

# Diplomarbeit

zur Erlangung des akademischen Grades

## Diplom-Informatiker (FH)

an der  
Fachhochschule Brandenburg

Fachbereich  
Informatik und Medien

**Thema:** Entwurf, Implementierung und  
Evaluation eines Verfahrens zur  
Detektion von Gesichtsmerkmalen in  
Einzelbildern mittels künstlicher  
neuronaler Netze

**Diplomand:** Holger Ripp, Brandenburger Straße 30,  
14542 Werder/Havel

**Betreuer:** Prof. Dr. rer. nat. Friedhelm Mündemann  
Dipl.-Inform. Ingo Boersch

**Eingereicht:** 8. November 2005



## Inhaltsverzeichnis

<b>1 Einleitung</b> .....	<b>1</b>
1.1 Anforderungen an Gesichtsdetektion/-lokalisierung.....	2
1.2 Zielsetzung und Aufbau der Arbeit.....	4
<b>2 Verfahren zur Gesichtsdetektion</b> .....	<b>5</b>
2.1 Wissensbasierter Ansatz.....	6
2.2 Merkmalsbasierter Ansatz.....	7
2.2.1 Gesichtsmerkmale.....	7
2.2.2 Farbe.....	9
2.3 Template Matching Ansatz.....	11
2.4 Ansichtsbasierter Ansatz.....	12
2.4.1 Eigenfaces.....	12
2.4.2 Künstliche neuronale Netze .....	14
2.4.3 Hidden-Markov-Modell .....	21
2.5 Zusätzliche Betrachtung der Bildvorverarbeitung.....	22
2.5.1 Kantenfilter.....	22
2.5.2 Beleuchtungskorrektur.....	27
2.5.3 Histogrammausgleich.....	27
2.6 Bewertung der Verfahren.....	28
<b>3 Lösungsansatz zur Detektion von Gesichtsmerkmalen</b> .....	<b>30</b>
3.1 BioID-Gesichtsdatenbank.....	31
3.1 Gesichtsmerkmale.....	33
3.1.1 Das Normgesicht.....	33
3.1.2 Größe der Gesichtsmerkmale.....	33
3.1.3 Extraktion des Bildausschnittes.....	34
3.2 Bildvorverarbeitung/Normierung.....	37
3.2.1 Beleuchtungskorrektur.....	37
3.2.2 Kantenfilter.....	39
3.3 Modellierung der Gesichtsmerkmale.....	40
3.4 Suchstrategie.....	43
<b>4 Implementierung</b> .....	<b>44</b>
4.1 Umgebung.....	44
4.2 Umsetzung der einzelnen Teile.....	44
4.2.1 Fenstertechnik.....	44
4.2.2 Bildvorverarbeitung.....	46
4.2.3 KNN – Training und Anwendung.....	48
4.2.4 Suche über das Bild.....	49
4.3 Zusätzliche Entwicklungen.....	50
4.3.1 Analyse der Suche.....	50
4.3.2 Visualisierung der Ausgabewerte der KNN über eine Musterdatei oder der Ausgabewerte einer SNNS-Result-Datei.....	51

<b>5 Experimentelle Untersuchungen.....</b>	<b>52</b>
5.1 Test der Fenstertechnik.....	52
5.2 Test der Beleuchtungskorrektur.....	55
5.3 Test des SUSAN-Kantenfilters.....	57
5.4 Training der künstlichen neuronalen Netze.....	58
5.5 Ergebnisse der Suchtests.....	60
<b>6 Zusammenfassung und Ausblick.....</b>	<b>67</b>
<b>Anhang A: Entfernte Bilder .....</b>	<b>68</b>
<b>Anhang B: Bilder je Person.....</b>	<b>70</b>
<b>Anhang C: Einzelheiten zum Training der KNN.....</b>	<b>72</b>
<b>Anhang D: Einzelheiten zu den einfachen Suchtests.....</b>	<b>73</b>
<b>Abbildungsverzeichnis.....</b>	<b>75</b>
<b>Literaturverzeichnis.....</b>	<b>78</b>

## 1 Einleitung

Das Gesicht ist, bis auf wenige Ausnahmen wie beispielsweise eineiige Zwillinge, seit Menschengedenken der Teil des Menschen der ihn unverwechselbar macht. Menschen können ihr Gegenüber daran nicht nur identifizieren, sondern auch meist dessen Gemütszustand erkennen. Ebenso lässt sich das Alter eines Menschen über das Gesicht mehr oder weniger genau schätzen. Ob sich aus dem Gesicht sogar Charaktereigenschaften ablesen lassen, ist immer wieder Gegenstand der Forschung. Der Mensch kann also ein Gesicht auf vielfältige Weise interpretieren und natürlich ist die Wissenschaft bestrebt, diese Fähigkeit des Menschen nachzubilden. Zum einen aus reinem Forscherdrang, zum anderen bieten der vermehrte Einsatz von Bildsensoren, welche die Bilddaten in automatisch verarbeitbarer Form speichern, und stetig leistungsstärker werdende Rechnersysteme immer neue Anwendungsszenarien.

Dabei beschäftigt sich die Wissenschaft vorrangig mit der Erkennung von Gesichtern zum Zwecke der Identifikation (z.B. in [TP91b]) bzw. Verifikation (z.B. in [TKP98]) von Gesichtern oder der Verfolgung von Gesichtern in Bildsequenzen (z.B. in [CB97]). Ein weiteres Forschungsfeld ist die Interpretation des Ausdrucks und der Mimik (z.B. in [DCPA02]).

Anwendung finden heutige Verfahren vorrangig um das Gesicht als biometrisches Merkmal (eine gute Einführung in biometrische Merkmale gibt [Br05]) für die Zugangskontrolle von Personen (z.B. in [NEC05], [Vi05]) an Eingängen zu Sicherheitsbereichen von Unternehmen und Institutionen zu nutzen. Ein Schritt für mehr Sicherheit für Staat und Gesellschaft sind Pässe mit integrierten biometrischen Merkmalen, doch gerade hier muss dem Thema Datenschutz und Privatsphäre besondere Aufmerksamkeit geschenkt werden. Kritisiert werden beispielsweise der geringe Sicherheitsgewinn, ungeklärte Kosten und fehlende Systemtests im Vorfeld [CCC05]. Wie bei anderen Wissenschaften auch gibt es Schattenseiten dieser Forschungsthemen, die sich durch eine sich verringernde Privatsphäre äußern, da gerade Sicherheitsunternehmen, Ermittlungsbehörden und Geheimdienste die Weiterentwicklung in dieser Wissenschaft forcieren. Umfangreiche Untersuchungen zum Einsatz der Gesichtserkennung als

biometrisches Merkmal führte das Fraunhofer Institut Graphische Datenverarbeitung (IGD) in Zusammenarbeit mit dem Bundesamt für Sicherheit in der Informationstechnik (BSI) durch. Die Ergebnisse wurden veröffentlicht in [BSI03].

Im privaten Bereich sind Produkte mit Gesichtserkennung bisher kaum anzutreffen. Doch auch hier nimmt die Zahl der digitalen Bild- und Videodaten in Form von privaten Aufnahmen stetig zu. Ein mögliches Anwendungsszenario wäre beispielsweise die automatische Verschlagwortung von Digitalfotos mit den Namen bekannter Personen.

Die automatisierte Verarbeitung von digitalen Bilddaten ist Teil der Informatik, insbesondere der Bereiche Bildverarbeitung und Künstliche Intelligenz. Das Thema Bildverstehen, die Analyse und Interpretation von visuellen Informationen, ist eines der schwierigsten Forschungsgebiete der Informatik. Wird nach komplexen, veränderlichen Mustern in komplexen Hintergründen gesucht, ist es schwierig, dafür ein einfaches Modell zu konstruieren.

Bevor das Gesicht einem Verfahren zur Weiterverarbeitung (Identifikation, Verifikation usw.) vorgelegt werden kann, ist in einem Vorverarbeitungsschritt zu ermitteln, ob und wo sich ein Gesicht im Bild befindet.

### **1.1 Anforderungen an Gesichtsdetektion/-lokalisierung**

Dieser Vorverarbeitungsschritt wird als Gesichtsdetektion bzw. Gesichtslokalisierung bezeichnet. In [YKA02] wird die Suche nach der Position *eines* Gesichtes als *Gesichtslokalisierung* definiert. Das Vorhandensein dieses Gesichtes wird meist vorausgesetzt, ebenso wie Größe und Orientierung.

Im Gegensatz dazu setzt die *Gesichtsdetektion*, nach der Definition von [YKA02], kein Wissen über das Vorhandensein eines Gesichtes voraus und es dürfen sich mehrere Gesichter im Bild befinden.

Die größte Herausforderung bei der Entwicklung neuer Verfahren ist, neben der reinen Detektionsleistung, eine geringe Zahl von Nebenbedingungen. Die Nebenbedingungen dienen dem Zweck, die Variabilität (Beispielhaft in Abbildung 1 dargestellt), bedingt durch nachfolgend beschriebene Eigenschaften, einzuschränken. Diese Eigenschaften erschweren eine automatische Verarbeitung von Gesichtsbildern und werden folgend als Problemeigenschaften im Sinne der Bildverarbeitung bezeichnet.

Nebenbedingungen sind hauptsächlich erforderlich, damit die Verfahren Gesichter in vertretbarer Zeit detektieren und weiterverarbeiten können. Insbesondere, wenn es sich nicht um die Verarbeitung von Einzelbildern, sondern von Bildsequenzen handelt.

Es gibt aktuell kein Verfahren, welches keine Nebenbedingungen festlegt und damit robust gegenüber allen Problemeigenschaften ist [YKA02].



*Abbildung 1 : Beispiele für die Variabilität in der Abbildung von Gesichtern (aus der CMU-MIT-Gesichtsdatenbank [CMU05]).*

Folgende Problemeigenschaften erschweren die Detektion und Weiterverarbeitung eines Gesichtes:

- a) Problemeigenschaften eines Gesichtes:
  - verschiedene Gesichtsausdrücke
  - Bart
  - Brille
  - Frisur
  - Bekleidung (Hut, Mütze, Kapuze, Schal)
  - Gesichtsfarbe
  - teilweise Verdeckung
  - Rotation des Kopfes (3 Achsen):
    - ◆ in der Bildebene (um die Achse, die senkrecht auf dem Bild steht)
    - ◆ außerhalb der Bildebene (Orientierung des Gesichtes: rechts, links, oben oder unten)

b) Problemeigenschaften des Aufnahmesensors:

- Auflösung
- Farbfehler
- Rauschen

c) Problemeigenschaften des Bildes:

- Komplexität des Hintergrundes
- verschiedene Beleuchtungssituationen

Folgende Problemeigenschaften erschweren die Detektion eines Gesichtes zusätzlich:

a) Problemeigenschaften eines Gesichtes im Bild:

- Position im Bild
- Größe im Bild

b) Problemeigenschaften des Bildes:

- Anzahl der Gesichter im Bild (bei mehreren Gesichtern)

## **1.2 Zielsetzung und Aufbau der Arbeit**

In dieser Arbeit wurde ein Verfahren zur Detektion von Gesichtsmarkmalen in digitalen Einzelbildern entwickelt. Mit Gesichtsmarkmalen sind folgend die gesichtsbildenden Merkmale Augen, Nase und Mund gemeint.

Bei der Entwicklung des Verfahrens wurde Wert darauf gelegt, dass es positions-, skalierungs- und rotationsinvariant ist. Es sollte robust gegenüber unterschiedlichen Beleuchtungssituationen sein und eine gute Generalisierungsfähigkeit aufweisen.

Verschiedene Ansätze im Bereich der Gesichtslokalisierung und –detektion sind in der Literatur publiziert worden. Einen Überblick über den aktuellen Stand der Forschung gibt Kapitel 2. Eine Bewertung der vorgestellten Verfahren, bezüglich der oben genannten Zielsetzung, erfolgt am Ende des Kapitels.

Aufbauend auf den Ergebnissen der Bewertungen aus Kapitel 2, wird in Kapitel 3 das Verfahren vorgestellt. Grundlage dieses Verfahrens ist die Modellierung der Gesichtsmarkmale mit künstlichen neuronalen Netzen.

In Kapitel 4 wird die konkrete Umsetzung des Verfahrens beschrieben.

Eine ausführliche Diskussion der Ergebnisse des Verfahrens erfolgt in Kapitel 5. Abgeschlossen wird die Arbeit mit einem Ausblick in Kapitel 6.



## 2 Verfahren zur Gesichtsdetektion

In der Literatur sind eine Vielzahl verschiedener Ansätze zur Detektion und Lokalisierung von Gesichtern publiziert worden. Einen umfassenden Überblick geben Yang u.a. in [YKA02].

Aufgabe der Verfahren ist es, ein Bild bzw. einen Bildausschnitt zu klassifizieren. Dabei handelt es sich um ein 2-Klassen-Problem, d.h. der Eingaberaum muss in „Gesicht“ und „Nichtgesicht“ klassifiziert werden.

Die Detektion eines Gesichtes erfordert zudem die Suche nach diesen Bildausschnitten.

Grob besteht die Detektion aus zwei Teilbereichen:

- die Selektion von Bildausschnitten
- die Klassifikation dieses Bildausschnittes

Wie klein diese Bildausschnitte sind hängt vom jeweiligen Verfahren ab. Ebenso erfolgt die Klassifikation mit ganz unterschiedlichen Ansätzen.

Wie groß der Eingaberaum  $\mathfrak{X}$  ist, lässt sich an einem Beispiel verdeutlichen. Ein 19 x 19 Pixel großes Grauwertbild hat die Dimension  $19 \cdot 19 = 361$ , somit enthält  $\mathfrak{X}^{361} : 256^{361} = 2^8 \cdot 361 = 2^{2888}$  darstellbare Grauwertkombinationen. Im Vergleich dazu betrug die Anzahl der Menschen auf der ganzen Welt im Jahr 2004:  $6.400.000.000 = 2^{32}$  [Ya04].

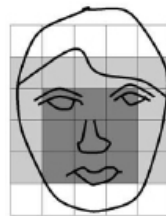
Aus der Perspektive des Eingaberaums spannen Gesichter darin nur einen sehr kleinen Unterraum, folgend Gesichtsunterraum genannt, auf, der sich auch relativ gut vom Rest trennen lässt. Betrachtet man jedoch die in Kapitel 1.1 aufgezählten Problemeigenschaften, ist dieser Gesichtsunterraum sehr groß. Jedes Verfahren zur Gesichtsdetektion muss diesen Gesichtsunterraum möglichst genau beschreiben. Doch schränken alle in [YKA02] beschriebenen Verfahren diesen durch Nebenbedingungen ein.

Dagegen müssen Verfahren zur Gesichtserkennung pro Gesicht nur einen Teil dieses Gesichtsunterraumes beschreiben, haben aber das Problem der Trennung von anderen Gesichtern.

### 2.1 Wissensbasierter Ansatz

Bei diesem Ansatz wird Wissen über charakteristische Eigenschaften des gesamten Gesichtes genutzt, um einen Gesichtskandidaten, das heißt ein mögliches Gesicht, zu extrahieren. Das Wissen um die Relationen zwischen den Gesichtsmerkmalen wird dann zur Validierung dieses Gesichtskandidaten benutzt.

Ein verblüffend einfaches Verfahren wurde in [YH94] vorgestellt. Dazu wurde in 3 Schritten die Auflösung des Bildes verändert. Im ersten Schritt konnten mögliche Gesichtskandidaten in einem grob aufgelösten Bild anhand einer vordefinierten Helligkeitsverteilung detektiert werden (siehe Abbildung 2).



*Abbildung 2: Gesichtskandidat nach Yang [YH94] (ausgehend von der Annahme, dass der Bereich von Augen, Nase und Mund dunkler als umliegende Gesichtsteile ist).*

Mit einem höher aufgelösten Bild wurde im zweiten Schritt der Bereich der möglichen Gesichtskandidaten mit Histogrammausgleich und Kantenfilter vorverarbeitet und mit einem Regelsatz validiert. Um die verbleibenden Gesichtskandidaten endgültig als Gesichter zu bestimmen, wurden in einem letzten Schritt Augen und Mund gesucht und mit einem weiteren Regelsatz abgeglichen. Das Wissen über das Gesicht ist bei diesem Verfahren in der Helligkeitsverteilung und den Regelsätzen kodiert.

Von 60 Gesichtsbildern konnte dieses Verfahren 83 % der Gesichter korrekt detektieren.

Davon motiviert, wandten Kotropoulos und Pitas in [KP97] ein ähnliches Verfahren an. Um jedoch die Verarbeitungszeit zu verringern, wurden Gesichtskandidaten mit Hilfe der Histogramme der Spalten- und Zeilen-grauwertsummen bestimmt (siehe Abbildung 3).

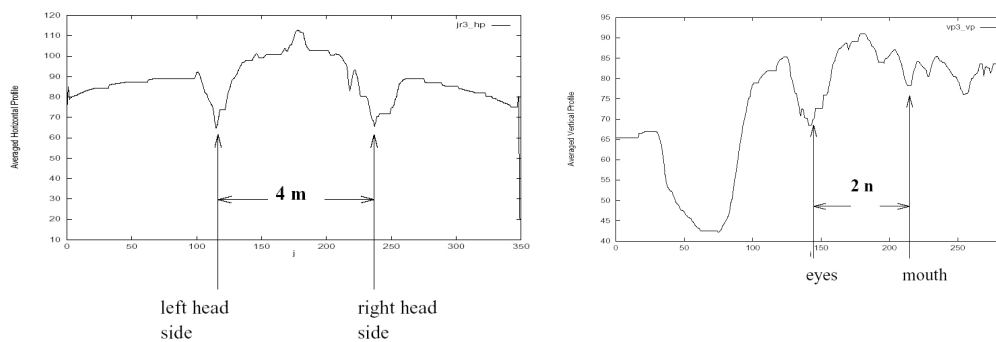


Abbildung 3: Die Histogramme der Spalten- (links) und Zeilengrauwertsummen (rechts) zur Bestimmung eines Gesichtskandidaten (aus [KP97]).

Danach konnten diese Kandidaten mittels Regelsätzen, angewandt auf Gesichtsteile wie Augen, Nase und Mund, validiert werden. Unterschiedlich groß skalierte Gesichter bereiten diesem Verfahren keine Probleme, dagegen eignet es sich nicht für komplexe Hintergründe. Dieses Verfahren wurde mit der M2VTS-Gesichtsdatenbank [Pi98] getestet und konnte 86,5 % der Gesichter korrekt detektieren.

## 2.2 Merkmalsbasierter Ansatz

Im Gegensatz zum wissenbasierten Ansatz wird beim merkmalsbasierten Ansatz die Idee verfolgt, dass es Teile des Gesichtes gibt, die unabhängig von Orientierung und Lichtverhältnissen immer detektierbar sind. Motiviert wird dieser Ansatz durch die Fähigkeit des Menschen Gesichter, unabhängig von den genannten Eigenschaften, zu erkennen, im Sinne von detektieren, oder sogar zu identifizieren.

### 2.2.1 Gesichtsmerkmale

Dem biologischen Vorbild des Menschen folgend, der Gesichter auch erkennt, wenn er nur Teile eines Gesichtes wie beispielsweise ein Auge sieht, verwenden eine Reihe von Verfahren Gesichtsmerkmale wie Augen, Nase, Mund und deren Relation zueinander.

In [LBP95] wurde ein Verfahren vorgestellt, das lokale Merkmale wie die Augen, die Nasenlöcher und die Nasen-Lippen-Verbindung sucht. Diese Merkmale konnten mit Gauß-Filter-Sets extrahiert werden. Ein Gauß-Filter-

Set besteht aus mehreren unterschiedlich rotierten und skalierten Gauß-Filtern. Die Relationen zwischen den Merkmalen waren als Gaußverteilung in einem Graphen modelliert. Bei diesem als Random Graph Matching bezeichneten Verfahren wird das Gesicht in Form eines etikettierten Graphen repräsentiert. Von 150 Gesichtsbildern konnte dieses Verfahren 86 % der Gesichter korrekt detektieren.

Yow und Cipolla verwendeten in [YC96] einen ähnlichen Ansatz. So wurden mit einem Gauß-Filter, angewandt auf das ganze Bild, dessen lokale Maxima als interessante Punkte und mögliche Kandidaten für Gesichtsmarkale bestimmt. Um die Punkte wurden Kanten gesucht und diese in Regionen gruppiert.

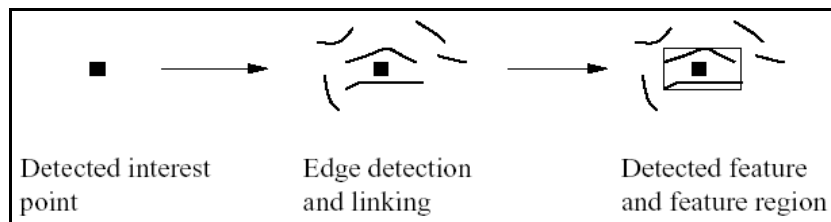


Abbildung 4: Detektierung von Gesichtsmarkalen (aus [YC96]).

Messbare Eigenschaften der Regionen wurden in Vektoren gespeichert. Überschritt die Mahalanobis-Distanz dieses Vektors zu einem der, für die jeweiligen Merkmale vorberechneten, Durchschnittsvektoren einen bestimmten Schwellwert, galt dieses Merkmal als erkannt. Die Vorberechnung der Durchschnittsvektoren erfolgte über eine Anzahl von Trainingsgesichtsbildern. Mit Hilfe eines Bayesschen Netzes wurden diese Merkmale schrittweise zu einem Gesicht gruppiert.

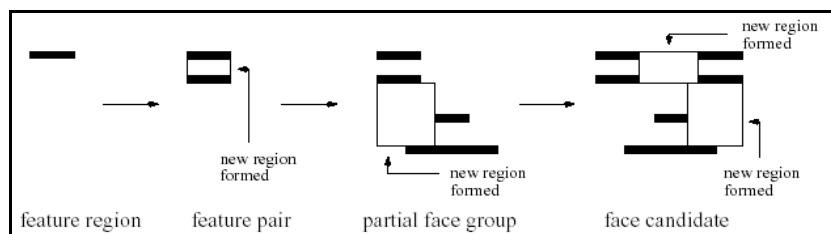


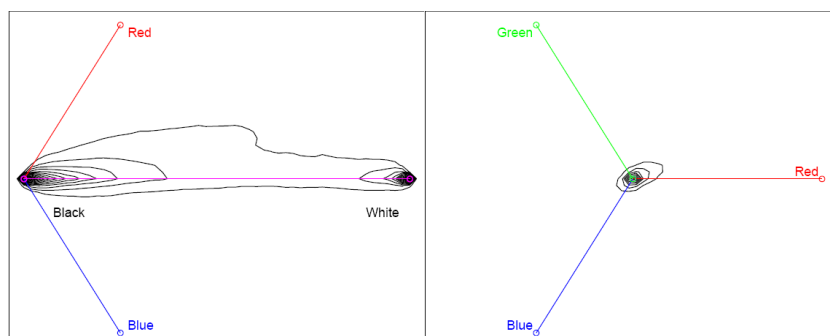
Abbildung 5: Schrittweise Gruppierung der Merkmale, beginnend bei der rechten Augenbraue (aus [YC96]).

Dieses Verfahren konnte Gesichter rotiert oder in verschiedenen Orientierungen detektieren, war aber nur für Gesichter größer 60 x 60 Pixel praktikabel. In 100 Gesichtsbildern wurden 92 % der Gesichter korrekt detektiert.

### 2.2.2 Farbe

Eine vielfach verwendete Eigenschaft des Gesichtes ist die Farbe. In [ATD01] wurde die Eignung verschiedener Farbräume für die Detektion von Haut betrachtet.

Jones und Rehg konstruierten in [JR99] einen histogrammbasierten Hautpixel-Klassifikator. Dazu erstellten sie auf Basis eines RGB-Histogramms über 18.696 Bilder, die dem Internet entstammten, ein allgemeines RGB-Farbmodell (siehe Abbildung 6).



*Abbildung 6: Darstellung der allgemeinen Farbverteilung von 18.696 Bildern aus dem Internet (links entlang der Grün-Magenta-Achse, rechts im RGB-Farbraum) (aus [JR99]).*

Von diesen 18.696 Bildern wurden 4675 Bilder die Haut enthielten manuell gekennzeichnet und über diese ein RGB-Histogramm und die entsprechende Farbverteilung der Hautpixel erstellt (siehe Abbildung 7).

