Gesichtsbasierte Geschlechtserkennung auf Bildsequenzen

Studienarbeit

Institut für Theoretische Informatik
Lehrstuhl Prof. Dr. A. Waibel

Fakultät für Informatik
Universität Karlsruhe (TH)

von

Clemens Siebler

29. Juli 2008

Betreuer:

Prof. Dr. Alex Waibel
Dr.-Ing. Rainer Stiefelhagen
Dipl.-Inf. Keni Bernardin

Hiermit erkläre ich, die vorliegende Arbeit selbständig erstellt und keine anderen als die angegebenen Quellen verwendet zu haben.

Karlsruhe, 29. Juli 2008 .....

Inhaltsverzeichnis

1 Einführung

1.1 Motivation

Der Mensch ist in der Lage, Gesichter auf eine Vielzahl von Arten zu verarbeiten. Nur so ist es ihm möglich, in seinem sozialen Umfeld Zurecht zu kommen. Viele Werke im Bereich der sozialen und kognitiven Psychologie belegen, dass wir dazu in der Lage sind aus Gesichtern Informationen zu erfassen. So können wir z.B. das Geschlecht, die Rasse oder den emotionalen Zustand unserer Mitmenschen binnen des Bruchteils einer Sekunde feststellen. Dies gilt sowohl für uns bekannte, als auch für uns unbekannte Gesichter.

Automatische Gesichtsverarbeitung per Computer ist jedoch, bedingt durch z.B. schlechte Bildqualität, Rotation, Verdeckung, Beleuchtung und Gesichtsausdruck, eine schwierige Aufgabe. Ein System, das automatisch das Geschlecht einer Person anhand ihres Gesichtes bestimmt, könnte z.B. für die Personenidentifikation verwendet werden. Im Gegensatz zur Haarfarbe oder der Kleidung verändert sich dieses Merkmal über die Zeit nicht. Auch könnten mit einer robusten Erkennung große Bilddatenbanken automatisch nach Bildern, auf denen z.B. nur Frauen gezeigt werden, automatisch durchsucht werden. Für den Handel wäre z.B. interessant zu wissen, wie viele Männer bzw. Frauen täglich eine bestimmte Abteilung in einem Kaufhaus betreten oder ein bestimmtes Produkt betrachtet haben.

1.2 Ziel der Arbeit

Ziel dieser Arbeit ist der Entwurf eines Systems zur gesichtsbasierten Geschlechtserkennung anhand von Bildsequenzen. An das System werden einige Anforderungen gestellt: Es soll auf einem handelsüblichen Laptop in Verbindung mit einer Webcam funktionieren. Da das System auch portabel sein soll, der Aufnahmeort also beliebig verändert werden kann, muss es mit variierenden Bedingungen, wie z.B. unterschiedlicher Beleuchtung, leichter Kopfrotation und unterschiedlichen Gesichtsgrößen im Bild zurechtkommen. Wichtig hierbei ist, dass das System ohne eine manuelle Anpassung an die neue Umgebung funktionieren soll. Dazu soll die Geschlechtserkennung nicht auf einem einzigen Bild durchgeführt werden, sondern auf einer kurzen Bildsequenz.

1.3 Stand der Forschung

Moghaddam et al. [1] verwendeten in ihrem Ansatz für die Klassifikation von Gesichtsbildern zur Geschlechtsbestimmung eine Support Vector Machine. Ihr Ansatz arbeitete mit $48 * 84$ Pixel großen Bildern als Eingabe für eine SVM und erreichte auf der FERET Datenbank einen durchschnittlichen Fehler von 3.4%. Die Trainings- und Testdaten unterlagen denselben, kontrollierten Bedingungen und es wurde frameweise klassifiziert. Es wurde gezeigt, dass die Klassifikation nur minimal schlechter wurde, wenn die Auflösung auf $12*21$ Pixel reduziert wurde.

Castrillón-Santana et al. [2] kombinierten ihren echtzeitfähigen Gesichtsdetektor ENCARA [3] mit Geschlechts- und Identitätsbestimmung. Die Experimente wurden auf Bildsequenzen, die mit einer Webcam aufgenommen wurden, durchgeführt. Ihr System lieferte gute Ergebnisse, jedoch bestand die Testdatenmenge nur aus wenigen Sequenzen.

Makinen et al. [4] evaluierten in ihrer Publikation aktuelle Methoden zur Geschlechtserkennung auf Gesichtsbildern in Kombination mit verschiedenen Alignmentmethoden in der Vorverarbeitung. Es stellte sich heraus, dass automatisches im Gegensatz zu manuellem Alignment nicht zur Verbesserung der Erkennungsraten führte. Aus diesem Ergebnis wurde geschlossen, dass die aktuellen Methoden zum automatischen Alignment von Gesichtern noch nicht ausgereift genug seien, da manuelles Alignment niedrigere Fehlerraten lieferte. Neben verschiedenen automatischen Alignmentmethoden wurden auch verschiedene Klassifikatoren getestet, wie z.B. Neuronale Netze, Support Vector Machines und AdaBoost basierte Klassifikatoren. Es stellte sich heraus, dass wie bei [1] die Auflösung der Bilder nur eine untergeordnete Rolle spielte. Support Vector Machines wiesen die geringsten Fehlklassifikationen auf, jedoch dicht gefolgt von Neuronalen Netzen und einem sehr schnellen AdaBoost-basierten Klassifikator. Alle Versuche wurden auf einer Untermenge der FERET Datenbank durchgeführt.

Das Problem bei allen oben genannten Ansätzen liegt darin, dass auf einfachen Daten oder auf nicht genug repräsentativen Testsätzen gearbeitet wurde. Gesichtsaufnahmen in der FERET Datenbank liegen zu größtem Teil in hoher Auflösung vor und wurden unter kontrollierten Bedingungen (z.B. Beleuchtung und Kopfpose) gemacht. Einige der Ansätze wurden auf eigenen, jedoch kleinen Testdatensätzen evaluiert, wodurch nur bedingt eine repräsentative Aussage für den allgemeinen Fall entstehen kann. Bis auf den Ansatz von Castrillón-Santana et al. [2] betrachteten alle Ansätze nur ein Frame um die Geschlechtserkennerung durchzuführen.

In der folgenden Arbeit wird ein neuer Ansatz zur Lösung der gesichtsbasierten Geschlechtserkennung vorgestellt, der auf einer Datenbank mit 100 Testpersonen evaluiert wurde. Anstatt auf einzelnen Bildern wird auf Bildsequenzen klassifiziert.

Dies hat den Vorteil, dass zusätzlich eine Sicherheit für die gemachte Klassifikation einer Person angegeben werden kann und dass nicht zwangsweise jedes Frame in der Bildsequenz einer Person richtig klassifiziert werden muss. Alle Videosequenzen im Testdatenset wurden mit einer Philips Webcam aufgenommen und variieren stark in Beleuchtung der Szene, Pose der Gesichter der Personen und ganz speziell im Aufnahmeort.

2 Theoretische Grundlagen

2.1 Support Vector Machines

2.1.1 Einführung

Der nachfolgende Text zu Support Vector Machines ("Stützvektormethode") basiert auf einem Tutorial von Burges [5]. Auch die Abbildungen wurden von [5] übernommen.

2.1.2 Lineare Support Vector Machines

Gegeben sei eine gelabelte Trainingsdatenmenge mit $\{\mathbf{x_i}, y_i\}, i = 1, ..., l, y_i \in \{-1, 1\}, \mathbf{x_i} \in R^{\mathbf{d}}$. Es werde angenommen, es existiere eine Hyperebene, die die positiven von den negativen Beispielen separiert. Die Punkte, die auf der Hyperebene liegen, genügen $\mathbf{w} \cdot \mathbf{x} + b = 0$, wobei $\mathbf{w}$ der Normalenvektor zur Hyperebene, $|b|/\|\mathbf{w}\|$ die rechtwinklige Distanz von der Hyperebene zum Ursprung und $\|\mathbf{w}\|$ die euklidische Norm von $\mathbf{w}$ ist. Sei d_+ (d_-) die kürzste Distanz von der separierenden Hyperebene zum nächsten positiven (negativen) Beispiel. Der Margin ("Rand") sei nun definiert als $d_+ + d_-$. Für den Fall der linearen Separierbarkeit berechnet der Stützvektoralgorithmus die Hyperebene mit dem größten Margin. Dies kann wie folgt formuliert werden: Angenommen alle Trainingsdaten genügen folgenden Bedingungen:

$$\mathbf{x_i} \cdot \mathbf{w} + b \geq +1 \text{ für } y_i = +1 \tag{2.1}$$

$$\mathbf{x_i} \cdot \mathbf{w} + b \leq -1 \text{ für } y_i = -1 \tag{2.2}$$

Diese beide Ungleichungen können zu einer zusammengefasst werden:

$$y_i(\mathbf{x_i} \cdot \mathbf{w} + b) - 1 \geq 0 \ \forall i \tag{2.3}$$

Betrachten wir nun die Punkte, für die Gleichheit in (2.1) gilt (die Existenz eines solchen Punktes kann durch die Wahl einer Skalierung für $\mathbf{w}$ und b gewährleistet werden). Diese Punkte liegen auf einer Hyperebene $H_1 : \mathbf{x_i} \cdot \mathbf{w} + b = 1$

mit der Normalen $\mathbf{w}$ und der rechtwinkligen Distanz $|1-b|/\|\mathbf{w}\|$ zum Ursprung. Analog dazu liegen die Punkte, für die Gleichheit in (2.2) gilt, auf einer Hyperebene $H_2 : \mathbf{x_i} \cdot \mathbf{w} + b = -1$ mit der Normalen $\mathbf{w}$ und der rechtwinkligen Distanz $|-1-b|/\|\mathbf{w}\|$ zum Ursprung. Folglich ist $d_+ = d_- = 1/\|\mathbf{w}\|$ und der Margin $2/\|\mathbf{w}\|$. Also sind H_1 und H_2 parallel (sie haben dieselbe Normale) und es fallen keine Trainingsbeispiele zwischen sie. Folglich können wir ein Hyperebenenpaar bestimmen, welches den größten Margin liefert, indem wir $\|\mathbf{w}\|^2$ unter der Nebenbedingung (2.3) minimieren. Dieses Minimierungsproblem wird im Folgenden als "primales Problem" bezeichnet.

Eine Veranschaulichung für den zweidimensionalen Fall wird in Abb. 2.1 gezeigt. Wie zu erkennen ist, würde das Entfernen der Trainingsbeispiele, für die Gleichheit in (2.3) gilt (diese liegen auf den Hyperebenen H_1 und H_2), die gefundene Lösung verändern. Man nennt diese Vektoren Stützvektoren. Diese sind in Abb. 2.1 zusätzlich eingekreist.

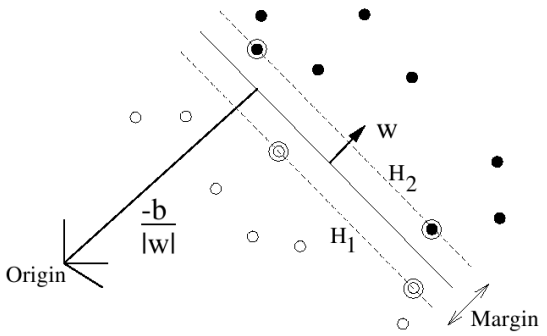

Abb. 2.1: Linear separiernde Hyperebene im separierbaren Fall. Die Stützvektoren sind eingekreist.

Das Optimierungsproblem wird nun nicht direkt, sondern in seiner dualen Form gelöst. Dieses Problem ist äquivalent zu dem zuvor eingeführten primalen Problem, in dem Sinne, dass alle Lösungen des dualen Problems auch Lösungen des primalen sind. Dies hat den Vorteil, dass die Trainings- bzw. Testdaten nur noch in Verbindung mit Skalarprodukten vorkommen. Für spätere Betrachtungen ist dies eine wichtige Voraussetzung. Die duale Form kann mit Hilfe von Lagrange-Multiplikatoren und Karush-Kuhn-Tucker-Bedingungen hergeleitet und gelöst werden. Im Detail kann dies in [5, Kap. 3.1] und [5, Kap. 3.2] nachgelesen werden.

