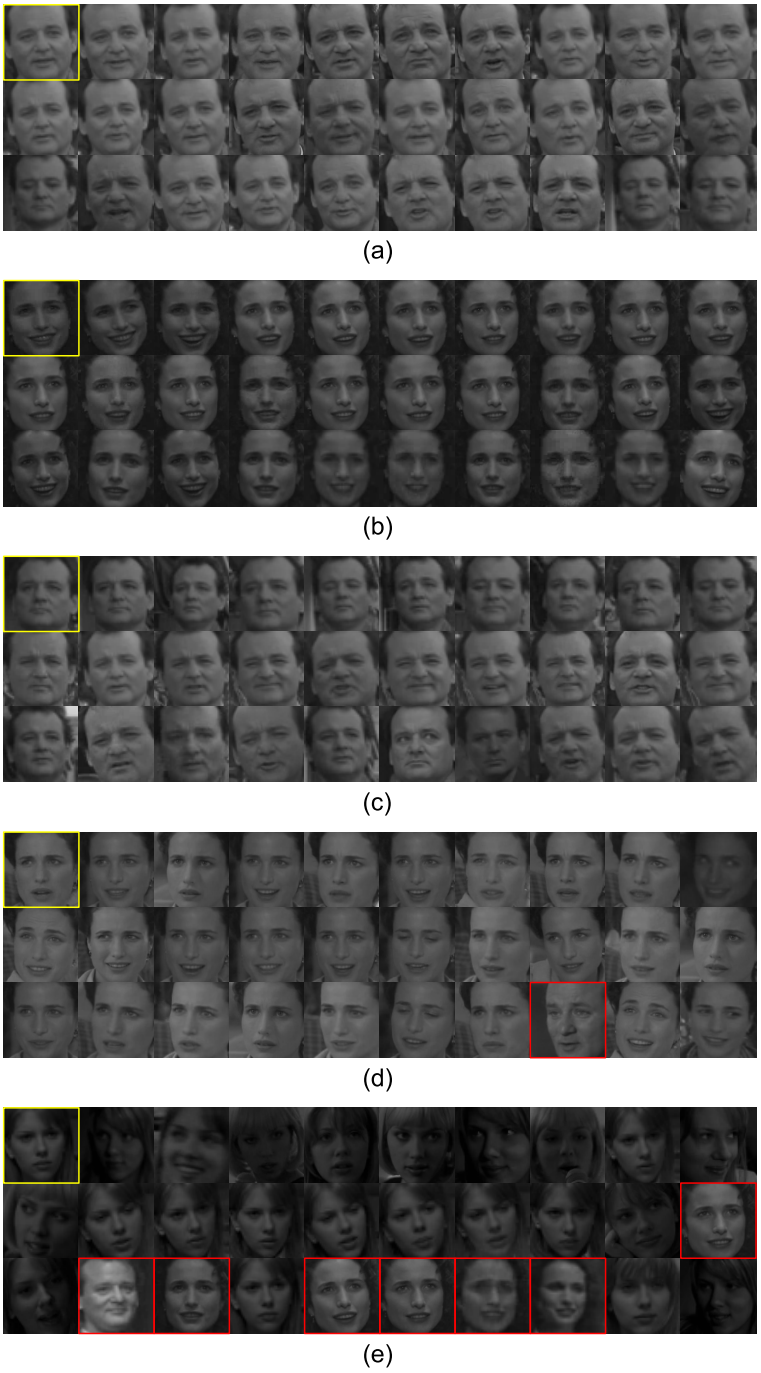


Abbildung 4.1: Beipiele für gute Ergebnisse der Ranglistenauswertung: (a) Maske 1, $H_1$, $L_1$-Norm. (b) Maske 2, $H_2$, $L_1$-Norm. (c) Maske 2, $H_2$, $L_2$-Norm. (d) Maske 3, $H_1$, $L_2$-Norm. (e) Maske 2, $H_1$, $L_1$-Norm. Bei allen Diagrammen sind das Eingabebild gelb und die Bilder von anderen Personen rot markiert.

