Faltungsnetzwerke zur Gesichtsdetektion unter Einbeziehung der Pose

Diplomarbeit

Fakultät für Informatik
Technische Universität Dortmund

vorgelegt von
Fabian Naße
10.12.2008

Gutachter:
Dr.-Ing. Christian Thurau
Professor Dr.-Ing. Gernot A. Fink

Inhaltsverzeichnis

Abbildungsverzeichnis

Kapitel 1

Einleitung

Bei dem Thema der Gesichtsdetektion geht es um das automatische Erkennen von Gesichtern in Bildern. Im Bereich der visuellen Mustererkennung ist die Gesichtsdetektion ein wichtiges Forschungsgebiet. Wenn sich ein System bzw. ein Roboter mit Hilfe von optischen Sensoren in seiner Umgebung zurechtfinden kann, spricht man in diesem Zusammenhang vom maschinellem Sehen (engl.: computer vision). Soll ein solches System mit Menschen interagieren, ist die Möglichkeit, Gesichter zu erkennen, von großem Vorteil. Es kann so auf die Anwesenheit von Personen geschlossen werden. Das System kann dann automatisch darauf reagieren und bspw. Dienstleistungen anbieten oder bestimmte Anwendungen ausführen. Im Bereich der Mensch-Maschinen-Interaktion gibt es zahlreiche zukunftsorientierte Konzepte dieser Art, um nicht mehr auf konventionelle Eingabegeräte wie Maus oder Tastatur zurückgreifen zu müssen, sondern Benutzerschnittstellen bereit stellen zu können, die einen bequemen und intuitiven Umgang mit der Technik erlauben. Bei der Realisierung solcher Schnittstellen werden häufig herkömmliche Videokameras eingesetzt, da diese vielseitig einsetzbar und günstig in der Anschaffung und Installation sind. Es gibt eine ganze Reihe konkreter Anwendungsfälle, bei denen ein Gesichtsdetektor benötigt wird oder zumindest von diesem profitiert werden kann. Ein wichtiges Beispiel ist die Identifizierung von Personen. Das Gesicht stellt ein eindeutiges biometrisches Identifizierungsmerkmal des Menschen dar. Dies kann bspw. im Rahmen einer Zugriffskontrolle ausgenutzt werden. Erkennt der Detektor im ersten Schritt ein Gesicht, kann anschließend versucht werden, die Identität der Personen mit Hilfe eines Gesichtsidentifizierungsverfahrens zu ermitteln. Ein weiteres Beispiel ist das Gesichtstracking. Ist ein Gesicht detektiert worden kann es mit Hilfe eines geeigneten Trackingverfahrens weiter verfolgt werden. Auf diese Weise kann bspw. festgestellt werden, ob eine Person einen Raum betritt oder verlässt. Anstatt sich nur auf das Gesicht zu beschränken, kann auch versucht werden, die gesamte Person zu detektieren. Da das Gesicht im Allgemeinen einfacher zu erkennen ist, als andere Körperregionen wie bspw. Arme oder Beine, kann ein Personendetektor im ersten Schritt das Gesicht suchen. Ist die Position des Gesichts erst bekannt, können andere Körperteile wesentlich leichter gefunden werden. Um

Abbildung 1.1: Der „intelligente Raum“ des Instituts für Roboterforschung der technischen Universität Dortmund. Zwei gegenüberliegende Deckenkameras werden hier zur Gesichtsdetektion eingesetzt.

Rückschlüsse auf den Gemütszustand einer Person zu ziehen, kann nach erfolgter Detektion eine Mimikerkennung zum Einsatz kommen. Das Ziel kann es bspw. sein, zu erkennen, ob ein Anwender auf eine bestimmte Fehlermeldung verärgert reagiert. Eine Anwendung könnte dann entsprechend darauf reagieren.

In dieser Arbeit wird das in [Osadchy u. a. 2005] und [Osadchy u. a. 2007] vorgestellte Verfahren zur Gesichtsdetektion unter Einbeziehung der Pose untersucht. Im Kern verwendet dieses Verfahren ein Faltungsnetz. Faltungsnetze stellen eine spezielle Form künstlicher neuronaler Netze dar und können deshalb mittels Backpropagation trainiert werden. Die Funktionsweise des Verfahrens wird von den Autoren sehr detailliert beschrieben. Dennoch bleiben einige Fragen offen. Es wird lediglich eine einzige, relativ große Netzstruktur vorgeschlagen und mit rund 60.000 Beispielbildern eine sehr große Trainingsdatenmenge verwendet. In dieser Arbeit wird deshalb untersucht, ob auch mit kleineren Faltungsnetzen und weniger Bildern gute Ergebnisse erzielt werden können. In diesem Kontext werden Zusammenhänge zwischen den verschiedenen Trainingsparametern, Netzstrukturen und Datenmengen herausgearbeitet. Das Ziel dabei ist es, eine allgemeine Strategie zu entwickeln, mit der schnelle Trainingsfortschritte erzielt werden können. Ein weiterer Schwerpunkt dieser Arbeit ergibt sich wie folgt. Das vorgestellt Verfahren liefert zunächst für ein kleines Eingabebild ein Label. Die Autoren beschreiben nur oberflächlich, wie sie das Labeling zu einem vollständigen Lokalisierungsverfahren für größere Bilder ausgebaut haben. Eine Zielsetzung dieser Arbeit ist es deshalb, diese Problemstellung etwas genauer zu betrachten und eine entsprechende Erweiterung zu entwickeln. Es wird insbesondere Wert auf eine hohe Ausführungsgeschwindigkeit gelegt. Die Implementierung des Verfahrens erfolgt deshalb für einen schnellen Graphikchip (nVidia 8 Series). Das Gesamtsystem wird schließlich mit den unterschiedlichen Netzstrukturen ausführliche evaluiert. Hierbei wird unter Anderem der Einsatz in einem Konferenzraum mit Deckenkameras getestet (siehe Abbildung 1.1).

Kapitel 2 geht zunächst allgemein auf die Problemstellung der Gesichtsdetektion ein und gibt eine kurze Übersicht über verschiedene Verfahren zu diesem Thema. Kapitel 3 erläutert das dieser Arbeit zugrunde liegenden Verfahren und beschreibt insbesondere, wie die Gesichtspose in den Detektionsprozess mit einbezogen wird. Kapitel 4 behandelt die Grundlagen neuronaler Netze, soweit sie zum Verständnis dieser Arbeit benötigt werden und geht dabei auf das Backpropagation-Verfahren und einige für diese Arbeit relevanten Erweiterungen zu diesem ein. Aufbauend auf Kapitel 4 wird in Kapitel 5 die Struktur und Funktionsweise von Faltungsnetzen erläutert. Kapitel 5 beschreibt die für diese Arbeit gewählte Vorgehensweise bei der Erweiterung des Labelings zu einem effizienten Lokalisierungsverfahren. Die durchgeführten Untersuchungen zum Training von Faltungsnetzen werden in Kapitel 7 aufgezeigt und ausgewertet. Eine ausführliche Evaluierung des Detektors erfolgt in Kapitel 8. Schließlich befindet sich in Kapitel 9 eine Zusammenfassung sowie ein Ausblick.

Kapitel 2

Gesichtsdetektion

Dieses Kapitel gibt eine Übersicht zu dem Thema der Gesichtsdetektion. In Abschnitt 2.1 wird das Problem genauer formuliert und erläutert, welche Faktoren es zu einer schwierigen Herausforderung machen. In Abschnitt 2.2 werden verschiedenen Gesichtsdetetkionsverfahren betrachtet. Da die Anzahl der Veröffentlichungen zu diesem Thema sehr hoch ist, wird hier kein Anspruch auf Vollständigkeit erhoben. Vielmehr werden anhand repräsentativer Beispiele verschiedene Herangehensweisen aufgezeigt. Schließlich werden in Abschnitt 2.3 verwandte Arbeiten betrachtet, die sich entweder speziell mit dem Problem der Posenvariation beschäftigen oder auf künstliche neuronale Netze zurückgreifen.

2.1 Problemstellung

In diesem Abschnitt wird die Problemstellung der Gesichtsdetetkion genauer beschrieben und es werden einige wichtige Begriffe in diesem Zusammenhang erläutert. Es wird in dieser Arbeit zwischen dem Labelproblem und dem Lokalisierungsproblem unterschieden. Beim Labelproblem (vgl. Abbildung 2.1 a) soll für ein relativ kleines Bild entschieden werden, ob dieses ein Gesicht zeigt, oder nicht. Zwei Arten von Fehlern können hierbei auftreten:

- Ein falsch-positives Ergebnis (engl.: false positive) liegt vor, wenn fälschlicher Weise ein Gesicht erkannt wurde.
- Ein falsch-negatives Ergebnis (engl.: false negative) liegt vor, wenn fälschlicher Weise kein Gesicht erkannt wurde.

Die visuelle Wahrnehmungsfähigkeit eines durchschnittlichen menschlichen Probanden kann hierbei als Referenz dienen. Wenn ein menschlicher Betrachter, bspw. aufgrund starker Rauscheinflüsse oder schlechter Lichtverhältnisse, ein Gesicht nicht mehr eindeutig erkennen kann, wird dies auch nicht von einem Computerprogramm erwartet.

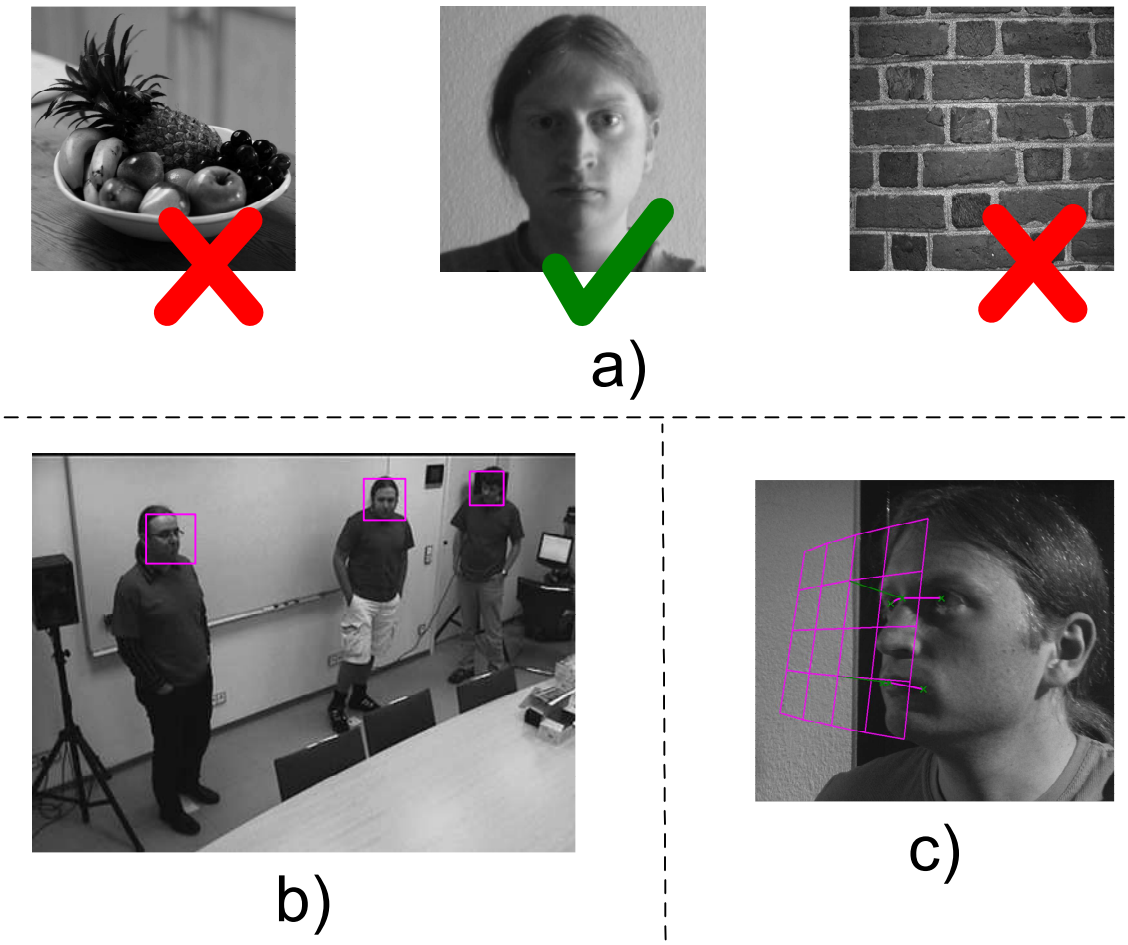

Abbildung 2.1: Betrachtete Problemstellungen: a) Labelproblem, b) Lokalisierungsproblem, c) Posenschätzung.

Das Lokalisierungsproblem (Abbildung 2.1 b) ist eng mit dem Labelproblem verwandt. Hier soll in einem größeren Bild (oder einer Bildsequenz) die Positionen und Größen aller abgebildeten Gesichter möglichst genau ermittelt werden. Die Größe eines Gesichts kann hierbei über markante Punkte definiert werden, die sich bei jedem Gesicht eindeutig wieder finden lassen (bspw. die Augenkoordinaten). Auch bei der Gesichtslokalisierung gibt es analog zum Labelproblem zwei Arten von Fehlern. Jedoch können hier im Allgemeinen wesentlich mehr falsch-positive Ergebnisse auftreten, da in einem Bild in der Regel Regionen überwiegen, die keine Gesichter zeigen. Des Weiteren wird bei der Bewertung der Ergebnisse ein gewisser Toleranzspielraum eingeräumt, d.h. die Position und die Größe eines Gesichts müssen zwar relativ genau, aber nicht exakt getroffen werden (vgl. später Kapitel 8).
Allgemein können in der Praxis viele erschwerende Faktoren das Problem der Gesichtsdetektion zu einer anspruchsvollen Herausforderung machen. Wie schwierig ein bestimmtes Gesicht im Einzellfall zu detektieren ist, hängt zum Einen vom konkreten Erscheinungsbild des Individuums ab und zum Anderen von den äußeren Umständen, unter denen die Aufnahme entstanden ist. Gesichter können von Fall zu Fall sehr unterschiedlich aussehen, je nachdem welches Alter, Geschlecht, Körpergewicht und welche ethnische Zugehörigkeit eine

Person hat. Darüber hinaus gibt es noch eine Reihe von weiteren Faktoren, von denen einige Wichtige im Folgenden aufgelistet werden:

- Pose - Einen erheblichen Einfluss auf das Erscheinungsbild hat die Pose, d.h. der Aufnahmewinkel der Kamera relativ zur Lage des Gesichts. Bspw. ändert sich das Aussehen eines Gesichts von der Frontalansicht bis hin zum Profil sehr stark. Im einfachsten Fall ist ein Gesichtsdetektor nur auf eine bestimmte Pose spezialisiert und toleriert Abweichungen nur in sehr geringem Umfang. In dieser Arbeit wird ein Schwerpunkt auf das Problem der Posenvariation gelegt. In diesem Zusammenhang ist der Begriff der Posenschätzung von Bedeutung (Abbildung 2.1 c). Hierbei soll für ein Bild, das ein Gesicht zeigt, die Pose dieses Gesichts bestimmt werden. Die Angabe kann dabei über einen Normalenvektor oder äquivalent mit Hilfe von Winkeln erfolgen. Die Pose eines Objekts kann allgemein mit drei Winkeln vollständig beschrieben werden. Es ist jedoch auch eine vereinfachte Problemstellung denkbar, bei der nur zwei oder ein Winkel betrachtet werden.

- Zusätzliche strukturelle Komponenten - Damit sind Objekte gemeint, die zum Gesicht gehören, aber nicht bei allen Personen vorkommen. Hierzu zählen in erster Linie Brillen und Gesichtsbehaarung. Beides kann in vielen unterschiedlichen Varianten vorkommen.

- Gesichtsausdruck - Insbesondere das Aussehen von Mund und Augen können durch unterschiedliche Mimiken stark variieren. Beispielsweise können die Augen geschlossen, die Augenbrauen oder Mundwinkel hochgezogen oder der Mund geöffnet sein. Bei letzterem können auch Teile des Gebisses sichtbar werden.

- Verdeckung - Teile des Gesichts können durch andere Objekte verdeckt sein. Ein typischer Fall ist, dass eine Person eine Hand vor sein Gesicht hält. Denkbar ist auch eine Maskierung durch einen Schal oder ähnliches. Eine andere Möglichkeit besteht darin, dass eine Person, die in einer Menschenmenge steht, durch andere Personen im Vordergrund teilweise verdeckt wird.

- Lichtverhältnisse - Stark unterschiedliche Situationen können sich durch die Lage, Intensität und den Spektralbereich der Lichtquellen ergeben. Je nach Situation kann bspw. ein ungünstiger Schattenwurf entstehen.

- Bildqualität - Diese hängt in erste Linie von der Aufnahmeapertur der Kamera ab. Des Weiteren können noch Einflüsse durch Rauschen oder einer verlustbehafteten Codierung hinzukommen. Wenn ein bestimmtes Verfahren auch bei einer moderaten Bildqualität noch gute Ergebnisse erzielt, kann dies die Kosten der Hardwareanschaffung reduzieren.