Wollen wir nun ein neues Testdatum $\mathbf{x}$ klassifizieren, so muss nur bestimmt werden, auf welcher Seite der separierenden Hyperebene, die genau in der Mitte zwi-

schen H_1 und H_2 verläuft, das Datum liegt. Die korrespondierende Klasse von $\mathbf{x}$ ist dann $sgn(\mathbf{w} \cdot \mathbf{x} + b)$.

Zusammengefasst ist zu sagen, dass die Stützvektormethode (im linear separierbaren Fall) alle Trainingsbeispiele bestimmt, die auf den Hyperebenen H_1 und H_2 liegen, welche wiederum so bestimmt werden, dass der Margin zwischen den Klassen maximiert wird. Entfernt oder bewegt man (jedoch nicht über H_1 bzw. H_2 hinaus) alle Nicht-Stützvektoren und löst das Problem erneut, so erhält man dieselbe separierende Hyperebene.

2.1.3 Der nicht-separierbare Fall

Der bisher angesprochene Ansatz liefert nur dann eine Lösung, wenn die Daten linear separierbar sind. Für nicht linear trennbare Daten liefert er keine Lösung. Die Idee ist nun, dass, falls nötig, die Bedingungen (2.1) und (2.2) gelockert werden. Dazu werden die Schlupfvariablen $\xi_i, i = 1, ..., l$ in die beiden Bedingungen eingefügt:

$$\mathbf{x_i} \cdot \mathbf{w} + b \geq +1 - \xi_i \text{ für } y_i = +1 \tag{2.4}$$

$$\mathbf{x_i} \cdot \mathbf{w} + b \leq -1 + \xi_i \text{ für } y_i = -1 \tag{2.5}$$

$$\xi_i \geq 0 \forall i \tag{2.6}$$

Findet nun ein Trainingsfehler statt, so muss das zugehörige ξ_i größer als 1 sein, womit $\sum_i \xi_i$ eine obere Schranke für die Anzahl der Trainingsfehler darstellt. Anstatt $\|\mathbf{w}\|^2$ zu minimieren, kann nun $\|\mathbf{w}\|^2 + C(\sum_i \xi_i)$ minimiert werden - dadurch fliesst ein Kostenparameter C mit in die Berechnung ein. Der Kostenparameter C wird vom Benutzer gewählt: Ein großes C korrespondiert mit einer großen Bestrafung für Fehler, während folglich ein kleines C Trainingsfehler zulässt. Durch Umformungen in eine duale Form (2.7) mit den Nebenbedingungen in (2.8) kann auch dieses Problem gelöst werden, wie in [5, Kap. 3.5] gezeigt wird. Die Hyperebene ergibt sich dann gemäß (2.9). Dabei sind α_i die Lagrangemultiplikatoren, $\mathbf{x}_i$ die Trainingsbeispiele, y_i die Klasse der Trainingsbeispiele und N_S die Anzahl der Stützvektoren.

$$\max L_D \equiv \sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \tag{2.7}$$

$$0 \leq \alpha_i \leq C, \ \sum_i \alpha_i y_i = 0 \tag{2.8}$$

$$\mathbf{w} = \sum_{i=1}^{N_s} \alpha_i y_i \mathbf{x}_i \tag{2.9}$$

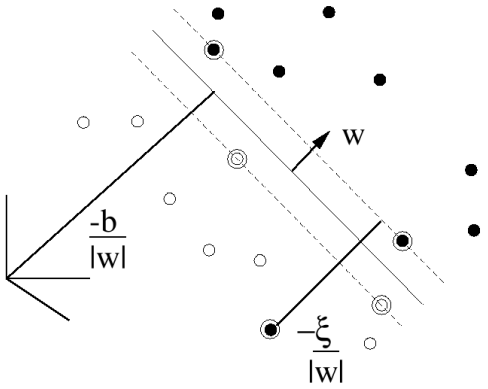

Abb. 2.2: Linear separiernde Hyperebene im nicht-separierbaren Fall.

2.1.4 Nichtlineare Support Vector Machines

Bisher wurden nur die Fälle betrachtet, in denen die gegebenen Trainingsdaten, von z.B. Ausreißern abgesehen, linear trennbar waren. Für Daten, deren zugrunde liegende Entscheidungsfunktion keine lineare Funktion ist, findet das Verfahren jedoch keine ausreichend gute Lösung. Mit einem einfachen Trick kann auch dieses Problem bewältigt werden. Wie in (2.7) zu sehen ist, kommen die Trainingsbeispiele nur in Verbindung mit Skalarprodukten $\mathbf{x}_i \cdot \mathbf{x}_j$ vor. Stellen wir uns vor, wir würden die Daten nun in einen anderen (ggf. auch unendlich dimensionalen) euklidischen Raum $\mathcal{H}$ über die Funktion Φ projizieren:

$$\phi : R^d \mapsto \mathcal{H} \tag{2.10}$$

Dadurch würde der Trainingsalgorithmus von Skalarprodukten auf den Daten in $\mathcal{H}$ abhängen, also von Ausdrücken der Form $\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$. Dazu würden die Daten über Φ in einen neuen Raum transformiert werden und dann dort über das Skalarprodukt verknüpft werden. In einem Theorem von Cover [6] wurde gezeigt, dass sich dadurch die Anzahl der möglichen linearen Trennmöglichkeiten erhöht. Gäbe es eine "Kernelfunktion" K sodass $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)$ gelten würde, so müsste nur K bekannt sein, um den Trainingsalgorithmus ausführen zu können. Φ müsste nie explizit bekannt sein. Ein Beispiel für eine solche Funktion ist:

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma^2} \tag{2.11}$$

In diesem speziellen Beispiel ist die Dimension von $\mathcal{H}$ unendlich groß. Es wäre somit unmöglich, mit Φ explizit zu rechnen. Ersetzt man im Trainingsalgorithmus jedes Vorkommen von $\mathbf{x}_i \cdot \mathbf{x}_j$ durch $K(\mathbf{x}_i, \mathbf{x}_j)$, so liefert das Verfahren eine Support Vector Machine, die in einem unendlich dimensionalen Raum arbeitet und dies noch in fast der selben Zeit, als wenn auf den nicht-projizierten Daten gearbeitet werden würde. Alle bisher gemachten Betrachtungen sind immer noch gültig, da wir weiterhin nur eine lineare Separation durchführen, aber dieses Mal in einem anderen Raum. Details zur Ersetzung von $\mathbf{x}_i \cdot \mathbf{x}_j$ durch $K(\mathbf{x}_i, \mathbf{x}_j)$ können in [5, Kap. 4] nachgeschlagen werden. Ebenso kann dort nachgelesen werden, welche Anforderungen an die Kernelfunktion K gestellt werden müssen und wie mit diesen umgegangen werden muss. Einige zur Mustererkennung gebräuchlichen Kernel sind z.B.:

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^p \tag{2.12}$$

$$K(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2/2\sigma^2} \tag{2.13}$$

$$K(\mathbf{x}, \mathbf{y}) = tanh(\kappa \mathbf{x} \cdot \mathbf{y} - \delta) \tag{2.14}$$

Das Verwenden von (2.12) führt zu einem polynomialen Klassifizierer vom Grad p. (2.13) ermöglicht einen gaußschen radiale Basisfunktion Klassifizierer und (2.14) liefert eine spezielle Art von zweischichtigem, sigmoidalen Neuronalen Netzwerk. Es sei noch angemerkt, dass die Stützvektormethode dahin erweitert werden kann, auch den Fall von mehreren Klassen zu lösen (s. [5, Kap. 4.3]).

2.1.5 Einschränkungen

Das wohl größte Problem beim Stützvektoransatz liegt in der Wahl des Kernels. Wird dieser, inklusive seiner Parameter (sofern der Kernel welche besitzt), festgehalten, hat ein SVM Klassifizierer nur noch einen Parameter, der vom Benutzer gewählt werden kann (der Parameter C zur Bestrafung für Trainingsfehler). In der Praxis sind etliche, leistungsfähige Kernels parametrisch, wodurch oft ein sehr großer Suchraum für den richtigen Kernel und dessen Parameter entsteht. Das Einbringen von A-priori Wissen über die Kernels, um so die Suche zu erleichtern, ist immer noch Gegenstand aktueller Forschung.
Während die Testphase rechentechnisch inzwischen gut zu bewältigen ist, stellt die Trainingsphase, speziell bei sehr hochdimensionalen und sehr umfangreichen Datensätzen (mehrere Millionen Stützvektoren) laut [5, Kap. 8] immer noch ein Problem dar. Inwiefern diese Aussage heute, nach 10 Jahren, noch gültig ist, müsste überprüft werden.

2.1.6 Zusammenfassung: Support Vector Machines

Die Stützvektormethode versucht die Daten in einen hoch- oder unendlichdimensionalen Raum zu projizieren um dort eine lineare Hyperebene zu finden, die das Zwei-Klassenproblem löst. Diese Projektionsfunktion muss nicht explizit bekannt sein, sondern wird durch einen entsprechenden Kernel realisiert. Durch einen Kostenparameter können beim Training Fehlklassifikationen kontrolliert zugelassen werden, um so z.B. Ausreißern in den Trainingsdaten entgegenwirken zu können. Die Wahl des Kernels, der die Daten in den höherdimensionalen Raum projiziert, und dessen Parameterfindung stellen dabei jedoch eines der größten Probleme dar.

Der Stützvektoransatz unterscheidet sich in zwei Punkten stark von vergleichbaren Ansätzen, wie z.B. Neuronalen Netzwerken: Das SVM Training findet immer das globale Minimum und die einfache geometrische Interpretierbarkeit erlaubt Untersuchungen für weitere Eingriffe, wie z.B. Parameteranpassung.

2.2 Klassifikatorkaskaden

2.2.1 Einführung

Um Objekte in Bildern oder Bildsequenzen erkennen zu können, gibt es etliche Ansätze. Paul Viola und Michael Jones veröffentlichten 2001 einen Ansatz [7], der es ermöglicht, Objekte schnell und robust zu detektieren. Basierend auf einem so genannten "Integralbild" können spezielle Features schnell berechnet werden. Der anschließend verwendete Lernalgorithmus, welcher auf AdaBoost basiert [8], wählt eine kleine Anzahl aussagekräftiger Features aus einer größeren Featuremenge aus, welche dann zum Bau von sehr leistungsfähigen Klassifikatoren genutzt werden. Um die Laufzeit des Algorithmus weiter zu verringern, wird eine so genannte "Kaskade" verwendet, die mehrere Klassifikatoren mit stetig zunehmender Komplexität kombiniert. So können z.B. Hintergrundbereiche in einem Bild schnell zurückgewiesen und dafür vielversprechendere Bildausschnitte ausführlicher untersucht werden.

2.2.2 Features

Die Objektdetektion basiert auf den Werten einfacher Features. Diese werden verwendet, da dadurch wesentlich mehr Domänenwissen erfasst wird (auf einer endlichen Trainingsdatenmenge), als wenn direkt mit Pixelwerten gearbeitet wird. Zudem ergibt sich speziell bei diesem Ansatz dadurch ein erheblicher Geschwindigkeitsvorteil, als wenn pixelbasiert gearbeitet werden würde.

Die Features basieren auf vier verschiedenen Rechteckfunktionen, die nach Haar benannten wurden, sogenannte Haarfunktionen, und werden für jede Skalierung und Position im Suchfenster berechnet. In Abb. 2.3 sind vier dieser Rechteckfunktionen abgebildet. Bei Features, die auf zwei Rechtecken basieren, berechnet sich der Wert eines Features aus der Differenz der Summen innerhalb der Rechteckflächen. Bei Features mit drei Rechtecken berechnet sich der Wert, indem die äußeren Rechtecke aufsummiert werden und anschließend diese Summe von der Summe des mittleren Rechtecks subtrahiert wird. Bei vier Rechtecken ergibt sich der Wert aus der Differenz der diagonal liegenden Flächen.