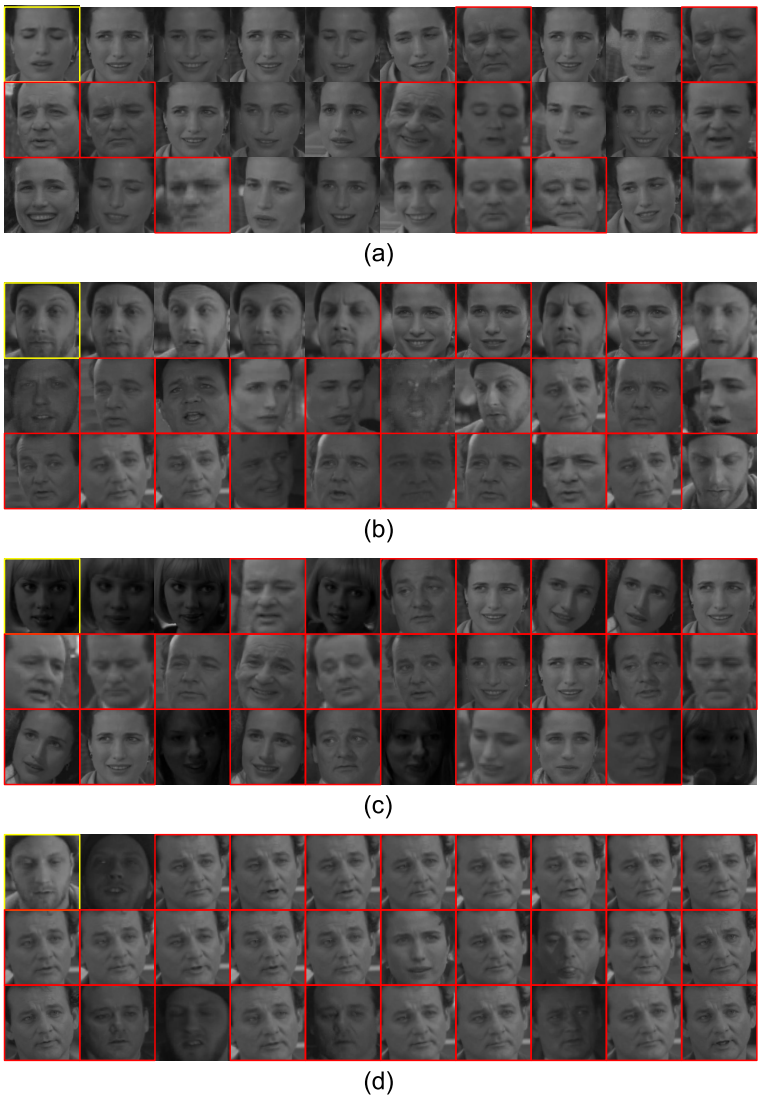


Abbildung 4.2: Beipiele für schlechte Ergebnisse der Ranglistenauswertung: (a) Maske 1, $H_2$, $L_1$-Norm. (b) Maske 2, $H_1$, $L_2$-Norm. (c) Maske 1, $H_2$, $L_2$-Norm. (d) Maske 3, $H_1$, $L_2$-Norm. Bei allen Diagrammen sind das Eingabebild gelb und die Bilder von anderen Personen rot markiert.

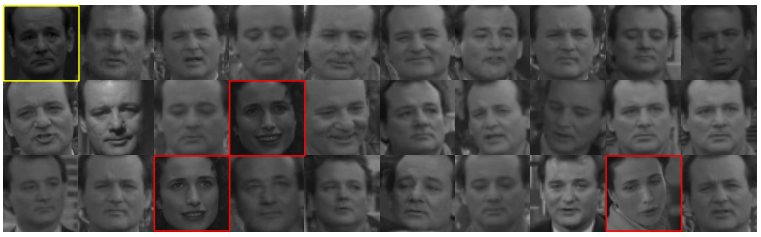

Abbildung 4.3: Die Gesichter mit der kleinsten Distanz zum Eingabebild (gelb). Als abfragende Signatur wurde ein Bild aus Film L mit *Bill* genommen und mit den Signaturen von *Bill* aus Film M zusammen mit denen der anderen Personen verglichen. Parameter: Maske 1, $H_1$, $L_2$-Norm.

der Mittelwert genommen, um die Rangliste zu erstellen. Dieser Ansatz konnte aus Zeitgründen in dieser Arbeit jedoch nicht weiter untersucht werden.

## 4.2 Klassifizierung

In diesem Abschnitt soll die Anwendung des vorgestellten Erkennungsverfahrens auf die korrekte Zuordnung eines Gesichtsbildes zu einer konkreten Person experimentell ausgewertet werden. Ziel ist es, die Auswirkungen der verschiedenen Parameter zu bestimmen. Hierfür werden jeweils *Precision* und *Recall* (siehe Formel 2.10 und 2.11) als Messkriterium für die *Performance* der Klassifizierung ermittelt.
Es wurden wiederum zunächst sechs Signaturen pro Person verwendet. Diese unterscheiden sich in der benutzten Maske zur Hintergrundentfernung und der Art der Helligkeitsnormalisierung. Die Signaturen wurden in einigen Experimenten zusätzlich durch die Anwendung einer PCA (siehe Abschnitt 3.1.3) in ihrer Dimension verrignert. Die gewählten Parametereinstellungen (Maske, Helligkeitsnormalisierung, Dimensionsreduzierung) werden für jedes Experiment in folgenden einzelnd genannt. Um den Einfluß dieser Parameter auf das Ergebnis zu ermitteln, wird der jeweilige Versuch für verschiedene Parameterkonfigurationen des jeweils untersuchten Parameters durchgeführt, während die anderen Parameter fest auf einen Wert eingestellt bleiben.
Weiterhin wurde in einigen Experimenten eine feste Anzahl $A \geq 1$ von Signaturen einer Person zu einem Modell zusammengefasst. Dieses Modell besteht dann aus einer Signatur, deren Vektorkomponenten sich aus den Mittelwerten der einzelnen Signaturen zusammensetzen.