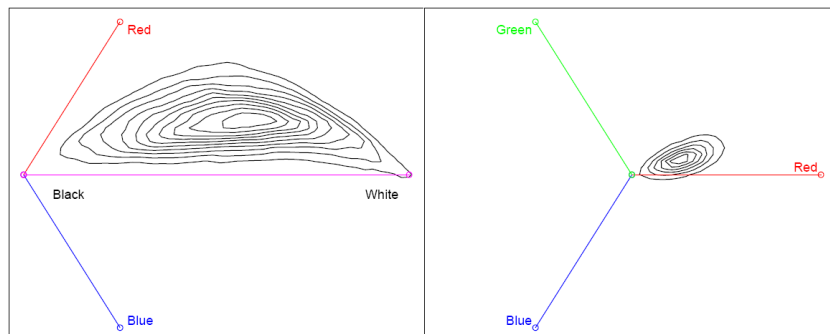


Abbildung 7: Darstellung der Farbverteilung von Hautpixeln über 4675 Bilder (aus [JR99]).

Aus weiteren 8965 Bildern die keine Haut enthielten, konnte das RGB-Histogramm und die Farbverteilung für Nicht-Hautpixel erstellt werden (siehe Abbildung 8).

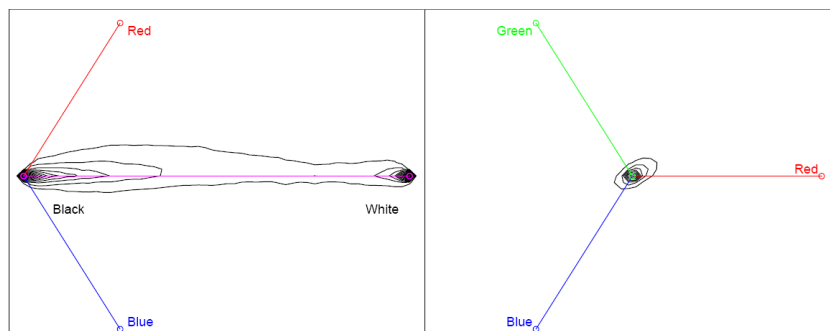


Abbildung 8: Darstellung der Farbverteilung von Nicht-Hautpixeln über 8965 Bilder (aus [JR99]).

Aus den Histogrammen wurden die Wahrscheinlichkeiten berechnet, mit der die Farbe eines Pixels Haut oder Nicht-Haut ist:

$$P(rgb|skin) = \frac{s[rgb]}{T_s} \quad P(rgb|\neg skin) = \frac{n[rgb]}{T_n} \quad ,$$

mit  $s[rgb]$  und  $n[rgb]$  als die Anzahl der Pixel des Farbwertes  $rgb$  der Histogramme von Haut- bzw. Nicht-Hautpixel und mit  $T_s$  und  $T_n$  als die Gesamtzahl der in den Histogrammen enthaltenen Werte. Die Histogramme wurden im  $256^3$  Format (256 Werte für jeden Kanal R, G und B) erstellt.

Es wurde festgestellt, dass 76,6 % der möglichen 16,7 Millionen Farben nicht in den Bildern vorkamen. Außerdem stellte man fest, dass von den 959.955

Farben, die als Hautfarbe bestimmt wurden, 933.275 ebenfalls als Nicht-Hautfarbe bestimmt wurden.

Mit diesem Verfahren konnten Jones und Rehg von 4999 Trainingsbildern 78 % und von 1909 Testbildern 77,5 % richtig klassifizieren.

Problematisch ist dieser Ansatz durch die unterschiedlichen Spektren der Hautfarbe verschiedener ethnischer Gruppen oder durch die Möglichkeit von Bemalung und Make-Up [AMD01] (siehe Abbildung 9).

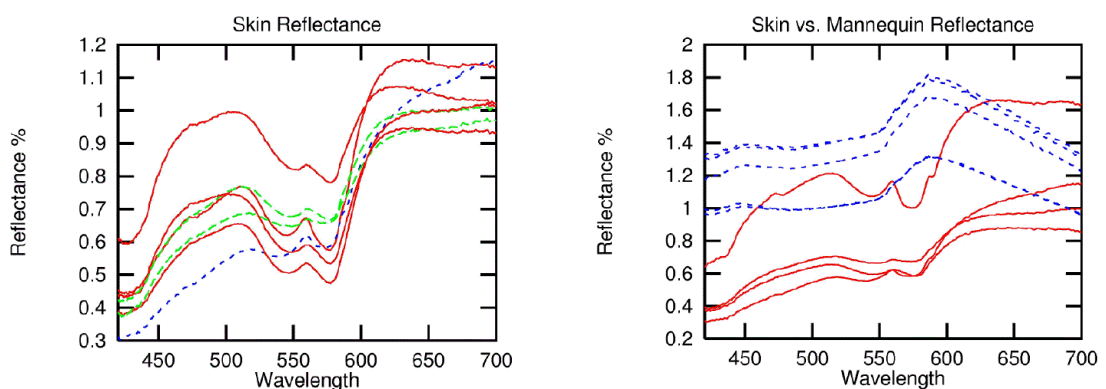


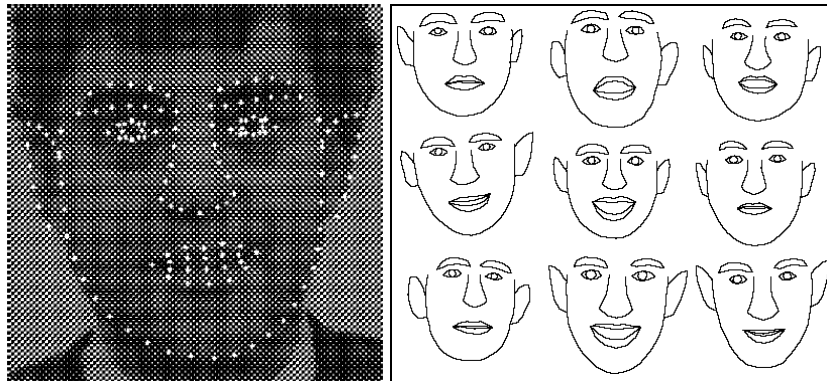
Abbildung 9: Spektren der Handrückenfarbe, links verschiedene ethnische Gruppen (Kaukasiern (rot), Asiaten (grün) und Inder (blau)), rechts normale Haut (rot) im Vergleich zu geschminkter Haut (blau) (aus [AMD01]).

### 2.3 Template Matching Ansatz

Bei Verfahren dieser Kategorie werden Muster/Templates der Gesichter bzw. Gesichtsteile vordefiniert. Diese Muster enthalten geometrische Informationen. Um ein Bild oder Bildausschnitt zu klassifizieren, wird die Korrelation zu einem Muster berechnet.

Zur Generierung der Muster werden zuerst markante Punkte festgelegt und ihre geometrischen Beziehungen untereinander oder zu einem Referenzpunkt über eine Menge von Trainingsbildern ermittelt.

In [LTC95] wurde ein Verfahren vorgestellt, welches mit deformierbaren *Face Shape Models*, sogenannten PDMs, arbeitete. Ein PDM wurde über 152 markante Punkte des Gesichtes (siehe Abbildung 10 links) und 160 Trainingsgesichtsbilder erzeugt.



*Abbildung 10: Die markanten Punkte (links) und Beispiele für ermittelte Gesichtsformen(rechts) (aus [LTC95]).*

Für jedes Trainingsgesichtsbild wurde über die Punktkoordinaten die Gesichtsform ermittelt. Mit Hilfe der Hauptkomponentenanalyse (siehe Kapitel 2.4.1) konnten die wichtigsten geometrischen Informationen extrahiert werden. Um ein Gesicht zu lokalisieren, wurde die Mahalanobis-Distanz zwischen dem PDM und einem Beobachtungsausschnitt berechnet. Da es sich um ein Verfahren zur Gesichtsidentifikation handelte, sind keine Testergebnisse bezüglich der Lokalisierung vorhanden.

## **2.4 Ansichtsbasierter Ansatz**

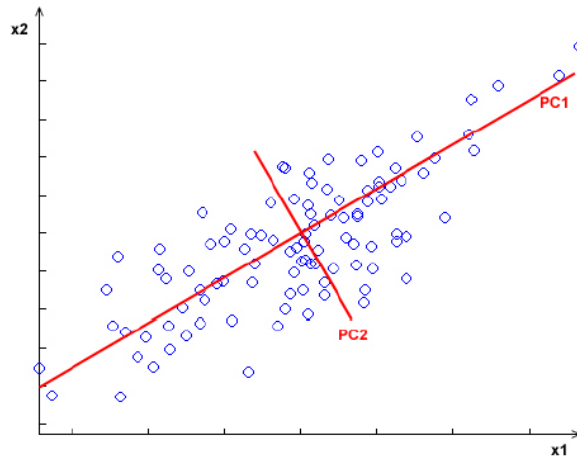
Ansichtsbasierte Verfahren benutzen keine vordefinierten Modelle der gesuchten Gesichter bzw. Gesichtsteile, sondern erzeugen diese über eine Anzahl von Trainingsgesichtsbildern. Dabei werden statistische Eigenschaften der Trainingsmuster extrahiert und gespeichert, meist verbunden mit einer zusätzlichen Informationsreduktion. Vorteil dieser Verfahren ist eine bessere Generalisierungsfähigkeit verbunden mit einer größeren Freiheit gegenüber den in Kapitel 1.1 beschriebenen Problemeigenschaften.

### **2.4.1 Eigenfaces**

Die Eigenfacemethode basiert auf der Hauptkomponentenanalyse (englisch: Principal Component Analysis (PCA)), Hauptachsentransformation oder Karhunen-Loève-Transformation. Bei diesem Verfahren werden aus mehreren ähnlichen n-dimensionalen Eingabevektoren die wichtigsten



gemeinsamen Faktoren extrahiert. Die Eingabevektoren stellen einen Unterraum im  $n$ -dimensionalen Eingaberaum  $\mathfrak{R}^n$  dar. Dieser Unterraum soll durch Vektoren, sogenannte Eigenvektoren, aufgespannt werden (siehe Beispiel in Abbildung 11).

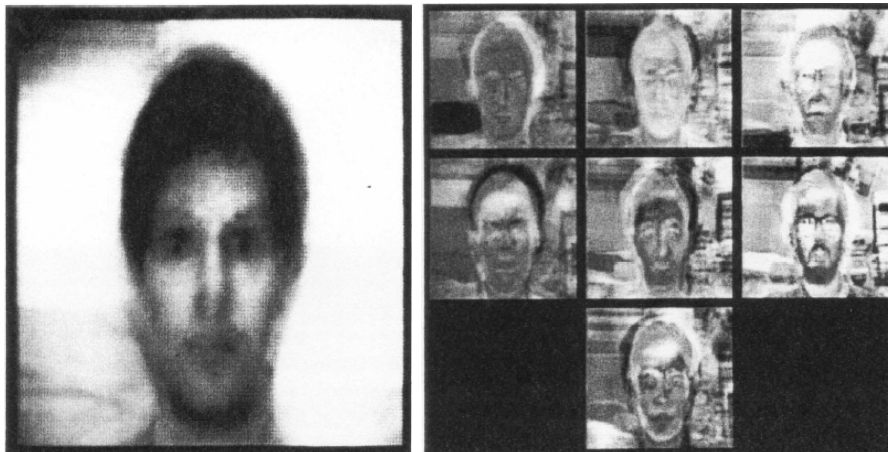


*Abbildung 11: Beispielfhafte Darstellung einer 2-dimensionalen Eingabemenge (blaue Kreise) im Eingaberaum  $\mathfrak{R}^2$  und die zwei zugehörigen Eigenvektoren (PC1 und PC2) (aus [GB05]).*

Anwendung findet dieses Verfahren beispielsweise in [TP91b]. Folgende Schritte werden dabei durchgeführt:

- Berechnung des Durchschnittsgesichtsbildes über alle  $M$  Gesichtsbilder
- Bilden der Differenzbilder der  $M$  Gesichtsbilder zum Durchschnittsgesichtsbild
- Berechnen der Eigenwerte und Eigenvektoren aus der Kovarianzmatrix der Differenzbilder. Eine ausgesuchte Menge von Eigenvektoren, mit den höchsten Eigenwerten, spannen den Unterraum (Gesichtsraum / Facespace) auf.
- Transformation der Gesichtsbilder in den Gesichtsraum. Die berechneten Vektoren werden als Referenzen gespeichert.

Um festzustellen, ob ein Bild ein Gesicht enthält, wird dieses Bild in den Gesichtsraum transformiert und der Abstand des resultierenden Vektors zu allen Referenzvektoren, also zum aufgespannten Gesichtsraum, berechnet und mit einem Schwellwert verglichen.



*Abbildung 12: Das Durchschnittsgesichtsbild (links) und die ersten 7 Eigenvektoren (rechts)(aus [TP91b]).*

Um die Eigenvektoren den gesuchten Unterraum möglichst genau aufspannen zu lassen, ist es erforderlich, die Trainingsgesichtsbilder zu normieren. So werden in [TP91b] weitere Vorverarbeitungsschritte, wie zum Beispiel die Entfernung des Hintergrundes, angewandt. Da es sich hier um ein Verfahren zur Gesichtsidentifikation handelte, sind keine Tests zur Gesichtslokalisierung durchgeführt worden.

### **2.4.2 Künstliche neuronale Netze**

Künstliche neuronale Netze, abgekürzt auch KNN genannt, sind Verfahren der künstlichen Intelligenz bei denen versucht wird, die Funktionsweise eines biologischen neuronalen Netzes zu simulieren.

Die Informationsweiterleitung erfolgt in einer Nervenzelle über das Axon an nachfolgende Nervenzellen, sobald die von den Vorgängerzellen über die Synapsen auf die Dendriten übertragenen Signale ein bestimmtes Potential überschreiten.

Analog zum biologischen Aufbau eines Neurons (Nervenzelle) aus Zellkern, Synapsen, Dendriten und Axon, besitzt auch das künstliche Neuron einen Zellkern, Dendriten (Verbindungen zu vorhergehenden Neuronen) und Axone (Verbindungen zu nachfolgenden Neuronen) (siehe Abbildung 13).

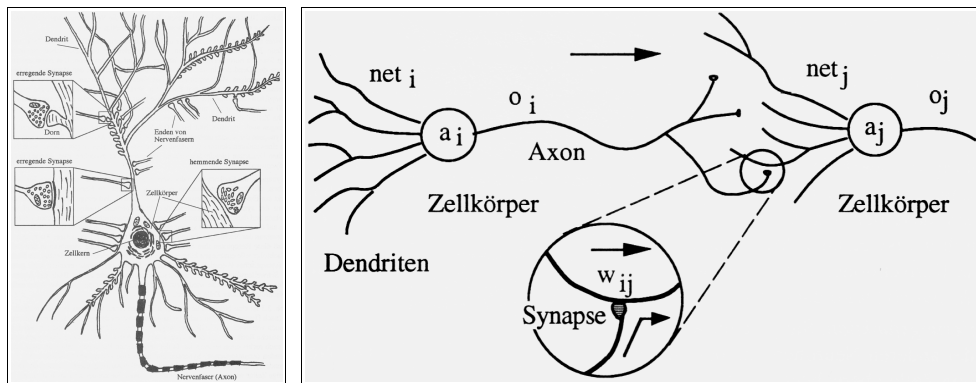


Abbildung 13: Links der schematische Aufbau einer biologischen Nervenzelle, rechts das abgeleitete Modell der künstlichen Neuronen (aus [Ze94]).

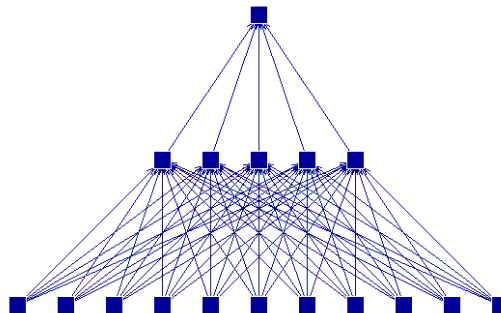
Konkret besteht ein neuronales Netz aus folgenden Bestandteilen (nach [Ze94]):

- **Neuronen (Zellen):**
  - Aktivierung  $a_i(t)$
  - Aktivierungsfunktion  $f_{act}$ . Diese gibt an, wie sich aus der alten Aktivierung und der Netzeingabe  $net_i(t)$  eine neue Aktivierung ergibt.  
Allgemeine Formel:  $a_i(t+1) = f_{act}(a_i(t), net_i(t), \Theta_i)$ ,  
mit  $\Theta_i(t)$ : Schwellenwert des Neurons i, ab dem es „feuert“, d.h. die Ausgabe weiterleitet
  - Ausgabefunktion  $f_{out}$ , welche die Ausgabe  $o_i = f_{out}(a_i)$  bestimmt.
  - Eingabeneuronen, die nur die Verbindung nach außen herstellen und keine Aktivierung, Aktivierungs- oder Ausgabefunktion benötigen, werden folgend inaktive Eingabeneuronen genannt. Demgegenüber sind alle Hidden- und Ausgabeneuronen aktive Neuronen. Hiddenneuronen sind zwischen Eingabe- und Ausgabeneuronen befindliche verdeckte Neuronen. Verdeckt deshalb, weil ein außenstehender Betrachter das KNN nur als Blackbox sieht.
- Das **Verbindungsnetzwerk** der Zellen stellt die gewichteten Verbindungen zwischen den Neuronen dar. Ein Gewicht der Verbindung von Neuron i nach Neuron j wird als  $w_{ij}$  bezeichnet. Die Matrix  $W$  beinhaltet alle Gewichte.
- Die **Propagierungsfunktion** gibt an, wie sich aus den Ausgaben der Vorgängerneuronen und den Verbindungsgewichten die Netzeingabe  $net$  eines Neurons berechnet. Für Neuron j berechnet sich  $net_j(t)$  wie folgt:  $net_j(t) = \sum_i o_i(t) w_{ij}$
- Als **Lernverfahren** wird der Algorithmus bezeichnet, mit dem das neuronale Netz lernt, für gegebene Eingaben eine gewünschte Ausgabe zu produzieren.

Lernverfahren werden allgemein in 3 Arten unterteilt:

1. *Überwachtes Lernen* (supervised learning). Für jedes Eingabemuster gibt es ein Ausgabemuster. Das Lernverfahren muss die Gewichte des KNN so anpassen, dass nach wiederholter Präsentation von Eingabe- und Ausgabemustern das Netz diese Assoziation selbstständig vornehmen kann. Außerdem sollte das KNN bei ähnlichen Eingabemustern auch ähnlich passende Ausgabemuster erzeugen (Generalisierung). Schnellste Lernmethode, aber biologisch unplausibel.
2. *Bestärkendes Lernen* (reinforcement learning). Statt eines Ausgabemusters je Eingabemuster erhält das KNN nur die Information, ob falsch oder richtig klassifiziert wurde. Langsamere Lernmethode, aber biologisch plausibler.
3. *Unüberwachtes Lernen* (unsupervised learning). Das KNN versucht die Eingabemuster selbstständig in Klassen zu trennen. Biologisch am plausibelsten, aber nicht für alle Problemstellungen geeignet.

Es gibt eine Vielzahl verschiedener Modelle künstlicher neuronaler Netze sowie verschiedene Arten von Verbindungsnetzwerken und verschiedene Aktivierungs- und Ausgabefunktionen.



*Abbildung 14: Schema eines KNN mit Eingabeneuronen (unten), Hiddenneuronen (mitte), Ausgabeneuron (oben) und einem vorwärtsgerichteten Verbindungsnetzwerk. Die Neuronen sind vollständig ebenenweise verbunden.*

Die Modelle unterscheiden sich durch die Topologie, welche die Verbindung und Reihenfolge der Neuronen beschreibt und lassen sich grob unterteilen in vorwärtsgerichtete, rekursive und selbstorganisierende KNN.

Von den vorwärtsgerichteten KNN ist das Perzeptron [MP69] das bekannteste und meistverwendete. Weitere Beispiele sind die RBF(Radiale-Basisfunktionen)-Netze [PG89] und die PNN (Probabilistische Neuronale Netze) [Sp90].

Zu den rekursiven KNN gehören beispielsweise Hopfield-Netze [Ho82] und der BAM (bidirektionaler Assoziativspeicher) [Kos87].

Die Vertreter der selbstorganisierenden KNN sind die SOM (self-organizing maps) von Kohonen [Ko90].

Rekursive Netze besitzen Verbindungsnetzwerke mit Rückkopplungen und benötigen zur Simulation eine zusätzliche Zeitbetrachtung sowie ein Abbruchkriterium. Diese Art von KNN ermöglicht die Berechnung einer Aktivierung unter Einbeziehung der Ausgabe nachfolgender Neuronen, vergleichbar mit einem Zwischenspeicher, beispielsweise bei der Verarbeitung einer Sequenz von Eingabesignalen.

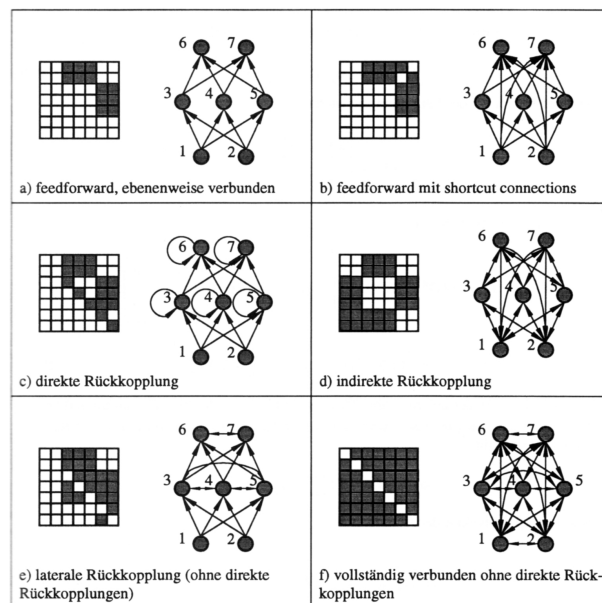
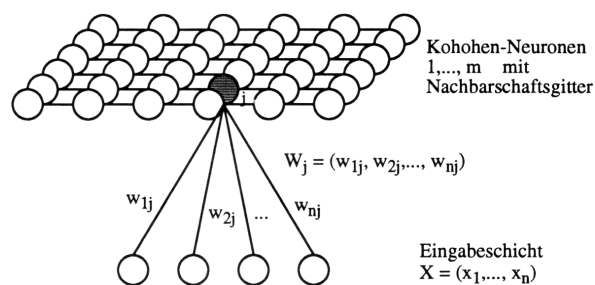


Abbildung 15: Beispiele für verschiedene Verbindungsnetzwerke und die entsprechenden Verbindungsmatrizen (aus [Ze94]).

Bei den selbstorganisierenden Karten (SOM) von Kohonen handelt es sich um KNN mit einer Schicht aktiver Neuronen, auch Kohonenkarte genannt, und einer Schicht inaktiver Eingabeneuronen. Die Eingabeneuronen sind

vollständig mit allen aktiven Neuronen verbunden. Zusätzlich existiert eine lokale Nachbarschaftsbeziehung zwischen den aktiven Neuronen. Als Lernverfahren werden unüberwachte Verfahren angewendet.

Nach Anlegen eines Eingabevektors wird in der Kohonenkarte das Neuron gesucht, dessen Gewichtsvektor dem Eingabevektor am ähnlichsten ist. Der Gewichtsvektor dieses als Gewinnerneuron bezeichneten Neurons wird so angepasst, dass er dem Eingabevektor noch ähnlicher wird. Die Gewichtsvektoren der innerhalb einer variablen Umgebung um das Gewinnerneuron liegenden Neuronen werden ebenfalls angepasst. Diese Anpassung kann beispielsweise in Form einer Gaußverteilung vorgenommen werden, d.h. mit zunehmender Entfernung der Umgebungsneuronen wird eine schwächere Anpassung der Gewichtsvektoren vorgenommen. Das KNN ist fertig trainiert, wenn ein stabiler Endzustand erreicht ist. Jede Klasse von Eingabevektoren wird von einem Bereich auf der Kohonenkarte repräsentiert (ein anschauliches Beispiel gibt [LGB05]).