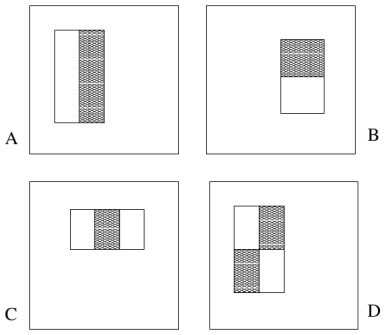

Abb. 2.3: Beispielhafte Positionierung der verschiedenen Rechtecke zur Featureberechnung.

In einem Bildfenster mit einer Auflösung von $24 * 24$ ergeben sich schon über 180,000 mögliche Features. Eine schnelle Berechnung einzelner Features ermöglicht das Integralbild. Das Integralbild liefert an der Stelle x, y die Summe der Pixelwerte über und links von x, y:

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

wobei $ii(x, y)$ das Integralbild und $i(x, y)$ das ursprüngliche Bild sind. Durch eine rekursive Berechnung kann das Integralbild in einem Durchgang berechnet werden [7, Abschnitt 2.1]. Nun kann die Summe eines beliebigen Rechtecks mit vier Arrayzugriffen berechnet werden und dadurch die eigentlichen Features mit 6 bis 9 Zugriffen.

2.2.3 Lernvorgang

Anhand eines gegebenen Sets an Features für positive und negative Bildbeispiele könnte nun ein beliebiger Klassifikator trainiert werden. In diesem Fall wird eine Variante von AdaBoost verwendet, die sowohl eine kleine Anzahl an Features aussucht, als auch den Klassifikator trainiert. In seiner ursprünglichen Form

kombiniert der AdaBoost Lernalgorithmus eine große Anzahl schwacher Klassifikatoren, um daraus einen leistungsfähigen Klassifikator zu erhalten.
Wie angesprochen gibt es pro Bildfenster mit $24 * 24$ Pixel Auflösung wesentlich mehr Features (über 180,000) als Pixel. Auch wenn jedes Feature effizient berechnet werden kann, so ist die vollständige Berechnung immer noch zu aufwändig. Um einen leistungsfähigen Klassifikator zu erstellen, reicht jedoch eine kleinere Anzahl dieser Features. Die Problematik liegt im Finden dieser aussagekräftigen Features.
Ein möglicher Lösungsansatz liegt darin, das Rechteck, bzw. das Feature zu finden, das die positiven und negativen Beispiele am besten separiert. Dies geschieht durch einen schwachen Lerner, der für jedes Feature den optimalen Schwellwert zur Klassifizierung bestimmt, sodass die Falschklassifikationsrate minimiert wird. Im Detail kann der AdaBoost Algorithmus in [7, Abschnitt 3] nachgelesen werden. In der Praxis kann ein einzelnes Feature keine Klassifikation mit niedrigem Fehler erreichen. Aussagekräftige Features haben eine Fehlerrate zwischen 0.1 und 0.3, nicht ganz so aussagekräftige zwischen 0.4 und 0.5.
Konstruiert man nun einen Detektor für frontale Gesichtsaufnahmen mit 200 Features, so lassen sich Detektionsraten von 95% bei einer False Positive Rate von 1 zu 14084 erreichen [7, Abschnitt 3.2]. Um höhere Genauigkeiten zu erreichen, können mehr Features betrachtet werden, was aber direkt die Berechnungszeit erhöht.

2.2.4 Kaskade

Um die Detektionsperformance weiter zu steigern und zugleich die Rechenzeit zu reduzieren wird eine Kaskade verwendet. Die Idee dahinter ist, etliche der negativen Bildausschnitte sofort abzuweisen, jedoch alle positiven zu behalten (indem z.B. der Schwellwert der Featureklassifikatoren so eingestellt wird, dass die False Negative Rate nahe bei Null liegt). Auf die verbleibenden Features können anschließend komplexere Klassifikatoren angewendet werden, um niedrige False Positive Rates zu erreichen.

Eine Kaskade besteht aus mehreren Stufen. In jeder Stufe findet eine Klassifikation auf einer bestimmten Anzahl von Features statt, die mit zunehmender Stufenzahl größer wird. Liefert die erste Stufe der Kaskade ein positives Ergebnis, so wird die Eingabe an die zweite Stufe übergeben und dort ausgewertet. Bei abermaligem positivem Ergebnis erfolgt eine Übergabe nach Stufe 3, usw. Liefert eine Stufe kein positives Ergebnis, so wird das Bildfenster verworfen und zum nächsten Fenster übergegangen. Liefert die letzte Stufe ein positives Ergebnis, so handelt es sich mit hoher Wahrscheinlichkeit um ein korrekt detektiertes Objekt. In Abb. 2.4 wird der Vorgang schematisch dargestellt. Die einzelnen Stufen werden so trainiert, dass sie die Anzahl an False Negatives minimieren. Die genaue Prozedur des Kaskadentrainings kann in [7, Abschnitt 4] nachgelesen werden.

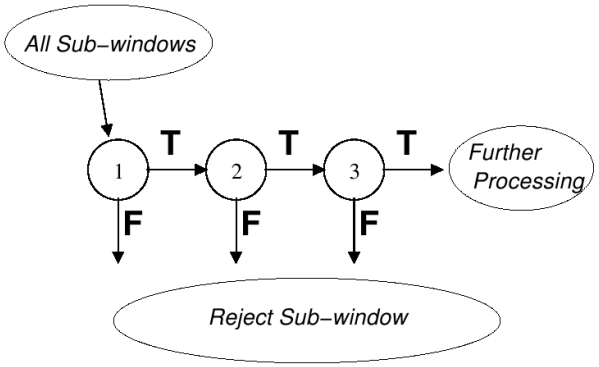

Abb. 2.4: Schematische Zeichnung der Klassifikatorkaskade

2.2.5 Zusammenfassung: Klassifikatorkaskade

Der Ansatz von Viola & Jones ermöglicht es, zuvor eingelernte Objekte in Bildern zu erkennen. Der Algorithmus hat sich in der Praxis speziell für die Gesichtsdetektion bewährt, was vor allem darauf zurückzuführen ist, dass er echtzeitfähig ist und zudem, gerade bei Gesichtern, akzeptable Erkennungsraten liefert.

3 Vorverarbeitung

Im folgenden Kapitel wird beschrieben, wie die Vorverarbeitung der gegeben Bilddaten stattfindet und wie daraus ein Featurevektor entsteht, der für das Training eines Klassifikators zur gesichtsbasierten Geschlechtsbestimmung verwendet werden kann.
Für alle Implementierungen wurde C bzw. C++ verwendet. Für Grafikoperationen wurde OpenCV in der Version 1.0.0 [9] eingebunden.

3.1 Anpassung der Gesichter

Ein Ziel der Vorverarbeitung ist es, Gesichter in einem Bild zu detektieren. Anschließend ist jedoch eine weitere Verarbeitung der Gesichter notwendig, um z.B. Rotation und Skalierung der Gesichter auszugleichen. Auch die durch unterschiedliche Aufnahmeorte gegebene Variation in der Beleuchtung soll in diesem Schritt gemindert werden.

3.1.1 Gesichtsdetektion

Um Gesichter in Bildern zu detektieren wird die mit OpenCV [9] gelieferte Gesichtskaskade verwendet. Diese wurde entsprechend Abschnitt 2.2 trainiert und stellt eine ausreichend hohe Erkennungsrate mit wenigen False Positives bereit. Neben Frontalaufnahmen entdeckt die Kaskade auch noch Gesichter, die um kleine Winkel beliebig in alle drei Dimensionen gedreht sein können. Auf Bildern der Größe $320 * 240$ Pixeln dauert der Suchlauf mit modernen Rechnern wenige Millisekunden, wodurch Echtzeitfähigkeit (15 Bilder/Sekunde) gewährleistet ist.
Für die hier vorliegenden Eingabebilder war gegeben, dass nur eine Person pro Bild sichtbar war. Detektiert die Kaskade trotzdem mehrere Gesichter (folglich False Positives), so wird die größte Detektion gewählt.

3.1.2 Augendetektion

Nachdem das Gesicht im Bild gefunden wurde, wird dazu übergegangen die Augen zu detektieren. Dies geschieht aus dem Grund, da, wie angesprochen, die Ge-

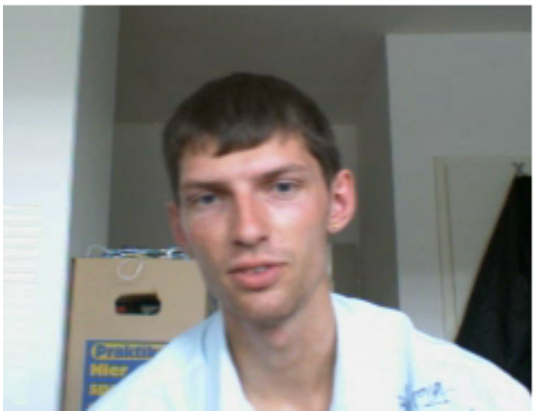

Abb. 3.1: Eingabebild

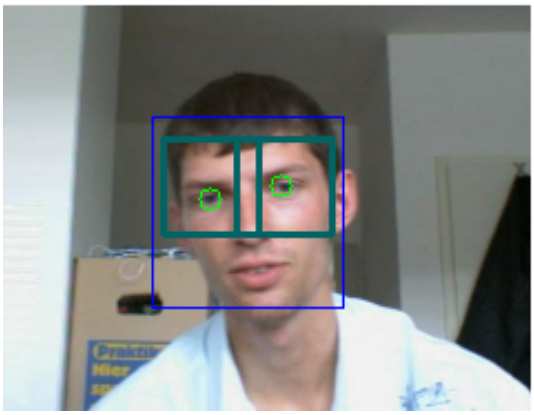

Abb. 3.2: Detektionen

sichtskaskade auch leicht gedrehte Gesichter detektiert. Speziell Rotation in der Gesichtsebene kann später bei der Klassifikation ein Problem darstellen.
Um die Augen zu detektieren wird in einem Rechtecktbereich im Augenbereich für das linke und rechte Auge jeweils eine eigenen Kasakade angesetzt. Die dafür verwendeten Kaskaden wurde von Castrillón-Santana et al. [10] erstellt. Ein typisches Eingabebild wird in Abb. 3.1 gezeigt. Der Suchbereich für die Augenkaskaden wurde experimentell ermittelt und ist in Abb. 3.2 abgebildet: Das gefundene Gesicht ist blau eingezeichnet, die Augensuchbereiche in türkis und die detektierten Augen sind hellgrün eingekreist. Eine Verwendung einer Region-of-Interest hat hierbei den Vorteil, dass der Suchraum klein gehalten wird, wodurch die Detektion beschleunigt wird. Zum anderen sind die Augen in den meisten Fotos nur wenige Pixel groß und schwer zu erfassen. Ein Anwenden der Detektoren auf das komplette Gesicht würden in vielen Fällen False Positives ergeben.

Die verwendeten Augendetektoren haben den Vorteil, dass sie die Augenposition sehr genau bestimmen, wodurch es später möglich wird, das Bild so zu drehen, dass beide Augen auf einer horizontalen Geraden liegen. Es ist jedoch anzumerken, dass auch diese Detektoren die Augenpositionen nicht fehlerfrei bestimmen. In den meisten Fällen reicht diese aber aus. Werden im Bild keine Augen gefunden, dann wird das Bild verworfen. Dies ist durchaus vertretbar, da später auf Bildsequenzen klassifiziert wird - das Verwerfen von wenigen Frames fällt bei einer ausreichend großen Gesamtbildzahl kaum ins Gewicht. Zudem ist ohne eine korrekte Augendetektion kein gutes Alignment, wie es im nächsten Kapitel besprochen wird, möglich.