Alle Signaturen/Modelle wurden entweder mit der $L_1$- oder $L_2$-Distanz miteinander verglichen. Das jeweils nächste Modell/Signatur wurde dann als die Klassifizierung angenommen.
In jedem Experiment wurden zufällig 50 Bilder für jede der vier Personen aus den vorhandenen Gesichtsbildern ausgewählt. Diese Bilder bilden dann die Testdaten. Weiterhin wurden aus den übrigen Bildern für jede Person $A$ Gesichtsbilder zur Signatur-/Modellerstellung zufällig herausgesucht. Jedes Experiment wurde für die jeweiligen Parametereinstellungen 20-mal wiederholt und der Mittelwert für Precision und Recall daraus gebildet.
Die True Positives einer Person sind in diesem Zusammenhang die richtig als positiv klassifizierten Gesichter der Testmenge für diesen Charakter, die False Positives analog dazu die fälschlicherweise benannten Bilder für die Person. Gesichtsbilder einer Person, die mit einer anderen Person assoziiert werden, werden als False Negatives bezeichnet. Korrekterweise falsch benannte Personen, sprich welche, bei denen die Gesichtsbilder zu einem anderen Charakter gehören und auch so klassifiziert werden, heissen True Negatives.
Im Folgenden werden die Ergebnisse der zahlreichen Experimente vorgestellt und ausgewertet.

## Distanz

Zum Vergleich zweier Signaturen wird die Distanz ermittelt. Dies kann entweder mittels $L_1$ oder $L_2$-Norm (siehe Kapitel 3.2) erfolgen. Um zu ermitteln, welche Distanznorm bessere Erkennungsergebnisse liefert, wurden Experimente durchgeführt, bei denen die jeweiligen Parametereinstellungen gleich blieben und nur die Distanznorm, die zur Berechnung herangezogen wurde, verändert wurde. In Abb. 4.4 sind die durchschnittlichen Werte für die Precision und den Recall für zwei verschiedene Parametereinstellungen abgebildet. Bei den durchgeführten Tests lagen die Werte für Precision und Recall für beide Distanzen meist relativ nah beieinander. Tendenziell bringt die $L_2$-Distanz bei der Precision bessere Ergebnisse. Da bei der $L_2$-Distanz die Differenz quadriert wird, wirken sich größere Farbunterschiede in den Bildern stärker auf das Ergebnis aus.

## PCA-reduzierte Signaturen

Durch eine PCA lassen sich die Signaturen in ihrer Dimension reduzieren. Dabei wird die neue Dimension $K$ vorher festgelegt. In den Diagrammen der Abb. 4.5

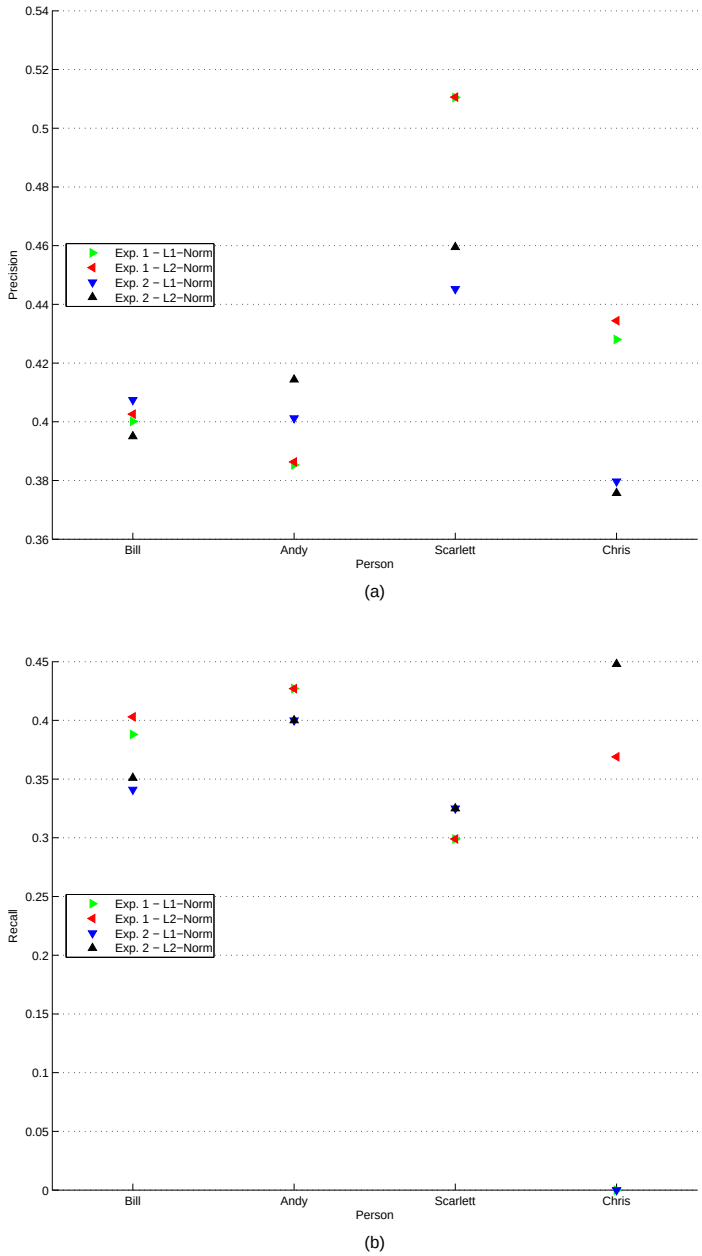


Abbildung 4.4: Auswertung von $L_1$ und $L_2$-Norm. (a) Precision, (b) Recall. Verwendete Parameter: Experiment 1: Modell 2, Maske 2, $H_2$, keine PCA. Experiment 2: Modell 2, Maske 2, $H_1$, keine PCA