*Abbildung 16: Darstellung einer selbstorganisierenden Karte in Form eines quadratischen Gitters (aus [Ze94]).*

Das bekannteste und weitverbreitetste Modell ist das Perzeptron [MP69] (siehe Abbildung 17), welches eigentlich eine Klasse von neuronalen Netzen beschreibt. Das Perzeptron besteht aus einer Schicht inaktiver Eingabeneuronen und mindestens einem Ausgabeneuron. Die Weiterentwicklung, das sogenannte MLP (Multi-Layer-Perzeptron), enthält zusätzlich beliebig viele Hiddenneuronen, die wiederum in verschiedene Schichten unterteilt sein können.

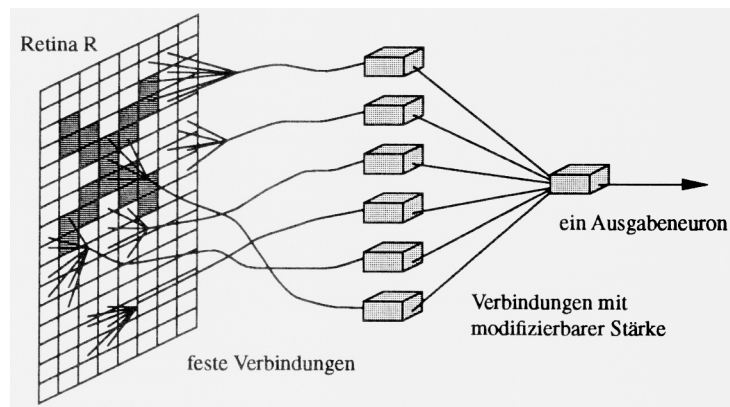


Abbildung 17: Struktur des Perzeptrons (aus [Ze94]).

Als Lernverfahren hat sich beim MLP das Backpropagationverfahren durchgesetzt. Es handelt sich dabei um ein Gradientenabstiegsverfahren, d.h. der Fehler jedes Trainingsmusters, ein Paar aus Eingabe- und Ausgabemuster, soll minimiert werden.

Die Anpassung der Gewichte der Verbindungen zu einem Neuron  $j$  für ein Muster  $p$  erfolgt nach

$$\Delta w_{ij} = \eta o_{pj} \delta_{pj} \quad ,$$

mit

$$\delta_{pj} = \begin{cases} f_{act}'(net_{pj})(t_{pj} - o_{pj}), & \text{falls } j \text{ ein Ausgabeneuron ist} \\ f_{act}'(net_{pj}) \sum_k \delta_{pk} w_{jk}, & \text{falls } j \text{ ein Hiddenneuron ist} \end{cases}$$

für alle  $k$  dem Hiddenneuron nachfolgenden Neuronen.

Der Trainingszustand eines MLP wird meist über den MSE (Mean-Squared-Error: gemittelter summierter Fehler über alle Trainingsmuster) beurteilt.

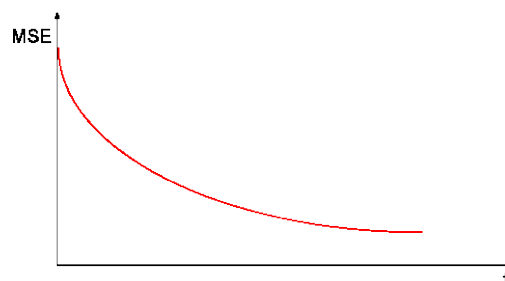


Abbildung 18: Verlauf der idealen Trainingskurve

Allerdings neigen KNN dazu, bei zu hoher Zykluszahl (Zyklus: Anlegen der gesamten Trainingsmustermenge) ihre Generalisierungsfähigkeit zu verlieren, was auch als Übertraining bezeichnet wird. Auch verläuft die Fehlerkurve nicht immer so kontinuierlich wie in Abbildung 18, so wird es schwierig das globale Minimum zu finden.

KNN wurden bereits in zahlreichen Verfahren zur Gesichtsdetektion angewandt.

In [Ro99] verwendete Rowley beispielsweise ein MLP mit mehreren nebeneinanderliegenden Rezeptiven Feldern und einer Schicht Hiddenneuronen zur Klassifikation von Bildausschnitten (siehe Abbildung 19). Rezeptives Feld bedeutet hier eine Eingabeschicht zur Betrachtung einzelner Teile des Bildausschnitts, zum Beispiel in horizontalen Streifen. Zu jedem rezeptiven Feld gehört ein Teil der Hiddenneuronen. Die Reaktionswerte der Hiddenneuronen wurden in einem gemeinsamen Ausgabeneuron verarbeitet. Die Grauwerte des Bildausschnitts wurden zuvor in Werte zwischen -1.0 und 1.0 umgewandelt. Über die Ausgabe des Ausgabeneurons wurde der Bildausschnitt klassifiziert.

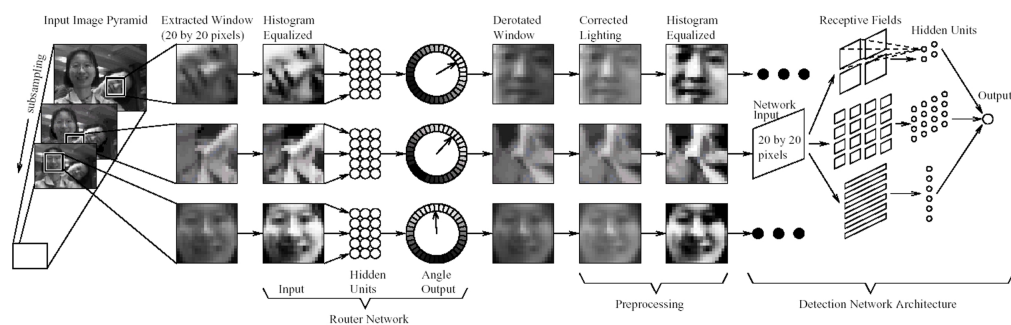


Abbildung 19: Funktionsweise des Verfahrens von [Ro99] (aus [Ro99]).

Um eine Positionsinvarianz zu erreichen, wurde das KNN an jedem Pixel des Bildes angewendet. Eine Skalierungsinvarianz erreichte man, indem das Bild in mehreren Schritten um den Faktor 1,2 kleiner skaliert wurde.

In der Vorverarbeitungsstufe wurden die Bildausschnitte beleuchtungskorrigiert und mit einem Histogrammausgleich kontrastverstärkt. Ein weiteres KNN wurde zur Derotation trainiert, um das Verfahren Rotationsinvariant zu machen (siehe Abbildung 19).



Das Verfahren erreichte eine durchschnittliche Detektionsrate von 76,6 % über verschiedene Testsets.

### 2.4.3 Hidden-Markov-Modell

Dieses bereits erfolgreich in der Spracherkennung und Handschrifterkennung eingesetzte Verfahren basiert auf der Beschreibung der statistischen Eigenschaften von Signalen. Ein Hidden-Markov-Modell, HMM, besteht aus einer Markov-Kette von Zuständen (siehe Abbildung 20).

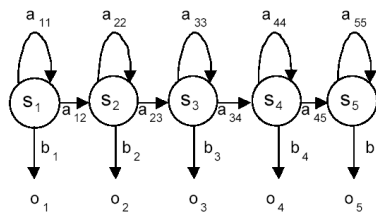


Abbildung 20: Aufbau eines Hidden-Markov-Modells (aus [Mi00]).

Für jeden Zustand  $s_i$  gibt es die Übergangswahrscheinlichkeit  $a_{ii}$  (in den gleichen Zustand  $s_i$ ),  $a_{ij}$  (in den nächsten Zustand  $s_j$ ) und die Ausgabewahrscheinlichkeit  $b_i$  für die Ausgabe  $o_i$ . Bei der Mustererkennung mit HMM geht man davon aus, dass die Erkennung ein über die Übergangswahrscheinlichkeiten parametrisierbarer Zufallsprozess ist. Dabei müssen Bereiche des Musters als Zustände definiert werden.

In [Sa93] wandte Samaria dieses Verfahren zur Gesichtsidentifikation an. Als Zustände definierte er die folgenden, als Bänder bezeichneten, horizontalen Bereiche (von oben nach unten): Stirn, Augen, Nase, Mund und Kinn (siehe Abbildung 21).

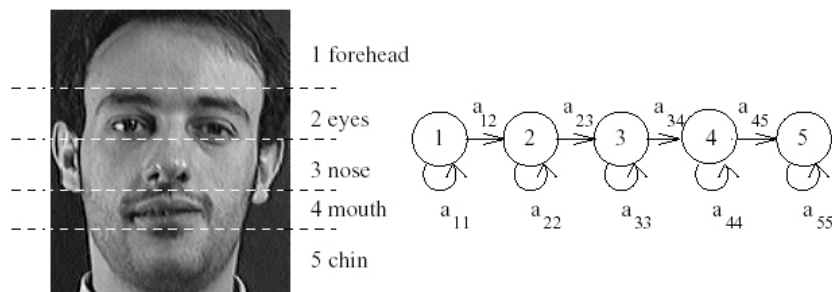


Abbildung 21: Darstellung der 5 relevanten Bereiche / Zustände (aus [Sa93]).

Über eine Anzahl von Bildern je Person wurde für jede Person ein HMM erzeugt. Das Modell einer Person  $k$  wird repräsentiert durch

$$\lambda^{(k)} = (A^{(k)}, B^{(k)}, \pi^{(k)}), 1 \leq k \leq F, \text{ mit:}$$

- $F$  : Anzahl Personen
- $A^{(k)}$  : Übergangswahrscheinlichkeiten in das nächste Band
- $B^{(k)}$  : Übergangswahrscheinlichkeiten in das gleiche Band
- $\pi^{(k)}$  : Anfangswahrscheinlichkeiten, werden festgelegt nach

$$\pi_1^{(k)} = 1 \text{ und } \pi_i^{(k)} = 0, \quad 1 < i \leq N, \quad N: \text{ Anzahl an Zuständen}$$

Für die Erkennung wurde ein HMM eines Testbildes erstellt und mit jedem gespeicherten Modell verglichen.

Da dieses Verfahren für die Gesichtsidifikation entwickelt wurde, war Generalisierungsfähigkeit über alle Gesichter nicht von Bedeutung. Für die Gesichtslokalisierung müsste man ein HMM über eine große Anzahl von Gesichtsbildern trainieren. Es wurden außerdem keine Tests zur Gesichtslokalisierung durchgeführt.

## **2.5 Zusätzliche Betrachtung der Bildvorverarbeitung**

### **2.5.1 Kantenfilter**

In verschiedenen Verfahren wurden Kantenfilter angewendet ([YC96], [YH94]), um beispielsweise geometrische Eigenschaften zu ermitteln oder eine Beleuchtungsinvarianz zu erreichen. Sie reduzieren die Bildinformation auf Helligkeitsübergänge.

Kantenfilter sind im Allgemeinen lokale lineare Differenzoperatoren und suchen Grauwertunterschiede zwischen Pixeln. Meist bestehen sie aus einer quadratischen  $m \times m$ -Matrix, mit ungeradem  $m$ , welche Faltungsmatrix oder Faltungskern genannt wird. Die Kantenfilter werden auf jedes Pixel des Bildes oder Bildausschnittes angewendet.

Nachfolgende werden die bekanntesten Kantenfilter vorgestellt.

### Sobel-Operator

Der Sobel-Operator ist ein richtungsabhängiger Operator 1. Ordnung. Das heißt, dass er die diskrete 1. Ableitung verwendet und Bilder nach horizontalen oder vertikalen Grauwertdifferenzen filtert (siehe Abbildung 22).

Der Sobel-Operator verwendet folgende 3 x 3 -Matrizen:

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A \quad \text{und} \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

mit:

- A: Originalbild
- $G_x$ : horizontales Gradientenbild
- $G_y$ : vertikales Gradientenbild

Eine Kombination beider Ergebnisse nach  $G = \sqrt{G_x^2 + G_y^2}$  ergibt richtungsunabhängige Informationen.

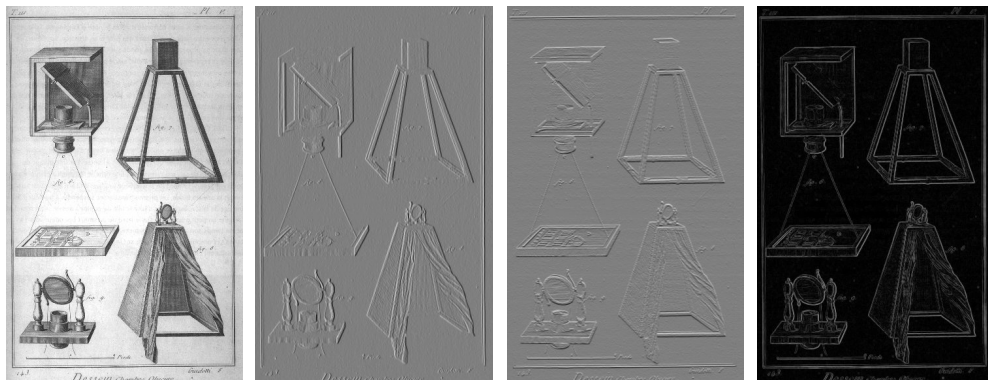


Abbildung 22: Beispiel für die Anwendung des Sobel-Operators (Nullwerte sind als mittleres Grau dargestellt) (v.l.): A,  $G_x$ ,  $G_y$ , G (aus [Wi05]).

Die Richtung einer Kante lässt sich nach  $\Theta = \arctan\left(\frac{G_y}{G_x}\right)$  berechnen.

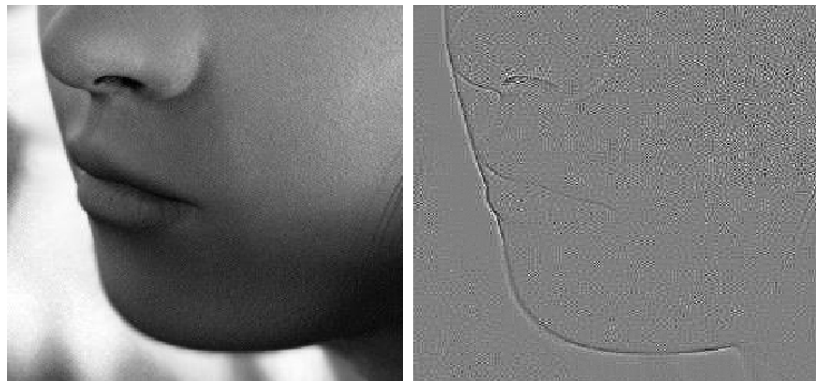
### **Laplace-Operator**

Der Laplace-Operator ist ein richtungsunabhängiger Operator 2. Ordnung. Das heißt, er verwendet die diskrete 2. Ableitung. Er kann im Bild beliebig orientierte Kanten hervorheben und verwendet dazu folgende Matrix:

$$G = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} * A$$

mit:

- A: Originalbild
- G: Gradientenbild



*Abbildung 23: Beispiel für die Anwendung des Laplace-Operators (aus [Ha03]).*

Dieser Kantenfilter ist allerdings sehr rauschempfindlich.

### **Canny-Algorithmus**

Der Canny-Algorithmus arbeitet mit verschiedenen Faltungsoperatoren. Im ersten Schritt glättet er das Bild mit einem Gaußfilter. Danach wird das Bild nach Kanten gefiltert. Abschließend werden die Pixel mittels zweier Schwellwerte gruppiert (Non-maxima Suppression).

Die Ergebnisse ist sehr gut, wenig rauschempfindlich, aber rechnerisch aufwändig.

**SUSAN-Kantenfilter**

Smith und Brady entwickelten in [SB95] das SUSAN-Prinzip und leiteten davon unter anderem einen Kantenfilter ab. Der SUSAN-Kantenfilter ist ein lokaler nichtlinearer Operator, der in Form einer runden Maske mit einem Zentrum, dem Nukleus, über das Bild bewegt wird (siehe Abbildung 24).

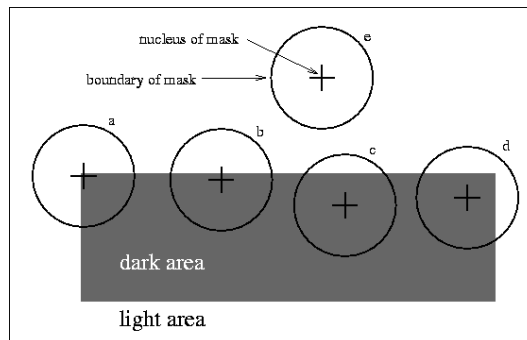


Abbildung 24: Funktionsweise des SUSAN-Kantenfilters (aus [SB95]).

SUSAN steht für „Smallest USAN“, wobei mit der USAN, „Univalued Segment Assimilating Nucleus“, alle dem Nukleus, bezüglich des Grauwertes, ähnlichen Pixel innerhalb der Maske gemeint sind. Betrachtet wird die Ähnlichkeit zwischen dem Pixel im Maskenmittelpunkt, dem Nukleus  $\vec{r}_0$ , und aller innerhalb der Maske liegender Umgebungspixel  $\vec{r}$  mit Hilfe des Vergleichsoperators  $c(\vec{r}, \vec{r}_0), 0 \leq c \leq 1$ .

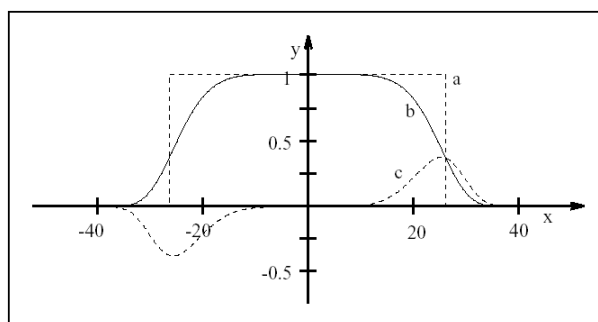


Abbildung 25: Funktionsdarstellung des Vergleichsoperators. Die X-Achse stellt die Grauwertdifferenz und die Y-Achse das Vergleichsergebnis dar. (mit einem Beispielkontrast  $t = 27$ ) (aus [SB95]).

Die Berechnung von  $c$  erfolgt entweder mit einer Schwellwertfunktion (Funktion a, siehe Abbildung 25) nach

$$c(\vec{r}, \vec{r}_0) = \begin{cases} 1, & \text{falls } |I(\vec{r}) - I(\vec{r}_0)| \leq t \\ 0, & \text{falls } |I(\vec{r}) - I(\vec{r}_0)| > t \end{cases}$$

oder mit einer glättenden Funktion (Funktion b, siehe Abbildung 25) nach

$$c(\vec{r}, \vec{r}_0) = e^{-\left(\frac{I(\vec{r}) - I(\vec{r}_0)}{t}\right)^6}$$

$I(\vec{x})$  ist der Grauwert von Pixel  $x$  und  $t$  wird als Kontrast bezeichnet.

Die USAN  $n$  berechnet sich nach  $n(\vec{r}_0) = \sum_{\vec{r}} c(\vec{r}, \vec{r}_0)$ .

Alle als ähnlich bestimmten Pixel gehen mit einem Wert zwischen 0 und 1 in die USAN ein, der Wert der USAN entspricht somit maximal der Anzahl der in der Maske vorhandenen Pixel.

Über den Wert der USAN wird eine Kante bestimmt. Wie in Abbildung 24 zu erkennen, müsste beispielsweise Maske b mit einer USAN von ca.  $\frac{1}{2}$  der maximal möglichen USAN-Größe eine Kante detektieren und Maske c mit der maximalen USAN-Größe sollte keine Kante detektieren.

Die USAN wird dazu mit einem Schwellwert  $g$  verglichen.

$$R(\vec{r}_0) = \begin{cases} g - n(\vec{r}_0), & \text{falls } n(\vec{r}_0) < g \\ 0, & \text{andererseits} \end{cases}$$

Entsprechend dem Namen SUSAN, Smallest USAN, wird  $n$  von  $g$  abgezogen, so dass der Reaktionswert  $R(\vec{r}_0)$  der Maske umso größer wird, je kleiner  $n$  wird.

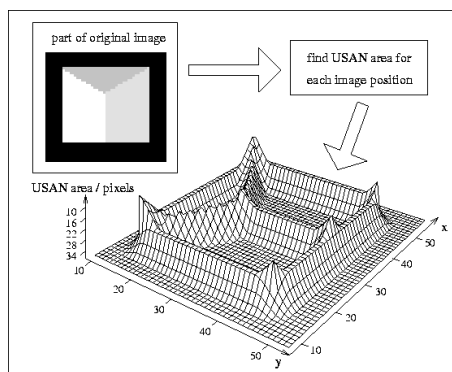


Abbildung 26: Die Reaktionswerte des SUSAN-Filters auf ein Beispielbild (aus [SB95]).

Hierbei wird deutlich, dass Ecken und Rauschen einen starken Reaktionswert erzeugen. Weitere Berechnungen sind zur Ermittlung der Kantenrichtung beschrieben (siehe [SM95]), diese können unter anderem zur Nicht-Maxima Unterdrückung (Non-maxima Suppression) verwendet werden.

### 2.5.2 Beleuchtungskorrektur

In [Ro99] wird zur Normierung des Bildausschnitts eine Beleuchtungskorrektur vorgenommen, dabei wird versucht, mögliche lineare Helligkeitsverläufe zu detektieren und zu eliminieren. Dieser Vorverarbeitungsschritt ist vor allem dann sinnvoll, wenn die Klassifikatoren direkt mit den Pixelwerten arbeiten. Doch auch Kantenfilter reagieren bei einer gleichmäßigen Kante unter einem Helligkeitsverlauf mit unterschiedlich starken Reaktionswerten.

### 2.5.3 Histogrammausgleich

Der Histogrammausgleich ist ein einfaches Verfahren zur Kontrastverstärkung. Das Histogramm enthält die Häufigkeit der einzelnen Grauwerte zwischen 0 (schwarz) und 255 (weiß). Ist das Bild beispielsweise zu dunkel, häufen sich die Grauwerte mit kleineren Beträgen.

Ermittelt man den kleinsten  $E_{min}$  und den größten  $E_{max}$  dargestellten Grauwert, so lassen sich nach

$$g_A(x, y) = \frac{255}{E_{max} - E_{min}} (g_E(x, y) - E_{min}),$$

$$0 \leq x \leq \text{Breite des Bildes}, 0 \leq y \leq \text{Höhe des Bildes},$$

mit

$g_A(x, y)$ : Grauwert des Ausgangsbildes an der Stelle  $x, y$ ,

$g_E(x, y)$ : Grauwert des Eingangsbildes an der Stelle  $x, y$

die Grauwerte auf den gesamten zur Verfügung stehenden Bereich abbilden (siehe Abbildung rechts). Dieser Verfahren wird auch lineares Dehnen bezeichnet.

## 2.6 Bewertung der Verfahren

Ein Vergleich der Ansätze bezüglich der Detektionsrate ist nicht möglich, da in den Verfahren unterschiedliche Gesichtsbilder zum Testen verwendet wurden. Unter alleiniger Betrachtung der Zahlenwerte lagen die Detektionsergebnisse zwischen 83 % [YH94] und 92 % [YC96].

Bewertet man die Ansätze bezüglich ihrer Robustheit gegenüber den eingangs beschriebenen Problemeigenschaften, wird deutlich, dass die *wissensbasierten Verfahren* aufgrund ihrer Einfachheit geringe Freiheiten besitzen. So können nur unverdeckte, nicht rotierte Gesichter in der Frontalansicht detektiert werden und die Beleuchtungssituation darf nur in geringem Maße variieren. Sie eignen sich eher zur Gesichtslokalisierung.

Alle anderen Ansätze ermöglichen eine stärkere Robustheit durch aufwendige Suchstrategien und Bildvorverarbeitung. Die gewählte Suchstrategie bestimmt beispielsweise, ob ein Gesicht oder mehrere Gesichter gefunden werden können. Sie bestimmt außerdem, ob die Gesichter rotiert sein, beliebig groß und sich an beliebigen Positionen befinden dürfen. Die Bildvorverarbeitung bestimmt die Unabhängigkeit von Problemeigenschaften des Aufnahmesensors und des Bildes.

*Erscheinungsbasierte Verfahren* besitzen, da ihre Modelle auf statistischen Eigenschaften basieren, eine sehr gute Generalisierungsfähigkeit. Im allgemeinen werden sie bei der Suche nach ganzen Gesichtern eingesetzt. Sie müssen, um invariant gegenüber verschiedenen Gesichtsausdrücken und den veränderlichen Problemeigenschaften zu sein, entweder mit einer größeren Anzahl von Trainingsgesichtsbildern mit verschiedenen Problemeigenschaften trainiert oder der Bildausschnitt muss so klein gewählt werden, dass Gesichtsausdrücke kaum noch erkennbar sind (20x20 Pixel, siehe [Ro99]). Sollen die Verfahren auch invariant gegenüber unterschiedlicher Orientierung des Gesichtes sein, müssen schon ab einem kleinen Winkel zusätzliche Modelle generiert und diese parallel angewendet werden (angewandt in [Ro99]).