3.1.3 Alignment

Da nun die Augenpositionen bekannt sind, kann das Bild in der Ebene rotiert werden, sodass die Augen auf einer horizontalen Linie zu liegen kommen. Bei der

alleinigen Gesichtsdetektion kommt es hin und wieder vor, dass ein unterschiedlich großer Bereich des Gesichts erkannt wird. So wird bei manchen Gesichtern deutlich mehr Hintergrund in das Detektionsfenster eingeschlossen als bei anderen. Dies ist natürlich nicht wünschenswert, da es sich bei dem Hintergrund nicht um "Gesichtsinformation" handelt. Da jedoch die Augenpositionen bekannt sind, wird nun ein neues Bild mit fester Größe erstellt, in dem die Augen in Zeile y zu liegen kommen, den Abstand d Pixel besitzen, und symmetrisch zur mittleren Bildspalte positioniert sind. Dazu wird das zuvor rotierte Bild so skaliert, dass die Augen den absoluten Abstand von d Pixel besitzen und anschließend mit der gewünschten Größe und Position so ausgeschnitten, dass die Augen an der entsprechenden Stelle zu liegen kommen. In Abb. 3.3 sieht man im Bild rechts das Resultat einer solchen Anpassung. Das Bild links zeigt das ursprüngliche Eingabebild, das Bild in der Mitte abermals die Detektionen mit den Suchbereichen für die Augen.

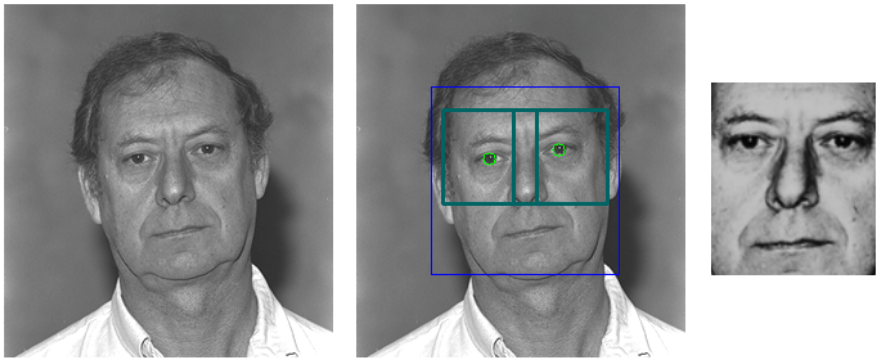

Abb. 3.3: Eingabebild, Detektionen und Suchbreiche, Angepasstes Gesicht

3.2 Merkmalsextraktion

Das zuvor angepasste Bild wird nun in ein Grauwertbild konvertiert und dann weiter verarbeitet, um daraus einen Featurevektor zu erstellen, der für das Training eines Klassifikators verwendet werden kann. An den Featurevektor wird die Anforderung gestellt, das zuvor erhaltene Bild möglichst gut in Hinsicht auf Trennbarkeit von Mann und Frau zu repräsentieren. Mögliche Repräsentationen können z.B. holistisch oder lokal fundiert sein. Auch sind verschiedene Methoden zur Dimensionsreduktion, wie z.B. DCT oder PCA denkbar. Im Folgenden werden die im nächsten Kapitel getesteten Techniken kurz angesprochen.

3.2.1 Holistisch

Die einfachste Möglichkeit das angepasste Eingabebild zu repräsentieren besteht darin, die Pixelwerte nacheinander auszulesen und diese zu einem Vektor zu konkatenieren. Da die verschiedenen Eingabebilder aber in Helligkeit und Kontrast stark variieren können, ist vor dem Konkatenieren eine weitere Vorverarbeitung wichtig. Hierfür bietet sich z.B. eine Histogrammegalisierung an, die auf das angepasste Gesichtsbild angewendet wird. Diese versucht die Summe der kumulierten Grauwerte zu linearisieren. In Appendix A.1 ist dieses Verfahren näher beschrieben. Anschließend kann aus dem egalisierten Bild der Featurevektor wie zuvor beschrieben extrahiert werden. Ein Vorteil der holistischen Methode liegt darin, dass bei der Repräsentation keine Information verloren geht. Andererseits ist dadurch im Bild auch noch irrelevante Information vorhanden. Trotz der Histogrammegalisierung ist dieser Ansatz kaum robust gegenüber ungleichen Beleuchtungsverläufen auf dem Gesicht, da diese sich sofort auf den Featurevektor auswirken. Ebenso findet keine Reduktion der Dimension oder Extraktion von relevanten Merkmalen statt, was unter Umständen dazu führen kann, dass die Klassifikation von Mann und Frau nur zufriedenstellend eingelernt werden kann, wenn sehr viele Trainingsbeispiele vorliegen. In Abb. 3.4 sind einige fertig verarbeitete Gesichter gezeigt. Die Auflösung beträgt $24 * 32$ Pixel pro Gesichtsbild. In der oberen Reihe wurden Frauen abgebildet, in der unteren ausschließlich Männer.

Abb. 3.4: Gesichtsbilder aus dem holistischen Ansatz

3.2.2 Holistisch auf Halbgesichtern

Die Gesichtshälften (vertikal geteilt) einer Person sind nicht vollkommen symmetrisch, enthalten also nicht nur redundante Information. Der Mensch ist aber dennoch in der Lage, auch auf einem Halbgesicht das Geschlecht bestimmen zu können. Es ist also denkbar, das angepasste Gesicht zu teilen und die linke und rechte Gesichtshälfte einzeln zu betrachten (eine davon sollte dann gespiegelt werden). Anschließend kann wie im zuvor genannten holistischen Verfahren vorgegangen werden: Es wird zuerst eine Histogrammegalisierung durchgeführt und danach werden die einzelnen Pixelwerte zu einem Featurevektor konkateniert. Eine stark seitliche Beleuchtung kann eventuell durch die Histogrammegalisierung

pro Seite besser ausgeglichen werden. Im Hinblick auf die Beleuchtung können dadurch eventuell einheitlichere Ergebnisse erzielt werden. Die Dimension der Daten halbiert sich im Vergleich zum holistischen Ansatz, während sich die Anzahl der Trainings- und Testdaten verdoppelt. Die Information, die in den gesamten Daten gespeichert ist, verdoppelt sich jedoch nicht, da es sich immer noch um die gleichen Bilder handelt. In Abb. 3.5 werden einige der fertig verarbeiteten Gesichter gezeigt. Die Auflösung pro Gesichtshälfte beträgt hier $12 * 32$ Pixel. Die obere Reihe zeigt ausschließlich Frauen, die untere nur Männer. Es wurden immer beide Gesichtshälften einer Person abgebildet.

Abb. 3.5: Gesichtsbilder aus dem holistischen Ansatz auf Halbgesichtern

3.2.3 Lokale DCT

Ist das angepasste Eingabebild quadratisch und hat eine Breite die durch 8 restlos teilbar ist, so lässt sich das Bild in $n * n$ Fenster unterteilen, die jeweils $8 * 8$ Pixel Ausmaß haben. Anschließend führt man für jedes Fenster lokal eine diskrete Cosinustransformation durch. Danach werden die für jedes Fenster, welches auch Block genannt wird, erhaltenen Koeffizienten nach einem vorgegebenen "Zick-Zack-Muster" konkateniert. Um eine Dimensionsreduktion durchzuführen werden je Block nur die ersten k Koeffizienten verwendet. Auch kann z.B. der erste Koeffizient, der den mittleren Grauwert beschreibt, weggelassen werden. Dies erhöht in manchen Fällen die Robustheit gegenüber schlechter Beleuchtung. In [11] wird die lokale DCT und die anschließende Featureextraktion im Detail beschrieben und für das Problem der Gesichtsidentifikation ausgewertet. Das gesamte Vorgehen ist in Abb. 3.6 vereinfacht schematisch dargestellt. Ein Vorteil dieser Methode liegt darin, dass die Dimension der Eingabedaten drastisch verkleinert werden kann, ohne dass zu viel relevante Information verloren geht.

3.2.4 Hauptkomponentenanalyse

Betrachtet man den holistischen Ansatz, so beschreibt jedes Gesichtsbild einen Punkt in einem hochdimensionalen Raum. Alle diese Bilder haben jedoch eine gewisse Ähnlichkeit und werden nicht zufällig verstreut in diesem Raum lie-

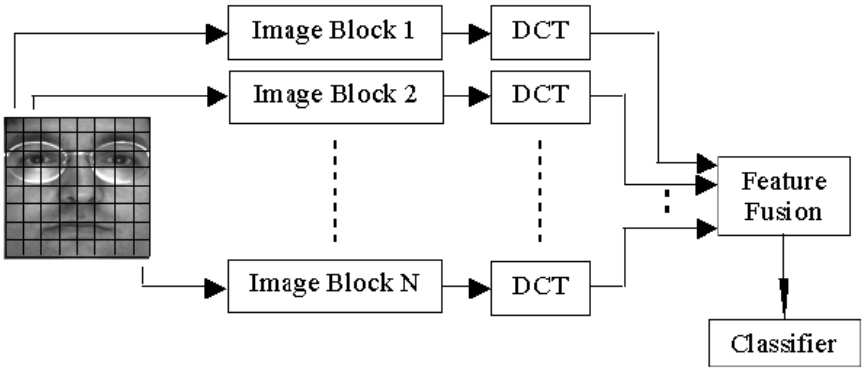

Abb. 3.6: DCT-Featureextraktion (Quelle: [11])

gen. Folglich können sie durch einen niedrigdimensionaleren Unterraum beschrieben werden. Die Grundidee der Hauptkomponentenanalyse liegt darin, diejenigen Vektoren zu finden, die die Verteilung der Gesichter in diesem Raum am Besten beschreiben. Diese Vektoren spannen dann einen Unterraum von Gesichtern auf und ein Gesichtsbild kann aus einer Linearkombination dieser neuen Vektoren zusammengesetzt werden. Im Detail sind diese Vektoren die Eigenvektoren der Kovarianzmatrix bzgl. der ursprünglichen Gesichter. Die ersten k Vektoren weisen die größte Varianz auf und beinhalten veranschaulicht somit den größten "Informationsgehalt". Werden nur die ersten k Vektoren verwendet, so kann ein niedrigdimensionaler Merkmalsraum erhalten werden, der jedoch noch ähnlich viel Information wie der zuvor betrachtete Raum enthält. Im Detail kann die Hauptkomponentenanalyse in Verbindung mit Gesichtsidentifikation in [12] nachgelesen werden.

4 Experimente

In diesem Kapitel werden die durchgeführten Experimente beschrieben und ausgewertet. Die verschiedenen Features, die in Abschnitt 3.2 vorgestellt wurden, werden nun auf drei verschiedenen Datensätzen evaluiert, um einen Überlick über ihre Funktionalität bezüglich der gesichtsbasierten Geschlechtserkennung zu erhalten.

4.1 Übersicht über die Datensätze

Es folgt eine kurze Übersicht über die einzelnen Datensätze, die für die Auswertung verwendet wurden.

4.1.1 FERET Datensatz

Die Bilder im FERET Datensatz wurden alle unter kontrollierten Lichtverhältnissen aufgenommen und zeichnen sich durch eine hohe Bildqualität aus. Pro Person wurden mehrere Aufnahmen gemacht, jedoch wurden für den hier zusammengestellten Datensatz nur die primären Frontalaufnahmen aus dem Unterdatensatz "fa" verwendet. Der erstellte Datensatz setzt sich aus 556 Männern und 397 Frauen zusammen und beinhaltet von jeder Person nur ein Bild. Einige Beispiele sind in Abb. 4.1 zu sehen.