sind die Ergebnisse eines Experiments dargestellt, bei dem alle Parametereinstellungen festgehalten wurden und nur die Dimension $K$ des Unterraums verändert wurde. Man erkennt, dass durch Verwendung einer PCA auf die Signaturen keine Verbesserung der Raten für Precision und Recall zustande kommt. Das Gegenteilige ist sogar der Fall: Die Ergebnisse werden schlechter. Dies kann darauf zurückzuführen sein, dass durch die Transformation wichtige Charakterstika der Daten verloren gehen, so dass die Gesichter weniger gut beschrieben werden als ohne Reduzierung.
Die Tests wurden zuerst mit einer Verkleinerung der Dimension auf sieben ausgeführt, danach auf elf und 23. Es kommt eine geringe Verbesserung durch die Erhöhung der Dimension zustande, was jedoch nicht immer zwangsläufig der Fall ist.

## Maske

Es wurden drei verschiedene Masken für die Tests verwendet (siehe Abb. 3.2). Diese bestanden aus mehr oder weniger großen ovalen Gesichtsbereichen. Der Einfluß der Wahl einer Maske auf das Ergebnis ist in Abb. 4.x dargestellt. Die Auswertungen zeigen, dass keine globale Aussage getroffen werden kann. Es kommen sowohl schlechte als auch gute Ergebnisse durch alle drei Masken zustande. Zuerst wird auf die Precision eingegangen: Bei *Bill* liefern die Masken 1 und 3 gute Ergebnisse, während bei *Andy* die ersten beiden Masken bessere Ergebnisse bringen als die dritte. Für *Scarlett* ist Maske 2 eindeutig die Beste, für *Chris* Maske 3. Die Diagramme in Abb. 4.6 zeigen den Sachverhalt. Beim Recall liefert *Bill* die besten Ergebnisse mit Maske 3 und Maske 1. Bei *Andy* stellt Maske 1 die beste Wahl dar. *Scarlett* und *Chris* haben mit Maske 3 die besten Werte. Die Qualität einer Maske in Bezug auf die Erkennungsraten ist abhängig von dem abgebildeten Gesicht des Bildes und somit auch personenabhängig.

## Helligkeitsnormalisierung

In der Arbeit werden zwei Verfahren zur Helligkeitsnormalisierung angewandt: die Methode $H_1$ von Rowley in [6] und $H_2$ in [1] (beide sind in Kapitel 3.1.2 detailliert erläutert). Der Einfluß der Methode der Helligkeitsnormalisierung auf das Ergebnis eines Experiments wird in Abb. 4.7 verdeutlicht. Wiederum wurden alle Parameter bis auf die Wahl des Verfahrens zur Helligkeitsnormalisierung fest gelassen und die Werte für Precision und Recall für die verschiedenen Personen