2.2 Verfahren zur Gesichtsdetektion

Trotz intensiver Forschung und zahlreicher Veröffentlichungen der letzten Jahre zum Thema Gesichtsdetektion, kann von einer abschließenden Lösung des Problems nicht die Rede sein, vergleicht man die Qualität der besten bekannten Verfahren mit der visuellen Wahrnehmungsfähigkeit des Menschen. In [Yang u. a. 2002] wird eine Einteilung unterschiedlicher Ansätze in vier Kategorien vorgenommen. Es wird dort zwischen Verfahren auf Basis von Expertenwissen, invarianten Merkmalen, standardisierten Vorlagen und Erscheinungsbildern unterschieden. Zwar lassen sich konkrete Methoden aufgrund thematischer Überschneidungen häufig nicht scharf einer Kategorie zuordnen, jedoch ist diese Einteilung geeignet, um generelle Strategien und Herangehensweisen zu verdeutlichen.

Verfahren auf Basis von Expertenwissen greifen auf menschliches Wissen über Gesichter zurück, um konkrete Regeln aufzustellen, die beim Detektionsprozess überprüft werden. Diese Regeln können sich bspw. auf geometrische Eigenschaften wie die Lage von Mund, Augen, Kinn und Nase beziehen. Das in [Yang und Huang 1994] vorgestellte Verfahren ist ein bekanntes Beispiel hierfür. Dort werden Regeln hierarchisch in drei unterschiedlichen Detailstufen aufgestellt. Wissensbasierte Verfahren haben den Nachteil, dass sie stark auf die Problemstellung zugeschnitten sind, sich also nicht ohne Weiteres zur Detektion anderer Objekte als Gesichter einsetzen lassen. Des Weiteren lassen sich zuverlässige Regeln nur schwer in einer ausreichend hohen Zahl aufstellen, so dass letztendlich die zur Verfügung stehenden Bildinformationen nur sehr begrenzt ausgenutzt werden.

Invariante Merkmale sind solche, die auch unter Variationen von Position, Pose und Beleuchtung existieren. Diese Merkmale können auf strukturellen, texturellen oder farblichen Informationen basieren. Mit einer allgemeinen Methode (bspw. Kantendetektion, Histogramme, Integralbilder oder Wavelets) werden zunächst unbestimmte Informationen aus dem Bild extrahiert. Anschließend wird mit einem statistischen Modell überprüft, ob diese Informationen auf ein Gesicht schließen lassen. Das verfahren in [Gundimada und Asari 2004] arbeitet mit skalierungs- und rotationsinvarianten Merkmalen auf Basis von Wavelets. Ein weiteres Beispiel in diesem Zusammenhang ist der Ansatz, Hautfarbe zu detektieren. In Kombination mit anderen Merkmalen bzw. Verfahren kann ein Hautfarbendetektor zu einem Gesichtsdetektor ausgebaut werden (bspw. [Hsu u. a. 2002], [Zhang und Izquierdo 2006], oder auch [Yang u. a. 2008]).

Eine standardisierte Vorlage besteht aus Vergleichsmustern, die sich auf das ganze Gesicht oder auf einzelne Partien wie Augen und Mund beziehen können. Verfahren auf dieser Basis überprüfen Korrelationen zwischen diesen Mustern und dem Eingabebild. Das Problem dieses Ansatzes ist, dass solche Vorlagen schon bei kleinen Variationen der Größe, Form oder Pose schlechte Resultate liefern können. Aktuell gibt es dennoch verschiedene Ansätze, die von Vorlagen Gebrauch machen und dabei versuchen dieses Problem zu bewältigen. Bei-

spielsweise werden in [Zhong u. a. 2007] in einem Vorverarbeitungsschritt zunächst auf Basis der Hautfarbe und mit Hilfe von morphologischen Operationen einheitliche Rechtecke extrahiert, die in einem zweiten Schritt mit einer Vorlage verglichen werden. In [Wang und Yang 2008] wiederum wird eine Vorlage im ersten Schritt eines hierarchischen Prozesses lediglich als schwacher Klassifizierer eingesetzt, um den Suchbereich einzuschränken.

Verfahren auf Basis des Erscheinungsbildes zeichnen sich durch eine holistische Verarbeitung des Eingabebildes aus: Um für ein Bild ein Label zu ermitteln, wird es als ganzheitlicher Informationsvektor aufgefasst und stellt somit eine Variable in einem hochdimensionalen Raum dar, die mit einem geeigneten Modell ausgewertet werden kann. Das benötigte Wissen, um ein solches Modell aufzustellen, wird in einem automatischen Trainingsprozess aus einer Menge von Beispielbildern bezogen. Die Beispielbilder teilen sich in Gesichter und Nicht-Gesichter auf und decken eine große Vielfalt an möglichen Erscheinungsformen ab. Eine häufig angewendete Vorgehensweise in diesem Zusammenhang ist die Durchführung einer Unterraumanalyse. Ein typisches Beispiel hierzu findet sich in [Yang u. a. 2000]. Ein weiteres Beispiel ist die Verwendung einer sogenannten Support-Vector-Maschine, deren Aufgabe es ist, während des Trainings im Vektorraum eine mehrdimensionale Hyperebene einzupassen, die als Trennfläche zwischen Gesichtern und Nicht-Gesichtern fungiert (bspw. in [Waring und Liu 2005]). Verfahren, die künstliche Neuronale Netze einsetzen, fallen typischerweise auch in die Kategorie erscheinungsbasierter Ansätze. Dazu gehört auch das in dieser Arbeit untersuchten Verfahren (vgl. Kapitel 3 bis 5). Des Weiteren können auch statistische Modelle, wie z.B. das Hidden Markov Modell (bspw. in [Dass und Jain 2001]) eingesetzt werden.

Von Viola und Jones wurde 2002 ein schnelles und qualitativ hochwertiges Verfahren vorgestellt [Viola und Jones 2002]. Mit einem Boosting-Algorithmus, dem sogenannten AdaBoost, wird mit Hilfe einer Menge von Trainingsbildern eine Reihe von schwachen Klassifizierern zu einem starken Klassifizierer ausgebaut. Die schwachen Klassifizierer haben eine geringe Trefferquote (etwas über 50%) und arbeiten mit einfachen Merkmalen, die aus der Differenz von Integralbildern aus benachbarten, rechteckförmigen Bildregionen gebildet werden. Eine Reihe von starken Klassifizierern wird wiederum zu einer Kaskade zusammengeschaltet, die dann zur Detektion eingesetzt wird. Aufgrund seiner Qualität und Geschwindigkeit, hat sich das Verfahren von Viola und Jones zu einem Standard etablieren können und wird in vielen Veröffentlichungen aus Gründen eines Qualitätsvergleiches referenziert. Das Verfahren hat jedoch auch Nachteile. Zunächst einmal sind die rechteckigen Merkmale nicht robust gegenüber Posenvariationen. Entsprechend ist es auch nicht möglich, mit der selben Kaskade gleichzeitig Gesichter in Frontal- und Profilansicht zu detektieren. Einen weiteren Nachteil bilden die langen Trainingszeiten, die das AdaBoost benötigt um eine vollständige Kaskade zu entwickeln. Das in dieser Arbeit betrachtete Verfahren hat diese Nachteile nicht (vgl. Kapitel 3 zum Thema Pose und Kapitel 7 zum Thema Trainingszeiten).

2.3 Verwandte Arbeiten

Im Folgenden werden themenverwandte Arbeiten betrachtet. In Abschnitt 2.3.1 geht es dabei speziell um das Thema der Posenvariation und in Abschnitt 2.3.2 um den Einsatz von Neuronale Netzen zur Gesichtsdetektion.

2.3.1 Posenproblem

Wie zuvor schon angemerkt wurde, hat die Pose einen erheblichen Einfluss auf das Erscheinungsbild von Gesichtern. Da die Flexibilität eines Systems erheblich gesteigert wird, wenn es Gesichter unabhängig von der Pose erkennen kann, gibt es entsprechend eine Reihe von Veröffentlichungen, die ein besonderes Augenmerk auf dieses Problem legen. In diesem Abschnitt werden einige typische Beispiele hierfür aufgezeigt.

Eine sehr naheliegende und deshalb häufig angewendete Methode ist es, mehrere Klassifizierer für unterschiedliche Posen zu trainieren und diese parallel zu betreiben. Nach einer bestimmten Regel wird dann zwischen den Ergebnissen der einzelnen Klassifizierer ausgewählt. Ein Beispiel hierzu findet sich in [Schneiderman und Kanade 2000], wo mehrere erscheinungsbasierte Klassifizierer eingesetzt werden. Bei solchen Ansätzen besteht auch die Möglichkeit, das Eingabebild zu rotieren und den selben Klassifizierer mehrmals zu verwenden. Dadurch können verschiedene Winkel in der Bildebene überprüft werden. Eine horizontale Spiegelung ist auf Grund der Symmetrieeigenschaften von Gesichtern ebenfalls möglich. Der Einsatz mehrere Klassifizierer hat jedoch Nachteile: Für jeden Klassifizierer muss ein eigenes Training durchgeführt werden. Des Weiteren erhöht sich der Rechenaufwand entsprechend der Anzahl der eingesetzten Klassifizierer um ein Vielfaches. Viola und Jones schlagen in [Viola und Jones 2003] eine Lösung für das letztgenannte Problem vor. Aufbauend auf ihrer Arbeit in [Viola und Jones 2002] werden dort mehrere Kaskaden für jeweils eine andere Pose trainiert. In einem vorangestellten Schritt entscheidet ein Posenschätzer, welche Kaskade in der jeweiligen Situation eingesetzt wird. Wird der Posenschätzer auf ein Nicht-Gesicht angewendet, kann die Wahl als zufällig betrachtet werden. In [Li u. a. 2000] wird ein ähnliches Vorgehen auf Basis einer Support-Vector-Maschine vorgeschlagen. Eine andere Vorgehensweise findet sich in [Seshadrinathan und Ben-Arie 2003]. Hier wird im ersten Schritt mit einem Farbdetektor für jeden Pixel ermittelt, ob dieser zu einer Hautregion gehört und mit diesen Informationen zusammenhängenden Regionen ermittelt. Es wird nun angenommen, dass Gesichter unabhängig von ihrer Pose eine annähernd elliptische Form haben. Diese Annahme wird in einem zweiten Schritt genutzt, um bestimmte Regionen herauszufiltern. Schließlich werden in einem dritten Schritt Gabor-Wavelets eingesetzt, die auf unterschiedliche Kopfposen ausgelegt sind.

Die bisher betrachteten Verfahren, haben gemeinsam, dass sie an das Posenproblem durch

mehrere separate Arbeitsschritte herangehen. Im Gegensatz dazu wird bei dem in dieser Arbeit untersuchte Verfahren nach [Osadchy u. a. 2007] nur ein einziger Klassifizierer benötigt. Dies gelingt durch eine Integration der Posenschätzung in den Klassifizierungsprozess (vgl. Kapitel 3). Hierdurch ergibt sich ein erheblicher Vorteil: Es müssen nicht mehrere Klassifizierer trainiert und betrieben werden. Stattdessen können das Training sowie die Detektion mit einem einheitlichen Prozess durchgeführt werden.

2.3.2 Künstliche Neuronale Netze

Schon seit einigen Jahren gibt es Veröffentlichungen, die Gesichtsdetektion mit Hilfe von künstlichen neuronalen Netzen zum Thema haben. Eine früher Ansatz findet sich bspw. in [Propp und Samal 1992]. Damit neuronale Netze bei Problemstellungen im Bereich der visuellen Mustererkennung gute Ergebnisse erzielen können, verfügen sie im Allgemeinen über eine komplexe Netzstruktur. Dies ist verbunden mit einer hohen Zahl an Multiplikationen und langen Rechenzeiten. Neuere Veröffentlichungen untersuchen deshalb häufig Methoden, um das Training oder die Detektion zu beschleunigen. Ein Ansatz hierzu ist es, zunächst in einem geeigneten Vorverarbeitungsschritt die Dimension der Eingabe zu reduzieren. In einem zweiten Schritt wird dann ein kleineres Netz eingesetzt, um den reduzierten Informationsvektor zu klassifizieren. Ein Beispiel hierzu findet sich in [Kobayashi und Zhao 2007], wo zur Informationsreduktion ein auf die lineare Diskriminanzanalyse aufbauendes Verfahren eingesetzt wird. In [El-Bakry und Stoyan 2004] wir die Möglichkeit erörtert, die Berechnungen des Netzes unter Einsatz der schnellen Fouriertransformation im Ortsfrequenzbereich durchzuführen: Wird ein neuronales Netz mehrfach auf die Teilbilder in einem größeren Bild angewendet, dann lassen sich die erforderlichen Multiplikationen formelmäßig als Kreuzkorrelation zwischen dem Bild und den Netzparametern darstellen. Dies funktioniert jedoch nur, wenn entweder das Bild oder die Netzparameter symmetrisch sind, was bestimmte Vorkehrungen nötig macht. Faltungsnetze, die eine spezielle Form neuronaler Netze darstellen und mitunter Thema dieser Arbeit sind (vgl. Kapitel 5), bieten eine etwas andere Möglichkeit Multiplikationen einzusparen. Durch bestimmte Strukturvorgaben an das Netz, wird erreicht, dass Berechnungen des Netzes formelmäßig als Faltungen dargestellt werden können. Bei wiederholter Anwendung auf überlappende Teilbilder entstehen auf diese Weise Rechenredundanzen, die eingespart werden können (vgl. Kapitel 6.2). Neben der in dieser Arbeit betrachteten Form von Faltungsnetzen (nach [Vaillant u. a. 1993], [LeCun u. a. 1998a] und [Osadchy u. a. 2007]) findet sich ein ähnliches Beispiel in [Tivive und Bouzerdoum 2003]. Nach einem ganz ähnlichen Prinzip wird hier ein neuronales Netz eingesetzt, um Gesichter aus einer Frontalansicht zu detektieren. Abschließend sein noch erwähnt, dass neuronale Netze auch eingesetzt werden können, um die schon in den letzten Abschnitten mehrfach erwähnte Hautfarbendetektion zu realisieren, so z.B. in [Seshadrinathan und Ben-Arie 2003] oder auch [Seow u. a. 2003].

2.4 Zusammenfassung

In diesem Kapitel wurde eine Übersicht zum Thema der Gesichtsdetektion gegeben. Nach einer konkreten Beschreibung der Problemstellung (Labeling und Lokalisierung) wurden besondere Schwierigkeiten und Herausforderungen diesbezüglich aufgezeigt. Dabei wurde insbesondere das Problem der Posenvariation hervorgehoben und der damit verbundene Begriff der Poseschätzung erläutert. Im Anschluss daran wurden allgemeine Herangehensweisen bei der Entwicklung eines Gesichtsdetektors betrachtet (Expertenwissen, invariante Merkmale, standardisierte Vorlagen und Erscheinungsbilder), die mit konkreten Beispielen aus der Literatur verdeutlicht wurden. Darüber hinaus wurden themenverwandte Arbeiten betrachtet, bei denen ein besonderer Schwerpunkt auf das Problem der Posenvariation gelegt wird, oder künstliche neuronale Netze verwendet werden. In diesem Rahmen wurden bereits wichtige Vorteile des in dieser Arbeit untersuchten Verfahrens angesprochen, die in den nachfolgenden Kapiteln noch vertieft werden (integrierte Posenschätzung, geringe Laufzeit des Trainings, Einsparen der Rechenredundanz).

Kapitel 3

Kombination von Gesichtsdetektion und Posenschätzung

Im letzten Kapitel wurde das Problem der Posenvariation hervorgehoben und die Vorzüge des Verfahrens nach [Osadchy u. a. 2007] angesprochen. Bei diesem Verfahren wird lediglich ein Klassifikator eingesetzt, bei dessem Entwurf bereits das Problem der Posenvariation durch die Integration eines Posenschätzers berücksichtigt wird. Auf diese Weise wird erreicht, dass das Verfahren robust gegenüber starken Posenvariationen ist. Auch bei nur kleinen Variationen können so bessere Ergebnisse erzielt werden, als mit einem vergleichbaren Verfahren ohne integrierter Posenschätzung. Die Vorgehensweise wird in den folgenden Abschnitten erläutert. Abschnitt 3.1 skizziert die Grundzüge des Verfahrens, während die nachfolgenden Abschnitte 3.2 und 3.3 auf wichtige Details eingehen.