Demgegenüber haben *merkmalsbasierte Verfahren* die nach *Gesichtsmerkmalen* suchen den Vorteil, dass die Relationen zwischen den



Gesichtsmerkmalen als elastisch betrachtet werden kann. So besitzen sie stärkere Robustheit gegenüber der Orientierung des Gesichtes [YC96], müssen aber auch ab einem bestimmten Winkel zusätzliche Modelle der Gesichtsmerkmale generieren. Problemeigenschaften wie Gesichtsausdrücke, Bekleidung oder Verdeckung spielen eine untergeordnete Rolle, da sich Gesichtsausdrücke und Bekleidung weniger auf die einzelnen Gesichtsmerkmale auswirken und Verdeckung durch die Kombination der Gesichtsmerkmale kompensiert werden kann. Ist beispielsweise das rechte Auge verdeckt, lässt sich das Gesicht über die Kombination von linkem Auge, Nase und Mund detektieren.

Die Detektion von Gesichtern mittels *Farbe* nimmt eine Sonderstellung ein. Ein Farbbild vorausgesetzt, ist dieses Verfahren robust gegenüber fast jeder Problemeigenschaft, außer der Gesichtsfarbe selbst. Denn sollte ein Gesicht bemalt sein, versagt dieses Verfahren. Weitere Schwierigkeiten bereiten unterschiedliche Spektren der Hautfarbe verschiedener ethnischer Gruppen und verschiedener Lichtquellen und der Hautfarbe ähnliche Oberflächen. Farbe eignet sich zudem nur zur Detektion von Gesichtskandidaten. Ob es sich wirklich um ein Gesicht handelt, muss mit einem der anderen Ansätze validiert werden.

*Template Matching Verfahren* haben, je nachdem ob sie nach ganzen Gesichtern oder nach Gesichtsteilen suchen, ähnliche Robustheitseigenschaften. Allerdings leidet die Generalisierungsfähigkeit durch die relativ einfache Modellierung der Muster mit geometrischen Werten. Sie sind daher wenig für die Gesichtsdetektion geeignet.

Am erfolgversprechendsten ist eine Kombination von merkmalsbasierten Verfahren (Gesichtsmerkmalen) mit erscheinungsbasierten Verfahren, dass heißt die Modelle der Gesichtsmerkmale werden über Trainingsgesichtsbilder generiert und enthalten statistische Eigenschaften. Damit kann sowohl eine große Generalisierungsfähigkeit, als auch eine starke Robustheit gegenüber einer großen Anzahl von Problemeigenschaften, einschließlich der Verdeckung, erreicht werden. Diese Schlussfolgerung führt zu dem in Kapitel 3 beschriebenen Lösungsansatz.

### 3 Lösungsansatz zur Detektion von Gesichtsmerkmalen

Ziel des in dieser Arbeit entwickelten Verfahrens ist die Detektion von Gesichtsmerkmalen, wie Augen, Nase und Mund. Damit wird das Gesicht nicht als Ganzes betrachtet, sondern als Kombination dieser Merkmale.

Motiviert wurde dieser Ansatz von den Ergebnissen der Betrachtungen in Kapitel 2. Forschungen haben zudem ergeben, dass der Mensch bei der Erkennung eines Gesichtes diese Merkmale präferiert [HFMDM00].

Zur Modellierung der Gesichtsmerkmale werden künstliche neuronale Netze eingesetzt. Sie besitzen eine sehr gute Generalisierungsfähigkeit und benötigen während der Anwendung keinen hohen Rechenaufwand. Um die KNN mit einer großen Anzahl Trainingsgesichtsbilder trainieren zu können, wird die BioID-Gesichtsdatenbank verwendet.

Der Suchvorgang über ein Bild erfolgt mit der Betrachtung einzelner Bildausschnitte. Diese Bildausschnitte werden durch ein Fenster repräsentiert, welches skaliert und gedreht werden kann.

In einer Vorverarbeitungsstufe werden die Pixelgrauwerte (Farbwerte werden vorher in Grauwerte umgewandelt) beleuchtungskorrigiert, kontrastverstärkt und von einem Kantenfilter verarbeitet.

Folgende Nebenbedingungen werden festgelegt:

- Gesichter dürfen nicht kleiner als 60 x 60 Pixel sein
- Gesichter müssen vollständig im Bild enthalten sein
- Gesichter dürfen nicht stärker als 5° aus der Frontalansicht gedreht sein (Orientierung)
- Gesichter mit Brillen und Bärten bleiben unberücksichtigt

Jedes Gesichtsmerkmal hat somit in jedem Bild folgende Freiheiten:

- unterschiedliche Größe (begrenzt in Relation zur Gesichtsgröße)
- beliebiger Rotationswinkel (in der Bildebene)
- beliebige Beleuchtungssituation
- beliebige Anzahl (mehrere Gesichter)

Um in einer späteren Verarbeitungsstufe die Gesichtsmerkmale zu einem Gesicht kombinieren zu können, ist es wichtig, dass sie möglichst positions-, skalierungs- und winkelgenau klassifiziert werden.

Die Effizienz bezüglich der Verarbeitungszeit soll keine Rolle spielen.

## 3.1 BioID-Gesichtsdatenbank



Abbildung 27: Beispiele von Gesichtsbildern der BioID-Gesichtsdatenbank (aus [Hu05]).

Die BioID-Gesichtsdatenbank [Hu05] zeichnet sich durch ihre Größe (1521 Grauwertbilder von 23 verschiedenen Personen), die komplexen Hintergründe und die zusätzlichen Punktwerte aus. Zu jeder Bilddatei gibt es eine zugehörige Datei mit den Koordinaten einzelner Punkte..

Die Punkte sind zeilenweise in reellwertigen Koordinatenpaaren gespeichert. Der linke Wert entspricht der x-Koordinate. Der Koordinatenursprung liegt in der linken, oberen Ecke des Bildes. Nachfolgend wird ein Beispiel anhand der Datei „bioid\_0000.pts“ gezeigt. Zum besseren Verständnis wurden die Entsprechungen hinzugefügt.

```
version: 1
n_points: 20
{
159.128 108.541 // rechte Pupille
230.854 109.176 // linke Pupille
164.841 179.633 // rechte Mundecke
223.237 178.998 // linke Mundecke
132.469 93.9421 // Außenende der rechten Augenbraue
183.883 94.5768 // Innenende der rechten Augenbraue
211.177 95.2116 // Innenende der linken Augenbraue
254.974 91.4031 // Außenende der linken Augenbraue
129.295 109.176 // rechte Schläfe
144.529 109.811 // Außenecke des rechten Auges
176.901 107.272 // Innenecke des rechten Auges
216.89 107.272 // Innenecke des linken Auges
246.088 110.445 // Außenecke des linken Auges
261.957 109.811 // linke Schläfe
196.578 139.009 // Nasenspitze
184.518 147.261 // rechtes Nasenloch
207.369 145.991 // linkes Nasenloch
195.943 175.189 // Mittelpunkt der Außenkante der Oberlippe
193.404 193.597 // Mittelpunkt der Außenkante der Unterlippe
192.769 229.143 // Kinnspitze
}
```

In Abbildung 28 sind die Punkte in das zugehörige Bild eingezeichnet.



Abbildung 28: Die Punkte aus Datei *bioid\_0000.pts* in Bild *BioID\_0000.jpg* (aus [Hu05]).

Die für das jeweilige Gesichtsmerkmal relevanten Punkte sind, für:

- *das rechte Auge*: rechte Pupille, Außenende der rechten Augenbraue, Innenende der rechten Augenbraue, Außenecke des rechten Auges, Innenecke des rechten Auges
- *das linke Auge*: linke Pupille, Außenende der linken Augenbraue, Innenende der linken Augenbraue, Außenecke des linken Auges, Innenecke des linken Auges

• *die Nase*: Nasenspitze, linkes Nasenloch, rechtes Nasenloch

4 *den Mund*: rechte Mundecke, linke Mundecke, Mittelpunkt der Außenkante der Oberlippe, Mittelpunkt der Außenkante der Unterlippe

Die Gesichtsbilder, auf denen die Personen Bärte oder Brillen tragen, wurden entfernt. (siehe Anhang A: Entfernte Bilder). Alle Bilder wurden einzeln betrachtet und beim Auftreten von Bart oder Brille aussortiert.

## 3.1 Gesichtsmerkmale

### 3.1.1 Das Normgesicht

Die Definition eines Normgesichtes hat für diese Arbeit den Zweck der Festlegung der Grenzen und der Mindestgröße eines Gesichtes. Anhand dieses Normgesichtes können die Größen der Gesichtsmerkmale ermittelt werden.

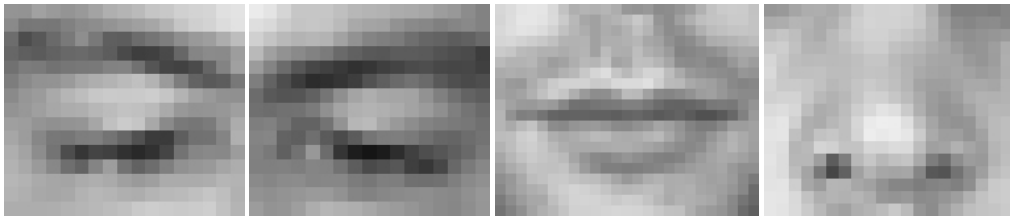
Das Normgesicht wurde auf der Basis des ersten Bildes der Bilddatenbank erstellt. Bei der Festlegung der Grenzen des Gesichtes wurden die Ergebnisse in [HFMDM00] berücksichtigt. Die Größe des Normgesichtes von 60 x 60 Pixel ergab sich aus der Überlegung, bis zu welcher Größe der Mensch in der Lage ist Gesichter zweifelsfrei zu identifizieren. Denn wann immer Gesichter automatisiert detektiert werden, sollen sie meist einer weiteren Verarbeitungsstufe zugeführt werden und ein 20 x 20 Pixel großes Gesicht ist schwierig von einem ähnlichen Gesicht zu unterscheiden.



*Abbildung 29: Entwicklung des Normgesichtes (v.l.): die Regionen des Gesichtes nach Priorität nummeriert (aus [HFMDM00]), Definiertes Normgesicht unter Berücksichtigung der wichtigsten Regionen, auf 60 x 60 Pixel verkleinertes Normgesicht*

### 3.1.2 Größe der Gesichtsmerkmale

Die Größe der einzelnen Gesichtsmerkmale wurde durch visuelle Kontrolle bestimmt. Dabei galt es zu beachten, die Ausschnitte einerseits nicht zu groß zu wählen, um nur wirklich relevante Bereiche einzuschließen. Andererseits durften sie nicht zu klein sein, um nicht Variationen des jeweiligen Gesichtsmerkmals auszuschließen.



*Abbildung 30: Darstellung der Größen der Gesichtsmerkmale im Normgesicht (v.l.):  
rechtes Auge, linkes Auge, Mund, Nase.*

In Abbildung 30 sind die ausgewählten Größen der Gesichtsmerkmale dargestellt. Konkret ergeben sich folgende Größen:

- rechtes und linkes Auge :  $17 \times 15 = 255$  Pixel
- Mund :  $25 \times 20 = 500$  Pixel
- Nase :  $20 \times 17 = 340$  Pixel

Bei der Auswahl des Bereichs um die Augen wurde Wert darauf gelegt, die Augenbrauen mit einzuschließen, um das Auge stärker zu charakterisieren und es damit dem KNN leichter zu machen, es von „Nichtaugen“ zu unterscheiden. Der Bereich um den Mund ist so hoch gewählt, um auch einen geöffneten Mund mit einzuschließen. Die Größe des Bereichs um die Nase ergibt sich aus möglichen unterschiedlichen Neigungen der Nase.

### **3.1.3 Extraktion des Bildausschnittes**

Der Bildausschnitt wird aus dem Bild extrahiert, indem ein Fenster über das Bild bewegt wird. Das Fenster besteht aus einem zweidimensionalen quadratischen Feld von Punkten. Die Seitenlängen dieses Feldes sind ungerade und jeder Punkt enthält nach der Initialisierung die Abstandskordinaten zum Fenstermittelpunkt, der selbst die Koordinaten 0,0 (x, y) hat. Ist das Fenster beispielsweise  $25 \times 25$  Punkte groß, so hat der Punkt links oben, da das Bild den Koordinatenursprung in der linken oberen Ecke hat, die Koordinaten -12, -12.

$$\begin{bmatrix} (-12,-12) & (-11,-12) & \dots & (0,-12) & \dots & (11,-12) & (12,-12) \\ (-12,-11) & (-11,-11) & \dots & (0,-11) & \dots & (11,-11) & (12,-11) \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ (-12,0) & (-11,0) & \dots & (0,0) & \dots & (11,0) & (12,0) \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ (-12,11) & (-11,11) & \dots & (0,11) & \dots & (11,11) & (12,11) \\ (-12,12) & (-11,12) & \dots & (0,12) & \dots & (11,12) & (12,12) \end{bmatrix}$$

Matrixdarstellung der Koordinaten der Punkte des Beispielfensters (25 x 25 Punkte).

Ausgehend vom Pixel im Bild, folgend Zentrumpixel genannt, an dem sich der Fenstermittelpunkt befindet, wird jedem Fensterpunkt ein Bildpixel zugeordnet. Dazu werden zu den Koordinaten des Zentrumpixels die Koordinaten des jeweiligen Fensterpunktes addiert.

$$\begin{bmatrix} (108,38) & (109,38) & \dots & (120,38) & \dots & (131,38) & (132,38) \\ (108,39) & (109,39) & \dots & (120,39) & \dots & (131,39) & (132,39) \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ (108,50) & (109,50) & \dots & (120,50) & \dots & (131,50) & (132,50) \\ \vdots & \vdots & & \vdots & & \vdots & \vdots \\ (108,61) & (109,61) & \dots & (120,61) & \dots & (131,61) & (132,61) \\ (108,62) & (109,62) & \dots & (120,62) & \dots & (131,62) & (132,62) \end{bmatrix}$$

Matrixdarstellung der Koordinaten der zugeordneten Bildpixel, die sich aus einem Zentrumpixel (120, 50) und obigem Beispielfenster ergeben.

Die Pixelgrauwerte dieser Bildpixel werden gespeichert und weiterverarbeitet. In Abbildung 31 sind das Fenster im Bild und der extrahierte Bildausschnitt dargestellt.



Abbildung 31: Rechts: Darstellung des Beispielfensters im Normgesicht mit Zentrumpunktkoordinaten 79,110. Die den Fensterpunkten zugeordneten Bildpixel sind blau gefärbt. Links: Der extrahierte Bildausschnitt.

Dieses Fenster kann skaliert werden, indem alle Koordinaten mit dem Skalierungsfaktor multipliziert werden.

Gedreht wird das Fenster mittels Koordinatenrotation um den Ursprung, hier den Fenstermittelpunkt, nach

$$x' = x \cdot \cos \phi + y \cdot \sin \phi \quad , \quad y' = y \cdot \cos \phi - x \cdot \sin \phi \quad ,$$

mit  $x', y'$ : neue Koordinaten des Punktes,  $x, y$ : alte Koordinaten des Punktes und  $\phi$ : Drehwinkel.

In Abbildung 32 ist die Funktionsweise an einem Beispiel mit einem Skalierungsfaktor und einem Winkelwerten dargestellt (Testergebnisse werden in Kapitel 5.1 vorgestellt).

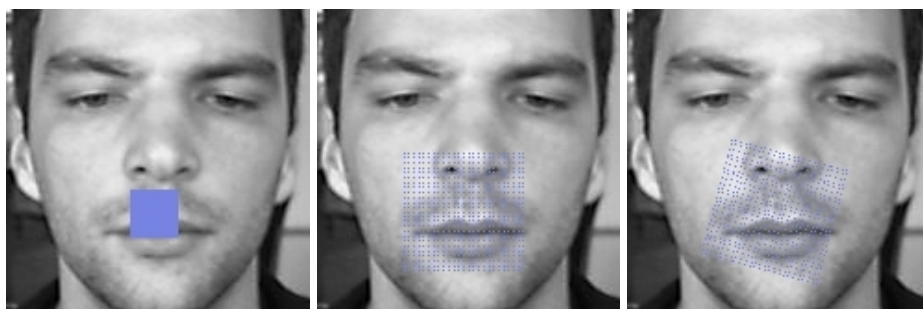


Abbildung 32: Darstellung der Skalierung und Rotation des Beispielfensters im Normgesicht mit Zentrumpunktkoordinaten 79,110. Die den Fensterpunkten zugeordneten Bildpixel sind blau gefärbt (v.l.): Skalierungsfaktor 1.0 und Winkel  $0^\circ$ , Skalierungsfaktor 2.7 und Winkel  $0^\circ$ , Skalierungsfaktor 2.7 und Winkel  $345^\circ = -15^\circ$ .



### **3.2 Bildvorverarbeitung/Normierung**

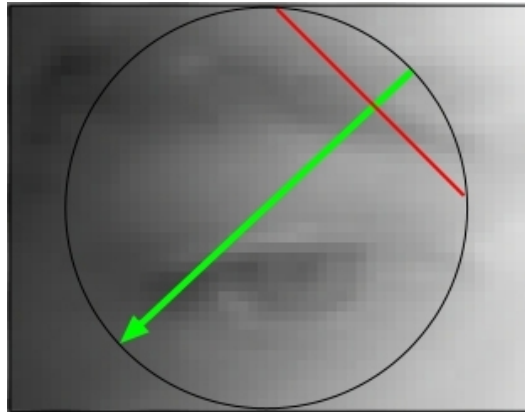
Um eine Beleuchtungsinvarianz des Verfahrens zu gewährleisten, ist es erforderlich den Bildausschnitt vorzuverarbeiten, d.h. mögliche Beleuchtungseinflüsse müssen minimiert werden. Dabei orientiert sich das hier beschriebene Verfahren an der Arbeit von H. Rowley in [Ro99], in welcher ganze Gesichter mittels neuronaler Netze detektiert werden. Die Bildausschnitte werden dort mit einer Beleuchtungskorrektur und einer anschließenden Kontrasterhöhung mittels Histogrammausgleich vorverarbeitet. Im hier beschriebenen Verfahren wird zusätzlich ein Kantenfilter angewendet.

#### **3.2.1 Beleuchtungskorrektur**

In diesem Schritt der Vorverarbeitung wird versucht, lineare Helligkeitsverläufe zu eliminieren. Die Ergebnisse der Beleuchtungskorrektur in [Ro99] konnten nicht überzeugen, speziell der Einsatz eines KNN führte beispielsweise dazu, dass eindeutige Nicht-Gesichter in Gesichter umgewandelt wurden. Deshalb wurde ein Verfahren entwickelt, welches nach linearen Helligkeitsverläufen sucht, d.h. eine Korrektur nicht in jedem Fall durchführt.

Dazu wird ein mittig innerhalb des Bildausschnittes liegender kreisrunder Teil, folgend Korrekturfenster genannt, betrachtet (siehe Abbildung 33). Dieses Korrekturfenster wird in verschiedenen Richtungen, folgend Prozessrichtung genannt, verarbeitet. Dabei wird orthogonal zur Prozessrichtung Pixelreihe für Pixelreihe durchlaufen und aus jeder Reihe der Mittelwert der 3 kleinsten Grauwerte gebildet und abgespeichert. Somit erhält man entlang der Prozessrichtung ein Grauwertfeld der Durchschnittswerte der kleinsten Grauwerte. Um Spitzen zu dämpfen, werden diese Werte über die jeweiligen Vorgänger und Nachfolger geglättet.

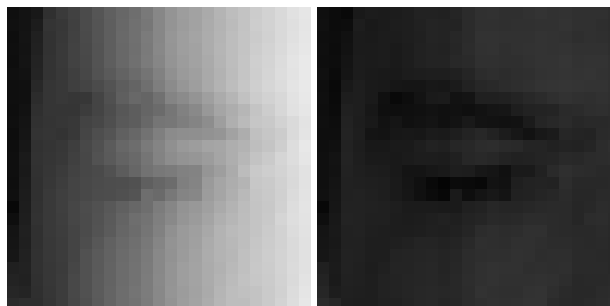
In Kapitel 5.2 werden die Ergebnisse der Untersuchungen zur Beleuchtungskorrektur vorgestellt.



*Abbildung 33: Korrekturfenster mit Prozessrichtung (grün) und Pixelreihe (rot).*

Dieser Vektor wird auf Linearität geprüft. Dazu wird aus dem Verhältnis vom ersten zum letzten Grauwert die theoretische Differenz zwischen zwei benachbarten Grauwerten berechnet. Bleibt die Differenz jedes im Vektor befindlichen benachbarten Grauwertpaars innerhalb dieser theoretischen Differenz (+/- einer festgelegten Toleranz), gilt der Verlauf als linear.

Übersteigt die höchste in allen Prozessrichtungen ermittelte Differenz vom ersten zum letzten Grauwert einen bestimmten Wert, wird ein Grauwertkeil, ein Vektor mit stufenweise erhöhten Grauwerten, berechnet und vom Bildausschnitt parallel zur Prozessrichtung abgezogen (siehe Abbildung 34). Dieser Grauwertkeil ist so lang, dass auch die außerhalb des Korrekturfensters befindlichen Randbereiche mit einbezogen werden.



*Abbildung 34: Beispielergebnis der Beleuchtungskorrektur (v.l.): Ausschnitt des Normgesichtes mit rechtem Auge und überlagertem Helligkeitsverlauf von links/dunkel nach rechts/hell, korrigierter Bildausschnitt*

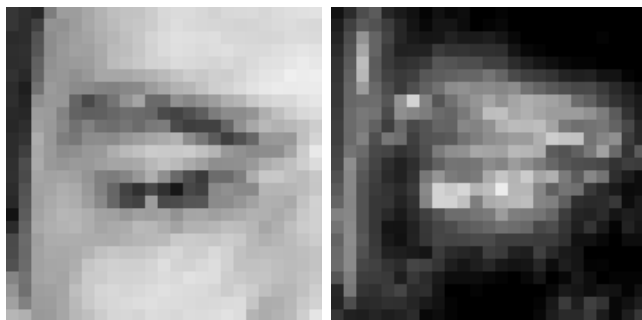
### 3.2.2 Kantenfilter

Für diese Arbeit wurde der in Kapitel 2.5.1 vorgestellte SUSAN-Kantenfilter verwendet. Grund dafür ist die geringe Größe der Gesichtsmerkmale. Bei der Binarisierung des Bildausschnittes würde einzelnen Pixeln bei unterschiedlichen Gesichtsbildern mal der Wert 255 (Kante) und mal der Wert 0 (Nichtkante) zugeordnet werden. Kanten müssten also weicher in homogene Bereiche auslaufen. Der SUSAN-Kantenfilter eignet sich aufgrund der großen Maske sehr gut dafür. In diesem Verfahren wurde er so modifiziert, dass die Reaktionswerte des Filters auf Werte zwischen 0 und 255 abgebildet werden. Das kommt auch den KNN entgegen, die gut mit kontinuierlichen Werten arbeiten können.

Das bedeutet, dass Pixel in homogenen Bereichen schwarz bleiben und umso heller werden, je weniger umliegende Pixel ähnlich dem Nukleus sind (siehe Abbildung 35).

Nach umfangreichen Tests (siehe Kapitel 5.3) wurde die Maskengröße mit 7, der Kontrast mit 27 und der Schwellwert mit 100%, also 37, festgelegt.

Als Vergleichsoperator wird Funktion b (siehe Kapitel 2.5.1) verwendet.



*Abbildung 35: Beispiel der Anwendung des modifizierten SUSAN-Kantenfilters.*

Das Bild wird somit nicht binarisiert, sondern jedes Pixel enthält die Information über die Anzahl ähnlicher umliegender Pixel. In Kapitel 5.3 werden die Testergebnisse zum SUSAN-Kantenfilter vorgestellt.