Abb. 4.1: Einige Beispiele aus dem FERET Datensatz

4.1.2 Internetdatensatz

Im FERET Datensatz ist die Auflösung der Bilder sehr hoch und die Beleuchtung bzw. die Kopfpose meist einheitlich. Um die Verfahren auf schlechteren Daten zu testen, wurde ein neuer Datensatz erstellt. Um in kurzer Zeit an viele Bildbeispiele zu gelangen wurde in der Google Bildersuche ([13]) nach "Gruppenfotos" (und ähnlichen Begriffen) gesucht und eine Vielzahl (ca. 150) hochaufgelöster Bilder im Jpeg Format gesammelt. Auf den Bildern sind zumeist etwa fünf bis 50 Personen abgebildet. Die Bilder entstanden an beliebigen Orten (Innen- und Außenbereich) und unterliegen stark variierender Beleuchtung. In Abb. 4.2 sind einige dieser Bilder in verkleinerter Form zu sehen (die ursprüngliche Größe pro Bild lag etwa zwischen 2000 und 3500 Pixeln Breite). Einige Beispiele der extrahierten Gesichter werden in Abb. 4.3 abgebildet. Etliche der Fotos weisen kamerabedingt starkes Rauschen, Bewegungsunschärfe und Kompressionsartefakte auf. Der endgültige Datensatz umfasst 642 Männer und 581 Frauen.

Abb. 4.2: Einige Gruppenfotos aus denen die Gesichter extrahiert wurden

Abb. 4.3: Einige extrahierte Gesichter für den Internetbilderdatensatz

Die Gesichter in den Bildern wurden mit einem kleinen Programm detektiert, ausreichend groß ausgeschnitten (um später ggf. eine Drehung bei der Vorverarbeitung vornehmen zu können) und dann von Hand gelabelt. Stark gedrehte oder durch die Linse verzerrte Gesichter wurden dabei von Hand entfernt. Auch wurden Gesichter von Personen, bei denen das Geschlecht nicht eindeutig bestimmt werden konnte, verworfen.

4.1.3 Webcam Datenbank

Die Datenbank mit an der Universität Karlsruhe gesammelten Bildersequenzen wurde ausschließlich zum Testen verwendet. Sie umfasst Bildsequenzen von 60 Männern und 40 Frauen. Jede Sequenz besteht aus 150 Bildern, in denen die

Datensatz	Bilder (m)	Bilder (f)	Beleuchtung	Bildqualität	Aufnahmeort
FERET	556	397	Kontrolliert	++	Fest
Internet	642	581	Ungleichmäßig	+/-	Beliebig
Webcam	60*	40*	Ungleichmäßig	+/-	Beliebig

*Anzahl der Personen (jeweils ca. 150 Bilder)

Tabelle 4.1: Die verschiedenen Datensätze

Gesichtskaskade ein Gesicht detektiert hat. Alle Aufnahmen wurden mit einer Philips Webcam gemacht. Die Aufnahmen wurden fast alle an verschiedenen Orten durchgeführt und variieren daher stark in der Beleuchtung und beinhalten teilweise Bewegungsunschärfe. Die einzelnen Gesichter in den Bildern sind unterschiedlich groß und reichen von ca. 20×20 bis 150×150 Pixel. In Abb. 4.4 sind einige Frames aus den Bildsequenzen von fünf verschiedenen Personen abgebildet.
Tabelle 4.1 stellt die verschiedenen Datensätze nochmal kurz gegenüber.

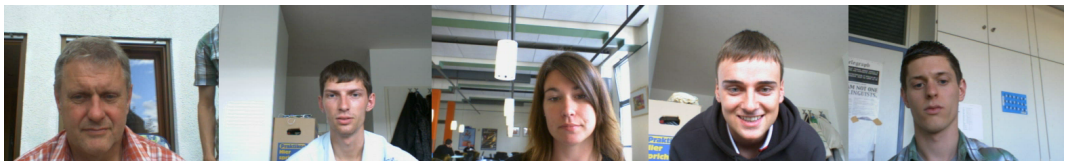

Abb. 4.4: Frames aus den Bildsequenzen des Webcam-Testdatensatz

4.2 Evaluation der verschiedenen Methoden

4.2.1 Einleitung

Die Erkennungsraten hängen von einer Vielzahl von Parametern ab. Davon das globale Optimum zu bestimmen ist ohne sehr großen Rechenaufwand nur schwer möglich. Einige Parameter, bei denen es plausibel erscheint, wurden daher unter festen Bedingungen maximiert um so den Suchraum einzuschränken. Dies wird im Folgenden näher erklärt.

Als Implementierung für die Support Vektor Machine wurde LIBSVM Version 2.86 [14] mit OpenMP-Unterstützung verwendet. Alle Werte der hier aufgezeigten Experimente wurden mit einem RBF-Kernel (4.1) ermittelt.

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2}, \gamma > 0 \tag{4.1}$$

d	y	Rate ohne HE*	Rate mit HE*	Rate** mit HE*
72	45	91.7	**93.6** (94.4m, 92.4f)	91.9 (92.4m, 91.4f)
72	32	91.8	92.2	91.2
60	45	91.6	92.9	92.8
60	32	92.1	93.4	92.6

*HE=Histogrammegalisierung ** 397 Männer, 397 Frauen

Tabelle 4.2: Erkennungsraten mit und ohne Histogrammegalisierung (auf FERET)

Die Erkennungsraten für den FERET- und Internetbilderdatensatz wurden in einer 5-maligen Kreuzvalidierung in Verbindung mit einer Gittersuche für den Kostenparameter C und den Kernelparameter γ ermittelt. Dazu wurden die Daten in fünf gleichgroße (sofern dies möglich war) Mengen zufällig aufgeteilt. Dann wurde jede Menge einmal zum Testen verwendet, während mit den übrigen vier Mengen trainiert wurde. Die Erkennungsrate A ergab sich dann gemäß (4.2). Für jeden der fünf Durchgänge wurden die Testbeispiele gemäß dem Verhältnis von Männern und Frauen im Trainingsset gewählt. Die Gittersuche wurde für alle Kombinationen $(C \times \gamma)$ mit $C \in (2^1, 2^2, ..., 2^{16})$ und $\gamma \in (2^{-15}, 2^{-14}, ..., 2^1)$ durchgeführt. C und γ wurden für jede hier vorgestellte Erkennungsrate neu bestimmt.

$$A = \frac{1}{2}\frac{\sum Correct_{Males}}{\sum Total_{Males}} + \frac{1}{2}\frac{\sum Correct_{Females}}{\sum Total_{Females}} \tag{4.2}$$

4.2.2 Holistische Methode

In einem ersten Versuch wurde geprüft, ob Histogrammegalisierung bei dem holistischen Ansatz von Vorteil ist. Da die Trainingsdaten nicht balanciert waren, wurde noch geprüft, ob es von Vorteil ist, gleich viele Männer wie Frauen für das Training zu verwenden. In den Klammern wurden die Erkennungsraten für Männer und Frauen angegeben. Das Bildfenster für die angepassten Gesichter im Alignmentschritt betrug $128*150$ Pixel und wurde nach erfolgter Vorverarbeitung auf $24 * 32$ Pixel verkleinert. In Tabelle 4.2 wurden die Erkennungsraten für einige Augenabstände d und Augenzeilen y im Alignmentschritt abgedruckt. Diese beiden Parameter wurden jedoch im Folgenden noch genauer ermittelt.

Wie zu erkennen ist, verbessert eine Histogrammegalisierung das Ergebnis. Im Folgenden wurde deshalb bei den holistischen Ansätzen stets mit Histogrammegalisierung gearbeitet. Auch scheint es nicht von Nachteil, unbalanciert mit mehr Daten anstatt balanciert, aber dafür mit weniger Daten, zu trainieren. Mehr Trainingsdaten ermöglichen vermutlich eine bessere Trennhyperebene der SVM, wo-

durch indirekt auch die weniger gut repräsentierte Klasse eingelernt wird.

In einem weiteren Versuch wurde überprüft, wie sich verschiedene Augenabstände d (und damit auch die Skalierung bzw. Größe des Gesichtes) in Kombination mit der vertikalen Augenposition y im Detail auf die Erkennungsraten auswirken. Ein Augenabstand von $d = 62$ Pixel und Augenzeile $y = 50$ lieferten die höchsten Erkennungsraten, jedoch war die Abweichung zu anderen Parameterkombinationen gering. Eine Tabelle mit den Erkennungsraten (für den Internetbilderdatensatz) befindet sich im Anhang in Abschnitt B.2. Für alle folgenden Experimente betrug ab nun $d = 62$ Pixel und $y = 50$.

Mit LIBSVM wurde ein Python Skript geliefert, dass für jedes Feature (in diesem Fall für jedes Pixel) seine Relevanz für die Klassifikationsaufgabe schätzt. Details zur Implementierung finden sich in [15]. Für den FERET Datensatz wurde eine solche Schätzung durchgeführt, welche in Abb. 4.5 visualisiert wurde. Helle Bereiche zeichnen sich durch einen hohen Score aus, beschreiben also eine hohe Relevanz, dunkle dagegen eine niedrige. Das linke Bild beschreibt die $24*32 = 768$ Features und wurde für eine bessere Erkennbarkeit vergrößert. In den vier Bilden rechts davon sind vier Beispielgesichter abgebildet, um zu verdeutlichen, welche Bereiche von Bedeutung sind.

Abb. 4.5: Links: Geschätzte Relevanzen der Pixel (hell $\hat{=}$ relevant)

Wie zu erkennen ist, sind die ausschlaggebenden Regionen z.B. die Stirnseiten, der Bereich zwischen Augen und Augenbrauen und der Mund- bzw. untere Nasenbereich. Kaum relevant sind dagegen z.B. die Wangen.

Zuletzt blieb zu untersuchen, wie sich verschiedene Auflösungen auf die Erkennungsraten auswirkten. Wie in Tabelle 4.3 zu erkennen ist, verändern sich die Erkennungsraten mit einer Reduzierung der Auflösung nicht drastisch. Moghaddam et al. [1] berichteten in ihrer Publikation ähnliche Ergebnisse.

Holistische Methode auf Gesichtshälften

In diesem Versuch wurde von jeder Person nur eine Gesichtshälfte für die Erkennung verwendet. Wie vermutet verschlechtert sich die Erkennungsrate leicht, wie

Auflösung	Rate (Feret)	Rate (Internet)
$8 * 12$	89.1	88.9
$16 * 24$	92.4	92.0
$24 * 32$	93.1	**93.5**
$32 * 48$	**93.7**	92.5[1]

Tabelle 4.3: Erkennungsraten bei verschiedenen Auflösungen (Holistischer Ansatz)

Datensatz	Halbgesicht	Gesamtes Gesicht
FERET	92.0	93.0
Internet	90.3	93.4

Tabelle 4.4: Erkennungsraten für den holistischen Ansatz auf Halbgesichtern

in Tabelle 4.4 zu sehen ist. Der Augenabstand d betrug bei diesem Experiment 62 Pixel und die Augenzeile y wurde wie zuvor ermittelt auf 50 gesetzt. Die Bildgröße einer Gesichtshälfte, die zur Klassifikation verwendet wurde, betrug $12 * 32$ Pixel. Zum Vergleich werden noch die Erkennungsraten für das gesamte Gesicht angegeben.

Lokale DCT

Für die lokale DCT wurden die angepassten Gesichter auf $32 * 32$ bzw. $64 * 64$ Pixel skaliert. Nach einer Histogrammegalisierung wurde auf $8 * 8$ Pixeln großen Fenstern eine DCT durchgeführt. Auch hier war eine Histogrammegalisierung von Vorteil. Die besten Erkennungsraten ergaben sich, wenn der erste Koeffizient entfernt wurde. Mit dem Entfernen von weiteren führenden Koeffizienten sank die Erkennungsrate jedoch ab. Experimente mit $n = \{4, 5, 8, 16\}$ Koeffizienten zeigten, dass ab fünf Koeffizienten die Erkennungsraten nicht mehr zunahmen. Die Ergebnisse der lokalen DCT mit fünf Koeffizienten pro Block sind in Tabelle 4.5 zu sehen. In der Spalte "*Bildgröße*" wurde noch in den Klammern die Dimension des Featurevektors angegeben.

[1] Ein Zurückgehen der Erkennungsrate lag eventuell daran, dass einige Gesichter im Datensatz sehr klein waren und im letzten Schritt somit nicht auf $32 * 48$ verkleinert, sondern vergrößert wurden.