3.1 Grundprinzip

Um eine Posenschätzung in den Detektor zu integrieren, muss ein einheitliches Modell entworfen werden, das das Label und die Pose einheitlich beschreiben kann. In [Osadchy u. a. 2007] wird hierzu eine innovative Vorgehensweise vorgeschlagen. Die Menge der verschiedenen Posen wird hier in einem euklidischen Raum abgebildet. Hierzu wird eine Mannigfaltigkeit definiert (siehe Abschnitt 3.2). Jeder Punkt dieser Mannigfaltigkeit repräsentiert eine bestimmte Pose. Ein Punkt, der sich auf oder in der Nähe der Mannigfaltigkeit befindet, repräsentiert ein Gesicht. Punkte, die einen bestimmten Maximalabstand von der Mannigfaltigkeit überschreiten, repräsentieren Nicht-Gesichter. Um nun ein Bild in einen Punkt in diesem euklidischen Raum zu überführen, wird ein Mapping-Modul verwendet. Dieses Modul ist als eine komplexe Funktion zu betrachten, deren Verhalten über eine Vielzahl von Parametern eingestellt werden kann. Die Dimension der Eingabe, d.h. die Anzahl der Pixel eines Bildes, ist dabei wesentlich höher als die der Ausgabe, die wiederum der Dimension des

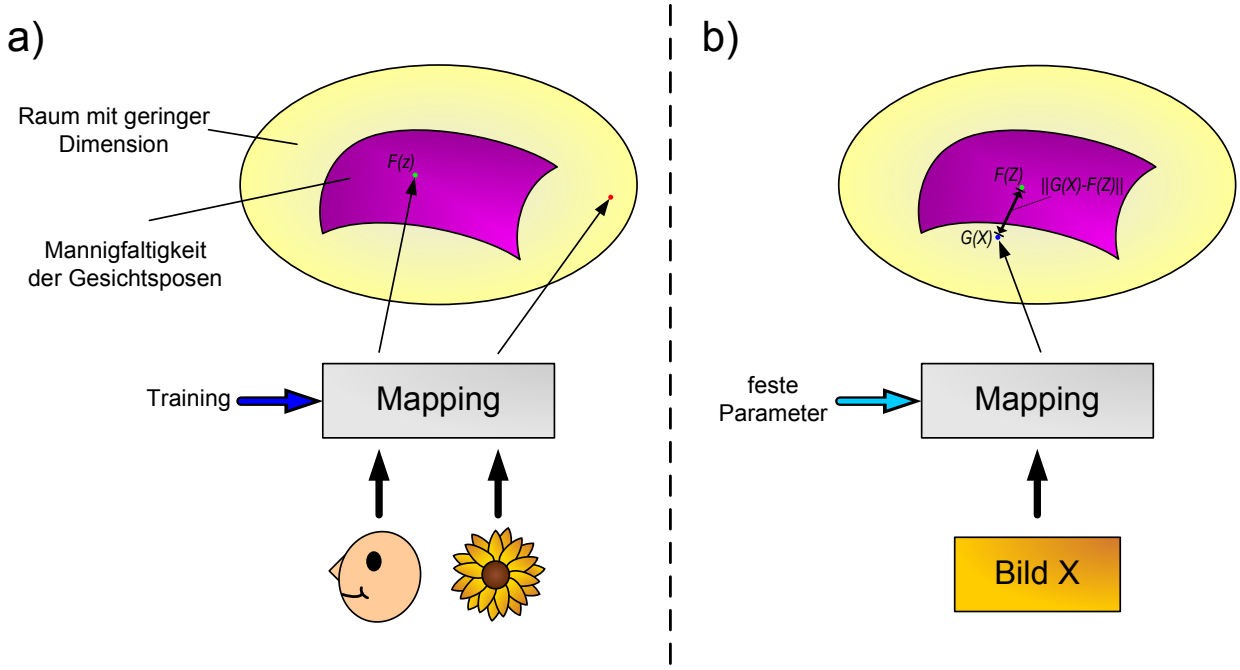

Abbildung 3.1: Mapping: a) Trainingsmodus, b) Gesichtserkennung und Posenschätzung nach [Osadchy u. a. 2007].

euklidischen Raumes entspricht. Abbildung 3.1 zeigt eine Übersicht des Gesamtsystems, das sich in eine Trainingsphase (Teil a) und in die eigentliche Detektionsphase (Teil b) aufteilt.

Das Ziel des Trainings ist es, geeignete Parameter für das Mapping-Modul zu entwickeln. Dazu erhält es nacheinander einzelne Trainingsbilder. Für jedes Trainingsbild ist bekannt, ob es sich um ein positives Beispiel, d.h. ein Gesicht, oder um ein negatives Beispiel, also ein Hintergrundbild oder ähnliches, handelt. Im Falle eines positiven Beispiels ist zusätzlich die Pose des Gesichts ebenfalls bekannt. Mit Hilfe eines bestimmten Trainingsverfahrens werden nun die Parameter des Mapping-Moduls für jedes Beispiel angepasst, und zwar so, dass die Ist-Ausgabe des Mappings die Soll-Ausgabe des jeweiligen Trainingsbeispiels besser erfüllt. Bei einem positiven Beispiel ist es das Ziel, ein Ergebnis möglichst nah an dem Punkt auf der Mannigfaltigkeit zu erzielen, der der angegebenen Pose entspricht. Bei einem negativen Beispiel hingegen soll das Ergebnis ein unbestimmter Punkt möglichst weit entfernt von der Mannigfaltigkeit sein. Um eine ausreichend hohe Qualität zu erzielen, wird die Trainingsmenge mehrfach durchlaufen.

Bei der Detektion wird zunächst das fragliche Eingabebild X dem Mapping-Modul übergeben. Die Ausgabe des Moduls sei der Punkt $G(X)$. Als nächstes wird der Punkt $F(Z)$ auf der Mannigfaltigkeit berechnet, der den kleinsten Abstand zu $G(X)$ hat. Z steht hier für die zugehörige Pose des Punktes. Ist der der Abstand zu $G(X)$ kleiner als ein bestimmter Schwellwert T, wird angenommen, dass X ein Gesicht mit der Pose Z zeigt. Andernfalls wird angenommen, dass X kein Gesicht darstellt.

In den folgenden Abschnitten wird auf wichtige Details des hier skizzierten Verfahrens eingegangen. Abschnitt 3.2 erläutert die Parametrisierung der Mannigfaltigkeit, während Abschnitt 3.3 die Zielvorgaben des Trainings genauer formuliert. Für das Mapping wird ein Faltungsnetz eingesetzt. Faltungsnetze werden später in Kapitel 5 behandelt.

3.2 Mannigfaltigkeit der Gesichtsposen

In diesem Abschnitt wird beschrieben, wie die Mannigfaltigkeit der Gesichtsposen parametrisiert wird und wie die zugehörige Abbildungsvorschrift aussieht. Ein klassisches Beispiel für eine Mannigfaltigkeit ist eine Weltkarte, die mit Hilfe einer bestimmten Vorschrift die Erdkugel abbildet. Jeder Punkt auf der Karte, angegeben durch eine horizontale und vertikale Koordinate, entspricht einem Punkt auf der Erdkugel, der wiederum durch einen Längen- und Breitengrad angegeben werden kann. Des Weiteren gilt, dass benachbarte Punkte auf der Weltkarte auch auf der Erdkugel benachbart sind. Das gleiche Vorgehen wird nun angewendet, um eine Pose mit euklidischen Koordinaten, anstelle von Winkeln angeben zu können. Die Pose eines Objekts kann allgemein durch drei Winkel vollständig beschrieben werden. Zunächst wird das Vorgehen bei der Abbildung eines einzigen Winkels Θ erläutert. Anschließend wird gezeigt, wie eine Erweiterung auf zwei bzw. drei Winkel erfolgt. Wie Abbildung 3.2 zeigt, werden zur Abbildung von Θ kosinusförmige Basisfunktionen (hier $F_1(\Theta)$ bis $F_3(\Theta)$) verwendet. Da lediglich der vordere Teil des Kopfes betrachtet werden soll, sind diese Funktionen nur im Bereich $-\frac{\pi}{2} \leq \Theta \leq \frac{\pi}{2}$ definiert (vgl. Abschnitt 7.1.1). Des Weiteren sind sie entsprechend Gleichung 3.1 gegeneinander phasenverschoben, um die Maxima über den Definitionsbereich zu verteilen (vgl. Abbildung 3.2 a).

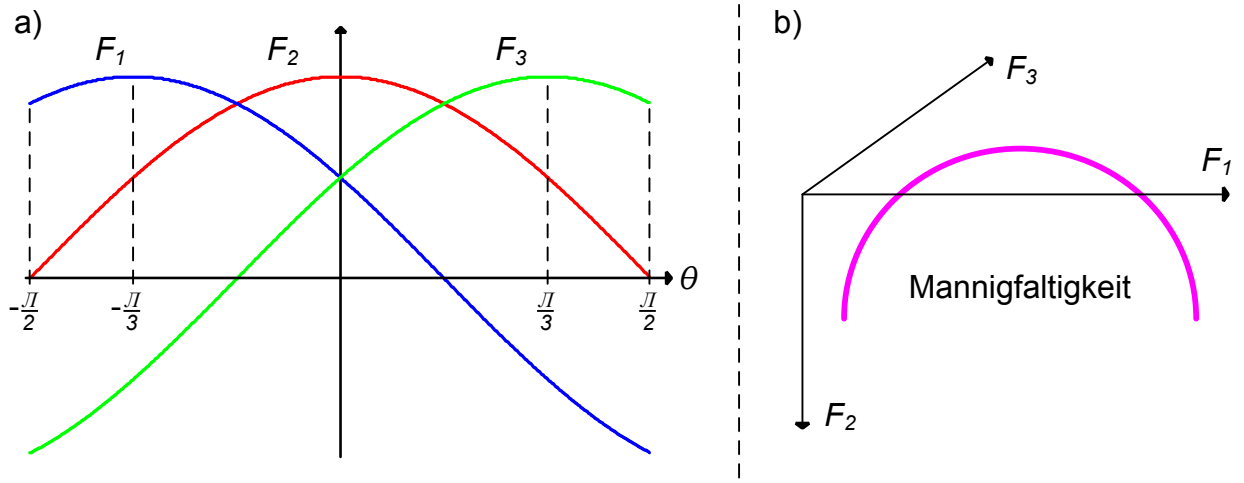

Abbildung 3.2: Parametrisierung mit einem Winkel nach [Osadchy u. a. 2007].

$$F_i(\Theta) = \cos(\Theta - \alpha_i); \quad i = 1, 2, 3; \quad \Theta = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]; \quad \alpha = \left\{-\frac{\pi}{3}, 0, \frac{\pi}{3}\right\} \tag{3.1}$$

Wie Abbildung 3.2 b) zeigt, entspricht jede Phase einer Achse des euklidischen Raumes. Die Mannigfaltigkeit ergibt sich nun, indem für jeden Wert für Θ der zugehörige Punkt $F(\Theta) = (F_1(\Theta), F_2(\Theta), F_3(\Theta))$ eingetragen wird. Es werden drei anstatt nur einer Funktion zur Abbildung des Winkels verwendet, um die Dimension des euklidischen Raumes zu erhöhen. Wird während des Trainings versucht, das Mapping durch Parametervariationen zu optimieren, ist bei einer höheren Dimension die Wahrscheinlichkeit geringer, bei einem lokalen Minimum zu verweilen, da die Chance erhöht wird, den Bereich um das Minimum herum über eine der zusätzlichen Achsen verlassen zu können (vgl. auch [LeCun u. a. 1998b]).

Von entscheidender Bedeutung ist, dass der Punkt auf der Mannigfaltigkeit mit dem kleinsten Abstand zum Mapping-Punkt $G(X)$ analytisch bestimmt werden kann. Hierzu wird zunächst mit Gleichung 3.2 die geschätzte Pose $\overline{\Theta}$ aus $G(X)$ extrahiert.

$$\overline{\Theta} = \arctan \frac{\sum_{i=1}^{3} G_i(X) \cos(\alpha_i)}{\sum_{i=1}^{3} G_i(X) \sin(\alpha_i)} \tag{3.2}$$

Der gesuchte Punkt auf der Mannigfaltigkeit ist nun $F(\overline{\Theta})$. Entsprechend den Ausführungen des letzten Abschnitts 3.1 kann das Label nun mit $\left\| G(X) - F(\overline{\Theta}) \right\|$ berechnet werden.

Um die oben aufgeführte Vorgehensweise auf einen zweiten Winkel Φ zu erweitern, werden Produktbildungen zwischen den Basisfunktionen von Θ und Φ durchgeführt. Die neuen Basisfunktionen Ergeben sich dann entsprechend Gleichung 3.3.

$$F_i(\Theta, \Phi) = \cos(\Theta - \alpha_i) \cos(\Phi - \beta_j); \quad i, j = 1, 2, 3 \tag{3.3}$$

Θ und Φ sollen im gleichen Winkelbereich definiert sein, d.h. es gilt $\beta_i = \alpha_i$. Gleichung 3.2 erweitert sich nun zu den Gleichungen 3.4 und 3.5.

$$\overline{\Theta} = \frac{1}{2} \left(\text{atan2}(cs + sc, cc - ss) + \text{atan2}(sc - cs, cc + ss) \right) \tag{3.4}$$

$$\overline{\Phi} = \frac{1}{2} \left(\text{atan2}(cs + sc, cc - ss) - \text{atan2}(sc - cs, cc + ss) \right) \tag{3.5}$$

Für die Symbole cs, sc, cc und ss gelten die Terme gemäß den Gleichungen 3.6 bis 3.9.

$$cc = \sum_{i,j} G_{i,j}(X)\cos(\alpha_i)\cos(\beta_j) \tag{3.6}$$

$$cs = \sum_{i,j} G_{i,j}(X)\cos(\alpha_i)\sin(\beta_j) \tag{3.7}$$

$$sc = \sum_{i,j} G_{i,j}(X)\sin(\alpha_i)\cos(\beta_j) \tag{3.8}$$

$$ss = \sum_{i,j} G_{i,j}(X)\sin(\alpha_i)\sin(\beta_j) \tag{3.9}$$

Der euklidische Raum erweitert sich also bei zwei Winkeln auf neun Dimensionen. Eine Erweiterung auf drei Winkel erfolgt analog mit 27 Dimensionen.

3.3 Detektion und Training

Das eingangs in Abschnitt 3.1 skizzierte Detektionsverfahren wird in diesem Abschnitt formal beschrieben. Aufbauend auf dieser Formalisierung erfolgt dann die Beschreibung der Zielfunktion, die während des Trainings optimiert werden soll.

Die Detektion soll so formuliert werden, dass die Pose und das Label ermittelt werden können, indem eine bestimmte Abstands- bzw. Bewertungsfunktion minimiert wird. In [Osadchy u. a. 2007] wird hierfür die Bezeichnung „Energiefunktion“ gewählt (vgl. auch [LeCun u. a. 2006]). Die Ausgabe dieser Funktion wird entsprechend als „Energie“ bezeichnet. Diese Terminologie wird hier aus Gründen der Einheitlichkeit übernommen. Aufbauend auf den Ausführungen der vorangegangenen Abschnitte zeigt Abbildung 3.3 das Schema der Energieminimierung für das hier betrachtete Verfahren. Die zugehörige Energiefunktion zeigt Gleichung 3.10.

$$E_W(Y, Z, X) = Y \left\| G_W(X) - F(Z) \right\| + (1 - Y)T \tag{3.10}$$

Z steht hier für die geschätzte Pose, X für das betrachtete Bild und W repräsentiert die aktuellen Parametersatz, der das Verhalten der Mapping-Funktion bestimmt. Y bezeichnet das Label und kann die Werte 0 und 1 annehmen. Für $Y = 1$ wird angenommen, dass X ein Gesicht zeigt, und der Energiewert wird durch die in den letzten Abschnitten erläuterte Abstandsberechnung bestimmt. Für Nicht-Gesichter ($Y = 0$) ist die Energie stets gleich einem Schwellwert T. Bei der Detektion eines Bildes X sind nun entsprechend Gleichung 3.11 die Parameter Y und Z so zu wählen, dass die Energie minimiert wird.

$$(\overline{Y}, \overline{Z}) = \text{argmin}_{Y,Z} E_W(Y, Z, X) \tag{3.11}$$

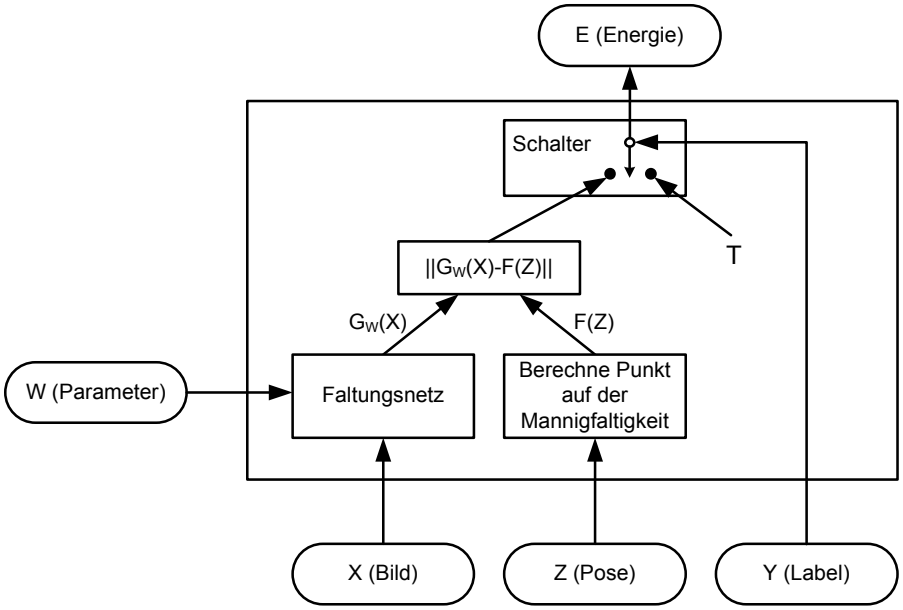

Abbildung 3.3: Schematischer Ablauf der Energieminimierung nach [Osadchy u. a. 2007].