### **3.3 Modellierung der Gesichtsmerkmale**

Die Gesichtsmerkmale werden von künstlichen neuronalen Netzen modelliert. Dazu werden in diesem Verfahren 2-schichtige (nach [Ze94]) MLPs (Multi-Layer-Perzeptron, siehe Kapitel 2.4.2) angewendet, d.h. zwei Schichten gewichtete Verbindungen. Sie bestehen aus einer Schicht inaktiver Eingabeneuronen, einer Schicht Hiddenneuronen und einer Schicht Ausgabeneuronen. Für jedes Gesichtsmerkmal wird ein MLP eingesetzt. Die Größe der Eingabeschicht entspricht der jeweiligen Größe des Gesichtsmerkmals. Die Hiddenneuronenschicht wurde mit verschiedenen Größen getestet (siehe Kapitel 5.4). Außerdem wurden verschiedene Tests mit einem oder zwei Ausgabeneuronen durchgeführt (siehe Kapitel 5.4).

Als Lernverfahren wird der Backpropagation-Algorithmus verwendet.

Um ein KNN zu trainieren sind Trainingsmuster notwendig. Zusätzlich lässt sich der Trainingsfortschritt mit Validierungsmuster bewerten. Wie diese Muster generiert werden, wird im Folgenden beschrieben.

#### **Generierung der positiven Trainings- und Validierungsmuster**

Die positiven Trainings- und Validierungsmuster werden aus der in Kapitel 2.6 beschriebenen BioID-Gesichtsdatenbank generiert. Dazu wird das entsprechende Gesichtsmerkmal mit dem in Kapitel 3.2.3 beschriebenen Fenster extrahiert. Dafür benötigte Werte wie Zentrumpunkt, Skalierungsfaktor und Winkel werden aus den, für das jeweilige Gesichtsmerkmal, relevanten Punktwerten (siehe Kapitel 2.6) nach folgenden Formeln ermittelt:

Folgende Abkürzungen wurden für die Punkte verwandt:

RP: rechte Pupille, ARA: Außenecke des rechten Auges,

IRA: Innenecke des rechten Auges, LP: linke Pupille,

ALA: Außenecke des linken Auges, ILA: Innenecke des linken Auges,

RM: rechte Mundecke, LM: linke Mundecke,

MO: Mittelpunkt der Außenkante der Oberlippe,

MU: Mittelpunkt der Außenkante der Unterlippe,

NS: Nasenspitze, RNL: rechtes Nasenloch, LNL: linkes Nasenloch

- rechtes Auge:
  - Zentrumpunkt  $z$  :  $z = RP$
  - Skalierungsfaktor  $s$  :  $s = \frac{\|ARA - IRA\| * 1,75}{17}$
  - Winkel  $w$  :  $w = \arctan\left(\frac{IRA_x - ARA_x}{ARA_y - IRA_y}\right)$
- linkes Auge:
  - Zentrumpunkt  $z$  :  $z = LP$
  - Skalierungsfaktor  $s$  :  $s = \frac{\|ILA - ALA\| * 1,75}{17}$
  - Winkel  $w$  :  $w = \arctan\left(\frac{ALA_x - ILA_x}{ILA_y - ALA_y}\right)$
- Nase:
  - Zentrumpunkt  $z$  :  $z = NS$
  - Skalierungsfaktor  $s$  :  $s = \frac{\|LNL - RNL\| * 2,5}{20}$
  - Winkel  $w$  :  $w = \arctan\left(\frac{LNL_x - RNL_x}{RNL_y - LNL_y}\right)$
- Mund:
  - Zentrumpunkt  $z$  :  $z_x = MO_x$  ,  $z_y = MU_y - \frac{MU_y - MO_y}{2}$
  - Skalierungsfaktor  $s$  :  $s = \frac{\|LM - RM\| * 1,2}{25}$
  - Winkel  $w$  :  $w = \arctan\left(\frac{LM_x - RM_x}{RM_y - LM_y}\right)$

Bei der Berechnung des Skalierungsfaktors für ein Gesichtsmerkmal ergibt sich der konstante Multiplikator aus dem Verhältnis des Abstandes der entsprechend dem Gesichtsmerkmal zur Berechnung nötigen Punkte im Normgesicht und der Breite des Gesichtsmerkmals. Der konstante

Divisionswert entspricht der jeweiligen Breite des Gesichtsmerkmals.

Nachdem das Gesichtsmerkmal aus dem Bild extrahiert wurde, durchläuft es die in Kapitel 3.3 beschriebenen Vorverarbeitungsstufen. Die vom SUSAN-Kantenfilter erhaltenen Grauwerte wurden anfänglich auf einen Wertebereich von 0.0 bis 1.0 abgebildet und in der Trainings- und Validierungsmusterdatei abgespeichert. Später wurde der Eingabewertebereich auf -1.0 bis 1.0 erweitert.

Für die Generierung lässt sich festlegen, wie viel Prozent der Gesichtsbilder der Bilddatenbank verwendet werden sollen. Um eine ausgeglichene Auswahl über alle in der Bilddatenbank vertretenen Personen zu gewährleisten, wurden die zu jeder Person gehörenden Gesichtsbilder ermittelt (siehe Anhang B: Bilder je Person).

Von jeder Person wird der festgelegte Prozentsatz von allen für die Person verfügbaren Bildern für die Generierung verwendet, davon 10 Prozent für Validierungsmuster und der Rest für Trainingsmuster.

Stellt man sich alle Bilder einer Person in 10 Teile zu je 10% getrennt vor, so können die Validierungsmuster aus 10 verschiedenen Bereichen generiert werden. Über einen Parameter lässt sich vor Beginn der Generierung der Bereich festlegen.

### ***Generierung der negativen Trainings- und Validierungsmuster***

Ein weiterer Parameter legt vor der Generierung fest, welchem Vielfachen der Anzahl der positiven Trainings- und Validierungsmuster die Anzahl der negativen Trainings- und Validierungsmuster entspricht.

Für die Generierung der negativen Trainings- und Validierungsmuster werden ebenfalls die Gesichtsbilder der Bilddatenbank verwendet. Im ersten Ansatz wurde ein um 90° gedrehtes Fenster verwendet, um Bildausschnitte zu extrahieren. Das Fenster war gedreht, um nicht mit einem Positivmuster zu kollidieren. Der Zentrumpunkt wurde für jedes Muster zufällig ermittelt und der Skalierungsfaktor hatte den Wert 1.0. Es wurde nur ein Gesichtsbild der Bilddatenbank verwendet. Nach diversen Tests (siehe Kapitel 5.5) ließ sich feststellen, dass die negativen Trainings- und Validierungsmuster wesentlich zu einem guten Klassifikationsergebnis beitragen. In einem zweiten Ansatz wurde nach jedem 10-ten generierten Negativmuster ein neues Gesichtsbild zufällig aus der Bilddatenbank ausgewählt. Bei jedem

neuen Bild wurde der Skalierungsfaktor und der Winkel des Fensters mit je einem Zufallswert neu festgelegt. Der Zentrumpunkt wurde für jedes Negativmuster zufällig ermittelt. Ein mit Schwellwerten festlegbarer Bereich um den Zentrumpunkt des Fensters des Positivmusters wurde dabei ausgespart.

In einem dritten Ansatz wurden zu jedem Positivmuster eine bestimmte Anzahl Negativmuster, folgend Positivvarianten genannt, generiert. Der Zentrumpunkt des Fensters der Positivvarianten entspricht dem des Fensters des Positivmusters. Die Positivvarianten wurden mit mehreren Skalierungsfaktoren und Winkeln generiert. Diese wurden bis zu bestimmten Grenzwerten an den Skalierungsfaktor und den Winkel des Fensters des Positivmusters angenähert. Hintergrund dieser Erweiterung ist die Forderung, dass die Gesichtsmerkmale möglichst Positions-, Skalierungs- und Winkelgenau zu detektieren sind.

### **3.4 Suchstrategie**

Um eine Skalierungs- und Rotationsinvarianz des Verfahrens zu gewährleisten, wurde die in Kapitel 3.2.3 beschriebene Fenstertechnik entwickelt.

Das Bild wird durchsucht, indem man das Fenster mit dem größtmöglichen Skalierungsfaktor skaliert und dann stufenweise verkleinert. Je Skalierungsstufe bewegt sich das Fenster mit einem bestimmten Translationswert (berechnet aus einem Translationsparameter und dem Skalierungsfaktor) über das Bild und pro Translationsschritt wird das Fenster mit einer von einem Rotationsparameter bestimmten Anzahl Winkelschritten um  $360^\circ$  gedreht.

Der größtmögliche Skalierungsfaktor wird bestimmt durch die Forderung, dass das Gesicht vollständig im Bild enthalten sein muss.

Beim kleinsten Skalierungsfaktor von 1.0 entsprechen die Gesichtsmerkmalsgrößen den aus dem Normgesicht definierten Größen. Aus der Forderung nach einem vollständigen Gesicht im Bild ergibt sich außerdem, dass das Fenster den Rand des Bildes nicht überschneidet, denn die Gesichtsmerkmale liegen innerhalb des Gesichtes. D.h. der Zentrumpunkt liegt nie am Bildrand.

## 4 Implementierung

In diesem Kapitel wird die Umsetzung des im vorigen Kapitel beschriebenen Lösungsansatzes ausführlich beschrieben.

### 4.1 Umgebung

Entwickelt wurde das Verfahren in der Programmiersprache Java. Grund dafür ist die Verwendbarkeit des entstandenen Programms auf vielen unterschiedlichen Plattformen in Verbindung mit der Verfügbarkeit kostenfreier, professioneller Entwicklungsumgebungen. Ein Mangel ist die gegenüber anderen, nativen Programmiersprachen geringere Verarbeitungsgeschwindigkeit.

Die Implementierung des Verfahrens erfolgte Objektorientiert. Entwickelt wurde unter MS-Windows mit der Entwicklungsumgebung Eclipse [Ec05].

Für die Einbindung der trainierten KNN in das Programm, wurde zudem die C++-Entwicklungsumgebung DEV-C++ [Dev05] verwendet.

### 4.2 Umsetzung der einzelnen Teile

Eigenschaften, die in vielen Klassen benötigt werden und/oder anpassbar sein sollen, sind in den Klassen *StaticProperties*, *NeuralNetworkProperties*, *PathProperties* und *Training-Properties* gespeichert, wobei letztere nur für die Generierung und die Tests benötigt werden.

Die abgebildeten Klassendiagramme demonstrieren die Abhängigkeiten der jeweils beteiligten Klassen, ein Verweis aus dem Text ist nicht immer nötig.

#### 4.2.1 Fenstertechnik

Das Fenster wird realisiert durch die Klasse *Mask*. Eine Instanz dieser Klasse kann nur über eine der zwei statischen Methoden *buildCircleMask()* (für runde Fenster, beispielsweise für die Beleuchtungskorrektur) und *buildRectangleMask()* (für rechteckige Fenster, beispielsweise für die Extraktion der Bildausschnitte) erzeugt werden. Diese Methoden sorgen für die Erzeugung aller erforderlichen Attribute, setzen die Parameter und rufen die Methode *distributePoints()* auf, die die initiale Koordinatenberechnung



der Punkte in Abhängigkeit vom Skalierungsfaktor durchführt. Die Punkte sind in einem eindimensionalen Vektor gespeichert. Um die Position des Punktes im zweidimensionalen Fenster zu markieren, wird das zweidimensionale Feld *distributeTable* verwendet. Es enthält für jede Position im Fenster den Index des Punktes im Vektor.

Dieses Feld markiert auch den Unterschied zwischen rundem und rechteckigem Fenster, denn beide sind bei der Verarbeitung rechteckig, d.h. beide werden Zeile für Zeile, Spalte für Spalte durchlaufen. Das runde Fenster enthält aber in den Eckbereichen keine Punkte. Als Lösung wird in *distributeTable* an diesen Positionen ein Index von -1 gespeichert.

Das Fenster kann mit den Methoden *scaleMask()* und *turnMask()* skaliert und gedreht werden. Die Extraktion der Farb- oder Grauwerte erfolgt mit der Methode *getGreyValuesFromBufferedImage()*.

Aufgrund nicht vertretbar langer Verarbeitungszeiten wurden für die Rotationsberechnung Lookup-Tabellen für jeweils 360 Sinus- und Kosinuswerte generiert. Diese sind über statische Attribute der Klasse *TrigLUT* abrufbar.

Zur weiteren Beschleunigung der Drehung, die sich als Flaschenhals herausstellte, wurde *Mask* um das dreidimensionale Feld *turnLut* erweitert. Dieses enthält für eine festlegbare Anzahl von Drehschritten die Koordinaten der Punkte. Zur Extraktion der Pixelwerte musste eine weitere Methode *getGreyValuesFromBufferWithTurnLut()* entwickelt werden, der zusätzlich der gewünschte Drehschritt übergeben wird.

Um die Erhöhung der Anzahl der Drehschritte bei der Suche ohne eine erhebliche Erhöhung der Verarbeitungszeit zu ermöglichen, wurde die Klasse *MaskPartDimension* entwickelt. Diese enthält Größeninformationen für einen Teilbereich des Fensters. Mit der Methode *getFeature-PartGreyValues-FromBufferWithTurnLut()* können die Grauwerte eines Teilbereiches des Fensters extrahiert werden.

### 4.2.2 Bildvorverarbeitung

Die Schritte der Bildvorverarbeitung wurden in der Klasse *Filter* realisiert, welche 4 Methoden besitzt.

Die Methode *imagePreProcessing()* ruft alle Vorverarbeitungsmethoden in der Reihenfolge wie in Kapitel 3.3 angegeben auf. In *cornerDetector()* wurde der modifizierte SUSAN-Kantenfilter implementiert. Die Methode *histogramBalancer()* implementiert den Histogrammausgleich und die Methode *lightingCorrector()* die Beleuchtungskorrektur. Alle Methoden bekommen eine Referenz eines eindimensionalen Feldes mit Grauwerten übergeben und bearbeiten dieses. Nötige Informationen, wie beispielsweise die Breite und Höhe des Bildes, das von diesem Grauwertfeld repräsentiert wird, werden dem Konstruktor bei der Erzeugung der Klasse übergeben.

Alle Methoden sind auf schnelle Verarbeitung optimiert, denn sie haben den größten Einfluss auf die Verarbeitungszeit der Suche über das Bild.

#### **Die Beleuchtungskorrektur**

Das Korrekturfenster wird von einer Instanz der Klasse *Mask* repräsentiert. Die ermittelten Minimalgrauwerte werden im Feld *intensityBuffer* gespeichert. Wird in einer Prozessrichtung ein fast linearer Grauwertverlauf festgestellt und überschreitet die Differenz vom ersten zum letzten Grauwert einen bestimmten Wert, wird im Feld *correctIntensityBuffer* ein linearer Grauwertverlauf gespeichert. Der *correctIntensityBuffer* hat die Länge der Diagonale des durch das Grauwertbild repräsentierten Bildes, um alle Pixel bei beliebiger Prozess-richtung bearbeiten zu können (siehe Abbildung 36). Die Grauwerte vor *firstCorrectPartIndex* haben ein negatives Vorzeichen, d.h. bei der folgenden Differenzbildung werden zu den Pixelgrauwerten diese Grauwerte hinzuaddiert. Die Grauwerte ab *lastCorrectPartIndex* können unter Umständen Werte über 255 zugewiesen bekommen, diese werden bei der Differenzbildung abgeschnitten.

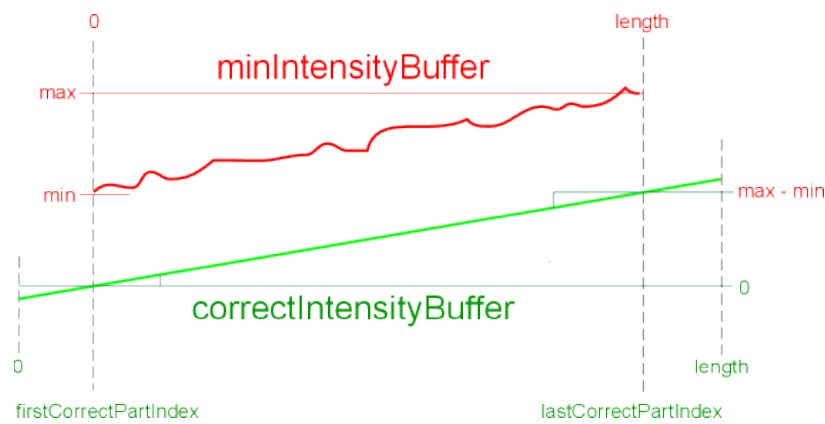


Abbildung 36: Prinzip der Erzeugung von `correctIntensityBuffer`.

Für die Differenzbildung wird das Grauwertfeld nach Zeilen und Spalten durchlaufen. Für jeden Grauwert wird der Abstand zu einer virtuellen Achse, die sich orthogonal zur Prozessrichtung am Ende des `correctIntensityBuffer` befindet, parallel zur Prozessrichtung berechnet (siehe Abbildung 37). Dieser Wert ergibt den Index in `correctIntensityBuffer`. Da die Prozessrichtungen bekannt sind, lässt sich für jede Pixelposition und jede Prozessrichtung eine Lookup-Tabelle dieser Abstände generieren, welche `distToPLut` bezeichnet wird.

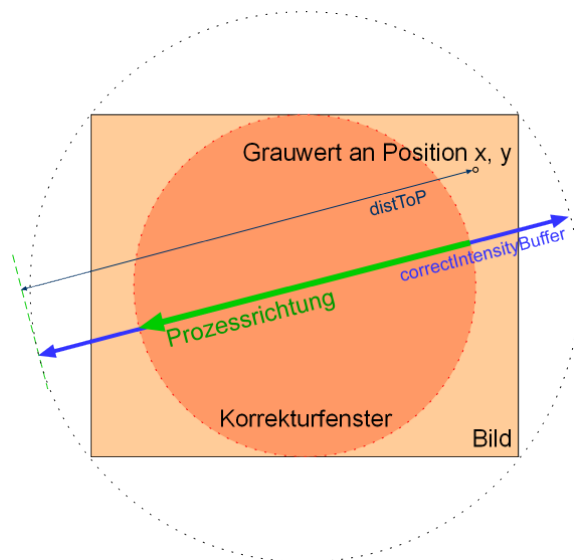


Abbildung 37: Schema der Verhältnisse der Größe des Korrekturfensters zur Bildgröße und der Länge des `correctIntensityBuffer` zur Bildgröße. Zusätzliche Darstellung der Ermittlung des Index in `correctIntensityBuffer` zur Korrektur eines Pixelgrauwertes

### ***Der SUSAN-Kantenfilter***

Die USAN des SUSAN-Kantenfilters wird ebenfalls von einer Instanz der Klasse *Mask* repräsentiert. Für den SUSAN-Kantenfilter wird für die 256 x 256 möglichen Differenzwerte, die sich aus der Formel b (siehe Kapitel 2.5.1) ergeben, eine dreidimensionale Lookup-Tabelle *corrDiffLut* erzeugt.

Eine weitere Lookup-Tabelle, die als *usanMapper* bezeichnet ist, enthält für alle im Bild befindlichen Pixel den Index der umliegenden zu USAN gehörenden Pixel im Grauwertfeld. Zusätzlich wird in der Lookup-Tabelle *usanBorderChecker* gespeichert, ob sich der jeweilige Punkt der USAN außerhalb des Bildes befindet, in diesem Fall erhält *usanBorderChecker* eine 1, sonst eine 0.

### **4.2.3 KNN – Training und Anwendung**

Für die Erzeugung und das Training der KNN, wurde der SNNS [SNNS05], der „Stuttgart Neural Network Simulator“, verwendet. Dieser wurde nicht für die Windowsplattform entwickelt, kann aber mit einem X-Window-Server unter Windows betrieben werden.

Die Einbindung der erzeugten KNN in das Programm erfolgt mit dem Java-Framework JNI (Java-Native-Interface). Dazu ist es nötig, die KNN zuerst in C-Dateien und dann in vom JNI verwendbare Funktionsbibliotheken im DLL-Format umzuwandeln. Zur Umwandlung eines KNN in eine C-Datei liefert der SNNS ein Kommandozeilenprogramm mit und für die Erzeugung einer DLL wurde die freie C/C++ Entwicklungsumgebung DEV-C++ verwendet. Zum Einbinden der DLL-Datei wurde die Klasse *NativeNetwork* entwickelt, deren Methode *getNetResponse()* die in der DLL-Datei enthaltene Funktion aufruft.

Um die Trainings- und Validierungsmuster aus der BioID-Gesichtsdatenbank zu generieren, wurden die Klassen *BioIDImageProcessor* und *BioID-PointProcessor* entwickelt. Diese stellen alle benötigten Methoden für den Zugriff auf die Gesichtsbilder und die entsprechenden Punktwerte zur Verfügung. Die in Anhang B: Bilder je Person beschriebene Aufzählung der Gesichtsbilder je Person ist in einer Datei gespeichert. Diese wird von der Methode *getImagesPerPerson()* der Klasse *PersonImageRelation* ausgelesen und die in einem Vektor gespeicherten Bildnummern je Person zurückgegeben. Um unterschiedliche Musterdatei-Formate generieren zu

können, wurde die Schnittstelle *PatternGenerator* entwickelt. Diese wird von der Klasse *PatternGeneratorImpl* realisiert, von der alle Klassen, die eine Musterdatei generieren sollen, abgeleitet werden müssen, wie im Fall von SNNS-Musterdateien mit der Klasse *SNNSPatterGenerator*. Die Erzeugung eines Mustergenerators ist nur mit der Methode *getFormat()* der Klasse *PatternGeneratorFactory* möglich.

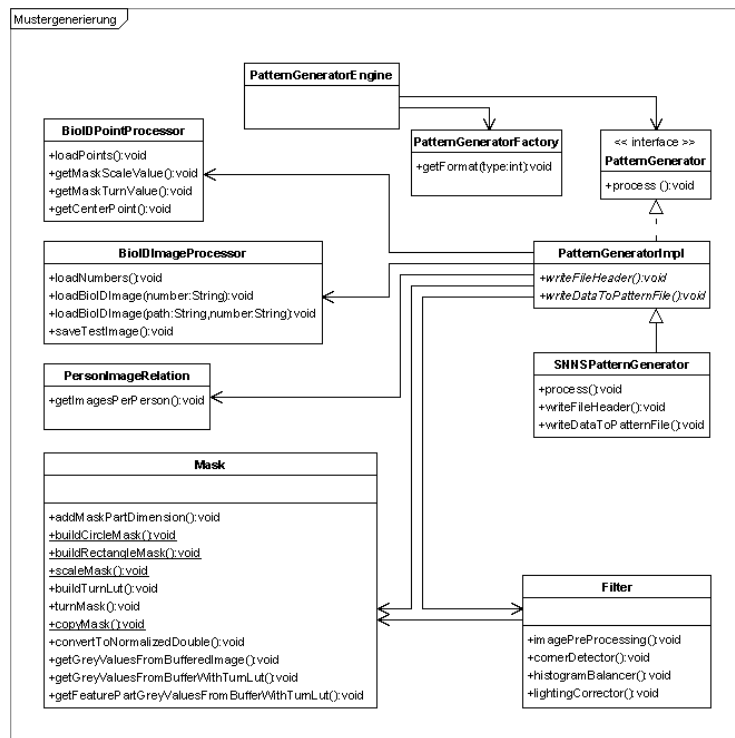


Abbildung 38: Architektur der Mustergenerierung

#### 4.2.4 Suche über das Bild

Für die Suche werden außer den Klassen für das Fenster und die Bildvorverarbeitung die Klassen *NativeNetwork* für die Verbindung zum KNN und die Klasse *ImageIO* zum Laden des zu durchsuchenden Bildes benötigt (siehe Abbildung 39).

In einer ersten Implementierung wurden die extrahierten Bildausschnitte für jeden Fensterwinkel vorverarbeitet und an das KNN zur Klassifikation weitergegeben. Um jedoch die Verarbeitungsgeschwindigkeit zu erhöhen,

wurde das Fenster auf den maximal möglichen Diagonalwert eines Gesichtsmerkmals, der Mund mit 25 x 20 Pixel, vergrößert. Damit können beliebig rotierte Teildbildausschnitte aus diesem Fenster entnommen werden. Der Bildausschnitt des Fensters wird also je Skalierungs- und Translationsschritt einmal extrahiert und vorverarbeitet.