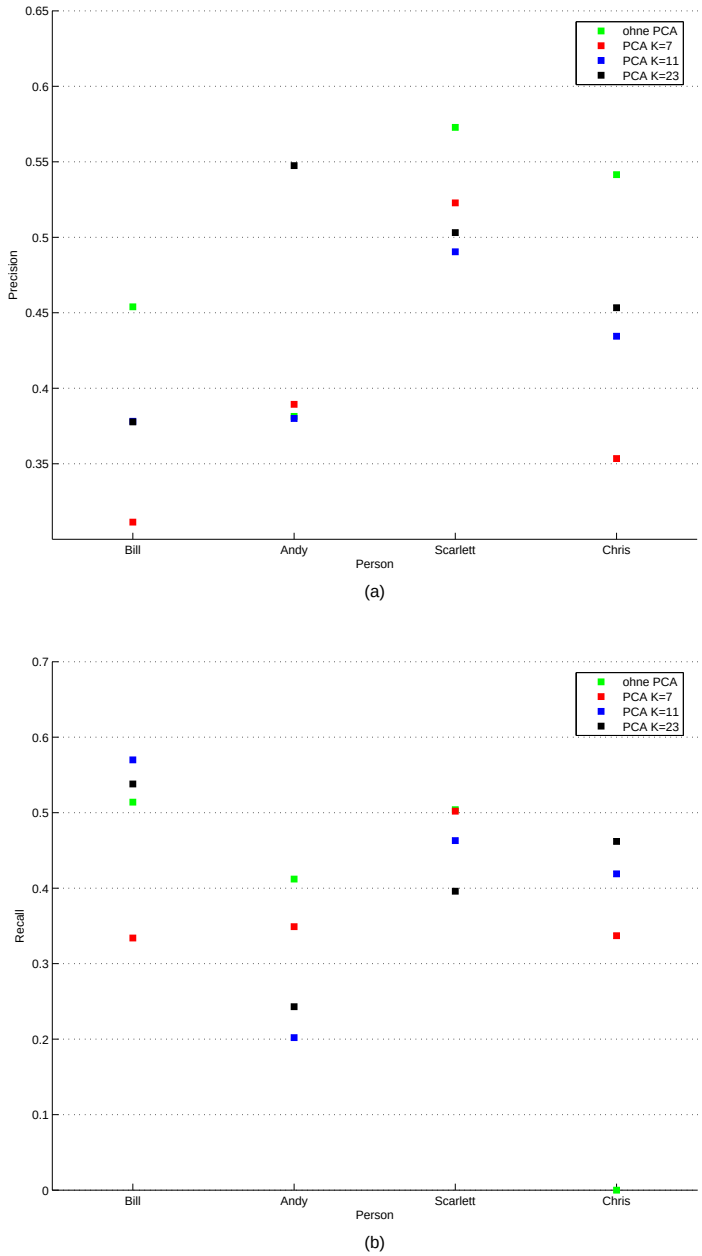


Abbildung 4.5: Auswertung der Anwendung einer PCA. (a) Precision, (b) Recall. Verwendete Parameter: Maske 1, $H_1$, Modell 4, $L_1$-Norm

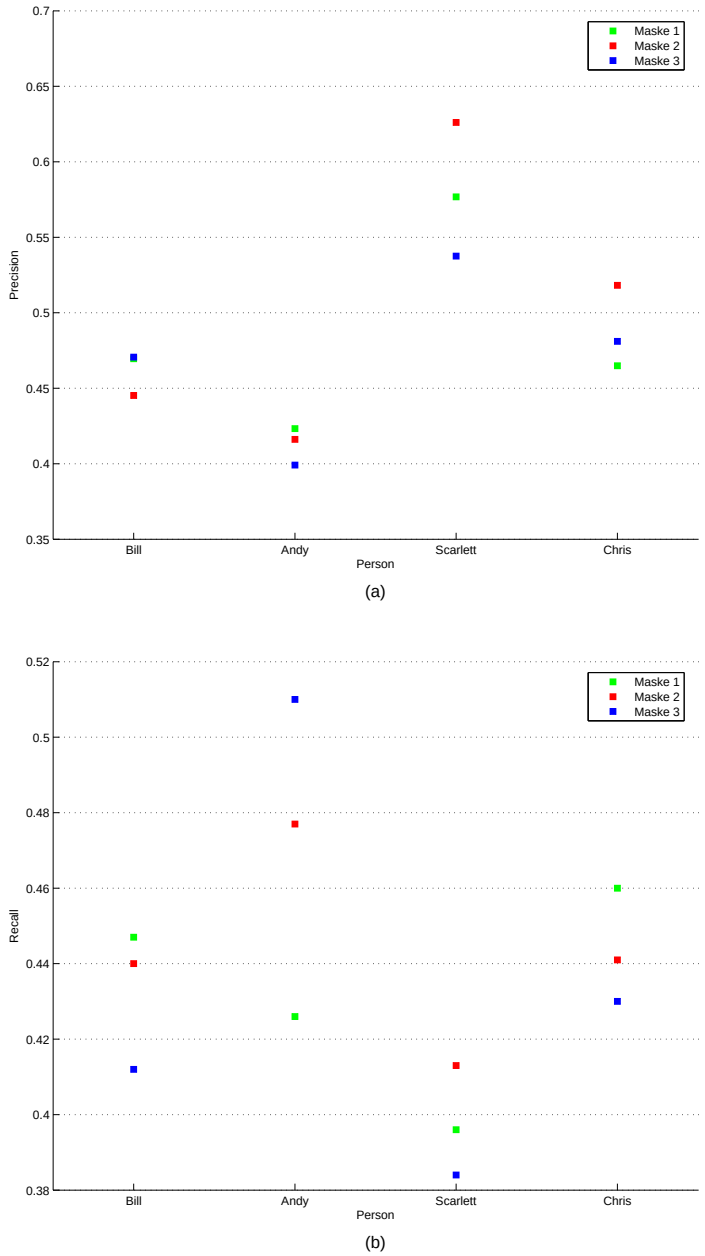


Abbildung 4.6: Auswertung von verschiedenen Masken. (a) Precision, (b) Recall. Verwendete Parameter: $H_1$, Modell 2, $L_2$-Norm, keine PCA

miteinander verglichen. Man erkennt, dass beide Methoden für die einzelnen Personen unterschiedliche Ergebnisse für die Precision liefern. Während bei *Bill* und *Scarlett* das zuerst beschriebene Verfahren $H_1$ bessere Ergebnisse liefert, sind die Werte bei *Andy* und *Chris* bei $H_2$ höher. Beim Recall ist die erste Methode tendenziell besser. Dies ist bei *Andy*, *Scarlett* und *Chris* der Fall.
Bei der Helligkeitsnormalisierung kommt es auf das jeweilige Gesichtsbild und die verwendete Maske an. Wenn noch viele Hintergrundpixel im eigentlich hintergrundentfernten Bild vorhanden sind, so kann dies Auswirkungen auf die Normalisierung der Helligkeit haben. Dies resultiert daraus, dass die Pixelwerte im Hintergrund oft im Kontrast zum Gesicht stehen.