In der Praxis erfolgt die Minimierung durch die im letzten Abschnitt 3.2 erläuterte analytische Berechnung des kürzesten Abstands zur Mannigfaltigkeit und einem anschließenden Vergleich mit dem Schwellwert T. Während des Trainings gilt es, die Parameter aus W so zu optimieren, dass eine Ziel- bzw. Trainingsfunktion $\mathcal{L}$ einen möglichst kleinen Wert annimmt. In [LeCun u. a. 2006] wird diese Funktion als Loss-Funktion („loss“ engl. für „Verlust“) bezeichnet. Gleichung 3.12 zeigt die Loss-Funktion für den hier betrachteten Anwendungsfall.

$$\mathcal{L}(W, \mathcal{S}) = \frac{1}{|\mathcal{S}_1|} \sum_{P \in \mathcal{S}_1} L_1(W, Z_P, X_P) + \frac{1}{|\mathcal{S}_0|} \sum_{P \in \mathcal{S}_0} L_0(W, Z_P, X_P) \tag{3.12}$$

Die Funktion wird über die Menge aller Trainingsbeispiele $\mathcal{S}$ gebildet, wobei die Menge der positiven Beispiele $\mathcal{S}_1$ und die Menge der negativen Beispiele $\mathcal{S}_0$ separat betrachtet werden. L_1 bzw. L_0 formulieren nun die Anforderungen, die das Mapping für jedes positive bzw. negative Trainingsbeispiel möglichst gut erfüllen soll. Für ein positives Beispiel $P = (Y_P = 1, Z_P, X_P)$ wird hierfür zunächst die Bedingung nach Gleichung 3.13 aufgestellt.

$$E_W(Y_P = 1, Z_P, X_P) < E_W(Y, Z, X_P) \quad \text{für} \quad Y \neq Y_P \quad \text{oder} \quad Z \neq Z_P \tag{3.13}$$

Sie besagt, dass der Energiewert für die korrekten Parameter kleiner sein muss, als bei allen anderen möglichen Eingabekombinationen. Dies ist gleichbedeutend mit den Bedingungen nach Gleichung 3.14.

$$E_W(1, Z_P, X_P) < T \quad \text{und} \quad E_W(1, Z_P, X_P) < \min_{Z \neq Z_P} E_W(1, Z, X_P) \tag{3.14}$$

Sie fordern, dass der Energiewert in jedem Fall kleiner sein muss als T. Des Weiteren muss die korrekte Pose im Vergleich zu anderen Möglichkeiten mit positvem Label ($Y = 1$) den kleinsten Energiewert liefern. Zur Erfüllung dieser Bedingungen wird L_1 für das Training entsprechend Gleichung 3.15 gewählt.

$$L_1(W, Z, X) = E_W(1, Z, X)^2 \tag{3.15}$$

Formal entspricht dies dem Problem der Minimierung eines quadratischen Fehlers (vgl. Kapitel 4.3.1).

Für ein negatives Trainingsbeispiel $P = (Y_P = 0, X_P)$ besteht im Umkehrschluss die Forderung, dass gemäß Gleichung 3.16 für die inkorrekte Annahme $Y = 1$ in jedem Fall ein Energiewert größer als T ausgegeben werden muss, damit das korrekte Label $Y = 0$ den kleinsten Energiewert ($= T$) erzeugt.

$$E_W(1, Z, X_P) > T \quad \forall Z \tag{3.16}$$

Es genügt dabei wieder, die Betrachtung auf die analytisch zu ermittelnde Pose $\overline{Z}$ (gem. Gl. 3.2 bzw. 3.4 und 3.5) zu reduzieren. Diese liefert per Definition den kleinsten Energiewert, da der zugehörige Punkt auf der Mannigfaltigkeit den kürzesten Abstand zum gemappten Punkt hat. Damit ändert sich die Bedingung entsprechend Gleichung 3.17.

$$E_W(1, \overline{Z}, X_P) > T \quad \text{mit} \quad \overline{Z} = \text{argmin}_z E_W(1, z, X_P) \tag{3.17}$$

Zur Erfüllung dieser Bedingung wird L_0 gemäß Gleichung 3.18 gewählt.

$$L_0(W, X_P) = K \exp\left(-E_W(1, \overline{Z}, X_P)\right) \tag{3.18}$$

K ist hier eine positive Konstante. Das negative Vorzeichen des Exponenten sorgt dafür, dass die Ausgabe von $\mathcal{L}$ umso kleiner wird, je weiter der gemappte Punkt von der Mannigfaltigkeit entfernt ist.

3.4 Zusammenfassung

In diesem Kapitel wurde eine Übersicht zu dem in dieser Arbeit untersuchten Detektionsverfahren gegeben. Dabei wurde insbesondere erläutert, wie die Posenschätzung in den Detektionsprozess miteinbezogen wird. Das Grundkonzept besteht darin, eine Mannigfaltigkeit und ein Mapping-Modul zu verwenden, um das Label und die Pose in einem euklidischen Raum gleichzeitig beschreiben zu können. Für ein tiefergehendes Verständnis wurden anschließend die einzelnen Aspekte des Verfahrens im Detail beschrieben. Es wurde gezeigt, wie die Mannigfaltigkeit mit Hilfe von kosinusförmigen Basisfunktionen parametrisiert wird.

Der entscheidende Vorteil dieser Vorgehensweise ist die Möglichkeit, Abstandberechnungen analytisch durchführen zu können. Aufbauend auf der Parametrisierung der Mannigfaltigkeit wurde erläutert, wie der Detektionsprozess mit Hilfe einer Energiefunktion formal beschrieben werden kann. Diese Formalisierung wird wiederum benötigt, um die Trainingsfunktion zur Optimierung des Mappings aufzustellen. In den späteren Kapiteln wird auf diese Funktion zurückgegriffen, wenn es darum geht, das Mapping mit Hilfe eines Faltungsnetzes zu realisieren.

Kapitel 4

Künstliche neuronale Netze

Künstliche neuronale Netze werden im Bereich der künstlichen Intelligenz eingesetzt. Sie bestehen aus einer Vernetzung von künstlichen Neuronen. Ihr Aufbau und ihre Funktionsweise ist von biologischen Nervensystemen inspiriert worden. Der Begriff „künstlich“, der im Folgenden der Einfachheit halber entfällt, stellt also eine Abgrenzung zu biologischen neuronalen Netzen dar. Künstliche Neuronen sind Funktionen, die mehrere Eingabewerte auf einen Ausgabewert abbilden. Diese Funktionen sind im Allgemeinen sehr einfach gestaltet, jedoch können durch die Vernetzung einer Vielzahl von Neuronen sehr komplexe Funktionen approximiert werden. Ebenso ist es auf diese Weise möglich, intelligentes Verhalten zu simulieren. Ein Neuron verfügt hierzu über eine gewisse Anzahl von frei einstellbaren Parametern, durch die sein Verhalten bestimmt werden kann. Für eine konkrete Problemstellung wird dann ein bestimmtes Optimierungsverfahren eingesetzt, das geeignete Werte für diese Parameter ermittelt. Der nächste Abschnitt 4.1 beschreibt zunächst den Aufbau eines Neurons. Danach wird in Abschnitt 4.2 die Vernetzung von Neuronen betrachtet. In Abschnitt 4.3 wird das Backpropagation-Verfahren mit einigen Erweiterungen beschrieben. Es handelt sich hierbei um ein Optimierungs- bzw. Trainingsverfahren für neuronale Netze. Die Betrachtungen dieses Kapitels beschränken sich lediglich auf Strukturen und Methoden, wie sie in dieser Arbeit verwendet wurden. Für ein weitergehendes Verständnis sei an dieser Stelle auf einschlägige Literatur verwiesen (bspw. [Bishop 1995]).

4.1 Künstliche Neuronen

Abbildung 4.1 zeigt den Aufbau eines Neurons nach [Rosenblatt 1958]. Ein Neuron hat n Eingabewerte und einen Ausgabewert. Die Eingabewerte x_0 bis x_{n-1} werden mit den Parametern w_{0j} bis $w_{(n-1)j}$ gewichtet und anschließend aufsummiert. Zusätzlich wird eine Konstante, die durch den Parameter b_j festgelegt wird, hinzuaddiert. Das Ergebnis wird einer sog. Aktivierungsfunktion $g(a_j)$ übergeben, die die Ausgabe des Neurons y_j bestimmt. Netze, die

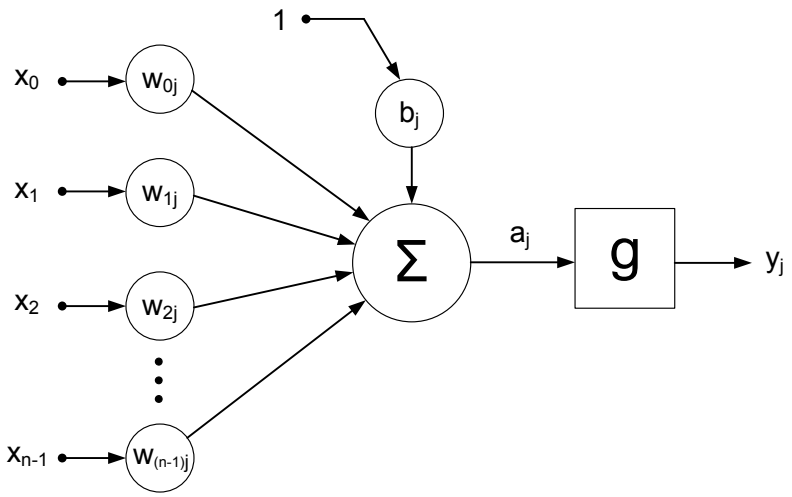

Abbildung 4.1: Künstliches Neuron nach [Rosenblatt 1958].

aus Neuronen dieser Art bestehen, werden in der Literatur als Perzeptrons bezeichnet. Für die Aktivierungsfunktion $g(a_j)$ wird im einfachsten Fall eine Schwellwertfunktion verwendet. Die Ausgabe kann in dem Fall nur zwei Werte annehmen. Wenn die Summe der gewichteten Eingaben einen bestimmten Wert überschreitet, „feuert“ das Neuron. Schwellwertfunktionen sind technisch einfach zu realisieren, jedoch sind sie auf Grund ihrer Unstetigkeit im Sprungpunkt nicht durchgängig differenzierbar. Dies macht sie für Optimierungsverfahren auf Basis eines Gradientenabstiegs (vgl. Abschnitt 4.3) ungeeignet. Eine Alternative bieten sogenannte Sigmoidfunktionen (sigmoid ≅ s-förmig). Gleichung 4.1 ist ein Beispiel für eine solche Funktion. Abbildung 4.2 zeigt den zugehörigen Graphen für $A = 1,7159$ und $S = {}^2/_3$ (so verwendet in [LeCun u. a. 1998a]). Die Funktion konvergiert in beide Richtungen gegen A bzw. $-A$. Gleichung 4.2 zeigt die erste Ableitung von $g(a_j)$, die sich praktischer Weise durch das Quadrat der Stammfunktion ausdrücken lässt.

$$g(a_j) = A \cdot tanh(B \cdot a_j) \tag{4.1}$$

Der Parameter b_j wird als Bias bezeichnet („to bias“ engl. für „verzerren“). Er wirkt sich konstant auf das Ausgabeverhalten eines Neurons aus. Erst durch diese „Verzerrung“ wird es möglich, mit neuronalen Netzen nichtlineare Funktionen abzubilden.

$$\dot{g}(a_j) = \frac{B}{A} \cdot (A^2 - g^2(a_j)) \tag{4.2}$$

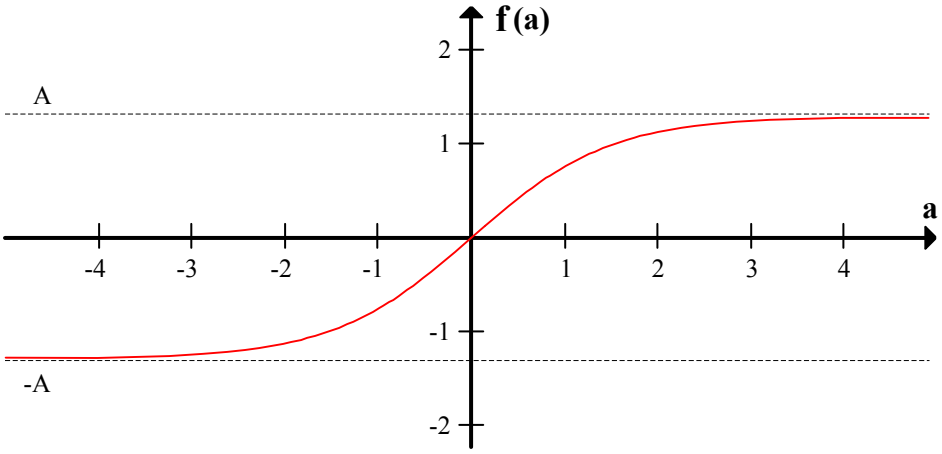

Abbildung 4.2: Beispiel für eine Sigmoidfunktion.

4.2 Netztopologie

Neuronale Netze können anhand ihrer Netztopologie in verschiedene Klassen mit unterschiedlicher Komplexität eingeteilt werden. In diesem Abschnitt werden diesbezüglich einige wichtige Unterscheidungen erläutert.

Ein neuronales Netz kann übersichtlicht als gerichteter Graph dargestellt werden, wobei die Neuronen durch Knoten und die Verbindungen zwischen diesen durch Kanten repräsentiert werden. Bei der Klasse der vorwärtsgerichteten Netze (engl.: Feed Forward Networks) wird bei diesen Graphen auf Zyklen und Schleifen verzichtet. Auf diese Weise wird die Komplexität, aufgrund fehlender Rekursionen, stark beschränkt. Eine weitere Vereinfachung besteht darin, die Neuronen in Schichten einzuteilen, wobei Verbindungen nur zwischen Neuronen benachbarter Schichten erlaubt sind. Die erste Schicht erhält dann die Netzeingaben und wird entsprechend als Eingabeschicht bezeichnet. Die letzte Schicht bildet den Ausgabevektor und wird folglich Ausgabeschicht genannt. Die übrigen, mittleren Schichten werden als versteckte Schichten bezeichnet. Der Eingabevektor wird nun sukzessiv über mehrere Schichten zum Ausgabevektor transformiert. Werden neuronale Netze zur Mustererkennung eingesetzt, gibt der Ausgabevektor dann Aufschluss darüber, ob in der Eingabe ein bestimmtes Muster erkannt wurde.

Abbildung 4.3 zeigt ein einfaches Beispiel für ein vorwärtsgerichtetes Netz mit vier Schichten, zwei Eingabewerten und einem Ausgabewert. Die Parameter der Neuronen sind als Kantengewichte eingetragen, wobei für die Biasparameter zusätzliche Kanten mit einem konstanten Eingabewert von Eins eingetragen wurden. Auch bei Faltungsnetze, die im nächsten Kapitel behandelt werden, handelt es sich um mehrschichtige, vorwärtsgerichtete Netze dieser Art.

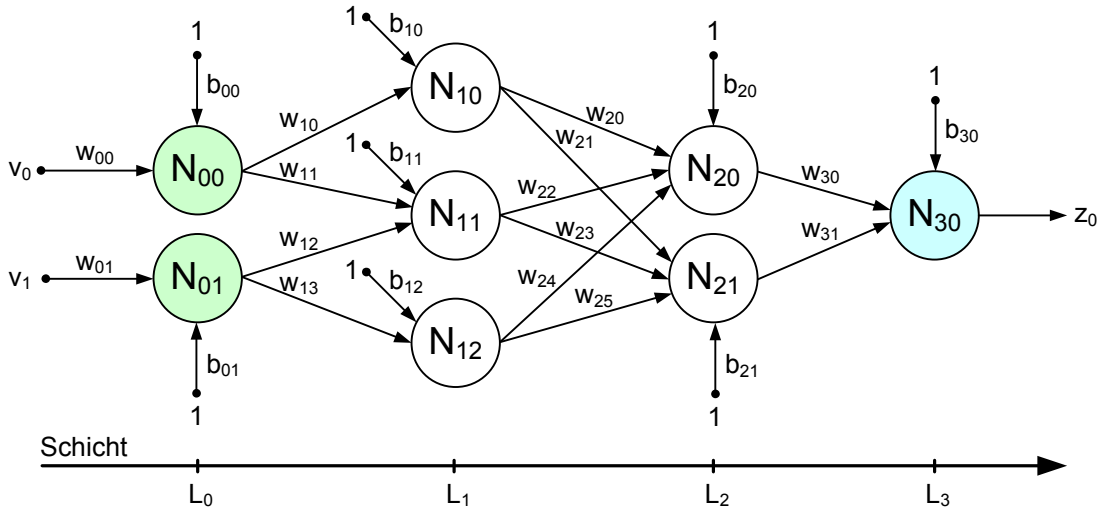

Abbildung 4.3: Beispiel für ein vorwärtsgerichtetes neuronales Netz. Grün: Eingabeschicht, Weiß: versteckte Schichten, Blau: Ausgabeschicht.

4.3 Backpropagation-Verfahren

Das Backpropagation-Verfahren ist ein Trainingsverfahren für neuronale Netze. Es handelt sich dabei um einen Spezialfall des allgemeineren Gradientenabstiegsverfahrens. Abschnitt 4.3.1 erläutert die Grundlagen dieses Verfahrens. Die Abschnitte 4.3.2 und 4.3.3 gehen auf Erweiterungen ein, die für diese Arbeit von Bedeutung sind.

4.3.1 Grundlagen

Das Backpropagation-Verfahren benutzt das Prinzip der Fehlerrückführung um einen Quadratischen Fehler nach Gleichung 4.3 zu minimieren. Hier bezeichnet $\mathcal{S}$ die Menge der Trainingsbeispiele, o_P die Ist-Ausgabe (engl. output) und t_P die Soll-Ausgabe (engl. teacher) des Netzes für das jeweilige Trainingsbeispiel $P \in \mathcal{S}$.

$$E = \frac{1}{2} \sum_{P \in \mathcal{S}} (t_P - o_P)^2 \tag{4.3}$$

Zunächst werden die Parameter mit zufälligen Werten in einer sinnvollen Größenordnung initialisiert. Das Verfahren führt dann nacheinander für jedes Trainingsbeispiel folgende Schritte durch:

1. Das Eingabemuster, d.h. das Trainingsbild, wird an den Eingang des Netzes gelegt, um die Ist-Ausgabe zu ermitteln.

2. Aus der Differenz zwischen der Ist- und der Soll-Ausgabe wird der Fehler bestimmt.

3. Der Fehler wird rückwärts durch das Netz propagiert. Durch schrittweises Bilden der partiellen Ableitungen δ_i wird dabei der Anteil jedes Neurons am Fehler ermittelt.