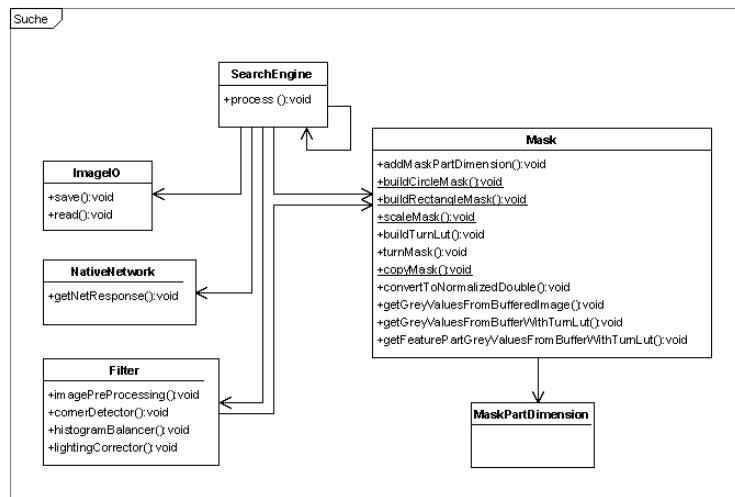


Abbildung 39: Klassendiagramm der Suche über das Bild.

### 4.3 Zusätzliche Entwicklungen

#### 4.3.1 Analyse der Suche

Um die Reaktionswerte der KNN während der Suche über ein Bild zu analysieren, wurden die Klassen *TestImage*, *TestImageSaver*, *SearchData-Logger* und *WindowDrawer* entwickelt. Die Klasse *TestImage* enthält die Positionswerte der Gesichtsmerkmale aller Gesichter ausgesuchter Testbilder. Die Klasse *TestImageSaver* stellt nur die Methode *saveImage()* zur Verfügung, mit der der extrahierte Bildausschnitt gespeichert werden kann. Die Klasse *SearchDataLogger* enthält sogenannte *analyzeAreas*, Analysebereiche um das Zentrum des Gesichtsmerkmals, in einstellbarer Anzahl. Bei jedem Suchschritt wird der Reaktionswert des KNN abhängig von Gesicht, falls sich mehrere Gesichter im Bild befinden, Skalierungsfaktor, Drehwinkel des Fensters und Analysebereich gespeichert.

Dazu werden der Methode *saveOutputData()* die entsprechenden Werte übergeben. Nach der Suche wird die Methode *writeOutputData()* aufgerufen, um die gespeicherten Daten sortiert in eine Logdatei zu schreiben. Mit Hilfe dieser Datei kann das Suchergebnis analysiert werden (siehe Kapitel 5.5).

### **4.3.2 Visualisierung der Ausgabewerte der KNN über eine Musterdatei oder der Ausgabewerte einer SNNS-Result-Datei**

Weil während des Trainings der KNN merkwürdige Gleichmäßigkeiten in den Ergebnisse auftraten (siehe Kapitel 5.4), wurde ein Programm zur Visualisierung der Ausgabewerte der KNN entwickelt. In diesem Programm können beliebige KNN auf beliebige Musterdateien angewendet werden. Bedingung ist die gleiche Anzahl von Eingabe- und Ausgabeneuronen. Alternativ können die gespeicherten Ausgabewerte aus einer SNNS-Result-Datei extrahiert werden. Die Ausgabewerte werden in einem Graphen visualisiert.

Die Benutzerschnittstelle wurde nach dem MVC-Muster implementiert. So erzeugt die Klasse *NetworkAnalyzer* die grafische Oberfläche und die Klasse *NetworkAnalyzerData* realisiert den Zugriff auf die KNN, die Musterdateien und die SNNS-Result-Dateien. Benutzereingaben werden von der Klasse *NetworkAnalyzerController* verarbeitet, einzig die Verarbeitung der Mauseingaben im Bereich des Graphen wurde der Einfachheit wegen in der entsprechenden Komponente implementiert. Für die Anwendung der KNN auf die in der Musterdatei enthaltenen Muster wurde die Klasse *AnalyzeNetwork* entwickelt, sie wird von der Methode *getAnalyzedPattern()* der Klasse *NetworkAnalyzerData* verwendet. Die Methode *getAnalyzedResultfile()* der Klasse *NetworkAnalyzerData* verwendet die Klasse *AnalyzeSNNSResultFile* zum extrahieren der Ausgabewerte aus der SNNS-Result-Datei. Die ermittelten Ausgabewerte werden in einer Instanz der Klasse *OutputInformations* gespeichert.

## 5 Experimentelle Untersuchungen

In diesem Kapitel werden die Untersuchungen der einzelnen Teile des Verfahrens diskutiert. Alle Untersuchungen wurden ausschließlich am rechten Auge durchgeführt. So sind alle KNN mit dem rechten Auge trainiert worden und dementsprechend erfolgte die Suche über das Bild auch nach dem rechten Auge.

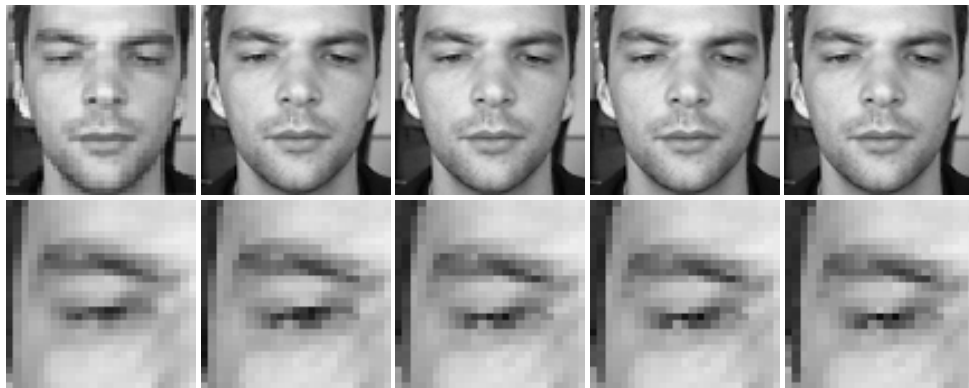
Es wurde nur dieses eine Gesichtsmerkmal verwandt, weil es einerseits die stärkste Charakteristik und andererseits keine große Variabilität aufweist. Das linke Auge ist vergleichbar mit dem rechten Auge. Die Charakteristik der Nase ist wesentlich geringer und die Variabilität des Mundes deutlich größer. Würde sich das Verfahren für das rechte Auge als nicht erfolgreich herausstellen, so ist anzunehmen, dass es auch bei der Nase und dem Mund nicht erfolgreich ist. Auch das biologische Vorbild des Menschen, der zur Identifizierung eines anderen Gesichtes zuerst nach dem rechten Auge sieht, war ein Grund dieser Entscheidung.

### 5.1 Test der Fenstertechnik

Für die Tests wurde das Normgesicht mit fünf verschiedenen Skalierungsfaktoren skaliert (1; 2,83; 5,83; 8,33; 28,33) und der Bildausschnitt, mit der rechten Pupille als Zentrumpunkt, mit einem entsprechend skalierten 25 x 25 Pixel Fenster extrahiert.

In einem ersten Test wurde geprüft, ob bei skaliertem Gesichtsbild und skaliertem Fenster Unterschiede in den extrahierten Bildausschnitten auftreten (siehe Abbildung 40).



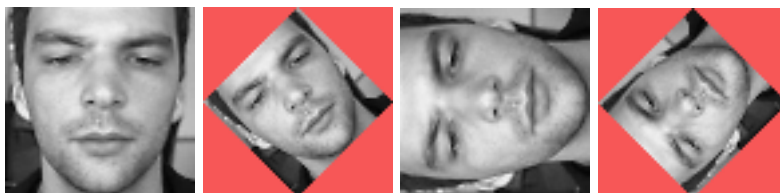


*Abbildung 40: Ergebnisse des ersten Tests. Oben das Normgesicht, darunter das zugehörige extrahierte Gesichtsmerkmal (v.l.): 60 x 60 Pixel großes Normgesicht / Skalierungsfaktor 1,0, 170 x 170 Pixel großes Normgesicht/ Skalierungsfaktor 2,83, 350 x 350 Pixel großes Normgesicht/ Skalierungsfaktor 5,83, 500 x 500 Pixel großes Normgesicht/ Skalierungsfaktor 8,33, 1700 x 1700 Pixel großes Normgesicht / Skalierungsfaktor 28,33.*

Die Ergebnisse machen deutlich, dass die Skalierung des Fensters keinen Einfluss auf das extrahierte Gesichtsmerkmal hat.

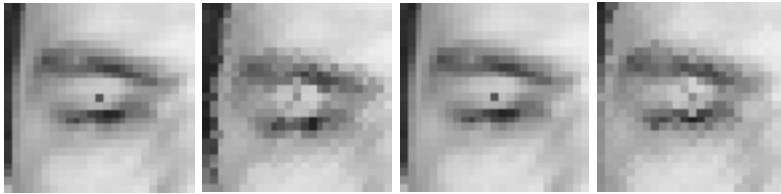
Im zweiten Test wurde geprüft, ob sich bei skaliertem und gedrehtem Gesichtsbild bzw. Fenster Unterschiede in den extrahierten Bildausschnitten ergeben.

Dazu wurden die fünf unterschiedlich großen Normgesichter um jeweils vier Winkel gedreht ( $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ ) und abgespeichert (am Beispiel des 500 x 500 Pixel großen Normgesichtes in Abbildung 41 dargestellt).



*Abbildung 41: Darstellung des gedrehten Normgesichtes (am Beispiel des 500 x 500 Pixel großen Normgesichtes) (v.l. :  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ ). Die Abbildungen wurden hier zur besseren Übersicht auf eine Größe skaliert, die Seitenlänge der um  $45^\circ$  bzw.  $135^\circ$  gedrehten Normgesichter entspricht der Diagonale der um  $0^\circ$  bzw.  $90^\circ$  gedrehten Normgesichter.*

Nachfolgend die Ergebnisse der Extraktion der Bildausschnitte entsprechend den Winkeln der Normgesichter (v.l.) 0°, 45°, 90° und 135°:



*Abbildung 42: Aus dem 60 x 60 Pixel großen Normgesicht.*



*Abbildung 43: Aus dem 170 x 170 Pixel großen Normgesicht.*



*Abbildung 44: Aus dem 350 x 350 Pixel großen Normgesicht.*



*Abbildung 45: Aus dem 500 x 500 Pixel großen Normgesicht.*



*Abbildung 46: Aus dem 1700 x 1700 Pixel großen Normgesicht.*

Die Ergebnisse zeigen, dass sich die extrahierten Bildausschnitte trotz Skalierung und Rotation des Fensters nicht wesentlich unterscheiden. Einzig die extrahierten Bildausschnitte des gedrehten 60 x 60 Pixel Normgesichtes weisen eine leichte Artefaktbildung auf.

Die Ergebnisse der Tests der Fenstertechnik führen zu dem Schluss, dass die Skalierung und Rotation wie erwartet funktioniert und eine Betrachtung der umliegenden Pixel eines zu einem Fensterpunkt korrespondierenden Pixels nicht notwendig ist.

### **5.2 Test der Beleuchtungskorrektur**

Um den Algorithmus der Beleuchtungskorrektur zu testen, wurde das 170 x 170 Pixel große Normgesicht im Bereich des rechten Auges mit einem Schwarz-Weiß-Verlauf überlagert. Dieser Verlauf wurde in acht Richtungen über das Normgesicht gelegt und das jeweilige Normgesicht abgespeichert. Die Verläufe wurden in Richtung des weißen Verlaufsendes bezeichnet mit Nord, Nordost, Ost, Südost, Süd, Südwest, West und Nordwest. Zusätzlich wurde das Normgesicht einmal abgedunkelt und einmal aufgehellt abgespeichert.

In den kommenden Abbildungen sind die unterschiedlichen Beleuchtungssituationen nach folgender Matrix sortiert:

$$\begin{bmatrix} \textit{dunkel} , & \textit{hell} , & \textit{Nord} , & \textit{Nordost} , & \textit{Ost} \\ \textit{Südost} , & \textit{Süd} , & \textit{Südwest} , & \textit{West} , & \textit{Nordwest} \end{bmatrix}$$



*Abbildung 47: Der extrahierte Bildausschnitt vor der Beleuchtungskorrektur.*

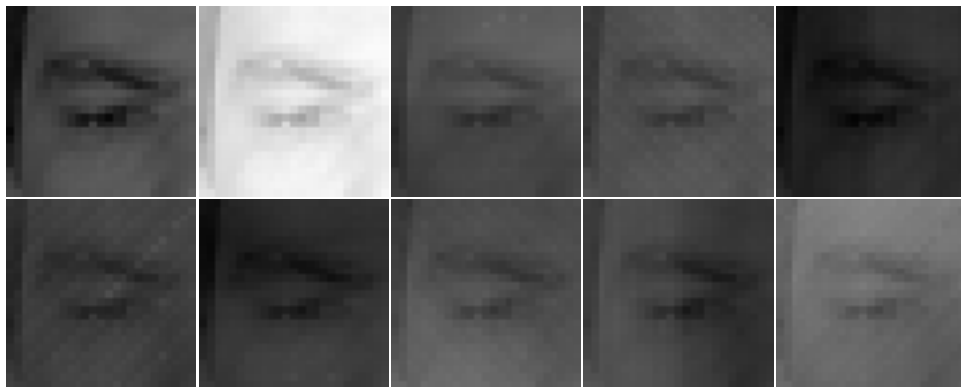


Abbildung 48: Der extrahierte Bildausschnitt nach der Beleuchtungskorrektur.



Abbildung 49: Der extrahierte Bildausschnitt nach der Beleuchtungskorrektur und dem Histogrammausgleich.

Die Ergebnisse der Beleuchtungskorrektur weisen große Unterschiede auf. So sind die Ergebnisse in den horizontalen und vertikalen Verlaufsrichtungen bis auf einen orthogonal zur Verlaufsrichtung liegenden Helligkeitssprung recht gut. Die Ergebnisse der diagonalen Verlaufsrichtungen zeigen jedoch eine starke lineare Strukturbildung orthogonal zur Verlaufsrichtung. Ein Grund könnte die Nichtlinearität der Pixelreihen bei gedrehtem Fenster sein. Das die Beleuchtungskorrektur trotzdem eingesetzt werden kann, zeigen die folgenden Ergebnisse des SUSAN-Kantenfilters.

### 5.3 Test des SUSAN-Kantenfilters

Ziel der Tests des SUSAN-Kantenfilters war es, eine brauchbare Kombination von Parametern zu finden. Mit Parametern sind die Größe der USAN, der Kontrast und der Schwellwert gemeint. Die Brauchbarkeit ergibt sich durch die Unempfindlichkeit gegenüber den Ergebnissen der Beleuchtungskorrektur bei möglichst kleiner USAN. Denn die Verarbeitungszeit wächst mit größer werdender USAN exponentiell.

Basis für die Tests waren die in Abbildung 49 dargestellten Ergebnisse der Beleuchtungskorrektur.

Es wurden zahlreiche Tests gemacht mit USAN-Größen von 5, 7, 9 und 11 und Kontrasten von 10, 15 und 27. In Abbildung 50 sind die Testergebnisse der Kombination von Parametern, die für das Verfahren gewählt wurden, abgebildet. Sie sind in derselben Weise sortiert wie die Abbildungen in Kapitel 5.2.

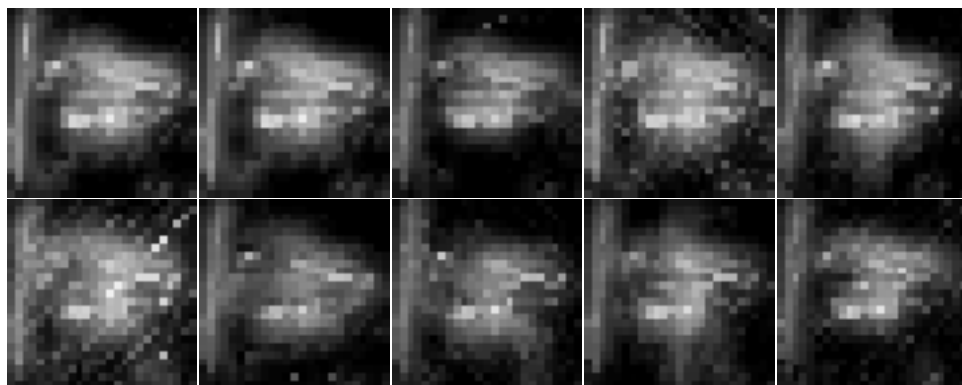


Abbildung 50: USAN-Größe 7, Kontrast 27 und Schwellwert 100%

Die Ergebnisse lassen sich folgendermaßen interpretieren. Je größer die USAN ist, um so unempfindlicher wird das Filterergebnis, denn der Einfluss einzelner Punkte verringert sich. Mit kleiner werdendem Kontrast wird der USAN-Wert kleiner, es werden mehr Kanten gefunden. Um möglichst nur signifikante Kanten zu filtern, muss der Kontrast also hoch sein. Ist er allerdings zu hoch, wird der Filter empfindlich gegenüber leichten Helligkeitsveränderungen. Den besten Kompromiss ergab die Kombination einer USAN-Größe von 7 und eines Kontrastes von 27.

#### **5.4 Training der künstlichen neuronalen Netze**

Die Beurteilung des Trainings der KNN erfolgte einerseits durch die Interpretation der Lernkurve und andererseits durch die Anwendung des KNN bei der Suche über ein Bild. Somit stehen die hier beschriebenen Ergebnisse in engem Zusammenhang mit den Ergebnissen im folgenden Kapitel 5.5.

Es wurden mehrere KNN für die Klassifikation des rechten Auges trainiert. Dabei wurde einerseits mit diversen Topologien und andererseits mit verschiedenen Musterdateien experimentiert. Die Experimente mit unterschiedlichen Topologien spielten nur eine untergeordnete Rolle, so wurde beispielsweise mit unterschiedlich großen Hiddenneuronenschichten trainiert. Eine größere Bedeutung hatten die verschiedenen Musterdateien, bei denen die Anzahl der Muster und das Verhältnis von Positiv- zu Negativmustern variierten. Die schrittweise Verbesserung der Generierung der Negativmuster hatte den größten Einfluss auf die besser werdenden Klassifikationsergebnisse der KNN.

Eine Tabelle mit den Einzelheiten zu allen KNN befindet sich in Anhang C: Einzelheiten zum Training der KNN.

Die eingestellte Zyklenanzahl betrug meist eintausend. War jedoch zu erkennen, dass sich die Lernkurven auf konstante Werte einpegelten, wurde das Training meist abgebrochen. Das Training eines KNN dauerte zwischen 4 und 12 Stunden, abhängig von der Topologie und der Anzahl der Muster. Die Interpretation der Lernkurve war schwierig. Grundsätzlich sollte sie absteigend sein und die Lernkurve der Validierungsmuster sollte, weil wesentlich weniger Muster für die Validierung verwandt werden, unterhalb der Lernkurve der Trainingsmuster liegen. Steigt die Lernkurve der Validierungsmuster wieder an, beginnt das Netz mit dem Übertraining. Ist die Hiddenneuronenschicht zu klein, sinkt die Lernkurve zu langsam und kann anfangen zu oszillieren. Ist sie dagegen zu groß, sinkt die Lernkurve zu schnell und die Lernkurve der Validierungsmuster wird relativ früh wieder ansteigen, ein sicheres Zeichen für Übertraining. Leider sind diese Eigenschaften keine Gesetzmäßigkeiten, sondern eher Anhaltspunkte für die Interpretation.

Durch eine zu geringe Zahl an Mustern waren die ersten Netze schnell übertrainiert. Durch eine Erhöhung der Musteranzahl trat diese Problematik bei nachfolgenden Netzen nicht mehr auf.

Im allgemeinen zeigten die Lernkurven ein mehr oder weniger stark oszillierendes Einschwingen zu Beginn, bevor sie sich auf relativ konstante Werte einpegelten (siehe Abbildung 51).

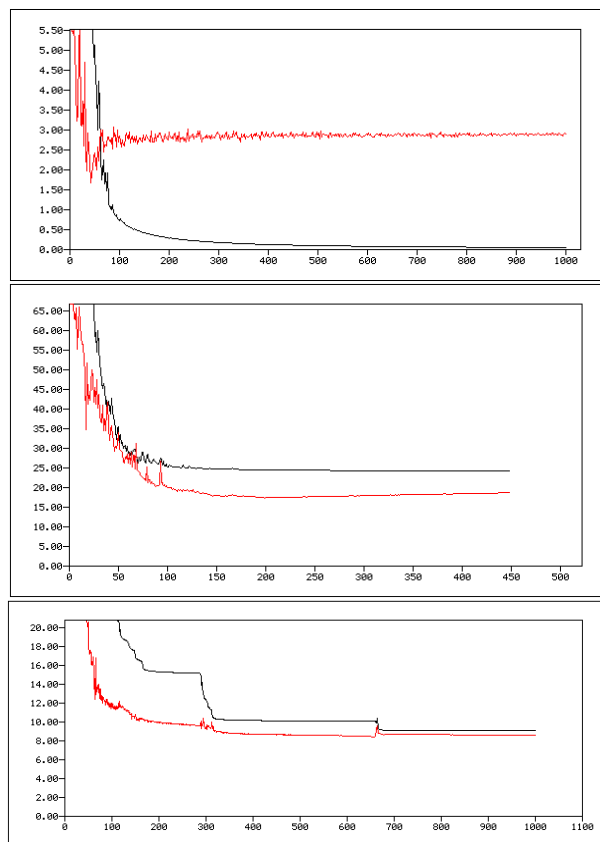


Abbildung 51: Beispiele für den Verlauf der Lernkurven (v.o.: Netz 24, Netz 38, Netz 41).

Zu Beginn wurde ein über einige Lernkurven gleichbleibender summierter Fehler zwischen 2 und 3 bemerkt. Da dieses Phänomen trotz unterschiedlicher Musteranzahl auftrat, wurde vermutet, dass die Ursache dafür immer dieselben Muster sein könnten. Diese Vermutung führte zur Entwicklung des in Kapitel 4.4.2 beschriebenen Programms, wurde jedoch nicht bestätigt.

Ab Netz 32 wurde mit erheblich verbesserten Negativmustern (aus mehreren Bildern generiert, siehe Kapitel 3.4) und ab Netz 41 mit noch weiter verbesserten Negativmustern trainiert (Positivvarianten und größerer Eingabewertebereich, siehe Kapitel 3.4).

Die Netze 38 und 42 wurden mit zwei Ausgabeneuronen trainiert. Das eine Ausgabeneuron sollte auf Gesichter und das andere Ausgabeneuron auf Nichtgesichter mit 1.0 reagieren.

Gemessen an den Lernkurven sollten die Netze 32 bis 42, ausgenommen die Netze 34 bis 36, die besten Klassifikationsergebnisse bei der Suche über ein Bild erwarten lassen.

Doch sind die Lernkurven kein sicherer Hinweis auf die Fähigkeiten der KNN, erst die Anwendung bei der Suche über ein Bild lässt eine Eignung der KNN beurteilen.

### **5.5 Ergebnisse der Suchtests**

Die Suchtests wurden über Gesichtsbilder zweier verschiedener Testsets durchgeführt. Das erste Testset (siehe Abbildung 52) besteht aus Gesichtsbildern der BioID-Gesichtsdatenbank bzw. Teilen der Gesichtsbilder und wurde zur Beurteilung der allgemeinen Klassifikationsfähigkeit der KNN genutzt.

Die Gesichtsbilder wurden nach folgenden Kriterien ausgewählt:

- unterschiedliche Beleuchtungssituationen
- Variationen des rechten Auges

Das zweite Testset (siehe Abbildung 53) besteht aus Gesichtsbildern der CMU-MIT-Gesichtsdatenbank und privaten Gesichtsbildern und wurde zur Beurteilung der Generalisierungsfähigkeit der KNN genutzt.

Die Gesichtsbilder wurden nach folgenden Kriterien ausgewählt:

- verschiedene Auflösungen
- mehrere Gesichter im Bild
- unterschiedliche Beleuchtungssituationen





Abbildung 52: Testset 1: Die Gesichtsbilder für die Tests der allgemeinen Klassifikationsfähigkeit der KNN : (oben v.l.) 170.jpg, 170\_b.jpg, BioID\_0000.jpg, BioID\_0113.jpg, (unten v.l.) BioID\_0488.jpg, BioID\_0643.jpg, BioID\_0181.jpg.