## Modell

Um die Personen klassifizieren zu können, werden Vergleichsmodelle für die einzelnen Personen erstellt. Ein Modell besteht aus einer Signatur, deren Vektorkomponenten sich aus den Mittelwerten von Signaturen der jeweiligen Person zusammensetzen. Der Wert für $m$ wird vorher festgelegt. Es ist zu erwarten, dass bei steigender Zahl an Eingabebilder für das Modell auch die Präzisionsrate ansteigt. Um dieses zu testen, fand eine Auswertung mit verschiedenen Werten für $m$ statt. Da bei *Chris* insgesamt nur eine Bilderzahl von 54 gegeben war und die Anzahl der Testbilder 50 betrug, verliefen die Tests mit Werten für $m$ von eins, zwei und vier. Ab einem Wert $m \geq 5$ wurde *Chris* aus dem Test herausgenommen, so dass nur Raten für Precision und Recall der anderen drei Personen berechnet wurden. In Abb. 4.8 sind die Ergebnisse dargestellt. In den meisten Fällen steigt der Wert der Precision bei zunehmenden Wert von $m$. Der Recall liefert ein eindeutigeres Ergebnis. Hier liefert das Modell mit $m = 8$ Signaturen für alle Personen die besten Ergebnisse.

## Gesamtauswertung

Unter Verwendung der Ergebnisse der vorhergehenden experimentellen Auswertung wurde ein weiteres Experiment mit den vorher ermittelten optimalen Parametereinstellungen durchgefhrt. Die Parametereinstellungen wurden somit wie folgt gewählt: Für das Modell jeder Person wurden $m = 10$ Signaturen verwendet. Da bei *Chris* nur 54 Signaturen vorlagen, wurden nur die anderen drei Personen getestet. Die Signaturen wurden mit dem ersten Verfahren zur Helligkeitsnormalisierung und der dritten Maske vorverarbeitet. Es wurde keine Reduzierung der

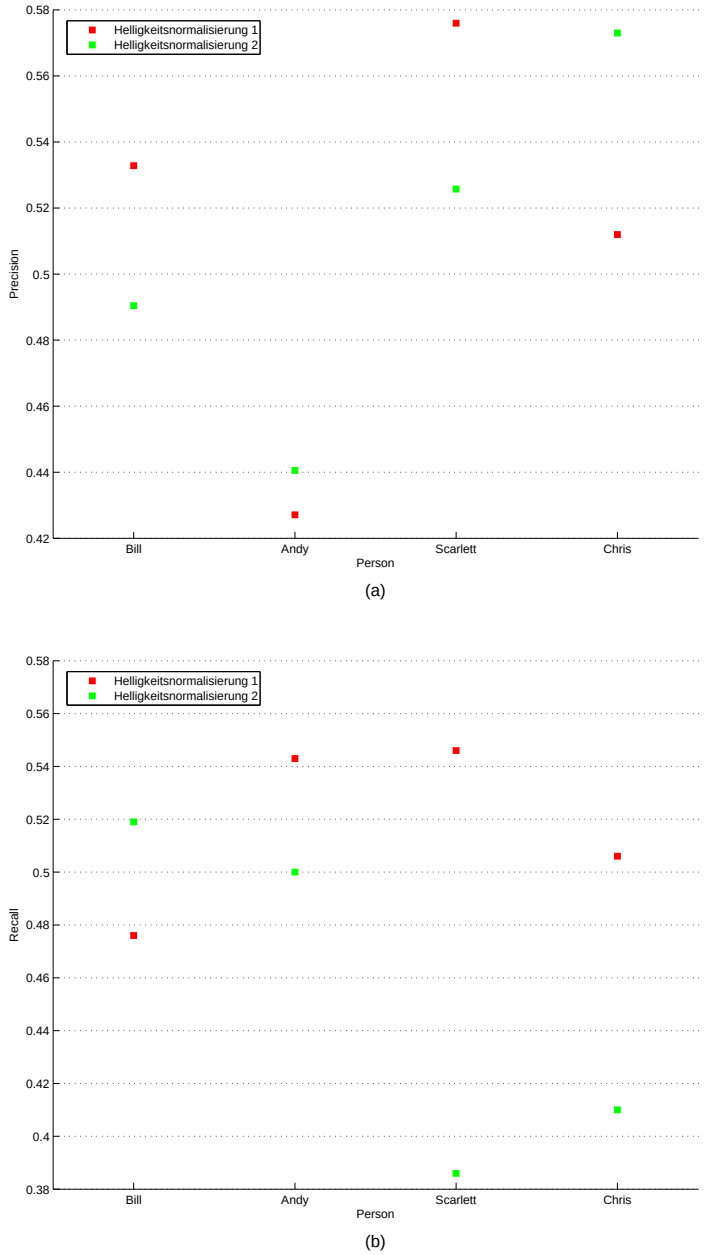


Abbildung 4.7: Auswertung der Methoden zur Helligkeitsnormalisierung. (a) Precision, (b) Recall. Verwendete Parameter: Modell 4, Maske 3, keine pca, $L_2$-Norm