4. Mit einer bestimmten Lernrate η werden die Gewichte w_{ij} der Neuronen entsprechend ihrer Fehleranteile angepasst.

Bei der Rückpropagierung werden zunächst die partiellen Ableitungen der Ausgabeneuronen entsprechend Gleichung 4.4 bestimmt. Die Ableitungen der übrigen Neuronen werden durch Anwendung der Kettenregel nach Gleichung 4.5 ermittelt. Hierbei ist k der Index für die Neuronen der vorangehenden Schicht, die mit dem Neuron j verbunden sind. Entsprechend bezeichnet w_{kj} das zugehörige Gewicht der jeweiligen Verbindung.

$$\delta_j = t_j - o_j \tag{4.4}$$

$$\delta_j = \dot{g}(a_j) \sum_k \delta_k w_{kj} \tag{4.5}$$

Die Anpassung der Gewichte erfolgt schließlich nach Gleichung 4.6. E_P bezeichnet hier den Fehler für ein einzelnes Trainingsbeispiel $P \in \mathcal{S}$ und x_i den ursprünglichen Eingabewert der Verbindung, der bei der Berechnung der Ist-Ausgabe ermittelt wurde. Die gesamte Trainingsmenge wird mehrfach durchlaufen, um ein besseres Trainingsergebnis erzielen zu können.

$$\Delta w_{ij} = -\eta \frac{\partial E_P}{\partial w_{ij}} = \eta \delta_j x_i \tag{4.6}$$

4.3.2 Hesse-Diagonale

In diesem Abschnitt wird beschrieben, wie die Konvergenzgeschwindigkeit durch Bildung individueller Lernraten für jedes Gewicht erheblich beschleunigt werden kann.

Um eine schnelle Konvergenz zu erreichen, ist die Wahl der richtigen Lernrate entscheidend. Abbildung 4.4 zeigt den Zusammenhang für den einfachen Fall einer quadratischen Fehlerfunktion E, die lediglich von einem einzigen Parameter w abhängt. Ist die gewählte Lernrate η kleiner als die optimale Lernrate η_{opt}, bedarf es mehrere Durchläufe (Teil a), während bei einer optimalen Lernrate lediglich ein Durchlauf erforderlich ist (Teil b). Bei der Wahl einer zu großen Lernrate wiederum, oszilliert w um den optimalen Wert w_{min} (Teil c). Für $\eta > 2\eta_{opt}$ divergiert w dabei (Teil d). Die optimale Lernrate lässt sich nach Gleichung 4.7 berechnen (siehe [LeCun u. a. 1998b]). Hier bezeichnet w_c den aktuellen Parameterwert.

$$\eta_{opt} = \left(\frac{d^2 E(w_c)}{dw^2} \right)^{-1} \tag{4.7}$$

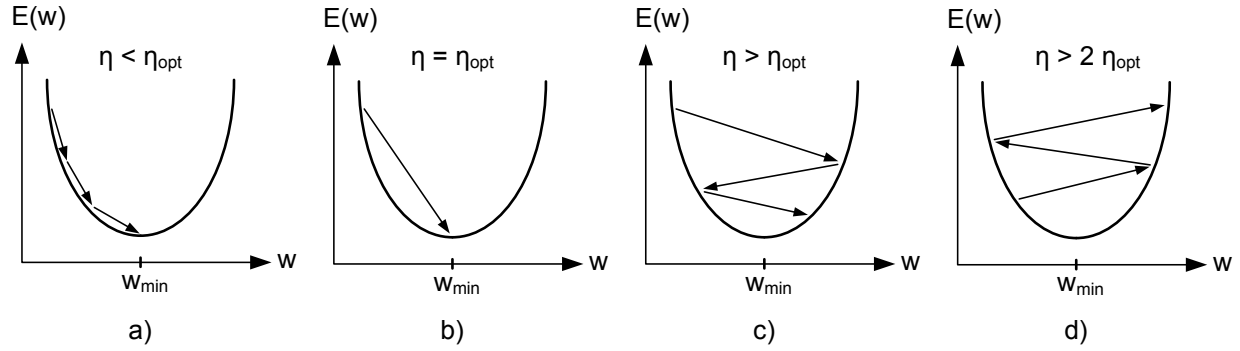

Abbildung 4.4: Zusammenhang zwischen der Lernrate und der Konvergenzgeschwindigkeit nach [LeCun u. a. 1998b].

Bei mehr als einem Parameter erweitert sich der rechte Teil von Gleichung 4.7 zu einer Matrix in der Form H^{-1}. H ist die sogenannte Hesse-Matrix, deren Komponenten sich über der Trainingsmenge entsprechend Gleichung 4.8 berechnen lassen. N ist hierbei die Anzahl der Trainingsbeispiele in $\mathcal{S}$.

$$h_{ij} = \frac{1}{N} \sum_{P \in \mathcal{S}} \frac{\partial^2 E_P}{\partial w_i \partial w_j} \tag{4.8}$$

Mit Hilfe der Hesse-Matrix lässt sich für jedes Gewicht eines neuronalen Netzes eine individuelle Lernrate bestimmen, die nicht nur den Anteil am Fehler berücksichtigt, sondern auch, wie stark sich die Änderung eines Parameters auf die Netzausgabe auswirkt. Das Problem ist jedoch, dass die Komplexität bei der Berechnung der Hesse-Matrix quadratisch mit der Anzahl der Gewichte steigt. Eine Vereinfachung besteht darin, nur die Elemente der Hauptdiagonalen h_{kk} zu bestimmen. Die Anpassung eines Gewichts w_k erfolgt dann nach der Vorschrift gem. Gleichung 4.9.

$$\Delta w_k = -\frac{\eta}{\mu + h_{kk}} \cdot \frac{\partial E_P}{\partial w_k} \tag{4.9}$$

Hier bezeichnet η eine globale Konstante, die für alle Gewichte gleich ist. Die Konstante μ soll verhindern, dass Δw_k bei sehr kleinem h_{kk} zu groß wird. In der Praxis erfolgt die Berechnung der zweiten Ableitung bezüglich eines Gewichts $w_{ij} = w_k$ durch eine modifizierten Form des Backpropagation-Verfahrens. Analog zu Gleichung 4.5 wird mit Hilfe einer abgewandelten Vorschrift nach Gleichung 4.10 eine Näherungslösung bestimmt.

$$\frac{\partial^2 E_P}{\partial w_{ij}^2} x_j^2 = \frac{\partial^2 E_P}{\partial a_i^2} = \dot{g}^2(a_j) \sum_k w_{kj}^2 \frac{\partial^2 E_P}{\partial a_k^2} x_j^2 \tag{4.10}$$

Die individuellen Lernraten werden in regelmäßigen Abständen neu berechnet, bspw. vor jedem neuen Durchlauf durch die Trainingsmenge. Eine weitere Vereinfachung kann darin

bestehen, die Elemente h_{kk} nicht jedes Mal über die gesamte Trainingsmenge zu berechnen, sondern nur über eine zufällig gebildete Teilmenge.

4.3.3 Geteilte Gewichte

Mit der Formulierung „geteilte Gewichte“ wird eine Vorgehensweise bezeichnet, bei der zwei oder mehr Neuronen die selben Gewichte verwenden. Bei Faltungsnetzen wird von dieser Möglichkeit Gebrauch gemacht, um trotz einer hohen Anzahl von Neuronen mit relativ wenigen Parametern auszukommen (vgl. Kapitel 5.2). Um geteilte Gewichte verwenden zu können, müssen Anpassungen beim Backpropagation-Verfahren vorgenommen werden, da die Gewichte eines Neurons nicht mehr unabhängig modifiziert werden können, ohne dabei das Verhalten anderer Neuronen zu verändern. Die diesbezüglichen Besonderheiten werden im Folgenden aufgezeigt. Zunächst wird für jedes Gewicht w_k eine Menge V_k definiert, die sich aus einer Anzahl von Wertepaaren (i, j) zusammensetzt. Jedes Wertepaar adressiert zwei Neuronen, zwischen denen eine Verbindung besteht. Das Gewicht dieser Verbindung sei u_{ij}. Es gilt nun Gleichung 4.11.

$$u_{ij} = w_k \quad \forall (i,j) \in V_k \tag{4.11}$$

Das bedeutet also, das V_k genau die Verbindungen adressiert, die sich das Gewicht w_k teilen. Die Anpassung der Gewichte nach Gleichung 4.6 ändert sich entsprechend Gleichung 4.12. Die partielle Ableitung bezüglich eines Parameters w_k ergibt sich nun durch die Aufsummierung der partiellen Ableitungen bezüglich der einzelnen u_{ij}.

$$\Delta w_{ij} = -\eta \sum_{(i,j)\in V_k} \frac{\partial E_P}{\partial u_{ij}} \tag{4.12}$$

Analog ändert sich die Berechnung der Hesse-Diagonalen entsprechend Gleichung 4.13.

$$h_{kk} = \frac{1}{N} \sum_{P\in\mathcal{S}} \sum_{(i,j)\in V_k} \frac{\partial^2 E_P}{\partial u_{ij}^2} \tag{4.13}$$

4.4 Zusammenfassung

Da Faltungsnetze eine Spezialform neuronaler Netze darstellen, wurden in diesem Kapitel die für diese Arbeit wichtigen Grundlagen neuronaler Netze aufgezeigt. Zunächst wurde der Aufbau neuronaler Netze erläutert. Dabei wurden insbesondere die Eigenschaften der für diese Arbeit relevanten Klasse der vorwärtsgerichteten, mehrschichtigen Perzeptrons aufgezeigt. Im Anschluss daran wurde das Backpropagation-Verfahren beschrieben, mit dem neuronale

Netze trainiert werden können. Als eine Erweiterung hierzu wurde gezeigt, wie für die einzelnen Netzparameter individuelle Lernraten bestimmt werden können. Von dieser Möglichkeit wird später noch im Rahmen der Untersuchung der Trainingsfortschritte Gebrauch gemacht (Kapitel 7). Die zweite Erweiterung, die hier betrachtet wurde, ist das Prinzip der geteilten Gewichte. Durch sie ist es möglich, trotz einer hohen Anzahl an Neuronen mit relativ wenig Gewichten auszukommen. Wie im nächsten Kapitel noch zu sehen sein wird, ist diese Erweiterung aufgrund der speziellen Struktur von Faltungsnetzen erforderlich.

Kapitel 5

Faltungsnetze

Faltungsnetze stellen eine Spezialform von neuronalen Netzen dar. Für Problemstellungen im Bereich der visuellen Mustererkennung wurde ein Faltungsnetz erstmals in [Vaillant u. a. 1993] eingesetzt. Die Eingabe des Netzes ist ein Graustufenbild in einer festgelegten Größe. Als Ausgabe liefert es einen Informationsvektor niedriger Dimension, der Aufschluss darüber gibt, ob im Eingabebild ein bestimmtes Muster erkannt wurde. Vereinfacht ausgedrückt besteht ein Faltungsnetz aus einer Vernetzung von Faltungsoperationen, die unterschiedliche Filtermasken verwenden. Ausgehend von dieser Betrachtung können Faltungsnetze zunächst losgelöst vom Hintergrund neuronaler Netze beschrieben werden. Eine entsprechende Beschreibung von Struktur und Funktionsweise erfolgt in Abschnitt 5.1. Abschnitt 5.2 erläutert anschließend, wie ein Faltungsnetz als neuronales Netz modelliert wird, um das im letzten Kapitel beschriebenen Trainingsverfahren anwenden zu können. In Abschnitt 5.3 wird schließlich aufbauend auf den Ausführung in Kapitel 3 die Verwendung eines Faltungsnetzes als Mapping-Modul beschrieben. Abschließend werden in Abschnitt 5.4 anhand einiger Beispiel die generischen Eigenschaften von Faltungsnetzen hervorgehoben, die sie zum universellen Werkzeug für die verschiedensten Problemstellungen der visuellen Mustererkennung machen.

5.1 Struktur und Funktionsweise

In diesem Abschnitt wird der Aufbau von Faltungsnetzen erläutert. Hierzu dient im Folgenden das in [Vaillant u. a. 1993] verwendete Netz als Beispiel (siehe Abbildung 5.1). Das Netz ist aus praktischen Gründen in mehreren Schichten S_1 bis S_5 unterteilt. Eine Schicht besteht aus mehreren Feldern gleicher Größe. Die Eingabe des Netzes ist ein Graustufenbild mit 20x20 Bildpunkten. Die Felder der Schichten S_1 bis S_4 stellen die Zwischenergebnisse innerhalb des Netzes dar. Die letzte Schicht S_5 entspricht dem Ausgabevektor, der in diesem einfachen Beispiel aus nur einem Wert besteht. In [Vaillant u. a. 1993] wird angenommen,

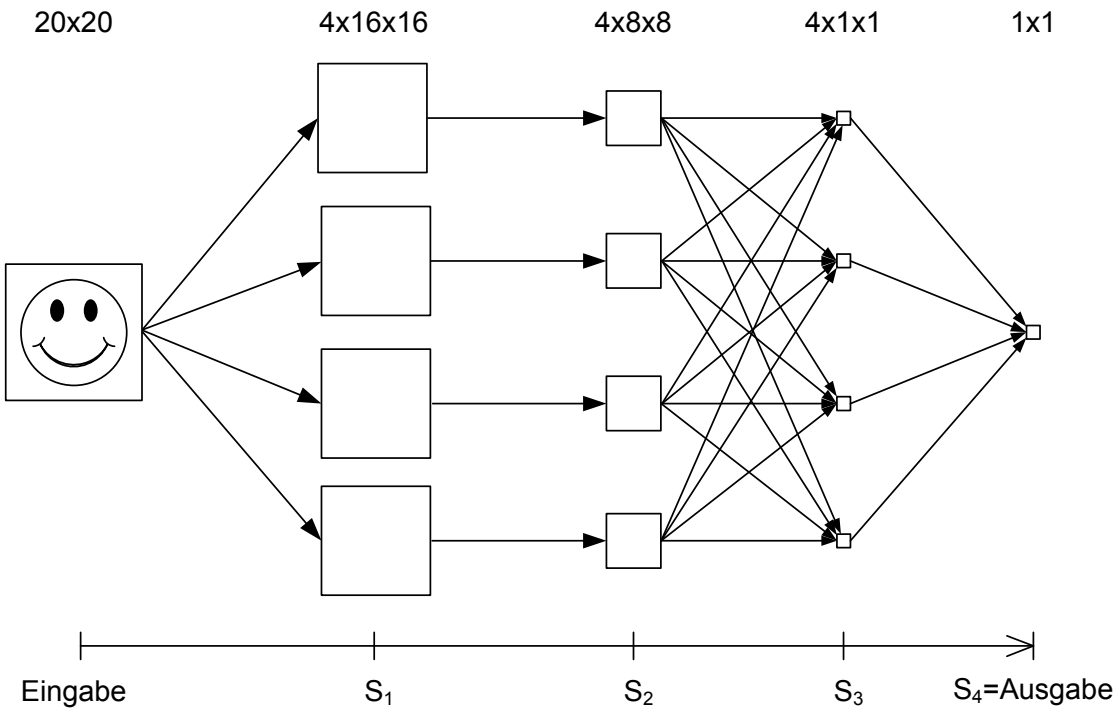

Abbildung 5.1: Faltungsnetz nach [Vaillant u. a. 1993].

dass das Bild ein Gesicht zeigt, wenn dieser Wert größer als Null ist. In den folgenden Teilabschnitten werden die Schritte beschrieben, die zwischen den einzelnen Schichten durchgeführt werden.

5.1.1 Faltung

In der ersten Schicht S_1 werden parallel vier Faltungen mit unterschiedlichen Filtermasken auf das Eingabebild angewendet. S_1 wird folglich als Faltungsschicht bezeichnet. Eine Filtermaske ist gem. Gleichung 5.1 ein endlicher, zweidimensionaler Koeffizientensatz. Hier steht x jeweils für die horizontale und y für die vertikale Bildkoordinate. N_{hx} und N_{hy} bezeichnen die Anzahl der Koeffizienten a_{xy} in horizontaler bzw. vertikaler Richtung. Gleichung 5.2 zeigt die Faltung eines Eingabebildes s mit einer Filtermaske h. Hier bezeichnet „$**$“ den zweidimensionalen Faltungsoperator und g das Ausgabebild.

$$h(x,y) = \begin{cases} 0 & für \quad x < -\lfloor \frac{(N_{hx}-1)}{2} \rfloor \vee y < -\lfloor \frac{(N_{hy}-1)}{2} \rfloor \\ 0 & für \quad x > \lfloor \frac{(N_{hx}-1)}{2} \rfloor \vee n_y > \lfloor \frac{(N_{hx-1})}{2} \rfloor \\ a_{xy} & sonst. \end{cases} \tag{5.1}$$

$$g(x,y) = s(x,y) * *h(x,y) = \sum_{m_x=-\infty}^{\infty} \sum_{m_y=-\infty}^{\infty} s(m_x, m_y) \cdot h(x - m_x, y - m_y) \tag{5.2}$$

Der Einfachheit halber wird bei den Faltungen keine Randbehandlung durchgeführt. Das bedeutet, es werden nur Ausgabewerte berechnet, bei denen sämtliche Filterkoeffizienten mit Bildpunkten der Eingabe multipliziert werden können. Abbildung 5.2 verdeutlicht dies anhand eines Beispiels. Hier wird ein 6x6-Feld der Schicht S_n mit einer 4x4-Filtermaske gefaltet und das Ergebnis in einem 3x3-Feld der Schicht S_{n+1} festgehalten. Beispielhaft sind die Berechnung von drei Ausgabewerten (a: links-oben, b: mitte-oben, c: rechts-unten) dargestellt. Aufgrund der fehlenden Randbehandlung, sind die Dimensionen des Ausgabebildes kleiner, als die der Eingabe. Für die Dimensionen der Ausgabe N_{gx} und N_{gy} gilt Gleichung 5.3. Hierbei sind N_{sx} und N_{sy} die Dimensionen des Eingabebildes. Im betrachteten Beispiel aus Abbildung 5.1 haben die vier Faltungsergebnisse der Schicht S_1 entsprechend die Größe 16x16.