Abbildung 53: Testset 2: Die Gesichtsbilder für die Tests der Generalisierungsfähigkeit der KNN: (oben v.l.) hendrix2.jpg, newsradio1.jpg, realTest\_2.jpg, (unten v.l.) cnn1260.jpg, married.jpg (bis auf realTest\_2.jpg alle Gesichtsbilder aus der CMU-MIT-Gesichtsdatenbank [CMU05])

Zur Beurteilung der Klassifikations- bzw. Generalisierungsfähigkeit der KNN wurde ein Schwellwert festgelegt, ab dem der Reaktionswert der KNN ein rechtes Auge bedeutete. Das Fenster am entsprechenden Zentrumpunkt wurde als Rechteck in eine Kopie des Bildes eingezeichnet und diese Kopie nach Beendigung der Suche abgespeichert.

Die Qualität der KNN wurde durch die Verteilung der eingezeichneten Rechtecke geschätzt. Erwartet wurde eine hohe Anzahl Rechtecke um das rechte Auge und eine möglichst geringe Anzahl im Rest des Bildes.

Bis Suchtest 47 war eine Anzahl von Drehschritten von 8 festgelegt, was einer Rotationsschrittweite von  $45^\circ$  entspricht. Durch die Optimierung der Suche (siehe Kapitel 4.2.4) wurde eine Erhöhung der Anzahl der Drehschritte ohne wesentliche Erhöhung der Verarbeitungszeit möglich. Für Suchtest 48 wurde die Anzahl der Drehschritte auf 16 ( $22,5^\circ$ ) und für alle nachfolgenden Suchtests auf 32 ( $11,25^\circ$ ) festgelegt.

### **Die einfachen Suchtests**

Mit einfachen Suchtests sind die Tests gemeint, die nur die Reaktionswerte der KNN über dem Schwellwert speichern und keine umfassende Analyse ermöglichen.

Eine Übersicht über alle einfachen Suchtests befindet sich in Anhang D: Einzelheiten zu den einfachen Suchtests.

Die Ergebnisse der Tests über Bilder von Testset 1 zeigen eine sehr gute Klassifikationsfähigkeit der KNN (siehe Abbildung 54), so wurden bei über einer Million Suchschritten maximal 1350 Bildausschnitte als rechte Augen klassifiziert. Das gesuchte rechte Auge wurde meist mehrfach klassifiziert.



*Abbildung 54: Ergebnisse der Tests zur Klassifikationsfähigkeit (Suchtestnr. v.l.: 29, 47, 51).*

Die Ergebnisse der Tests über Testset 2 zeigen dagegen eine schwache bis gute Generalisierungsfähigkeit der KNN (siehe Abbildung 55).

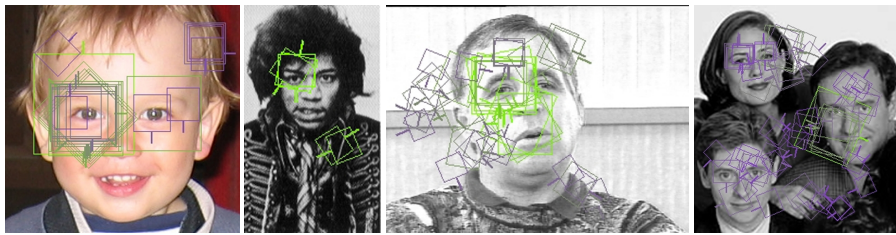


Abbildung 55: Ergebnisse der Tests zur Generalisierungsfähigkeit (Suchtestnr. v.l.: 31, 54, 56).

Die hohe Zahl falsch klassifizierter Bildausschnitte ist gerade in Hinblick auf den hohen Schwellwert von über 0,9 unbefriedigend.

Um diese Zahl zu verringern, wurden die in Kapitel 3.4 beschriebenen Verbesserungen zur Generierung der Negativmuster vorgenommen. Die verbesserten Negativmuster führten dazu, dass sich die Klassifikationsschwelle senkte. Deshalb haben die Suchtests, die mit Netzen ab Nummer 32 vorgenommen wurden, scheinbar schlechtere Klassifikationsergebnisse.

### **Die analysierbaren Suchtests**

Um die Zahl der klassifizierten Bildausschnitte in verschiedenen Schwellwertbereichen zu ermitteln, wurden die in Kapitel 4.4.1 beschriebenen Erweiterungen in der Implementierung der Suche vorgenommen. Die Reaktionswerte der Netze wurden in verschiedenen Bereichen um das rechte Auge gespeichert. Damit ist eine Aussage über die Positionsgenauigkeit der klassifizierten Bildausschnitte möglich.

Es wurden zuerst die Netze 5, 7 bis 10 und 14 bis 42 auf die Gesichtsbilder 170.jpg, BioID\_0000.jpg und BioID\_0643.jpg angewendet.

Nach der Analyse wurden die Netze 32 und 37 bis 42 als besonders geeignete Netze ausgewählt. Diese wurden nun auf die Gesichtsbilder BioID\_0488.jpg, cnn1260.jpg, hendrix2.jpg, married.jpg, newsradio1.jpg und realTest\_2.jpg angewendet.

Die Ergebnisse der Analyse dieser Tests sind in Tabelle 1 aufgeführt.

Gesichtsbild / Netz	32	37	38	39	40	41	42
BioID_0488.jpg	0,14	0,00	0,56	0,12	0,00	0,23	0,40
cnn1260.jpg	0,71	0,53	0,16	0,68	2,01	1,67	1,19
hendrix2.jpg	0,57	0,46	0,20	0,35	0,71	0,40	0,30
married.jpg Gesicht 1	0,25	0,00	0,26	0,00	0,00	0,00	0,45
married.jpg Gesicht 2	0,11	0,00	0,32	0,15	0,00	0,00	0,00
newsradio1.jpg Gesicht 1	0,16	0,05	0,07	0,76	1,45	0,22	0,34
newsradio1.jpg Gesicht 2	0,19	0,00	0,15	0,35	0,07	0,00	0,00
newsradio1.jpg Gesicht 3	0,10	0,64	0,09	0,54	0,41	0,23	0,17
Mittelwert	0,28	0,21	0,23	0,37	0,58	0,34	0,36
Standardabweichung	0,23	0,28	0,16	0,27	0,77	0,56	0,38
Mittelwert / Standardabweichung	1,21	0,75	1,43	1,35	0,76	0,62	0,94
realTest_2.jpg	0,51	15,23	2,44	8,74	6,80	3,95	0,98

Tabelle 1: Ergebnisse der Analyse der Positionsgenauigkeit (für nähere Erläuterungen, siehe Text).

Es wurden alle Reaktionswerte über 0,5 betrachtet. Die Werte in der Tabelle entsprechen dem Verhältnis der Wahrscheinlichkeiten des Auftretens eines Reaktionswertes in einem Bereich um den Zentrumpunkt des rechten Auges (Pupille, Radius 11 Pixel) und im Rest des Bildes, bezogen auf die Gesamtanzahl an Reaktionswerten im jeweiligen Bereich. Mit diesen Werten lässt sich die Qualität der KNN bezüglich falsch klassifizierter Bildausschnitte bewerten. Je höher dieser Wert ist, umso geringer ist die Wahrscheinlichkeit des Auftretens eines Reaktionswertes im Rest des Bildes. Nullwerte bedeuten, dass die KNN keine Bildausschnitte im Bereich um den Zentrumpunkt des rechten Auges klassifizieren konnten. Im oberen Teil der Tabelle (einschließlich Zeile Mittelwert) sind zur besseren Übersicht Nullwerte rot und Werte über 0,5 grün gekennzeichnet.

Die Ergebnisse der Analyse über Bild „realTest\_2.jpg“ wurden aufgrund von extremen Spitzen nicht in die Berechnung einbezogen.

Mit Hilfe des Verhältnisses von Mittelwert und Standardabweichung wurden die drei besten Netze ausgewählt (Werte über 1,0, grün gekennzeichnet).

Diese wurden anschließend auf Skalierungs- und Winkelgenauigkeit untersucht (siehe Tabelle 3).

Es wurden nur Reaktionswerte im Bereich um den Zentrumpunkt des rechten Auges betrachtet. Für die Analyse der Skalierungs- und Winkelgenauigkeit wurden die Reaktionswerte in eine Tabelle eingetragen (siehe Tabelle 2). In jeder Zelle steht die Anzahl der Reaktionswerte bei entsprechendem Skalierungsfaktor bzw. Drehwinkel.

Die korrekten Skalierungsfaktoren und Winkel wurden zuvor ermittelt. Der Wert der Zelle, die diesen korrekten Werten am nächsten war (siehe Tabelle 2, dunkelgrün), und die Werte der unmittelbar darum liegenden Zellen (siehe Tabelle 2, grün) wurden für die Berechnung der Wahrscheinlichkeit der Winkel- und Skalierungsgenauigkeit (siehe Tabelle 3 , R + S) verwandt. Die Werte der Zellen, die nur dem korrekten Winkel am nächsten waren (siehe Tabelle 2, dunkelgelb, horizontal) und die der nächstliegenden Winkel (siehe Tabelle 2, gelb, horizontal), wurden in die Berechnung der Wahrscheinlichkeit der Winkelgenauigkeit (siehe Tabelle 3 , R) einbezogen. Analog dazu wurden Zellen die nur dem korrekten Skalierungsfaktor am nächsten waren und die der nächstliegenden Skalierungsfaktoren in die Berechnung der Skalierungsgenauigkeit (siehe Tabelle 3 , S) einbezogen.

Winkel / Skalierungs-faktor	3,80	3,60	3,40	3,20	3,00	2,80	2,60	2,40	2,20	2,00	1,80	1,60	1,40
0													1
11							1	1	1				
22							1					1	1
33								1					1
45									1	1			1
56							2						1
67													2
78							3	2					2
90					1	2							1
101							2		1				1
112					1		1	1					1
123						1		1					1
135													
146						1							2
157					2								3
168					2	1	1		1		1	4	1
180													
191	1			1	3	1		1					
202		1	1	2	1	2	1						1
213				1	1	2	1			1			
225				1	1								1
236							1						2
247													1
258								1					2
270								1					3
281					1								1
292					1			1	1		1	1	2
303					1								1
315						1	1						1
326							2				2	1	2
337								2	1		2	1	
348													4

Tabelle 2: Verteilung der Reaktionswerte der KNN über 0,5, am Beispiel der Anwendung des Netzes 32 auf das Bild „cnn1260.jpg“.

Netz	32			38			39		
	R	S	R + S	R	S	R + S	R	S	R + S
Gesichtsbild									
BioID_0488.jpg	0,050	0,30	0,00	0,00	0,54	0,00	0,00	1,00	0,00
cnn1260.jpg	0,054	0,031	0,023	0,00	0,00	0,00	0,029	0,14	0,029
hendrix2.jpg	0,015	0,87	0,015	0,033	0,97	0,033	0,087	0,61	0,087
RealTest_2.jpg	0,14	0,00	0,00	0,11	0,00	0,00	0,037	0,00	0,00
married.jpg Gesicht 1	0,020	0,80	0,00	0,097	0,22	0,022	0,00	0,00	0,00
married.jpg Gesicht 2	0,15	0,39	0,10	0,007	0,42	0,00	0,00	0,00	0,00
newsradio1.jpg Gesicht 1	0,098	0,044	0,024	0,00	0,41	0,00	0,21	0,82	0,18
newsradio1.jpg Gesicht 2	0,028	0,086	0,028	0,039	0,55	0,00	0,00	0,40	0,00
newsradio1.jpg Gesicht 3	0,10	0,17	0,033	0,00	0,29	0,00	0,00	0,39	0,00
Mittelwert	0,07	0,30	0,03	0,03	0,38	0,01	0,04	0,37	0,03
Standardabweichung	0,05	0,33	0,03	0,04	0,30	0,01	0,07	0,37	0,06
Mittelwert / Standardabweichung	1,40	0,90	0,79	0,73	1,25	0,49	0,57	1,00	0,53

Tabelle 3: Ergebnisse der Analyse der Skalierungs- und Winkelgenauigkeit (R: winkelgenau, S: skalierungsgenau, R + S: winkel- und skalierungsgenau).

Die Werte in Tabelle 3 entsprechen den Wahrscheinlichkeiten des Vorkommens eines Reaktionswertes im entsprechenden Bereich (R, S und R + S), bezogen auf alle Reaktionswerte im Bereich um den Zentrumpunkt des rechten Auges. Zur besseren Übersicht wurden Nullwerte rot und Werte über 0,1 grün gekennzeichnet.

Der Mittelwert und die Standardabweichung sind hier kaum aussagekräftig, von größerer Bedeutung sind die Nullwerte. Gut zu erkennen ist, dass keines der Netze 100%ig skalierungs- und winkelgenau ist. Das Netz 38 hat eine sehr gute Skalierungsgenauigkeit, wogegen Netz 32 eine konstante Winkelgenauigkeit aufweist. Das Netz 39 kann trotz einiger sehr guter Werte insgesamt nicht überzeugen. Interessant ist, dass keines der Netze einen skalierungs- und winkelgenauen Treffer in den Bilder BioID\_0488.jpg und realTest\_2.jpg erzielen konnte.

In der Summe aller Tests zeigt Netz 32 die besten Ergebnisse und wird als Grundlage für die nachfolgende Zusammenfassung verwendet.

## 6 Zusammenfassung und Ausblick

Die Leistungsfähigkeit des Verfahrens wurde am Beispiel des rechten Auges getestet. Dieses kann in beliebig rotierten und skalierten Gesichtsbildern detektiert werden. Es darf sich außerdem an beliebigen Positionen befinden. Die Ergebnisse der Tests lassen darauf schließen, dass, nach entsprechendem Training der KNN, die Gesichtsmerkmale Nase und Mund ebenso sicher detektiert werden können.

Mit Hilfe der im Verfahren angewandten Bildvorverarbeitung konnte die geforderte Robustheit gegenüber unterschiedlichen Beleuchtungssituationen erreicht werden.

Die künstlichen neuronalen Netze erfüllen die von ihnen geforderte Modellierung der Gesichtsmerkmale ausgezeichnet und sorgen für eine gute Generalisierungsfähigkeit des Verfahrens.

Die in der Formulierung des Lösungsansatzes geforderte Skalierungs- und Winkelgenauigkeit konnte nicht 100%ig erreicht werden, somit ist eine nachfolgende Kombination der Gesichtsmerkmale zu einem Gesicht angesichts der hohen Zahl falsch klassifizierter Bildausschnitte schwierig, wenn nicht sogar unmöglich.

Mit einer Verarbeitungszeit von ca. einer Stunde für ein 384 x 286 Pixel-Bild befindet sich das Verfahren im experimentellen Stadium und eignet sich nicht für einen produktiven Einsatz.

Zusammenfassend kann festgestellt werden, dass das entwickelte Verfahren die einleitend geforderten Ziele erreicht hat. Gesichtsmerkmale können positions-, skalierungs- und rotationsinvariant detektiert werden. Das Verfahren ist robust gegenüber unterschiedlichen Beleuchtungssituationen und weist eine gute Generalisierungsfähigkeit auf.

Nachfolgende Forschungen müssen vor allem die Skalierungs- und Winkelgenauigkeit erhöhen, denn nur damit ist eine spätere Kombination der Gesichtsmerkmale zur Detektion eines vollständigen Gesichtes möglich.

Mit Hilfe statistischer Verfahren könnte die in diesem Verfahren vernachlässigte Verarbeitungszeit verringert werden. So ist vorstellbar, das Bild rekursiv in Cluster zu unterteilen, diese zu durchsuchen und die Suche gegebenenfalls vorher abubrechen.

---

## Anhang A: Entfernte Bilder

Die Nummerierung entspricht der Nummerierung in der BioID-Gesichtsdatenbank. Bei zusammenhängenden Bildbereichen werden das erste und letzte Bild mit einem Bindestrich verbunden. Nichtzusammenhängende Bilder bzw. Bildbereiche sind mit einem Semikolon getrennt.

Gesamtanzahl der Gesichtsbilder: 1521

Bilder, die entfernt wurden, weil die Person eine Brille trug:

10; 90-105; 108-112; 115-145; 147-169; 322-337; 341-342; 344-357; 361; 363-364; 366; 368-372; 377-392; 396; 398-410; 494-563; 567-589; 594-595; 599-602; 652-653; 741-778; 810-860; 973-1014; 1174-1211; 1213; 1319-1336; 1341-1342; 1346-1358; 1441; 1451-1472; 1486-1492; 1494

Rest der Gesichtsbilder: 1038

Bilder, die entfernt wurden, weil die Person einen Bart trug:

16-17; 338-367; 374; 393-417; 590-593; 596-598; 634-642; 1212; 1293-1318

Rest der Gesichtsbilder: 972

Somit können noch 972 Gesichtsbilder für die Generierung der Trainings- und Validierungsmuster eingesetzt werden.

Sechs Personen sind auf Grund der Aussortierung nicht mehr in den zur Generierung einsetzbaren Gesichtsbildern vertreten (in Klammern ist die Gesamtanzahl an Gesichtsbildern der Person angegeben):

Person 1 (2) 16-17




Person 2 (90) 115-145; 519-555; 1451-1472






Person 3 (150) 322-372; 374;  
377-417; 586-  
602; 1174-1213




Person 4 (35) 634-642; 1293-  
1318



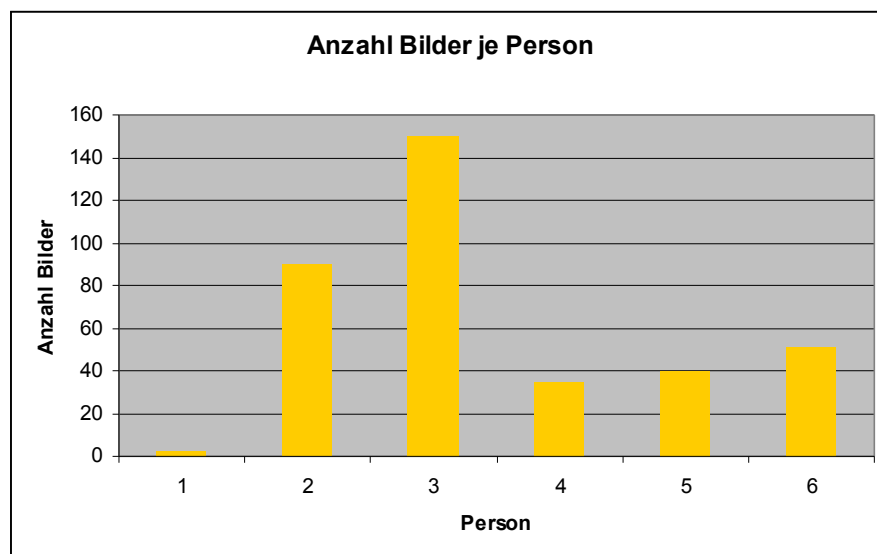
Person 5 (40) 652-653; 741-778



Person 6 (51) 810-860



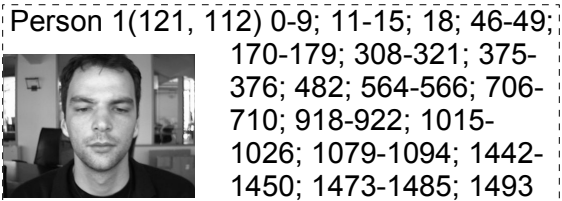
Von den 23 Personen, die in der BiID-Gesichtsdatenbank vertreten sind, bleiben somit noch 17 Personen, deren Gesichtsbilder für die Generierung der Trainings- und Validierungsmuster verwendet werden können.



---

## Anhang B: Bilder je Person

In Klammern ist die Gesamtanzahl (vor, nach der Aussortierung) der Gesichtsbildern der Person angegeben.



Person 2(59,59) 19-24; 25-45;  
373; 942-972



Person 3(78, 78) 50-75; 418-448; 1153-  
1173



Person 4(88, 88) 76-89; 711-740;  
861-873; 1122-  
1152



Person 5(60, 4) 106-107; 113-114



Person 6(61, 61) 180-221; 923-  
941



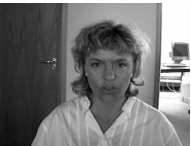
Person 7(71, 71) 222-240; 1027-1078



Person 8(106, 106) 241-259; 449-  
481; 1387-1440



Person 9(99, 99) 260-307; 649-651; 874-  
892; 1214-1242



Person 10(94, 69) 483-493; 779-  
809; 1095-1121



Person 11(51, 51) 603-633; 1495-1514



Person 12(40, 40) 643-648; 1359-  
1386; 1515-1520



Person 13(6, 6) 654-659



Person 14(46, 46) 660-705



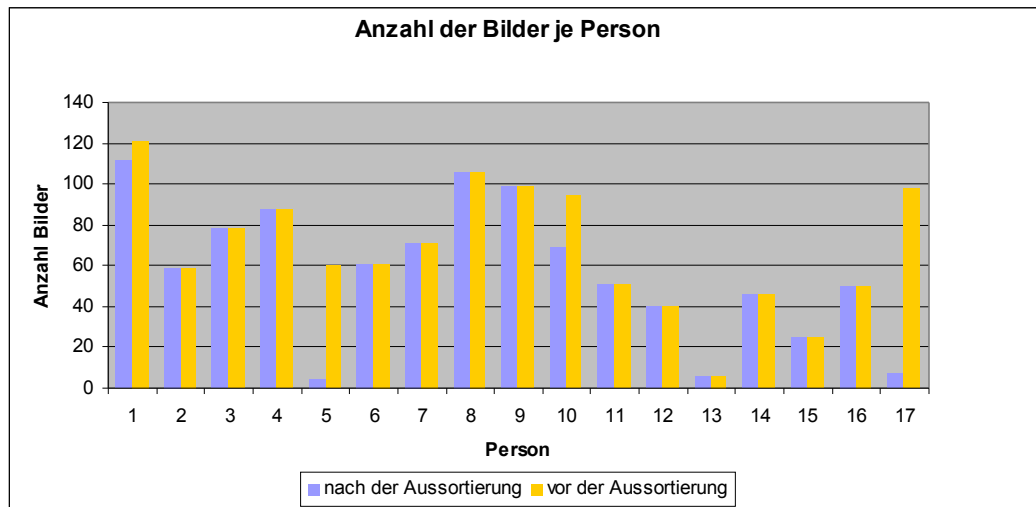
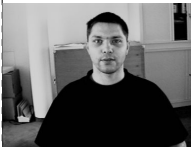
Person 15(25, 25) 893-917



Person 16(50, 50) 1243-1292



Person 17(98, 7) 1337-1340; 1343-1345



## Anhang C: Einzelheiten zum Training der KNN

Da nur KNN für das rechte Auge trainiert wurden, beträgt die Größe der Eingabeneuronenschicht immer 255.

KNN-Nr.	Anzahl der		Musterdatei			Trainingsergebnis	
	Hidden-neuronen	Ausgabe-neuronen	Nr.	Anzahl Positivmuster Training / Validierung	Anzahl Negativmuster Training / Validierung	Anzahl Zyklen	MSE der Validierungsmuster
1	10x10	1	1	60 / 7	600 / 70	1000	
2	10x10	1	1	60 / 7	600 / 70	ca. 780	
3	10x10	1	1	60 / 7	600 / 70	1000	
4	10x10	1	1	60 / 7	600 / 70	ca. 5500	
5	10x10	1	1	60 / 7	600 / 70	385	0,00000
6	10x10	1	3	220 / 26	1760 / 208	516	0,11111
7	10x10	1	3	220 / 26	1760 / 208	504	0,00853
8	10x10	1	4	441 / 49	3528 / 392	407	0,00458
9	10x10	1	5	441 / 49	3528 / 392	950	0,00348
10	5x5	1	6	175 / 19	17500 / 1900	1000	0,00287
11	10x10	1	1	220 / 26	1760 / 208	897	0,09091
12	10x10	1	6	175 / 19	17500 / 1900	300	0,00990
13	10x10	1	6	175 / 19	17500 / 1900	661	0,00990
14	12x12	1	6	175 / 19	17500 / 1900	326	0,00161
15	7x7	1	3	220 / 26	1760 / 208	619	0,01071
16	12x12	1	3	220 / 26	1760 / 208	767	0,01181
17	7x7	1	3	220 / 26	1760 / 208	1000	0,01054
18	7x7	1	3	220 / 26	1760 / 208	1000	0,01444
19	5x5	1	3	220 / 26	1760 / 208	1000	0,00876
20	12x12	1	3	220 / 26	1760 / 208	319	0,00896
21	3x5x5	1	3	220 / 26	1760 / 208	1000	0,01367
22	6x5x5	1	3	220 / 26	1760 / 208	789	0,01399
23	3x3	1	3	220 / 26	1760 / 208	1000	0,01187
24	3x3	1	4	441 / 49	3528 / 392	1000	0,00655
25	5x5	1	4	441 / 49	3528 / 392	1000	0,00333
26	5x5; 2x3x3; 3x3	1	4	441 / 49	3528 / 392	1000	0,00590
27	7x7	1	4	441 / 49	3528 / 392	737	0,00407
28	10x10	1	4	441 / 49	3528 / 392	1000	0,00563
29	7x7	1	8	132 / 16	3300 / 400	1000	0,00548
30	7x7	1	9	220 / 26	11000 / 1300	1000	0,00202
31	7x7	1	10	349 / 38	27920 / 3040	1328	0,00098
32	7x7	1	12	220 / 26	11000 / 1300	1000	0,00200
33	5x5	1	12	220 / 26	11000 / 1300	1000	0,00153
34	5x5	1	13	220 / 26	11000 / 1300	336	0,00311
35	7x7	1	13	220 / 26	11000 / 1300	302	0,00232
36	3x3	1	13	220 / 26	11000 / 1300	520	0,00362
37	7x7	1	16	220 / 26	220000 / 26000	956	0,00019
38	7x7	2	17	349 / 38	349000 / 38000	447	0,00048
39	7x7	1	18	220 / 26	110000 / 13000	426	0,00046
40	7x7	1	19	349 / 38	174500 / 19000	246	0,00032
41	7x7	1	20	220 / 26	220000 / 26000	1000	0,00033
42	2x5x5	2	21	262 / 29	262000 / 29000	1000	0,00083

Tabelle 4: Einzelheiten der KNN

## Anhang D: Einzelheiten zu den einfachen Suchtests.

Ein Suchschritt entspricht einem zu klassifizierendem Bildausschnitt. Suchtests vor Nr. 24 waren nur Probetests.