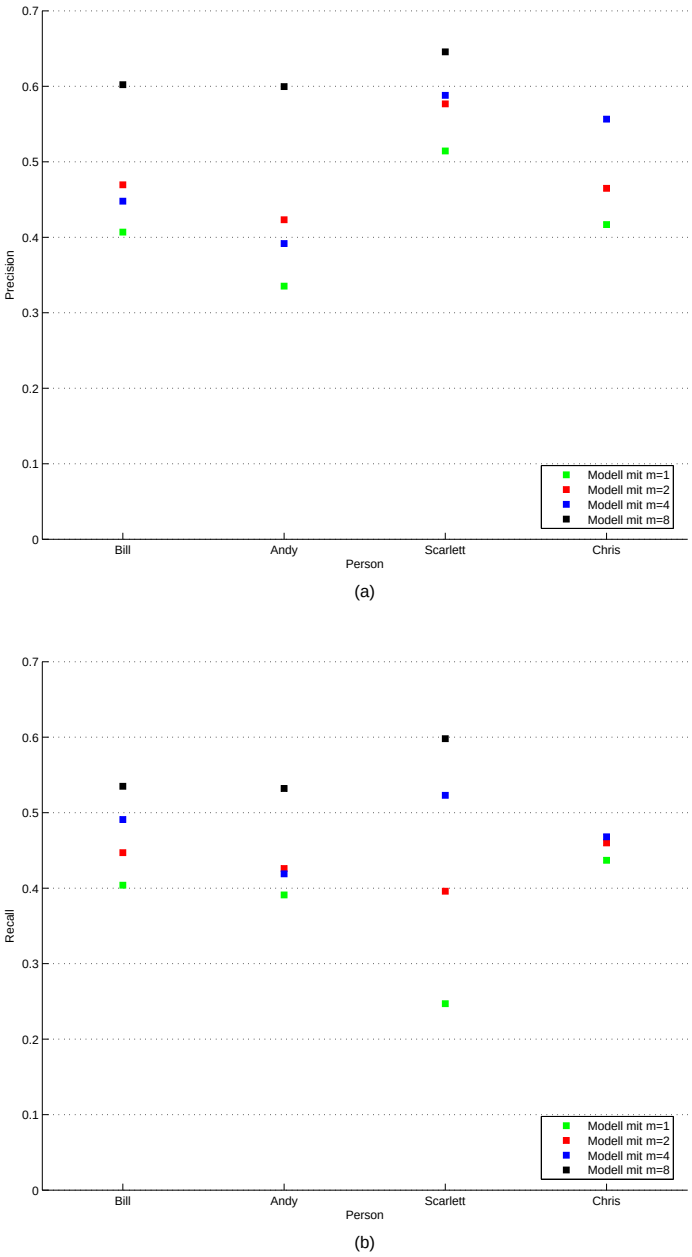


Abbildung 4.8: Auswertung der Modellerstellung. (a) Precision, (b) Recall. Verwendete Parameter: $H_1$, Maske 3, keine pca, $L_2$-Norm

Signaturen mittels PCA vorgenommen.
Für die Auswertung wurden 100 Testbilder pro Person genommen. Es wurde der gleiche Testaufbau wie in den vorherigen Experimenten verwendet. In Abb. 4.9 sind der jeweilige Mittelwert für Precision unf Recall aus den 20 Einzeltest sowie das Minimal- und Maximalergebnis dargestellt. Die Raten liegen bei der Precision im Durchschnitt bei ca. 60%, beim Recall bei über 50%. Dies ist ein vielversprechendes Ergebnis.
Eine weitere Verbesserung der Ergebnisse wäre durch die Verwendung einer *knn*-Klassifizierung zu erreichen. Dabei wird das Ergebnis der Klassifizierung nicht nur durch das nächste bekannte Gesicht bestimmt wie in dieser Arbeit, sondern durch die Mehrzahl der nächsten $k$ Elemente.