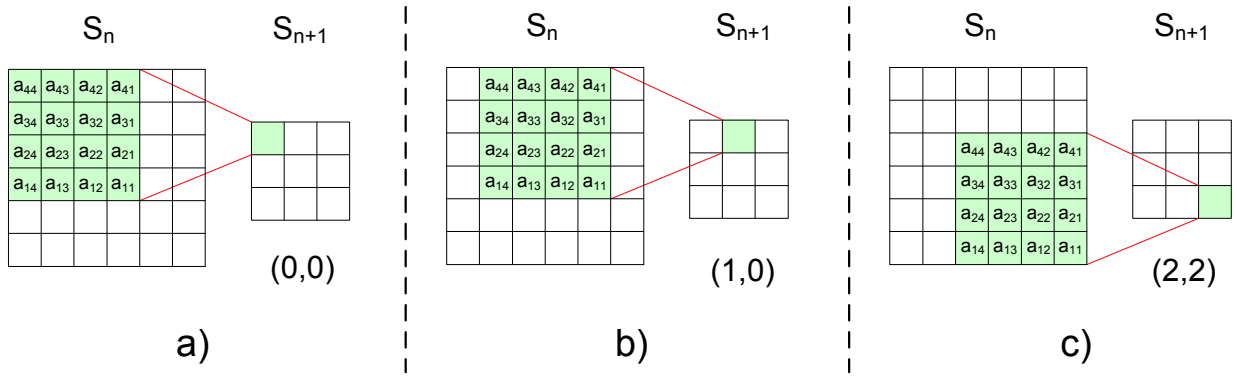

Abbildung 5.2: Beispiel für eine Faltung.

$$N_{gx} = N_{sx} - N_{hx} + 1; \qquad N_{gy} = N_{sy} - N_{hy} + 1 \tag{5.3}$$

Durch die Faltungen sollen die Signalanteile, die zur Beschreibung des Gesichts wichtig sind, hervorgehoben bzw. irrelevante Signalanteile unterdrückt werden. In diesem Sinne können Faltungen als nichtrekursive Filter aufgefasst werden. In der Signal- und Bildverarbeitung ist der Einsatz von Filtern zur Signalformung eine typische Vorgehensweise. Das gleiche Grundprinzip lässt sich auch unter Verwendung geeigneter Filtermasken auf Problemstellungen der Mustererkennung anwenden. Eine Faltung korrespondiert im Frequenzbereich mit einer Gewichtung der einzelnen Frequenzanteile mit unterschiedlichen Faktoren. Mit Faktoren größer als Eins werden so die relevanten Frequenzanteile hervorgehoben, während die übrigen Anteile mit Faktoren kleiner als Eins unterdrückt werden. Durch die Verwendung mehrer Filtermasken auf parallelen Pfade können dabei unterschiedliche Aspekte betrachtet werden.

5.1.2 Unterabtastung

Die Dimension des Ausgabevektors eines Faltungsnetzes (hier nur ein Wert) ist im Vergleich zu der Anzahl der Bildpunkte der Eingabe (hier vierhundert Werte) in der Regel gering. Es müssen also Vorkehrungen getroffen werden, um die Dimension der Eingabe zu reduzieren. In Faltungsnetzen geschieht dies durch Unterabtastungen. Im hier betrachteten Beispiel werden auf allen vier Pfaden Unterabtastungen im Übergang von Schicht S_1 nach Schicht S_2 durchgeführt. S_2 wird dem entsprechend als Unterabtastungsschicht bezeichnet. Eine schnelle und deshalb häufig angewendete Methode ist die bilineare Unterabtastung. Dabei werden die Pixel auf dem neuen, gröberen Bildraster durch die Mittelung der vier nächstliegenden Pixel des ursprünglichen, feineren Rasters berechnet. Bei Faltungsnetzen wird der Einfachheit halber stets ein Unterabtastungsfaktor von Zwei gewählt. Die Folgende Abbildung 5.3 verdeutlicht das Vorgehen anhand eines Beispiels. Hier wird ein 6x6-Feld der Schicht S_n unterabgetastet und das Ergebnis an die Schicht S_{n+1} weitergereicht. Beispielhaft sind die Berechnungen von drei Ausgabewerten (a: links-oben, b: mitte-oben, c: rechts-unten) dargestellt. Vier Pixel werden jeweils mit dem Faktor b_n gewichtet und addiert. Der Wert von b_n kann von 0.25 abweichen, um implizit den Kontrast des Bildes zu manipulieren. Bei der Unterabtastung ist es wichtig, dass möglichst viele Informationen erhalten bleiben, die für die Problemstellung relevant sind. Je mehr irrelevante Informationen zuvor durch eine vorangeschaltete Faltung herausgefiltert wurden, desto höher ist die Redundanz in der Datenmenge und umso weniger relevante Informationen gehen bei der Unterabtastung verloren.

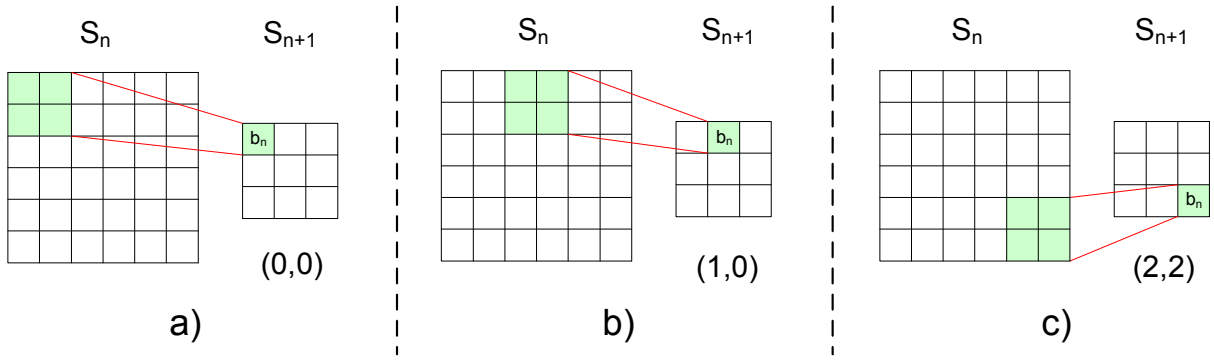

Abbildung 5.3: Beispiel für eine Unterabtastung.

5.1.3 Schwellwertfunktion und Gleichanteil

Zusätzlich zu den Unterabtastungen erfolgt eine weitere Informationsreduktion durch die Verwendung einer Schwellwertfunktion. Diese kommt jeweils am Ende einer Unterabtastungsschicht zum Einsatz. Die Pixel aller Felder werden dann auf einen von zwei möglichen Ausgabewerten abgebildet. Es wird bei allen Pixeln die gleiche Schwellwertfunktion verwendet. Jedoch lässt sich der Wertebereich durch eine konstante Manipulation des Gleichanteils

verschieben. Das bedeutet, dass zu jedem Pixel eines Feldes der selbe Wert hinzuaddiert wird. Erst danach durchlaufen die Pixel die Schwellwertfunktion. Die Möglichkeit, den Gleichanteil zu manipulieren besteht nicht nur bei den Feldern der Unterabtastungsschichten, sondern generell im gesamten Netz. Jedes Feld verfügt hierfür über einen zusätzlichen Parameter.

5.1.4 Überlagerung und volle Verbindungen

Als nächstens sei der Blick auf den Übergang von Schicht S_2 nach Schicht S_3 gerichtet. Hier steht jede Verbindung für eine Faltung mit einer 8x8-Filtermaske. Anders als bei den vorangegangenen Schichten, führen hier jeweils mehrere Verbindungen zum selben Zielfeld. In diesem Fall werden die Faltungsergebnisse überlagert. Das bedeutet, dass die Ergebniswerte der unterschiedlichen Faltungen, die bezüglich des Zielfeldes die gleichen Koordinaten haben, aufsummiert werden. Auf diese Weise können Zwischenergebnisse, die auf unterschiedlichen Pfaden ermittelt wurden miteinander kombiniert werden. Im Beispiel hat jedes Feld der Schicht S_2 Verbindungen zu allen Feldern der Schicht S_3. Des Weiteren haben die Filtermasken und die Felder der Schicht S_2 die gleiche Größe. Daraus resultiert, dass bei der Berechnung eines jeden Pixels der Schicht S_3 sämtliche Pixel der Schicht S_2 beteiligt sind. Der Übergang von S_2 nach S_3 wird entsprechend als „volle Verbindung“ bezeichnet. Beim letzten Übergang von S_3 nach S_4 handelt es sich ebenfalls um eine volle Verbindung. Es werden parallel vier Faltungen mit einer 1x1-Filtermaske durchgeführt (Filtermasken mit nur einem Koeffizienten entsprechen einer einfachen Faktorgewichtung). Durch Aufsummierung der vier Werte wird schließlich der (eindimensionale) Ergebnisvektor berechnet. Volle Verbindungen werden stets in den letzten Schichten eingesetzt, um sämtliche Teilergebnisse zu kombinieren und den Ausgabevektor zu bilden. Jedes Element des Ausgabevektors entspricht dabei einem 1x1-Feld.

5.1.5 Gesamtprozess

Faltungsnetze enthalten allgemein alle für eine Mustererkennung wichtigen Komponenten. Hinsichtlich eines Gesamtprozesses müssen die relevanten Informationen zunächst extrahiert und anschließend reduziert werden, um eine Klassifizierung durchführen zu können. Im Folgenden werden noch einmal zusammenfassend die einzelnen Faltungsnetzkomponenten hinsichtlich ihrer diesbezüglichen Funktion aufgeführt:

1. Extraktion - Die relevanten Informationen werden mit Hilfe von Filtermasken extrahiert. Dies geschieht parallel auf verschiedenen Pfaden, um unterschiedliche Aspekte betrachten zu können. Die verschiedenen Teilergebnisse werden später durch Überlagerungen miteinander kombiniert.

2. Reduktion - Die Reduktion der Informationen erfolgt explizit durch Unterabtastungen und implizit durch den Verzicht auf eine Randbehandlung bei den Faltungen. Eine weitere Informationsreduktion erfolgt durch den Einsatz einer Schwellwertfunktion jeweils am Ende einer Unterabtastungsschicht.

3. Klassifizierung - Der einfach auszuwertende Ausgabevektor wird in den letzten Schichten durch volle Verbindungen gebildet, wobei jeder Vektoreintrag einem 1x1-Feld entspricht.

5.2 Faltungsnetze als Neuronale Netze

In diesem Abschnitt wird beschrieben, wie geeignete Werte für die frei wählbaren Parameter der zuvor beschriebenen Faltungsnetzkomponenten ermittelt werden. Die Parameter setzen sich aus den Koeffizienten der Filtermasken a_{xy} entsprechend Gleichung 5.1, den bei den Unterabtastung verwendeten Gewichtungen b_n entsprechend Abbildung 5.3 und den Werten für die additiven Gleichanteile entsprechend Abschnitt 5.1.3 zusammen. Der Versuch, geeignete Parameterwerte manuell zu ermitteln, würde sich als äußerst schwieriges und langwieriges Unterfangen herausstellen, bei dem, aufgrund verschiedener erschwerender Faktoren (vgl. Kapitel 2.1), sehr viele Sonderfälle berücksichtigt und ausgiebig getestet werden müssten. Der in [Vaillant u. a. 1993] vorgestellt Ansatz, begreift Faltungsnetze als eine Spezialform neuronaler Netze. Auf diese Weise wird eine automatisierte Ermittelung der Parameter durch das Backpropagation-Verfahren ermöglicht. Im Folgenden wird diese Vorgehensweise im Detail beschrieben.

- Die Grundstruktur ist ein vorwärtsgerichtetes, mehrschichtiges neuronales Netz (vgl. Kapitel 4.2), wobei jede Schicht des Faltungsnetzes einer Schicht im neuronalen Netz entspricht.

- Die Felder der einzelnen Schichten werden durch Neuronen dargestellt. Jeder Pixel wird dabei durch ein Neuron repräsentiert.

- Eine Faltung wird durch Verbindungen zwischen den Neuronen zweier Felder modelliert, so dass die Kantengewichte der Neuronen genau den Filterkoeffizienten a_{xy} der zugehörigen Filtermaske entsprechen. Bei einer Faltung wird zur Berechnung eines jeden Pixels stets der gleiche Koeffizientensatz verwendet. Für das neuronale Netz bedeutet das, dass sich die Neuronen der selben Felder ihre Kantengewichte teilen. Dies muss entsprechend den Ausführungen in Abschnitt 4.3.3 berücksichtigt werden.

- Unterabtastungen werden ebenfalls durch Verbindungen zwischen Neuronen modelliert. Für jedes Feld wird hierbei jeweils nur ein Parameter benötigt, d.h. alle Kanten

des selben Feldes teilen sich das selbe Gewicht. Dieses Gewicht entspricht dem Parameter b_n nach Abbildung 5.3.

- Um volle Verbindungen nachzubilden, werden sämtliche Neuronen zweier Schichten miteinander verbunden. Jede Kante hat dabei ihr eigenes Gewicht.
- Die Schwellwertfunktion nach Abschnitt 5.1.3 entspricht gerade der Aktivierungsfunktion g eines Neurons (vgl. Abschnitt 4.1). Entsprechend wird hier die Sigmoidfunktion nach Gleichung 4.1 eingesetzt. Zu beachten ist hierbei, dass Schwellwertefunktionen in Faltungsnetzen nur bei Unterabtastungsschichten verwendet werden. Für die übrigen Schichte gilt deshalb $g(a_j) = a_j$ und $\dot{g}(a_j) = 1$.
- Der additive Gleichanteil entspricht den Biasparametern. Hierbei ist wieder zu beachten, dass sich die Neuronen des selben Feldes die gleichen Biasparameter teilen.

5.3 Mapping mit Faltungsnetzen

In diesem Abschnitt wird beschrieben, wie ein Faltungsnetz aufgebaut sein muss, um es entsprechend den Ausführungen in Kapitel 3 als Mapping-Modul zur Gesichtsdetektion unter Einbeziehung der Pose verwenden zu können. Als Beispiel wird hier das in [Osadchy u. a. 2005] und [Osadchy u. a. 2007] verwendete Netz herangezogen, welches in Abbildung 5.4 zu sehen ist. Eine vollständige Strukturbeschreibung diese Netzes findet sich in Anhang A (bezeichnet als „Faltungsnetz F_A“).

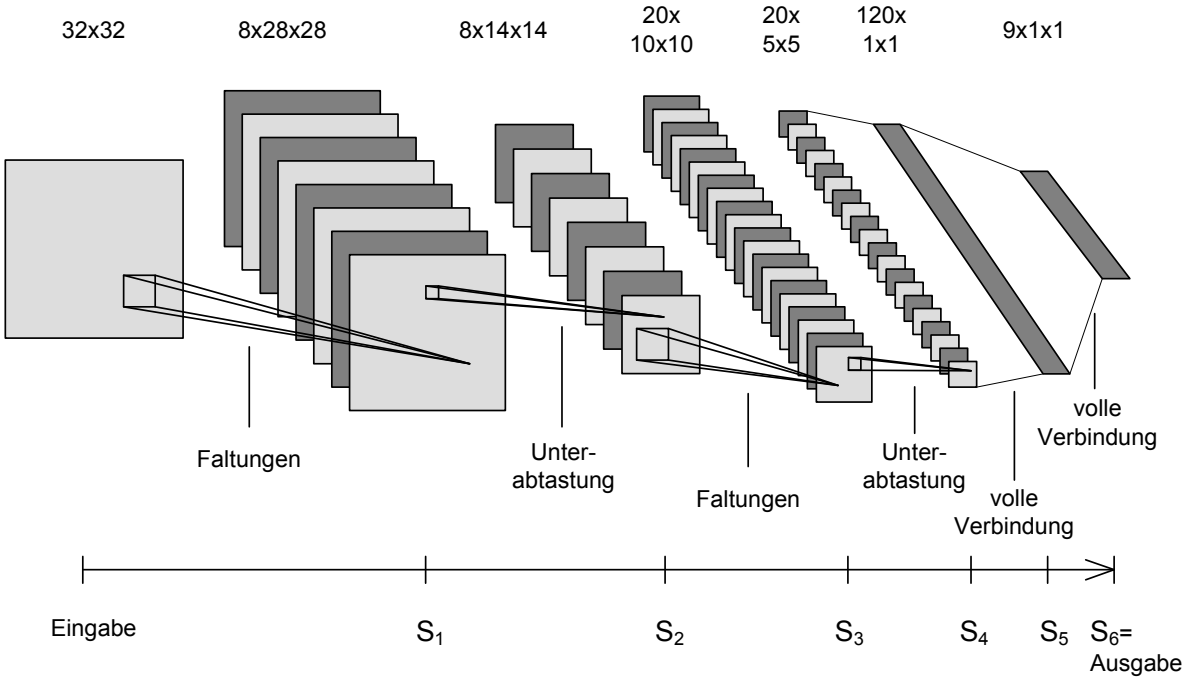

Abbildung 5.4: Faltungsnetz nach [Osadchy u. a. 2007].