Suchtest-Nr.	Gesichtsbildname	Schwellwert	Verarbeitungszeit in s	Anzahl Suchschritte	KNN-Nr.	Anzahl als rechtes Auge klassifizierte Bildausschnitte	Anzahl rechte Augen / wieviel erkannt
24	BiolD_0000.jpg	0,999	4855,375	2.103.904	5	154	1 / 1 (mehrfach)
25	170_b.jpg	0,999	1890,016	984.960	10	924	1 / 1 (mehrfach)
29	BiolD_0000.jpg	0,999	4198,734	2.103.904	7	64	1 / 1 (mehrfach)
30	BiolD_0000.jpg	0,999		2.103.904	8		1 / 1 (mehrfach)
31	realTest_2.jpg	0,999	1954,328	1.321.656	7	28	1 / 1 (mehrfach)
32	realTest_3.jpg	0,999	18639,359	13.785.328	7	118	2 / 1
33	170_b.jpg	0,999	1426,875	984.960	7	24	1 / 1 (mehrfach)
34	170_b.jpg	0,999	1409,657	984.960	8	91	1 / 1 (mehrfach)
35	realTest_2.jpg	0,999	1884,109	1.321.656	8	44	1 / 1 (mehrfach)
36	BiolD_0000.jpg	0,999	3015,453	2.103.904	9	126	1 / 1 (mehrfach)
37	170_b.jpg	0,999	1427,094	984.960	9	63	1 / 1 (mehrfach)
38	realTest_2.jpg	0,999	1986,250	1.321.656	9	52	1 / 1 (mehrfach)
39	BiolD_0000.jpg	0,999	3380,938	2.103.904	14	538	1 / 1 (mehrfach)
40	BiolD_0000.jpg	0,999	2034,953	2.103.904	26	1.350	1 / 1 (mehrfach)
41	realTest_2.jpg	0,999	1449,437	1.321.656	26	662	1 / 1 (mehrfach)
42	BiolD_0000.jpg	0,999	2263,922	2.103.904	25	338	1 / 1 (mehrfach)
43	BiolD_0000.jpg	0,999	3153,203	2.103.904	28	375	1 / 1 (mehrfach)
44	BiolD_0000.jpg	0,999	2372,750	2.103.904	19	201	1 / 1 (mehrfach)
45	BiolD_0000.jpg	0,999	3462,312	2.103.904	20	113	1 / 1 (mehrfach)
46	BiolD_0000.jpg	0,999	2842,766	2.103.904	18	200	1 / 1 (mehrfach)
47	BiolD_0000.jpg	0,999	1950,391	2.030.776	18	46	1 / 1 (mehrfach)
48	BiolD_0000.jpg	0,999	2371,844	4.061.552	18	95	1 / 1 (mehrfach)
49	BiolD_0000.jpg	0,999	3129,203	8.123.104	18	203	1 / 1 (mehrfach)
50	BiolD_0000.jpg	0,999	3070,515	8.123.104	29	32	1 / 1 (mehrfach)
51	BiolD_0000.jpg	0,999	2846,016	8.123.104	30	71	1 / 1 (mehrfach)
52	BiolD_0000.jpg	0,999	3318,641	8.123.104	31	83	1 / 1 (mehrfach)
53	realTest_3.jpg	0,999	18517,969	54.080.736	31	382	2 / 1
54	hendrix2.jpg	0,999	254,922	680.032	31	5	1 / 1 (mehrfach)
55	aerosmith-double.jpg	0,999	2669,875	7.764.384	31	91	5 / 0
56	cnn1260.jpg	0,999	1676,296	4.861.696	31	37	1 / 1 (mehrfach)
57	newsradio1.jpg	0,999	2562,063	7.041.472	31	61	3 / 2 (mehrfach)
59	BiolD_0000.jpg	0,500	2911,125	8.123.104	31	14.232	?
60	BiolD_0000.jpg	0,990	3082,938	8.123.104	33	135	1 / 0
72	170.jpg	0,999	449,250	1.058.560	38	1	1 / 0
73	170.jpg	0,999	440,953	1.058.560	41	1	1 / 0
74	newsradio1.jpg	0,999	3224,468	7.041.472	38	35	3 / 0
75	newsradio1.jpg	0,999	3426,063	7.041.472	41	5	3 / 0
76	newsradio1.jpg	0,990	3286,766	7.041.472	41	48	3 / 0
78	aerosmith-double.jpg	0,990	3538,640	7.251.296	41	57	5 / 0
79	newsradio1.jpg	0,990	5211,047	11.527.232	41	84	3 / 0

Tabelle 5: Einzelheiten zu den einfachen Suchtests.

---

## Tabellenverzeichnis

Tabelle 1: Ergebnisse der Analyse der Positionsgenauigkeit (für nähere Erläuterungen, siehe Text).....	64
Tabelle 2: Verteilung der Reaktionswerte der KNN über 0,5, am Beispiel der Anwendung des Netzes 32 auf das Bild „cnn1260.jpg“ .....	65
Tabelle 3: Ergebnisse der Analyse der Skalierungs- und Winkelgenauigkeit (R: winkelgenau, S: skalierungsgenau, R + S: winkel- und skalierungsgenau).....	66
Tabelle 4: Einzelheiten der KNN.....	72
Tabelle 5: Einzelheiten zu den einfachen Suchtests.....	73

---

## Abbildungsverzeichnis

Abbildung 1 : Beispiele für die Variabilität in der Abbildung von Gesichtern (aus der CMU-MIT-Gesichtsdatenbank [CMU05]).....	3
Abbildung 2: Gesichtskandidat nach Yang [YH94] (ausgehend von der Annahme, dass der Bereich von Augen, Nase und Mund dunkler als umliegende Gesichtsteile ist).....	6
Abbildung 3: Die Histogramme der Spalten- (links) und Zeilengrauwertsummen (rechts) zur Bestimmung eines Gesichtskandidaten (aus [KP97]).....	7
Abbildung 4: Detektierung von Gesichtsmarkmalen (aus [YC96]).....	8
Abbildung 5: Schrittweise Gruppierung der Merkmale, beginnend bei der rechten Augenbraue (aus [YC96]).....	8
Abbildung 6: Darstellung der allgemeinen Farbverteilung von 18.696 Bildern aus dem Internet (links entlang der Grün-Magenta-Achse, rechts im RGB-Farbraum) (aus [JR99]).....	9
Abbildung 7: Darstellung der Farbverteilung von Hautpixeln über 4675 Bilder (aus [JR99]).....	10
Abbildung 8: Darstellung der Farbverteilung von Nicht-Hautpixeln über 8965 Bilder (aus [JR99]).....	10
Abbildung 9: Spektren der Handrückenfarbe, links verschiedene ethnische Gruppen (Kaukasiern (rot), Asiaten(grün) und Inder(blau)), rechts normale Haut (rot) im Vergleich zu geschminkter Haut(blau) (aus [AMD01]).....	11
Abbildung 10: Die markanten Punkte (links) und Beispiele für ermittelte Gesichtsformen(rechts) (aus [LTC95]).....	12
Abbildung 11: Beispielhafte Darstellung einer 2-dimensionalen Eingabemenge (blaue Kreise) im Eingaberaum $\mathfrak{R}^2$ und die zwei zugehörigen Eigenvektoren (PC1 und PC2) (aus [GB05]).....	13
Abbildung 12: Das Durchschnittsgesichtsbild (links) und die ersten 7 Eigenvektoren (rechts)(aus [TP91b]).....	14
Abbildung 13: Links der schematische Aufbau einer biologischen Nervenzelle, rechts das abgeleitete Modell der künstlichen Neuronen (aus [Ze94]).....	15
Abbildung 14: Schema eines KNN mit Eingabeneuronen (unten), Hiddenneuronen (mitte), Ausgabeneuron (oben) und einem vorwärtsgerichteten Verbindungsnetzwerk. Die Neuronen sind vollständig ebenenweise verbunden.....	16
Abbildung 15: Beispiele für verschiedene Verbindungsnetzwerke und die entsprechenden Verbindungsmatrizen (aus [Ze94]).....	17
Abbildung 16: Darstellung einer selbstorganisierenden Karte in Form eines quadratischen Gitters (aus [Ze94]).....	18
Abbildung 17: Struktur des Perzeptrons (aus [Ze94]).....	19
Abbildung 18: Verlauf der idealen Trainingskurve.....	19
Abbildung 19: Funktionsweise des Verfahrens von [Ro99] (aus [Ro99]).....	20
Abbildung 20: Aufbau eines Hidden-Markov-Modells (aus [Mi00]).....	21
Abbildung 21: Darstellung der 5 relevanten Bereiche / Zustände (aus [Sa93]).....	21
Abbildung 22: Beispiel für die Anwendung des Sobel-Operators (Nullwerte sind als mittleres Grau dargestellt) (v.l.): A, Gx, Gy, G (aus [Wi05]).....	23
Abbildung 23: Beispiel für die Anwendung des Laplace-Operators (aus [Ha03]).....	24

---

Abbildung 24: Funktionsweise des SUSAN-Kantenfilters (aus [SB95]).....	25
Abbildung 25: Funktionsdarstellung des Vergleichsoperators. Die X-Achse stellt die Grauwertdifferenz und die Y-Achse das Vergleichsergebnis dar. (mit einem Beispielkontrast $t=27$ ) (aus [SB95]).....	25
Abbildung 26: Die Reaktionswerte des SUSAN-Filters auf ein Beispielbild (aus [SB95]).....	26
Abbildung 27: Beispiele von Gesichtsbildern der BioID-Gesichtsdatenbank (aus [Hu05]).....	31
Abbildung 28: Die Punkte aus Datei bioid_0000.pts in Bild BioID_0000.jpg (aus [Hu05]).....	32
Abbildung 29: Entwicklung des Normgesichtes (v.l.): die Regionen des Gesichtes nach Priorität nummeriert (aus [HFMDM00]), Definiertes Normgesicht unter Berücksichtigung der wichtigsten Regionen, auf 60 x 60 Pixel verkleinertes Normgesicht.....	33
Abbildung 30: Darstellung der Größen der Gesichtsmerkmale im Normgesicht (v.l.): rechtes Auge, linkes Auge, Mund, Nase.....	34
Abbildung 31: Rechts: Darstellung des Beispielfensters im Normgesicht mit Zentrumpunktkoordinaten 79,110. Die den Fensterpunkten zugeordneten Bildpixel sind blau gefärbt. Links: Der extrahierte Bildausschnitt.....	36
Abbildung 32: Darstellung der Skalierung und Rotation des Beispielfensters im Normgesicht mit Zentrumpunktkoordinaten 79,110. Die den Fensterpunkten zugeordneten Bildpixel sind blau gefärbt (v.l.): Skalierungsfaktor 1.0 und Winkel $0^\circ$ , Skalierungsfaktor 2.7 und Winkel $0^\circ$ , Skalierungsfaktor 2.7 und Winkel $345^\circ = -15^\circ$ . ....	36
Abbildung 33: Korrekturfenster mit Prozessrichtung(grün) und Pixelreihe(rot).....	38
Abbildung 34: Beispielergebnis der Beleuchtungskorrektur (v.l.): Ausschnitt des Normgesichtes mit rechtem Auge und überlagertem Helligkeitsverlauf von links/dunkel nach rechts/hell, korrigierter Bildausschnitt.....	38
Abbildung 35: Beispiel der Anwendung des modifizierten SUSAN-Kantenfilters.....	39
Abbildung 36: Prinzip der Erzeugung von correctIntensityBuffer.....	47
Abbildung 37: Schema der Verhältnisse der Größe des Korrekturfensters zur Bildgröße und der Länge des correctIntensityBuffer zur Bildgröße. Zusätzliche Darstellung der Ermittlung des Index in correctIntensityBuffer zur Korrektur eines Pixelgrauwertes.....	47
Abbildung 38: Architektur der Mustergenerierung.....	49
Abbildung 39: Klassen diagramm der Suche über das Bild.....	50
Abbildung 40: Ergebnisse des ersten Tests. Oben das Normgesicht, darunter das zugehörige extrahierte Gesichtsmerkmal (v.l.): 60 x 60 Pixel großes Normgesicht / Skalierungsfaktor 1,0, 170 x 170 Pixel großes Normgesicht/ Skalierungsfaktor 2,83, 350 x 350 Pixel großes Normgesicht / Skalierungsfaktor 5,83, 500 x 500 Pixel großes Normgesicht/ Skalierungsfaktor 8,33, 1700 x 1700 Pixel großes Normgesicht / Skalierungsfaktor 28,33.....	53

---



---

Abbildung 41: Darstellung des gedrehten Normgesichtes (am Beispiel des 500 x 500 Pixel großen Normgesichtes) (v.l. : 0°, 45°, 90°, 135°). Die Abbildungen wurden hier zur besseren Übersicht auf eine Größe skaliert, die Seitenlänge der um 45° bzw. 135° gedrehten Normgesichter entspricht der Diagonale der um 0° bzw. 90° gedrehten Normgesichter.....	53
Abbildung 42: Aus dem 60 x 60 Pixel großen Normgesicht.....	54
Abbildung 43: Aus dem 170 x 170 Pixel großen Normgesicht.....	54
Abbildung 44: Aus dem 350 x 350 Pixel großen Normgesicht.....	54
Abbildung 45: Aus dem 500 x 500 Pixel großen Normgesicht.....	54
Abbildung 46: Aus dem 1700 x 1700 Pixel großen Normgesicht.....	54
Abbildung 47: Der extrahierte Bildausschnitt vor der Beleuchtungskorrektur.....	55
Abbildung 48: Der extrahierte Bildausschnitt nach der Beleuchtungskorrektur.....	56
Abbildung 49: Der extrahierte Bildausschnitt nach der Beleuchtungskorrektur und dem Histogrammausgleich.....	56
Abbildung 50: USAN-Größe 7, Kontrast 27 und Schwellwert 100%.....	57
Abbildung 51: Beispiele für den Verlauf der Lernkurven (v.o.: Netz 24, Netz 38, Netz 41).....	59
Abbildung 52: Testset 1: Die Gesichtsbilder für die Tests der allgemeinen Klassifikationsfähigkeit der KNN : (oben v.l.) 170.jpg, 170_b.jpg, BioID_0000.jpg, BioID_0113.jpg, (unten v.l.) BioID_0488.jpg, BioID_0643.jpg, BioID_0181.jpg.....	61
Abbildung 53: Testset 2: Die Gesichtsbilder für die Tests der Generalisierungsfähigkeit der KNN: (v.l.) cnn1260.jpg, married.jpg, hendrix2.jpg, newsradio1.jpg, realTest_2.jpg.....	61
Abbildung 54: Ergebnisse der Tests zur Klassifikationsfähigkeit (Suchtestnr. v.l.: 29, 47, 51).....	62
Abbildung 55: Ergebnisse der Tests zur Generalisierungsfähigkeit (Suchtestnr. v.l.: 31, 54, 56).....	63

---

## Literaturverzeichnis

- [AMD01] E. Angelpoulou, R. Molana, K. Daniilidis, *Multispectral Skin Color Modeling*, Technical Report MS-CIS-01-22, University of Pennsylvania, 2001
- [ATD01] A. Albiol, L. Torrest, E.J. Delp, *Optimum Color Spaces for Skin Detection*, ICIP01, 2001
- [Br05] *BIOIDENTIFIKATION*, <http://www.bromba.com/faq/biofaqd.htm>, Oktober 2005
- [BSI03] C. Busch u.a., *BioFace - Vergleichende Untersuchung von Gesichtserkennungssystemen - Öffentlicher Abschlussbericht BioFace I & II*, BSI, 2003
- [CB97] J. Crowley, F. Berard, *Multi-modal tracking of faces for video communication*, Computer Vision and Pattern Recognition, 1997
- [CCC05] *Biometrische Merkmale in Ausweisen erhöhen Sicherheit nicht*, <http://www.ccc.de/biometrie/>, Oktober 2005
- [CMU05] *CMU Image Database: Frontal Face Images*, [http://vasc.ri.cmu.edu/idb/html/face/frontal\\_images/index.html](http://vasc.ri.cmu.edu/idb/html/face/frontal_images/index.html), Oktober 2005
- [DCPA02] M.N. Dailey, G.W. Cottrell, C. Padgett, R. Adolphs, *EMPATH: A Neural Network that Categorizes Facial Expressions*, Journal of Cognitive Neuroscience, vol. 14, no. 8, pp. 1158-1173, 2002
- [Dev05] *The Dev-C++ Resource Site*, <http://bloodshed.net/dev/devcpp.html>, Oktober 2005
- [Ec05] *Eclipse*, <http://www.eclipse.org/>, Oktober 2005
- [GB05] *Gesichter-Erkennung I Übersicht und Eigenfaces*, [www.informatik.uni-ulm.de/ni/Lehre/WS03/HSBiometrie/ausarbeitungen/GoldsteinRockmann.pdf](http://www.informatik.uni-ulm.de/ni/Lehre/WS03/HSBiometrie/ausarbeitungen/GoldsteinRockmann.pdf), Oktober 2005
- [Ha03] T. Haf, *Elementare Bildverarbeitung - Lokale Operatoren*, Proseminar Computer Vision - SS2003, 2003

- 
- [HFMDM00] J.M. Henderson, R. Falk, S. Minut, F.C. Dyer, S. Mahadevan, *Gaze Control for Face Learning and Recognition by Humans and Machines*, Michigan State University Eye Movement Laboratory Technical Report 2000, 2000
- [Ho82] J.J. Hopfield, *Neural Networks and physical systems with emergent collective computational abilities*, Proc. of the National Academy of Sciences, USA, vol. 79, pp. 2554-2558, 1982
- [Hu05] *BioID-Technology Research The BioID Face Database*, [www.bioid.de/downloads/facedb/index.html](http://www.bioid.de/downloads/facedb/index.html), Oktober 2005
- [JR99] M.J. Jones, J.M. Rehg, *Statistical Color Models with Application to Skin Detection*, International Journal of Computer Vision, vol. 46, no. 1, pp. 81-96, 1999
- [Ko90] T. Kohonen, *The Self-Organizing Map*, Proc. of the IEEE, vol. 78, no. 9, pp. 1464-1480, 1990
- [Kos87] B. Kosko, *Bi-directional associative memories*, IEEE Transactions on Systems, Man and Cybernetics, vol. 18, no. 1, pp. 49-60, 1987
- [KP97] C. Kotropoulos, I. Pitas, *Rule-based face detection in frontal views*, Proc. of IEEE Int. Conf. on Acoustics, Speech and Signal Processing, vol. 4, no. 4, pp. 2537-2540, 1997
- [LBP95] T.K. Leung, M.C. Burl, P. Perona, *Finding faces in cluttered scenes using random labeled graph matching*, 5. International Conference on Computer Vision, 1995
- [LGB05] *Pattern Recognition using Kohonen Featuremap*, <http://fbim.fh-regensburg.de/~saj39122/begrolu/kohonen.html>, Oktober 2005
- [LTC95] A. Lanitis, C.J. Taylor, T.F. Cootes, *Automatic Face Identification System Using Flexible Appearance Models*, IVC, vol. 13, no. 5, pp. 393-401, 1995
- [Mi00] S. Milch, *Videobasierte Fahreridentifikation in Kraftfahrzeugen*, Utz-Verlag München, ISBN 3-89675-741-5, 2000
- [MP69] M. Minsky, S. Papert, *Perceptrons: An Introduction to Computational Geometry*, MIT Press, 1969
- [NEC05] *High-end Range Finder Fiore*, [http://www.de.nec.de/productfiles/546\\_1Fiore\\_3D\\_d.pdf](http://www.de.nec.de/productfiles/546_1Fiore_3D_d.pdf), Oktober 2005

- 
- [PG89] T. Poggio, F. Girosi, *A theory of networks for approximation and learning*, A.I. Memo No.1140, MIT, 1989
- [Pi98] *M2VTS Project: MULTI-MODAL BIOMETRIC PERSON AUTHENTICATION*, <http://www.tele.ucl.ac.be/PROJECTS/M2VTS/>, Oktober 2005
- [Ro99] H.A. Rowley, *Neural network-Based Face Detection*, PhD thesis, Carnegie Mellon University, ISBN 0-599-52020-5, 1999
- [Sa93] F. Samaria, *Face Segmentation For Identification Using Hidden Markov Models*, BMVC, 1993
- [SB95] S.M. Smith, J.M. Brady, *SUSAN-A New Approach to Low Level Processing*, Technical Report TR95SMS1c, vol. 23, no. 1, pp. 45-78, 1995
- [SM95] S.M.Smith, J.M.Brady, *SUSAN-A New Approach to Low Level Processing*, Technical Report TR95SMS1c, vol. 23, pp. , 1995
- [SNNS05] *Stuttgart Neural Network Simulator*, <http://www-ra.informatik.uni-tuebingen.de/SNNS/>, Oktober 2002
- [Sp90] D.F. Specht, *Probabilistic neural networks and the polynomial adaline as complementary techniques for classification*, IEEE Transactions on Neural Networks, vol. 1, no. 1, pp. 111-121, 1990
- [TKP98] A. Tefas, C. Kotropoulos, I. Pitas, *Variants of Dynamic Link Architecture based on mathematical morphology for frontal face authentication*, Computer Vision and Pattern Recognition, 1998
- [TP91b] M. Turk, A. Pentland, *Eigenfaces for recognition*, Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71 - 86, 1991
- [Vi05] *Lösungen zur Personenidentifikation*, <http://www.viisage.com/ww/de/pub/home.cfm>, Oktober 2005
- [Wi05] *Sobel-Operator*, <http://de.wikipedia.org/wiki/Sobel-Operator>, Oktober 2005
- [Ya04] M. Yang, *Recent Advantages in Face Detection*, ICPR 2004, 2004
- [YC96] K.C. Yow, R. Cipolla, *A Probabilistic Framework for Perceptual Grouping of Features for Human Face Detection*, Proc. Second Int'l Conf. Automatic Face and Gesture Recognition, 1996

- 
- [YH94] G. Yang, T.S. Huang, *Human Face Detection in Complex Background*, Pattern Recognition, vol. 27, no. 1, pp. 53-63, 1994
- [YKA02] M. Yang, D. Kriegmann, N. Ahuja, *Detecting Faces in Images: A Survey*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 1, pp. 34-58, 2002
- [Ze94] Prof. Dr. Andreas Zell, *Simulation neuronaler Netze*, Addison Wesley, ISBN 3-486-24350-0, 1994

## **Erklärung**

Hiermit erkläre ich, dass die vorliegende Diplomarbeit von mir selbst ohne fremde Hilfe erstellt wurde. Alle benutzten Quellen sind im Literaturverzeichnis aufgeführt. Diese Arbeit hat in dieser oder ähnlicher Form noch keinem anderen Prüfungsausschuss vorgelegen.

Werder, 8. November 2005

Holger Ripp