Bildgröße (Dim)	Datensatz	Erkennungsrate
$32 * 32$ (80)	Feret	90.6
$64 * 64$ (320)	Feret	**94.1**
$32 * 32$ (80)	Internet	89.9
$64 * 64$ (320)	Internet	**92.2**
$64 * 64$ (320)	Feret & Internet	93.2

Tabelle 4.5: Erkennungsraten für den DCT-basierten Ansatz

Datensatz	# Eigenvektoren	Erkennungsrate
FERET	25	91.4
FERET	50	**92.9**
FERET	100	92.7
Internet	25	88.7
Internet	50	**91.6**
Internet	100	91.6

Tabelle 4.6: Erkennungsraten für den PCA-basierten Ansatz

Lokale PCA

Für die PCA wurden die ersten n Eigenvektoren verwendet. Wurde der erste Eigenvektor entfernt, so nahm die Erkennungsrate ab. Wie in Tabelle 4.6 zu sehen ist, erreicht das Verfahren trotz der sehr niedrigen Merkmalsraumdimension hohe Erkennungsraten.

4.2.3 Webcam-Datensatz

Im Folgenden wurde die gesammelte Webcam-Datenbank evaluiert. Es hat sich gezeigt, dass für hohe Erkennungsraten (über 90%) eine große Anzahl (weit über 100) an verschiedenen Personen im Trainingsset benötigt wird. Deshalb wurde auf dem FERET- bzw. Internetbilderdatensatz trainiert und dann auf den Webcamdaten getestet. Mit einer Kreuzvalidierung in Verbindung mit einer Gittersuche wurden die optimalen SVM-Parameter C und γ für die Trainingsdaten ermittelt. Anschließend wurden die Webcam-Testdaten mit dieser Support Vector Machine klassifiziert. Durch dieses Vorgehen ergaben sich unterschiedliche Bedingungen zwischen Trainings- und Testdaten. In Tabelle 4.7 sind die Erkennungsraten der verschiedenen Verfahren gezeigt. Bei dem holistischen Ansatz bot es sich an, die Trainingsdaten auch noch gespiegelt in den Trainingsdatensatz aufzunehmen. Auch wurden zusätzlich noch künstliche Trainingsdaten erzeugt, indem die detektierten Augen leicht verschoben wurden. Dazu wurde angenommen, dass beide

Trainingsset	Features	Rate Sub.	%Frames	Rate (M)	Rate (F)
Feret	Holistisch	85/100	75.8	81.0 (52/60)	69.8 (33/40)
Feret+Mirror	Holistisch	84/100	75.8	82.8 (53/60)	67.5 (31/40)
Feret+Mi+Mo	Holistisch	85/100	77.8	79.7 (51/60)	75.6 (34/40)
Internet	Holistisch	**86/100**	82.6	88.3 (56/60)	75.9 (30/40)
Internet+Mirror	Holistisch	85/100	82.5	84.2 (53/60)	80.5 (32/40)
Internet+Mi+Mo	Holistisch	85/100	**83.9**	85.6 (53/60)	82.0 (31/40)
Beide+Mi+Mo	Holistisch	84/100	82.7	85.8 (53/60)	79.1 (31/40)
Feret	Hol. Hälften	83/100	73.2	75.1 (52/60)	70.9 (31/40)
Feret+Moved	Hol. Hälften	85/100	74.1	75.9 (53/60)	71.9 (32/40)
Internet	Hol. Hälften	85/100	79.0	90.4 (57/60)	65.7 (28/40)
Internet+Moved	Hol. Hälften	87/100	80.5	90.4 (57/60)	69.0 (30/40)
Beide+Mo	Hol. Hälften	**89/100**	**81.7**	86.8 (55/60)	75.7 (34/40)
Feret	DCT	78/100	72.0	66.3 (42/60)	78.6 (36/40)
Internet	DCT	78/100	75.5	91.6 (55/60)	56.7 (23/40)
Feret&Internet	DCT	**87/100**	**80.7**	87.7 (54/60)	72.6 (33/40)
Feret&Internet	PCA	**83/100**	**77.8**	86.4 (54/60)	67.7 (29/40)

Tabelle 4.7: Erkennungsraten auf dem Webcam-Datensatz

Augen gleichzeitig jeweils ein Pixel links, rechts, oben bzw. unten, relativ zur Ausgangsposition, detektiert wurden. Anschließend wurde der Anpassungsalgorithmus für das Gesicht mit den verschobenen Positionen neu ausgeführt.
In Tabelle 4.7 ist das Ergebnis der Auswertung zu sehen. *Rate Sub.* zeigt, wie viele der Personen richtig klassifiziert worden sind. Eine Person galt als richtig klassifiziert, wenn mehr als 50% ihrer Frames richtig geschätzt wurden. *%Frames* beschreibt die Erkennungsrate auf der gesamten Bildmenge (12801 Frames). *Rate (M)* gibt den Prozentsatz der korrekt klassifizierten Frames der Männer wieder, *Rate (F)* entsprechend den, der Frauen. In den Klammern wurde noch angegeben, wie viele Männer bzw. Frauen richtig klassifiziert worden sind. Das Datensatzattribut "*Mirror*" (*Mi*) bedeutet, dass auch mit gespiegelten Trainingsdaten eingelernt wurde, "*Moved*" (*Mo*) bedeutet, dass mit den zusätzlichen, durch Verschiebung erzeugten, Daten trainiert wurde.

In Abb. 4.6 werden die Prozentangaben der korrekt klassifizierten Frames pro Person gezeigt. Es wurden einige der Kombinationen aus Tabelle 4.7 abgebildet.

Für den holistischen Ansatz auf Halbgesichtern wurde noch untersucht, wie sich weniger Frames auf die Erkennungsraten auswirken. Auf der horizontalen Achse ist in Abb. 4.7 die Anzahl der verwendeten Frames pro Person angegeben, auf der vertikalen die Anzahl der korrekt klassifizierten Personen. Es sei angemerkt, dass nicht bei jeder Person 150 Frames zur Klassifikation genutzt wurden, da in einigen Fällen kein zuverlässiges Alignment stattfinden konnte.

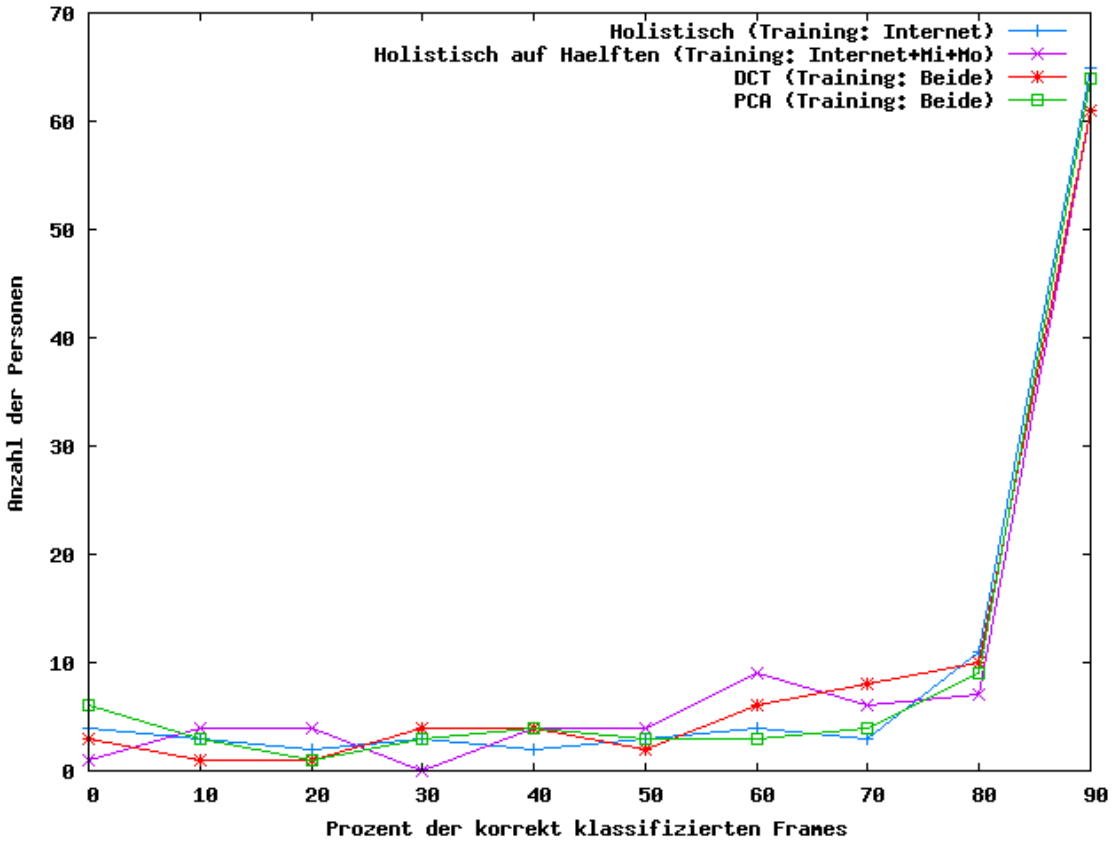

Abb. 4.6: Prozentangaben für die verschiedenen Ansätze

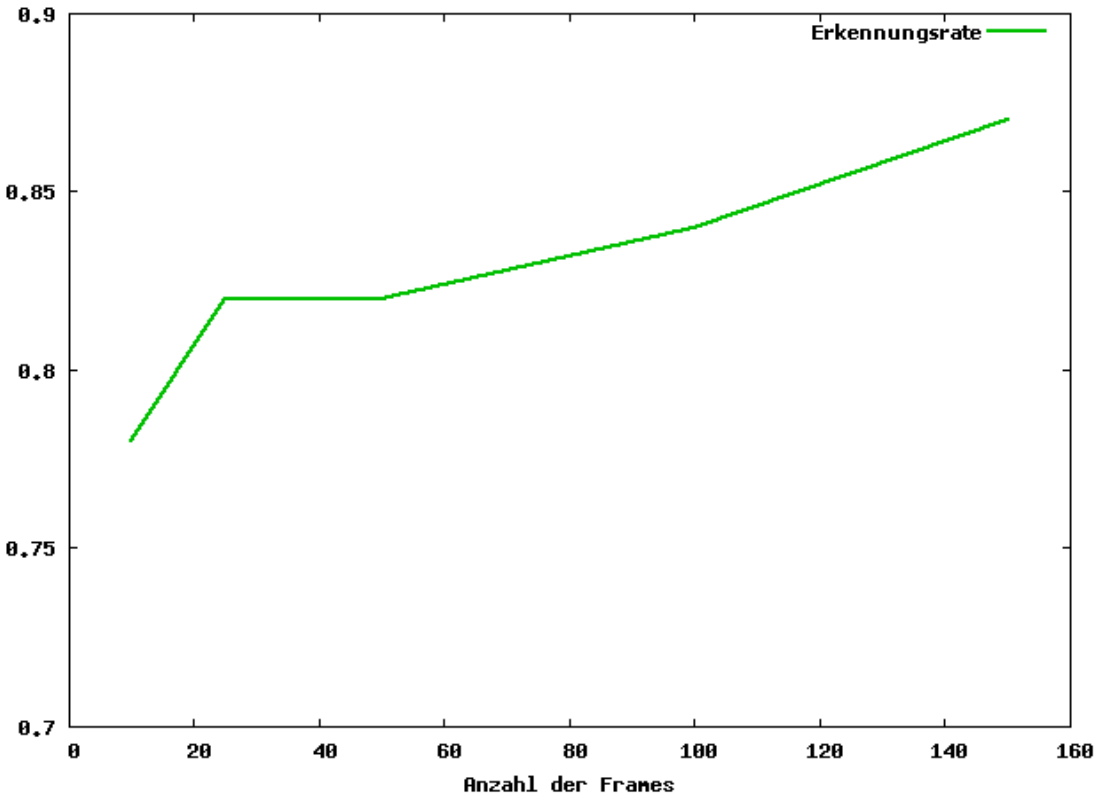

Abb. 4.7: Erkennungsrate in Abhängigkeit von der Frameanzahl

Trainingsset	Features	Rate Sub.	Score Sub.	%Frames
Feret & Internet	DCT+HF	**89/100**	87/100	81.3
Feret & Internet+Mi	DCT+HF	88/100	89/100	**82.6**

Tabelle 4.8: Erkennungsraten auf dem eigenen Datensatz mit homomorpher Filterung

Sowohl für den DCT-basierten, als auch den holistischen Ansatz wurde noch die homomorphe Filterung evaluiert. Die homomorphe Filterung wird in Abschnitt A.2 kurz beschrieben und kann im Detail in [16] nachgelesen werden. Bei dem holistischen Ansatz verschlechtern sich, sowohl auf den Trainingsdaten per Kreuzvalidierung, als auch auf den Testdaten die Werte. Bei dem DCT-basierten Verfahren ergeben sich leichte Verbesserungen, wie in Tabelle 4.8 zu erkennen ist.