Im Vergleich zu dem eingangs gezeigten Beispiel aus Abbildung 5.1, weist dieses Netz einige erhebliche Unterschiede auf. Zunächst sind die Dimensionen des Eingabebildes größer (32x32 statt 20x20). Im Bezug auf die Posenschätzung ist es wichtig, dass das Eingabebild groß genug ist, damit die Konturen des Kopfes vollständig zu sehen sind. Des Weiteren wurde das Netz konzipiert um implizit zwei Winkel der Pose zu schätzen. Es überführt also nach Gleichung 3.3 das Bild in einen Punkt in einem neundimensionalen euklidischen Raum. Entsprechend ist der Ausgabevektor des Netzes neundimensional. Aus ihm lassen sich nun analytisch entsprechend den Ausführungen in Abschnitt 3.2 die geschätzten Winkel und das Label berechnen. Ein Weiterer Unterschied besteht in der Anzahl der Parameter. Während das Netz aus Abbildung 5.1 mit rund 1100 Parametern auskommt, verfügt das hier betrachtete Netz über 63.493 Parameter. In Hinsicht auf die Leistungsfähigkeit moderner Prozessoren stellt diese Zahl einen Kompromis zwischen der Qualität und der Ausführungsgeschwindigkeit des Detektionsverfahrens dar. Das Netz hat eine zusätzliche Faltungs- und Unterabtastungsschicht und es enthält mehr und größere Felder. Insbesondere die Komplexität der vollen Verbindungen hat sich stark erhöht. Allein auf die volle Verbindung zwischen den Schichten S_4 und S_5 entfallen 60.120 Parameter. Ein weitere Aspekt, der hier hinzukommt, ist eine Symmetriebrechung nach [LeCun u. a. 1998a]. Bei einem symmetrisch aufgebauten Netz besteht die Gefahr, dass sich während des Trainings verschiedene parallele Pfade uniform entwickeln. Dies würde die Leistungsfähigkeit des Netzes mindern. Um dies zu vermeiden, besteht zwischen den Schichten S_2 und S_3 eine asymmetrische Verbindungsstruktur. Das bedeutet, die Felder der Schicht S_3 sind unregelmäßig mit jeweils unterschiedlich vielen Feldern der Schicht S_2 verbunden (siehe Tabelle A.2 in Anhang A für eine detaillierte Darstellung).

5.4 Faltungsnetze für andere Problemstellungen

Faltungsnetze sind im Bereich der visuellen Mustererkennung ein universell einsetzbares Werkzeug. Alle Operationen innerhalb eines Netzes werden stets in immer gleicher Form auf das Eingangsbild angewendet. Es wird nicht explizit nach problembezogenen Merkmalen gesucht und insbesondere von keinem Expertenwissen Gebrauch gemacht. Das Faltungsnetz bezieht sein gesamtes benötigtes Wissen aus den Trainingsdaten. Das Training wiederum funktioniert voll automatisch. Das hier behandelte Detektionsverfahren lässt sich dementsprechend nicht nur auf Gesichter anwenden, sondern auf alle Klassen dreidimensionaler Objekte, die ein gemeinsames Muster aufweisen. Es müssen lediglich geeignete Trainingsdaten zur Verfügung stehen. In [LeCun u. a. 2004] wird ein Faltungsnetz eingesetzt, dass zwischen einer Vielzahl von verschiedenen Objekttypen unterscheiden kann. Auch für gänzlich andere Problemstellungen lassen sich Faltungsnetze einsetzen. Zur Schrifterkennung wurde in [LeCun u. a. 1998a] ein erfolgreiches Verfahren ebenfalls auf Basis eines Faltungsnetzes vorgestellt. Ein weiteres Anwendungsbeispiel ist die biometrische Gesichtsidentifizierung in [Chopra u. a. 2005].

5.5 Zusammenfassung

In diesem Kapitel wurde die Struktur und Funktionsweise von Faltungsnetzen erläutert. Zunächst wurde losgelöst vom Hintergrund neuronaler Netze die verschiedenen Faltungsnetzkomponenten anhand eines Beispiels beschrieben (Faltungen, Unterabtastungen, Überlagerungen, volle Verbindungen, Schwellwertfunktionen und Gleichanteile). Dabei wurden die Aufgaben und das Zusammenspiel der verschiedenen Komponenten hinsichtlich eines Mustererkennungsprozesses verdeutlicht (Extraktion, Reduktion und Klassifizierung). Im Anschluss daran wurde erläutert, wie Faltungsnetze als neuronale Netze modelliert werden, um sie mit Hilfe des Backpropagation-Verfahrens trainieren zu können. Die Grundidee hierbei war es, die Pixel durch Neuronen darzustellen und die verschiedenen Operationen durch Verbindungen zwischen diesen Neuronen zu bilden. Hierbei wird von dem bereits im letzten Kapitel beschriebenen Prinzip der geteilten Gewichte Gebrauch gemacht. Schließlich wurde anhand eines weiteren Beispiels erläutert, wie ein Faltungsnetz aufgebaut sein muss, um es als Mapping-Modul für das in dieser Arbeit betrachtete Detektionsverfahren einsetzen zu können. Der wichtigste Aspekt hierbei ist die Bildung eines Ausgabevektors, der der Dimension des euklidischen Raumes entspricht, in dem die Mannigfaltigkeit liegt. Abschließend wurde die flexible Verwendbarkeit von Faltungsnetzen anhand von Beispielen für andere Problemstellungen verdeutlicht.

Kapitel 6

Erweiterung zu einer Lokalisierung

Bei den vorangegangenen Kapiteln 3 bis 5 stand das Labelproblem im Vordergrund. In diesem Kapitel wird erläutert, wie das dort betrachtete Labeling zu einer effizienten Lokalisierung erweitert werden kann. Im folgenden Abschnitt 6.1 wird zunächst ein einfacher, nahe liegender Ansatz betrachtet. Im darauf folgenden Abschnitt 6.2 wird gezeigt, wie eine erhebliche Beschleunigung durch Vermeidung von Rechenredundanzen möglich ist. Da diese Reduktion nur aufgrund der speziellen Form von Faltungsnetzen möglich ist, ist hierin der Vorteil von Faltungsnetzen gegenüber „herkömmlichen“ neuronalen Netzen zu sehen. Schließlich werden in Abschnitt 6.3 Möglichkeiten zur Parallelisierung des Lokalisierungsverfahrens betrachtet, um einen weiteren Geschwindigkeitszuwachs zu erreichen.

6.1 Einfacher Ansatz

Die einfachste Möglichkeit, das Labeling zu einer Lokalisierung zu erweitern, besteht darin, das Verfahren mehrfach auf Teilbildern an allen möglichen Positionen des Gesamtbildes durchzuführen. Da das Faltungsnetz mit Eingaben konstanter Größe arbeitet, muss das Bild skaliert werden, um auch Gesichter unterschiedlicher Größe finden zu können. Abbildung 6.1 illustriert das Vorgehen. Ein gleitendes Fenster (engl.: slide window) wandert auf mehreren Skalierungsstufen von links-oben nach rechts-unten. Ausgehend von der Originalgröße werden hierzu Unterabtastungen in mehreren Stufen mit dem Faktor $\sqrt{2}$ durchgeführt. Der Faktor ist aus Effizienzgründen relativ groß gewählt, damit nur wenige Skalierungsstufen durchlaufen werden müssen. Wie noch später in Kapitel 7.1.3 beschrieben wird, ist der Detektor robust gegenüber Größenvariationen der Gesichter in einem Bereich von 1 bis $\sqrt{2}$ mal der vorgegebenen Standardgröße. Dies wird im Training durch entsprechende Modifizierung der positiven Trainingsbilder erreicht. Auf diese Weise können bei der Lokalisierung auch Gesichter, deren Größen zwischen zwei Skalierungsstufen liegen, zuverlässig gefunden werden. Abhängig von einem festen Schwellwert ergibt sich so eine Menge mit positiven Labeln.

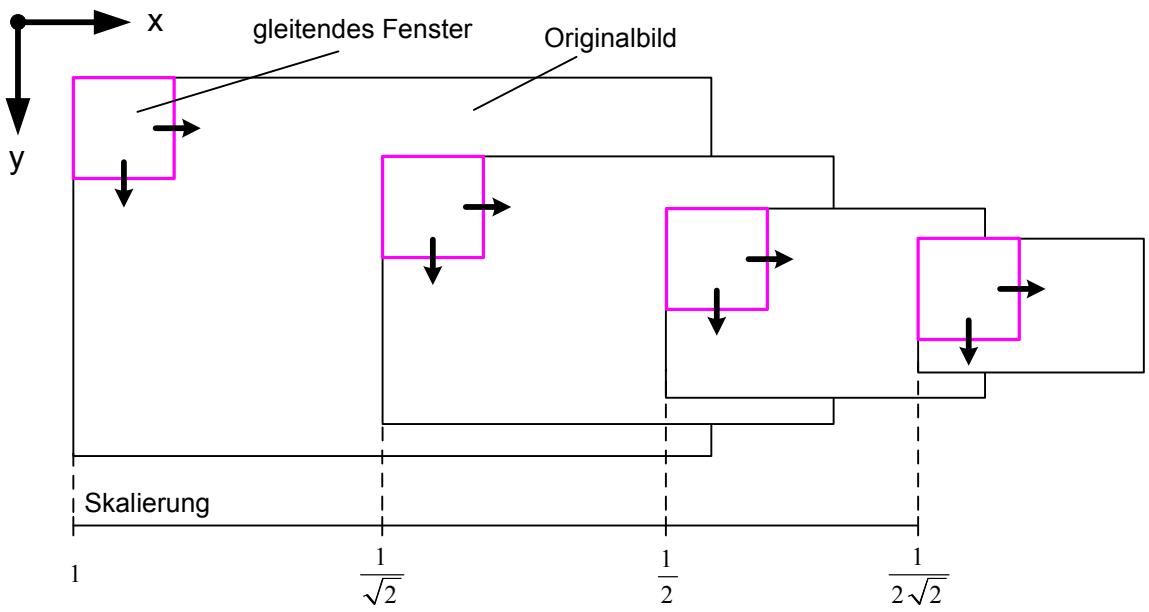

Abbildung 6.1: gleitendes Fenster auf mehreren Skalierungsstufen.

Die Größen und Positionen der einzelnen Treffer werden entsprechend der jeweiligen Skalierungsstufe auf die ursprüngliche Bildgröße umgerechnet. Die resultierende Treffermenge wird im Anschluss nachverarbeitet. Hierbei wird geprüft, ob sich mehrere Treffer auf das gleiche Gesicht beziehen. Ein Treffer H_x wird verworfen, wenn es mindestens einen weiteren Treffer $H_y \neq H_x$ gibt, so dass folgende Bedingungen erfüllt sind:

- H_y ist ein besserer Treffer als H_x, d.h. H_y hat einen kleineren Energiewert.
- H_y liegt auf der selben, oder einer benachbarten Skalierungsstufe von H_x.
- Bezogen auf die gleiche Bildgröße enthält die Teilbildfläche von H_y mindesten einen Überlappungsanteil von N der Teilbildfläche von H_x.

Bei kleinen Werten für N besteht die Gefahr, dass bei zwei nah beieinander liegenden Gesichtern nur eines erkannt und das andere verworfen wird, während sich bei großen Werten die Wahrscheinlichkeit erhöht, dass ein Gesicht mehrfach, leicht versetzt lokalisiert wird. Sinnvolle Werte für N liegen im Bereich um 50%.

6.2 Reduktion der Rechenredundanz

Bei der im letzten Abschnitt vorgeschlagenen Verwendung eines gleitenden Fensters entsteht, da die Berechnungen der Label unabhängig voneinander durchgeführt werden, eine hohe Rechenredundanz. Wird das das gleitende Fenster bspw. horizontal um einen Bildpunkt verschoben, müsste für eine Faltung in der ersten Schicht lediglich die hinzugekommene Reihe

neu berechnet werden. Auf ähnliche Weise lassen sich auch Redundanz bei den Unterabtastungen einsparen. In beiden Fällen muss hierzu auf bereits berechnete Werte zurückgegriffen werden können. Dabei sind folgende Punkte zu berücksichtigen:

- Redundanzen können sowohl bei einer horizontalen als auch bei einer vertikalen Verschiebung des gleitenden Fensters vermieden werden.
- Im Faltungsnetz erstreckt sich die Rechenredundanz über mehrere Schichten.
- Je nach Position des gleitenden Fensters müssen Unterabtastungen in unterschiedlichen „Phasen“ durchgeführt werden. Bei der Halbierung der Bildgröße werden stets vier Pixel zu Einem zusammengefasst. Dabei macht es jedoch einen Unterschied, ob mit einer geraden oder ungeraden Reihe bzw. Spalte im Bild begonnen wird. Es ist folglich zwischen vier verschiedenen Fällen zu unterscheiden.
- Auf wiederverwendete Werte muss aus der jeweiligen Situation heraus zielgerecht zugegriffen werden können, um den Verwaltungsaufwand möglichst gering zu halten. Dabei sollten möglichst wenig Daten im Arbeitsspeicher verschoben werden müssen.

Die Punkte zeigen, dass sich eine vollständige Reduktion der Rechenredundanz nur mit einigen Schwierigkeiten gestalten lässt. In dieser Arbeit wurde deshalb ein anderer Ansatz gewählt, der nicht auf ein gleitendes Fenster beruht. Stattdessen werden alle Label einer Skalierungsstufe gleichzeitig berechnet. Anstatt eine Faltung, Unterabtastung oder volle Verbindung nur auf einen kleinen Bildbereich anzuwenden, wird stets das gesamte Bild verarbeitet. Im Falle einer Unterabtastung werden dabei alle vier Phasen berücksichtigt, so dass sich eine Baumstruktur wie in Abbildung 6.2 ergibt. Hier wird für das Faltungsnetz A aus Abbildung 5.4 die Verarbeitung eines Bildes mit der Größe 640x480 px schematisch dargestellt. Die Anzahl der Bilder sowie deren Breite und Höhe ist für jede Schicht in den dargestellten Rechtecken eingetragen. Bei jeder Unterabtastungsschicht kommt es zu vier Verzweigungen. In Klammern ist jeweils angegeben, ob die Berechnungen bei einer geraden („g“) oder ungeraden („u“) Reihe bzw. Spalte beginnen. In der Ausgabeschicht S_6 gibt es für jeden Pfad neun Ausgabebilder. Aus diesen Bildern wird an jeder Position ein neundimensionaler Vektor gebildet. Jeder Vektor entspricht einem Ergebnis des ursprünglichen Faltungsnetzes. Die zugehörige Position eines Vektors bezogen auf das Eingabebild ergibt sich, indem die Position bezogen auf die Ausgabebilder mit vier multipliziert und anschließend der Versatz, der jeweils unterhalb der Pfade in Klammern angegeben ist, hinzuaddiert wird. Die gezeigte Vorgehensweise hat den Vorteil, dass keine Rechenredundanz mehr vorhanden ist und auch kein sonstiger Mehraufwand entsteht. Der einzige Nachteil ist der stark erhöhte Bedarf an Arbeitsspeicher. Beispielsweise müssen in der Schicht S_5 auf einem Pfad 120 Bilder der Größe 153x113 px zwischengespeichert werden, was einem Speicherbedarf von rund 8 MB entspricht

(Werte in Single Precision nach IEEE 754). Dies stellt jedoch für moderne Computersysteme i. d. R. keinen Engpass dar und wird hier für eine hohe Ausführungsgeschwindigkeit in Kauf genommen. Ein weiterer Geschwindigkeitszuwachs lässt sich erreichen, indem auf Kosten der Genauigkeit nur ein Teil der Pfade berechnet wird. Dabei wird ausgenutzt, dass die Detektion auch bei geringen Abweichungen vom Gesichtsmittelpunkt noch relativ kleine Energiewerte liefert.