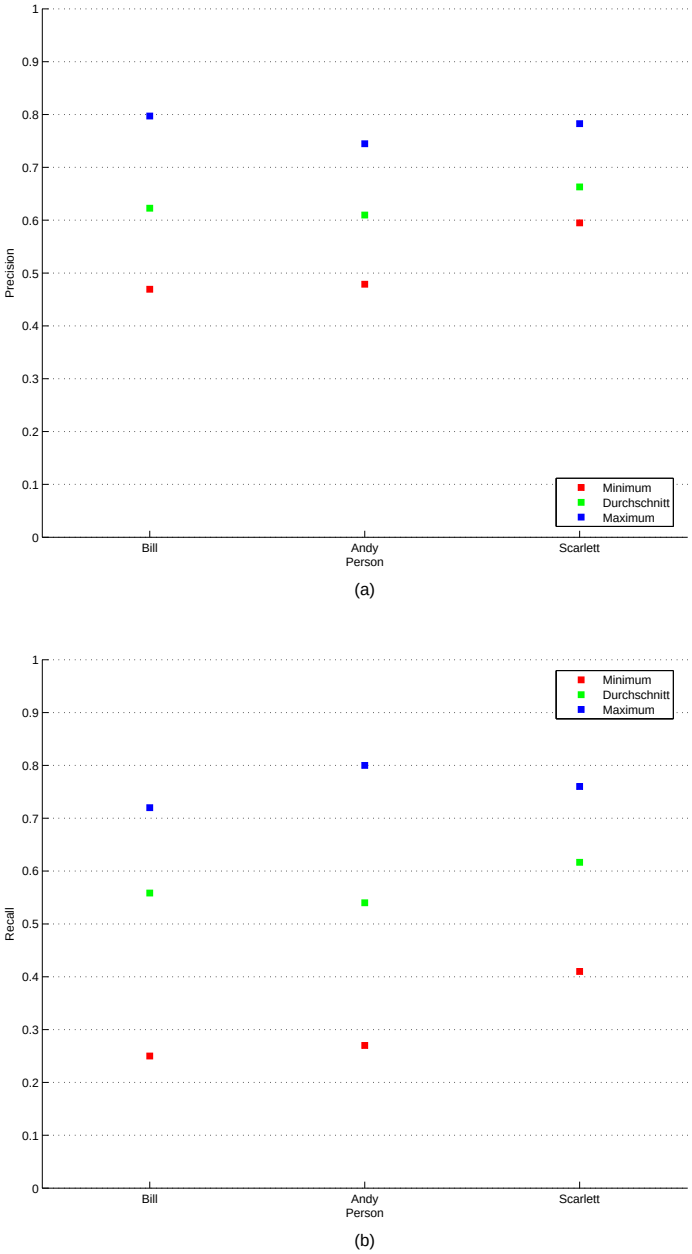


Abbildung 4.9: Auswertung der besten Ergebnisse. (a) Precision, (b) Recall. Verwendete Parameter: $H_1$, Maske 3, keine pca, $L_2$-Norm, Modell: 10

# Kapitel 5

# Zusammenfassung

In dieser Arbeit wurde eine Methode vorgestellt, die es ermöglicht Personen in Filmen anhand ihrer Gesichter zu erkennen. Hierzu wurden zunächst die Gesichter im zu untersuchenden Film detektiert. Hierfür wurde der Gesichtsdetektor aus OpenCV [5] verwendet.
Anschließend wurden aus den detektierten Gesichtern Signaturen erzeugt, die einen Vergleich zwischen unterschiedlichen Gesichtsbildern ermöglichen. Um diese Signature unhabhängig von den Einflüßen Position, Größe, Hintergrund und Helligkeitsverteilung sowie teilweise unabhängig gegenüber der Ausrichtung zu berechnen, wurden einige Vorverarbeitungsschritte durchgeführt.
Um das Gesicht in eine Standardposition und -größe bringen zu können wurden desweiteren Detektoren für die Gesichtsmerkmale Mund, linkes und rechtes Auge benötigt. Diese wurden mit Hilfe einer Trainingsmenge und dem statistischen Lernverfahren Boosting in Verbindung mit *Haar-like*-Merkmalen erzeugt. Die experimentelle Auswertung zeigte, dass die Merkmalsdetektoren gute Ergebnisse liefern.
Für die weiteren Vorverarbeitungschritte Hintergrundentfernung und Helligkeitsnormierung wurden zwei unterschiedliche Verfahren angewand. Die Pixelwerte der so vorverarbeiteten Gesichtsbilder wurden dann entweder direkt als Signatur verwendet oder optional in ihrer Dimension durch Anwendung einer PCA verringert. Um Signaturen miteinander zu vergleichen wurden $L_1$ bzw. $L_2$-Norm angewendet.
Das gesamte Gesichtserkennungsverfahren wurde in C/C++ implementiert und experimentell ausgewertet. Zur Auswertung wurden zwei unterschiedliche Anwendungen herangezogen.

Es hat sich gezeigt, dass die vorgestellte Methode gute Ergebnisse bei der Abfrage eines Gesichtsbildes erzielt. Die Klassifizierung von Gesichtern anhand von Modellen bekannter Personen lieferte ebenfalls gute Ergebnisse. Für optimale Parametereinstellungen wurde eine Precision von 60% und ein Recall von über 50% erreicht.

# Literaturverzeichnis


[1] Arandjelović, O. and Zisserman, A. *Automatic face recognition for film character retrieval in feature-length Films.* In Proc. IEEE Conference on Computer Vision and Pattern Recognition. 2005.

[2] Cristinacce, D. and Cootes, T. and Scott, I.. *A Multi-Stage Approach to Facial Feature Detection.* $15^{th}$ British Machine Vision Conference. 2004.

[3] Lienhart, R. and Kuranov, A. and Pisarevsky, V. *Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection.* MRL Technical Report. 2002.

[4] Lienhart, R. and Maydt, J. *An Extended Set of Haar-like Features for Rapid Object Detection.* Proceedings of the International Conference on Image Processing. 2002.

[5] Open Computer Vision Library. Version beta5. http://sourceforge.net/projects/opencvlibrary/

[6] Rowley, H. A. and Baluja, S. and Kanade, T. *Neural Network-Based Face Detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence. 1998.

[7] Schneiderman, H. *A Statistical Approach to 3D Object Detection Applied to Faces and Cars.* Robotics Institute, Carnegie Mellon University. 2000.

[8] Turk, M. A. and Pentland, A. P. *Face Recognition Using Eigenfaces.* In Proc. IEEE Conference on Computer Vision and Pattern Recognition. 1991.

[9] Viola, P. and Jones, M. *Rapid object detection using a boosted cascade of simple features.* In Proc. IEEE Conference on Computer Vision and Pattern Recognition. 2001.