Bisher wurde eine binäre Klassifikation pro Frame verwendet. Anstatt jeder Testperson pro Frame diese binäre Entscheidung zuzuordnen, kann aber auch z.B. ein Score verwendet werden. Hierfür bietet sich der Abstand zur Trennhyperebene der SVM an. Für die entgültige Entscheidung werden diese Werte über alle gegebenen Frames akkumuliert und es wird dann für die Klasse mit dem höheren Score entschieden. In der Spalte "*Score Sub.*" in Tabelle 4.8 wird die Erkennungsrate auf diese Weise gebildet.

4.3 Zusammenfassung

Betrachtet man alle Experimente auf den FERET- und Internetbilderdatensätzen, so lässt sich zusammengefasst feststellen, dass alle Methoden zur Featureextraktion ähnliche Ergebnisse liefern. Die besten Ergebnisse liefern das holistische und das DCT-basierte Verfahren. Jedoch sei angemerkt, dass der holistische Ansatz auf Halbgesichtern und der PCA-basierte Ansatz fast die selben Resultate erzielen.
Betrachtet man die Klassifikationsgeschwindigkeit, so schneidet die PCA am besten ab, da sie die geringste Merkmalsraumdimension aufweist. Die DCT-Version mit einem Featurevektor der Länge 320 liefert jedoch auch noch gute Klassifikationszeiten. Die holistischen Ansätze weisen die größte Merkmalsraumdimension auf und benötigen sowohl beim Training, als auch bei der Klassifikation die meiste Rechenzeit.
In den Experimenten hat sich ebenso herausgestellt, dass weder der Gesichtsausschnitt (siehe Tabelle B.2 im Anhang), noch die Auflösung der Gesichter die Geschlechtserkennung maßgeblich beeinflussen.
Auf dem an der Universität Karlsruhe gesammelten Testset wurden insgesamt

geringere Erkennungsraten erzielt. Dies ist eventuell auf die unterschiedlichen Bedingungen zwischen Trainings- und Testdatensatz zurückzuführen. So sahen auf dem FERET- und Internetbilderdatensatz fast alle Personen direkt in die Kamera. Auf dem Webcam-Datensatz ist zu sehen, dass die meiste Zeit fast alle Personen den Bildschirm betrachtet haben und somit nicht direkt in die Kamera sahen. Auch unterschieden sich die Gesichtsausdrücke zwischen den Datensätzen. Auf den Webcamdaten sind die Gesichtsausdrücke weitgehend neutral, während speziell auf dem Internetdatensatz fast immer gelächelt wurde.
Auf dem Webcam-Datensatz funktionieren der DCT-basierte Ansatz in Verbindung mit einer homomorphen Filterung und der holistische Ansatz auf Halbgesichtern am Besten. Diese ordneten 89 von 100 Personen das Geschlecht richtig zu. Der DCT-basierte Ansatz erreicht dieses Ergebnis jedoch mit wesentlich kürzeren Trainings- und Testzeiten und benötigt dazu nur die Hälfte an Stützvektoren. Am meisten Frames klassifiziert der holistische Ansatz richtig (83.9%). Unter Betrachtung aller Experimente kann auch hier festgehalten werden, dass keine der verwendeten Methoden zur Featureextraktion signifikant bessere Erkennungsraten liefert, bis auf die PCA, die ein wenig schlechter funktioniert hat.
Ob es von Vorteil ist über die Bildsequenz für jedes Frame einen Score zu berechnen oder nur eine binäre Entscheidung zu treffen, lässt sich nicht eindeutig beantworten. Die Erkennungsraten haben sich in einem Fall verbessert, in einem anderen aber verschlechtert. Hier wären weitere Untersuchungen notwendig.

5 Online-System

5.1 Einführung

Das Verfahren zur gesichtsbasierten Geschlechtserkennung wurde im Rahmen eines Online-Systems implementiert. Das Online-System bietet die Möglichkeit, mit einem Laptop und einer Webcam an beliebigen Orten in Echtzeit eine Klassifikation durchzuführen. Das Geschlechtserkennungssystem wurde in eine bestehende Implementierung eines portablen Gesichtsidentifikators, der auf Verfahren aus [11] basiert, integriert. In Abschnitt 5.1 wird die Oberfläche des Systems gezeigt. Rechts unten im Bildfenster wird die Prozentangabe der Scorebildung angezeigt.

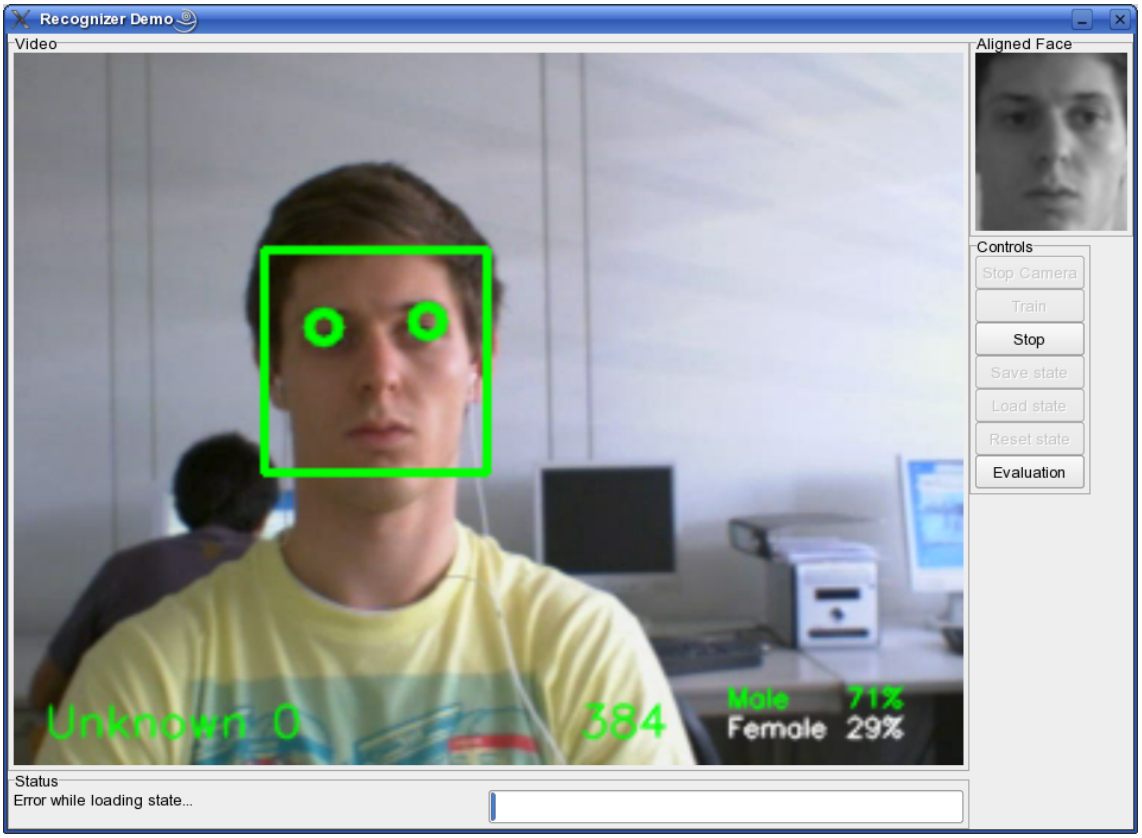

Abb. 5.1: Programmoberfläche des Online-Systems

5.2 Funktionsweise

Ein Trackingalgorithmus für Gesicht und Augen aus dem Gesichtsidentifikators wurde für die Bestimmung der groben Gesichtsposition verwendet. Um das selbe Alignment wie in den in Kap. 4 durchgeführten Experimenten zu erreichen, wurde das vom Tracker gefundene Gesicht ausreichend groß ausgeschnitten und darauf die Vorverarbeitung aus Kapitel 3 angewandt. Findet der Anpassungsalgorithmus nicht beide Augen, so werden die vom Tracker detektierten Augenpositionen verwendet. Dadurch ist eine Klassifikation auf jedem Frame möglich, jedoch kann sich das Ergebnis durch das ungenauere Alignment verschlechtern. Anschließend wird der Abstand des zu klassifizierenden Gesichts zur Trennhyperebene der Support Vector Machine bestimmt. Die dafür verwendete SVM wurde zuvor auf dem Internetdatensatz (mit Verschiebungen) trainiert. Der Abstand zur Hyperebene wird dann als Score verwendet. Die Konfidenzen für Mann bzw. Frau ergeben sich gemäß (5.1) bzw. (5.2).

$$\text{Konfidenz}_{\text{Mann}} = \frac{\sum Score_M}{\sum Score_M + \sum Score_F} \tag{5.1}$$

$$\text{Konfidenz}_{\text{Frau}} = \frac{\sum Score_F}{\sum Score_M + \sum Score_F} \tag{5.2}$$

Nachdem das Programm gestartet wurde, kann über den “Test” Button direkt in den Klassifikationsmodus gewechselt werden. Für jedes Frame wird daraufhin eine Klassifikation vorgenommen. Die Konfidenz der Geschlechtsbestimmung für die vor der Webcam positionierten Person wird rechts unten im Videofenster eingeblendet.

5.3 Zusammenfassung

Das Online-System liefert unabhängig von Beleuchtung, Aufnahmeort und Person gute Erkennungsraten. In den allermeisten Fällen ordnet es einer Person ihr Geschlecht korrekt zu. Die Implementierung entspricht den in dieser Arbeit vorgestellten Verfahren. Auch kann ohne großen Aufwand die Methode zur Merkmalsextraktion ausgetauscht werden.

6 Zusammenfassung und Ausblick

In dieser Arbeit wurde ein Ansatz zur gesichtsbasierten Geschlechtserkennung auf Bildsequenzen vorgestellt. Um die Gesichter in Bildern zu finden wurde eine Klassifikatorkaskade verwendet. Anschließend wurden die Augen detektiert, um so Rotation in der Gesichtsebene rückgängig zu machen. In einem Versuch wurde der Gesichtsausschnitt ermittelt, der die größte Erkennungsrate liefert. Es stellte sich heraus, dass der Gesichtssauschnitt jedoch nur wenig Auswirkung auf die Fehlerrate hatte, da verschiedene Alignments ähnliche Erkennungsraten lieferten. Auch führte ein Verkleinern der Bilder, die zur Klassifikation genutzt wurden, nur zu minimal schlechteren Fehlerraten.
Anschließend wurden verschiedene Methoden zur Merkmalsextraktion auf einer Untermenge der FERET-Datenbank und einem, aus Gruppenfotos aus dem Internet erstellten, Datensatz evaluiert. Alle Verfahren lieferten ähnliche Erkennungsraten, jedoch unterschied sich die Geschwindigkeit der einzelnen Ansätze untereinander teilweise sehr stark. Ein lokaler DCT Ansatz, ebenso ein PCA-basierter Ansatz benötigten aufgrund ihrer geringeren Merkmalsraumdimension weniger Trainings- und Testzeit als die holistischen Ansätze, die auf den beiden Datensätzen in etwa die selben Erkennungsraten lieferten.
In einem weiteren Versuch wurden die verschiedenen Verfahren auf einer Webcam-Datenbank mit 100 Personen getestet. Für jede Person lag eine Bildsequenz mit ca. 150 Frames vor. Da dieser Datensatz zu wenig verschiedene Personen aufwies, konnte damit keine Support Vector Machine trainiert werden. Hohe Erkennungsraten ergaben sich erst ab einer ausreichend großen Menge an verschiedenen Personen im Trainingsdatensatz. Es wurde deshalb auf dem FERET- und Internetbilderdatensatz trainiert. Die Erkennungsraten waren insgesamt für alle Verfahren leicht geringer. Dies lässt sich eventuell darauf zurückführen, dass Trainings- und Testset nun nicht mehr den selben Bedingungen unterlagen. Trotzdem konnten mit dem DCT-basierten und dem holistischen Ansatz auf Halbgesichtern auf den Bildsequenzen Erkennungsraten von ca. 89% erreicht werden. Das DCT-basierte Verfahren arbeitete jedoch mit einer geringeren Merkmalsrausmdimension und ermöglichte wesentlich schnellere Trainings- und Testzeiten. Es ist somit dem holistischen Ansatz vorzuziehen.
Pro Person lag eine Bildsequenz von ca. 150 Frames vor. Wurde auf weniger Frames klassifiziert, so nahm die Erkennungsrate ab. Würden mehr als 150 Frames

pro Person zu Verfügung stehen, so könnten eventuell die Erkennungsraten auf den Sequenzen noch etwas verbessert werden. Ob eine scorebasierte Entscheidung, z.B. der Abstand zur Trennhyperebene der SVM, einer binären Entscheidung für Mann oder Frau pro Frame überlegen ist, ließ sich nicht genau ermitteln. Hierzu wären weitere Versuche notwendig.
Zuletzt wurde ein Online-System entwickelt und vorgestellt, mit dem in Echtzeit die gesichtsbasierte Geschlechtserkennung auf einem Laptop mit einer Webcam durchgeführt werden kann. Durch den Einsatz der Algorithmen, die sich in den Experimenten als leistungsfähig erwiesen haben, erreicht das System vergleichbar gute Erkennungsraten auf Bildsequenzen von wenigen Sekunden.

6.1 Ausblick

In dieser Arbeit wurden etliche Experimente durchgeführt, die sich mit der gesichtsbasierten Geschlechtserkennung beschäftigen. Eine Robustheit des System gegenüber Gesichtsdrehungen (nach links und rechts bzw. oben und unten) konnte jedoch, aufgrund eines Mangels an Testdaten, nicht ermittelt werden.

Es gibt einige Methoden, die das Verfahren eventuell weiter verbessern könnten. Zwei davon sollten zum Abschluss noch kurz angesprochen werden:

Reduzierung irrelevanter Variation in den Bildern

Mit einer korrekten Munddetektion könnten die Gesichter so angepasst werden, dass nicht nur die Augen bei allen Personen an der selben Pixelposition liegen, sondern auch der Mund. Dadurch wäre weniger irrelevante Variation in den Bildern enthalten. Für den Menschen scheint eine Mann-Frau Einordnung auch noch möglich, wenn die Gesichter leicht gestaucht werden. Des Weiteren wäre es von Vorteil, wenn Beleuchtungseinflüsse minimiert werden könnten.

Modellbasierter Ansatz

Es wurden in dieser Arbeit nur ansichtsbasierte Ansätze getestet. Es wäre jedoch auch interessant zu prüfen, welche Erkennungsraten ein modellbasierter Ansatz liefern würde. Dafür ist jedoch eine zuverlässige und vor allem genaue Lokalisierung von Gesichtsregionen notwendig. Als Features könnten z.B. der Abstand zwischen Auge und Augenbrauen, Augenbrauendicke und -form (z.B. parametrisch), Nasendicke, Abstand von Mund und Nasenspitze und Kinnform (z.B. parametrisch) verwendet werden. Brunelli et al. [17] präsentierten einen solchen Ansatz, der jedoch eine Fehlerrate von 21% auf einem sehr kleinen Testset lieferte. Durch

den wissenschaftlichen Fortschritt und neue Techniken könnte heute eventuell ein besseres Ergebnis erzielt werden.

A Anhang

A.1 Histogrammegalisierung

Gegeben sei ein diskretes Grauwertbild und sei n_i die Anzahl der Pixel mit Grauwert i. Die Auftriffswahrscheinlichkeit für ein Pixel mit dem Grauwert i lautet

$$p(x_i) = \frac{n_i}{n}, i \in 0, ..., L-1 \tag{A.1}$$

wobei L die Anzahl der möglichen Grauwerte im Bild und n die gesamte Pixelanzahl beschreibt. Die Wahrscheinlichtkeitsverteilung p beschreibt das auf den Wertebereich $[0, 1]$ normalisierte Histogramm von dem gegebenen Bild.
Die kumulative Wahrscheinlichkeitsfunktion c in Abhängigkeit von p lässt sich wie folgt definieren:

$$c(i) = \sum_{j=0}^{i} p(x_j) \tag{A.2}$$

Gesucht ist nun eine Transformation in der Form $y = T(x)$, die einen Grauwert y für jeden Grauwert x liefert, sodass die kumulative Wahrscheinlichkeitsfunktion von y über ihren Wertebereich linearisiert wird. Diese Transformation ist durch

$$y_i = T(x_i) = c(i) \tag{A.3}$$

gegeben. Da T die Werte auf den Bereich $[0, 1]$ abbildet, kann das Ergebnis durch

$$y_i' = y_i \cdot (max - min) + min \tag{A.4}$$

wieder auf den ursprünglichen Wertebereich abgebildet werden. min bzw. max entsprechen dabei dem minimalen bzw. maximalen Grauwert im ursprünglichen Bild.

A.2 Homomorphe Filterung

Sei $s(\mathbf{x})$ die Reflektanz einer Oberfläche und $b(\mathbf{x})$ eine ungleichmäßige Beleuchtung. $b(\mathbf{x})$ sei im Gegensatz zu $s(\mathbf{x})$ zeitlich nur langsam veränderlich. Bildet man diese Szene mit einer Kamera zu einem Bild g($\mathbf{x}$) ab, so gilt vereinfacht folgende Gleichung:

$$g(\mathbf{x}) = s(\mathbf{x}) \cdot b(\mathbf{x}) \tag{A.5}$$

Transformiert man diesen Term in den Fourierraum, so ergibt sich:

$$G(\mathbf{f}) = S(\mathbf{f}) * *B(\mathbf{f}) \tag{A.6}$$

wobei $**$ der 2-dimensionalen Faltung entspricht. Wie zu erkennen ist, sind s und b multiplikativ im Ortsraum verknüpft. Nehmen wir nun an, s und b seien im Ortsraum additiv verknüpft, so wären S und B auch nach einer Tranformation in den Fourierraum additiv verknüpft. Anschließend könnte eine Hochpassfilterung angewandt werden, um das zeitlich langsam veränderliche Signal $b(\mathbf{x})$ zu unterdrücken. Anschließend könnte zurück in den Ortsraum transformiert werden. Die inhomogene Beleuchtung $b(\mathbf{x})$ wäre dadurch (zu einem großen Teil) entfernt worden.
Logarithmiert man $s(\mathbf{x}) \cdot b(\mathbf{x})$, ergibt sich folgende Gleichung:

$$ln(g(\mathbf{x})) = ln(s(\mathbf{x}) \cdot b(\mathbf{x})) = ln(s(\mathbf{x})) + ln(b(\mathbf{x})) \tag{A.7}$$

Transformiert man nun in den Fourierraum, so kann eine Hochpassfilterung angewendet werden, da die Signale nun additiv verknüpft werden. Anschließend wird wieder in den Ortsfrequenzraum gewechselt und dort die Logarithmierung über die e-Funktion rückgängig gemacht.

B Verwendete Parameterkonfigurationen

B.1 SVM Parameter

Der Kostenparameter wurde in den Experimenten auf dem Webcam-Datensatz stets auf 65536 gesetzt, damit die SVM keine Trainingsfehler macht. Dies ergab die kleinsten Fehlerraten. Der Kernelparameter γ war stark vom Trainingsdatensatz abhängig. Für den FERET Datensatz war γ in etwa halb so groß wie bei dem Internetbilderdatensatz. In Tabelle B.1 sind die Parameter für den Internetbilderdatensatz abgebildet.

B.2 Erkennungsraten für verschiedene Gesichtsausschnitte

In Tabelle B.2 werden die Erkennungsraten für den Internetbilderdatensatz in Verbindung mit verschiedenen Gesichtsausschnitten angegeben. d beschreibt den Augenabstand in Pixel, y die Zeile, in der die Augen lagen. Die Fensterauflösung für das Alignment betrug $128 * 150$ Pixel.

Merkmalsextraktion	Kostenparameter C	Kernelparameter γ
Holistisch	65536	0.002
Holistisch auf Gesichtshälften	65536	0.015
DCT	65536	0.07
PCA	65536	0.5

Tabelle B.1: SVM Parameterkonfigurationen (Internetbilderdatensatz)

$\downarrow y \; d \rightarrow$	60	62	64	66	68	70	72
30	90.9	91.7	91.4	92.2	91.5	92.6	92.6
35	91.4	91.6	92.1	92.1	92.1	92.2	92.2
40	91.3	91.8	91.9	92.2	91.7	92.5	92.1
45	91.3	92.2	92.0	92.5	92.2	92.6	92.2
50	91.8	**93.5**	92.5	92.6	92.3	92.2	92.3

Tabelle B.2: Erkennungsraten für verschiedene Verschiebungen (Internetbilderdatensatz)

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[1] B. Moghaddam and M.-H. Yang. Learning gender with support faces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):707–711, May 2002.

[2] M. Castrillón-Santana, O. Deniz, D. Hernandez, and A. Dominguez. Identity and gender recognition using the encara real-time face detector. Conferencia de la Asociación Española para la Inteligencia Artificial, 2003.

[3] M. Castrillón-Santana. *On Real-Time Face Detection in Video Streams. An Opportunistic Approach.* PhD thesis, Universidad de Las Palmas de Gran Canaria, 2003.

[4] E. Makinen and R. Raisamo. Evaluation of gender classification methods with automatically detected and aligned faces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(3):541–547, March 2008.

[5] Christopher J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[6] B. Scholkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press Cambridge, 2001.

[7] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition, 2001.*, 1:I–511–I–518 vol.1, 2001.

[8] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37, 1995.

[9] *Open Source Computer Vision Library, http://www.intel.com/technology/computing/opencv/index.htm.*

[10] M. Castrillón Santana, J. Lorenzo Navarro, O. Déniz Suárez, and A. Falcón Martel. Multiple face detection at different resolutions for perceptual user interfaces. In *2nd Iberian Conference on Pattern Recognition and Image Analysis*, Estoril, Portugal, June 2005.

[11] H.K. Ekenel and R. Stiefelhagen. Analysis of local appearance-based face recognition: Effects of feature selection and feature normalization. *Computer Vision and Pattern Recognition Workshop, 2006 Conference on*, pages 34–34, June 2006.

[12] M.A. Turk and A.P. Pentland. Face recognition using eigenfaces. *Computer Vision and Pattern Recognition.*, pages 586–591, Jun 1991.

[13] *Google Image Search, http://images.google.com.*

[14] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[15] Y. W. Chen and C. J. Lin. Combining SVMs with various feature selection strategies. In *Feature extraction, foundations and applications.* Springer, 2006.

[16] R. Gonzalez and R. Woods. *Digital Image Processing.* Addison-Wesley, 1992.

[17] R. Brunelli and R. Poggio. Hyperbf networks for gender classification. In *Proceedings of the DARPA Image Understanding Workshop*, pages 311–314, 1992.