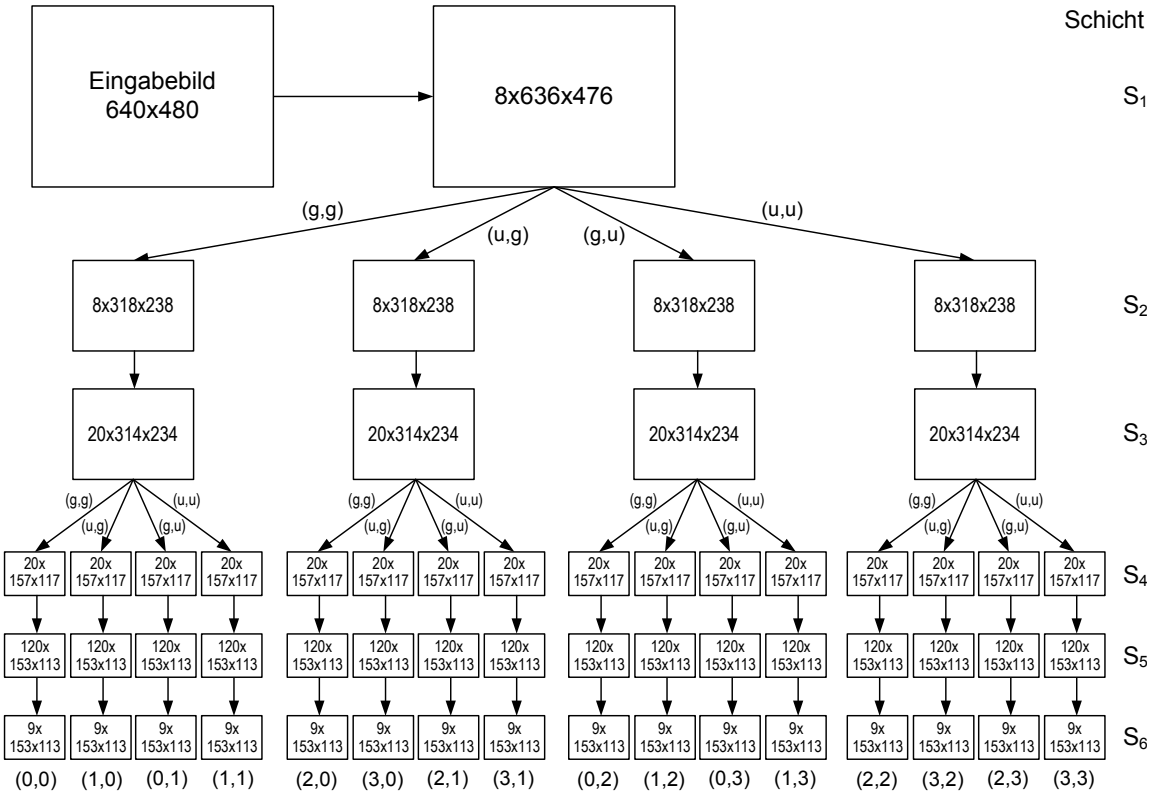

Abbildung 6.2: Verarbeitung des Gesamtbildes.

6.3 Parallelisierung

Bei der Parallelisierung des im letzten Abschnitts vorgestellten Detektionsverfahrens kann grundsätzlich zwischen zwei verschiedenen Strategien unterschieden werden:

- Globale Parallelisierung - In jeder Schicht des Faltungsnetzes müssen mehrere Bilder berechnet werden. Innerhalb einer Schicht bestehen dabei keine Abhängigkeiten, so dass jedes Bild in einem eigenen Thread berechnet werden kann.
- Lokale Parallelisierung - Bei der Berechnung eines Bildes müssen gleichartige Operationen für eine Vielzahl von Pixeln durchgeführt werden. Die einzelnen Pixel können unabhängig voneinander berechnet werden. Im Falle einer Faltung können hierbei jedoch Latenzzeiten beim Lesezugriff entstehen, da zur Berechnung benachbarter Pixel

Zugriff auf überlappende Bildbereiche der Eingabe erforderlich ist. Auch beim gleichzeitigen Zugriff auf einzelne Filterkoeffizienten können Verzögerungen entstehen. Der maximal zu erreichende Parallelisierungsgrad ist jedoch entsprechend der Anzahl an Bildpunkten sehr hoch.

Welche Strategie vorzuziehen ist, hängt von der verwendeten Hardware ab. Im Falle eines Multiprozessorsystems ist eine globale Parallelisierung vorteilhaft, da die einzelnen Prozessoren völlig unabhängig voneinander operieren können und keine Latenzzeiten entstehen. Lediglich vor jeder neuen Schicht muss eine Synchronisation stattfinden. Bei der Verwendung von SIMD-Bausteinen (Single Instruction, Multiple Data) ist eine lokale Parallelisierung sinnvoll, da die Architektur solcher Bausteine in der Regel verlangt, dass sich die gleichzeitig bearbeiteten Daten hintereinander im Speicher befinden. Dies ist für gewöhnlich bei (meist horizontal) benachbarten Pixeln der Fall.

Im Rahmen dieser Arbeit wurde das Detektionsverfahren für eine Graphikprozessor (nVidia GeForce 8 Series) implementiert. Abbildung 6.3 zeigt die Architektur des Chips in vereinfachter Form. Eine Reihe von Threads sind wie dargestellt in Blöcken organisiert, wobei jeder Thread über einen eigenen Registersatz verfügt und seinen Programmfluss unabhängig von anderen Threads steuern kann. Auf den großen Hauptspeicher ist ein globaler Zugriff möglich. Da er jedoch verhältnismäßig langsam ist, verfügt jeder Block zusätzlich über eine kleine, schnelle Speicherbank. Zu Beginn befinden sich das Eingabebild sowie die Netzparameter im

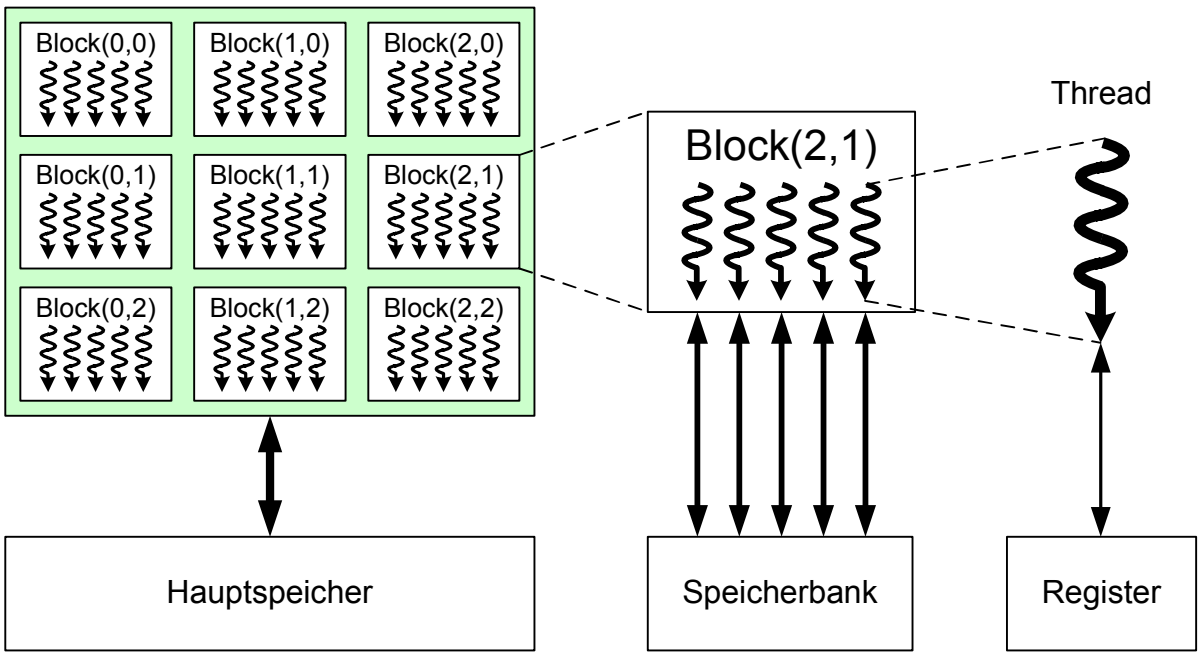

Abbildung 6.3: Speicherhierarchie des Grafikchips (nVidia GeForce 8 Series).

Hauptspeicher der Grafikkarte. Da die Speicherbanken zu klein sind, um ein ganzes Bild zu fassen, wird die Eingabe in mehrere Rechtecke zerlegt, von denen eins in jede Speicherbank

kopiert wird. Die Pixel eines jeden Rechtecks werden dann parallel von mehreren Threads des selben Blocks bearbeitet. Im Falle einer Faltung wird die aktuell verwendete Filtermaske ebenfalls in jede Speicherbank kopiert, um auch hier einen schnellen Zugriff zu erlauben. Des Weiteren ist bei Faltungen zu beachten, dass benachbarte Rechtecke nur unabhängig von einander bearbeitet werden können, wenn sie sich entsprechend Gleichung 5.3 um N_{gx} Pixel in horizontaler bzw. N_{gy} Pixel in vertikaler Richtung überschneiden. Es lässt sich also nicht vermeiden, dass einige Pixel doppelt bzw. vierfach aus dem relativ langsamen Hauptspeicher gelesen werden müssen. Die Rechtecke werden deshalb möglichst groß gewählt, so dass die Kapazität der Speicherbanken vollständig genutzt wird. Durch die beschriebene Vorgehensweise konnte ein hohe Ausführungsgeschwindigkeit erreicht werden. Die genauen Laufzeiten für unterschiedliche Bildgrößen werden in Kapitel 8.3 angegeben.

6.4 Zusammenfassung

In diesem Kapitel wurde gezeigt, wie für diese Arbeit das Labeling zu einem Detektionsverfahren erweitert wurde. Hierzu wurde zunächst ein naiver Ansatz auf Basis eines Silde-Fensters beschrieben, das mehrere Skalierungsstufen des Eingabebildes durchläuft. Anhand dieses Ansatzes wurde verdeutlicht, welche Möglichkeiten bestehen, Rechenredundanzen einzusparen. Die Geschwindigkeitssteigerung, die sich dadurch erreichen lässt, stellt den entscheidenden Vorteil von Faltungsnetzen gegenüber gewöhnlichen vorwärtsgerichteten neuronalen Netzen dar. Es wurde hier der Ansatz gewählt, die verschiedenen Operationen stets auf das gesamte Eingabebild anzuwenden. Auf Kosten eines erhöhten Speicherbedarfs wurde hierdurch die größtmögliche Redundanzreduktion erreicht. Eine weitere Möglichkeit der Effizienzsteigerung, die untersucht wurde, ist die Parallelisierung des Verfahrens. Hierzu wurden zunächst generelle Strategien vorgeschlagen und im Anschluss daran, die konkrete Umsetzung für eine Graphikkarte beschrieben. Hierbei wurde gezeigt, dass der maximal zu erreichende Parallelisierungsgrad entsprechend der Anzahl der Pixel des Eingabebildes sehr hoch ist.

Kapitel 7

Training

In diesem Kapitel wird beschrieben, wie das Training des Gesichtsdetektors im Rahmen dieser Arbeit umgesetzt wurde. Abschnitt 7.1 geht auf die Zusammensetzung der Trainingsdaten ein, erläutert die verwendeten Konventionen zur Beschreibung der Position, Größe und Pose von Gesichtern und zeigt, wie diese Daten für die Trainingsbilder ermittelt wurden. In Abschnitt 7.2 wird dann auf die konkrete Durchführung des Trainings eingegangen. Es werden hier verschiedene Versuche mit unterschiedlich großen Faltungsnetzen und Trainingsmengen beschrieben.

7.1 Trainingsdaten

In diesem Abschnitt erfolgt eine Beschreibung der Trainingsdaten, die für die Versuche im Rahmen dieser Arbeit verwendet wurden. Die Daten bestehen aus einer Menge von positiven und negativen Beispielbildern (Gesichter und Nicht-Gesichter). Im Folgenden wird ein Übersicht über die Zusammensetzung dieser Mengen gegeben. Die Abschnitte 7.1.1 und 7.1.2 gehen dabei auf die Details betreffend der Beschreibung und Annotierung der benötigten Zusatzinformationen zu den Gesichtsbildern ein. Schließlich beschreibt Abschnitt 7.1.3 wie die Bilder vorverarbeitet werden, bevor sie an den Eingang des Faltungsnetzes gelegt werden.

Um eine Menge von Gesichtsbildern in einer angemessenen Größe zusammenzustellen, bietet es sich häufig an, auf eine frei verfügbare Gesichtsdatenbank zurückzugreifen. Viele solcher Datenbanken weisen jedoch keine große Posenvielfalt auf, da sie für Verfahren konzipiert wurden, die nur in begrenztem Maß robust gegenüber Posenvariationen sind. Wieder andere Datenbanken beinhalten zwar mehrere Posen, diese jedoch nur von relativ wenigen Individuen, die mit mehreren Kameras aus unterschiedlichen Winkeln aufgenommen wurden. Eine hohe Zahl unterschiedlicher Gesichter ist jedoch wichtig, um eine Überanpassung des Detektors zu vermeiden. Mit einer Überanpassung ist eine zu starke Spezialisierung gemeint, die

Abbildung 7.1: Auszug aus der Trainingsmenge.

durch ein zu einseitiges Training verursacht wird. Aufgrund dieser Probleme, wurden die hier verwendeten Trainingsbeispiele aus einem individuell zusammengestellten Bildersatz gewonnen (bspw. von Hör- und Theatersälen, Plenarsitzungen und Militärparaden). Abbildung 7.1 zeigt einen Auszug aus der resultierenden Trainingsmenge. Die Eigenschaften dieser Menge werden im Folgenden aufgelistet:

- Metainformationen - Die Größen, Posen und die Mittelpunkte aller Gesichter sind bekannt. Abschnitt 7.1.1 beschreibt im Detail welche Konventionen zur Beschreibung dieser Informationen verwendet wurde und Abschnitt 7.1.2 erläutert, wie diese Daten in der Praxis ermittelt wurden.

- Gesamtanzahl - Da die Annotierung der Trainingsdaten mit einem gewissen Zeitaufwand verbunden ist (siehe Abschnitt 7.1.2), konnten nicht beliebig viele Bilder verwendet werden. Mit rund 6.700 Gesichtern steht dennoch eine angemessene Anzahl an Gesichtsbildern zur Verfügung (zum Vergleich: bei [Osadchy u. a. 2007] wurden rund 30.000 Bilder verwendet, mit denen ähnlich hohe Detetkionsraten wie bei Viola und Jones ([Viola und Jones 2003]) erzielt werden konnten).

- Posenvielfalt - Die verwendeten Gesichtsbilder weisen eine starke Variation betreffend der Pose auf. Insbesondere die Werte für den Gierwinkel (vgl. Abschnitt 7.1.1) sind über die gesamte Trainingsmenge annähernd gleichverteilt. Die Variation des Nickwinkels fällt kleiner aus. Hinsichtlich des Einsatzes von Deckenkameras wurden jedoch ausreichend Aufnahmen aus einem entsprechend steileren Winkel verwendet. Eine Gleichverteilung des Nickwinkels liegt jedoch nicht vor.

- Individuenzahl - Die Anzahl der abgebildeten Individuen ist sehr hoch. Nur in zufälligen Ausnahmefällen sind mehrere Aufnahmen vom selben Gesicht vorhanden.

Die negativen Trainingsbeispiele wurden zu einem großen Teil zufällig aus einem Satz Hintergrundbilder gewonnen. Hierzu wurden an zufälligen Positionen im jeweiligen Bild passende Rechtecke ausgeschnitten. Zu einem kleinen Anteil (ca. 15%) wurden problembezogene Trainingsbeispiele gesammelt. Bei verschiedenen Testläufen haben sich bestimmte Situationen ergeben, in denen vermehrt falsch-positive Treffer aufgetreten sind. Beispiele hierfür sind Streifenmuster (bspw. in Jalousien), bestimmte Textilmuster, Hände, Hälse und Schriftzüge. Diese speziellen Trainingsbilder wurden teilweise manuell und teilweise halbautomatisch mit vortrainierten Faltungsnetzen gesammelt. Insgesamt stehen genauso viele negative wie positive Beispiele zur Verfügung.

7.1.1 Metainformationen

In diesem Abschnitt werden die Konventionen erläutert, die in dieser Arbeit zur Beschreibung der Metainformationen zu den Gesichtsbildern festgelegt wurden. Die Informationen beziehen sich für jedes positive Beispiel auf die Pose, Größe und Position des Gesichts. Sie sind wichtig für eine normalisierte Darstellung der Trainingsbilder. Darüber hinaus werden die Posen für die in Kapitel 3 beschriebenen Trainingsfunktionen benötigt. Abbildung 7.2 zeigt eine Übersicht. Um die Größe einheitlich beschreiben zu können, werden zwei Punkte festgelegt, die sich bei jedem Gesicht leicht wiederfinden lassen. Die Augenkoordinaten können dabei nicht verwendet werden, da in der Profilansicht nicht beide Augen zu sehen sind. Stattdessen werden hier (wie in [Osadchy u. a. 2007] vorgeschlagen) der Mittelpunkt zwischen den Augen A sowie der Mittelpunkt des Mundes B genommen. Die beiden Punkte befinden sich unabhängig von der jeweiligen Perspektive stets auf einer Linie mit dem Nasenbein. Aus ihnen ergibt sich nun entsprechend Gleichung 7.1 die Größe G und der Mittelpunkt M des Gesichts. Für die Trainingsbilder werden die Gesichter so skaliert, dass G ca. 30% der Bildhöhe entspricht. Der Punkt M liegt in [Osadchy u. a. 2007] immer mittig im Bild. Der Nachteil dort ist jedoch, dass bei einer Profilansicht das Gesicht nur ca. die Hälfte der Bildfläche einnimmt. Bei der hier verwendeten Trainingsmenge erfolgt deshalb eine horizontale Verschiebung des Mittelpunktes, die vom Sinus der Gierwinkels abhängt. Wie in Abbildung 7.1 zu sehen war, wird dadurch erreicht, das stets ein Großteil der Bildfläche durch das Gesicht eingenommen wird.

$$G = ||B - A||; \quad M = \frac{B + A}{2} \tag{7.1}$$

Die Pose eines Gesichts wird, wie in Bild 7.2 zu sehen ist, durch drei Winkel beschrieben.

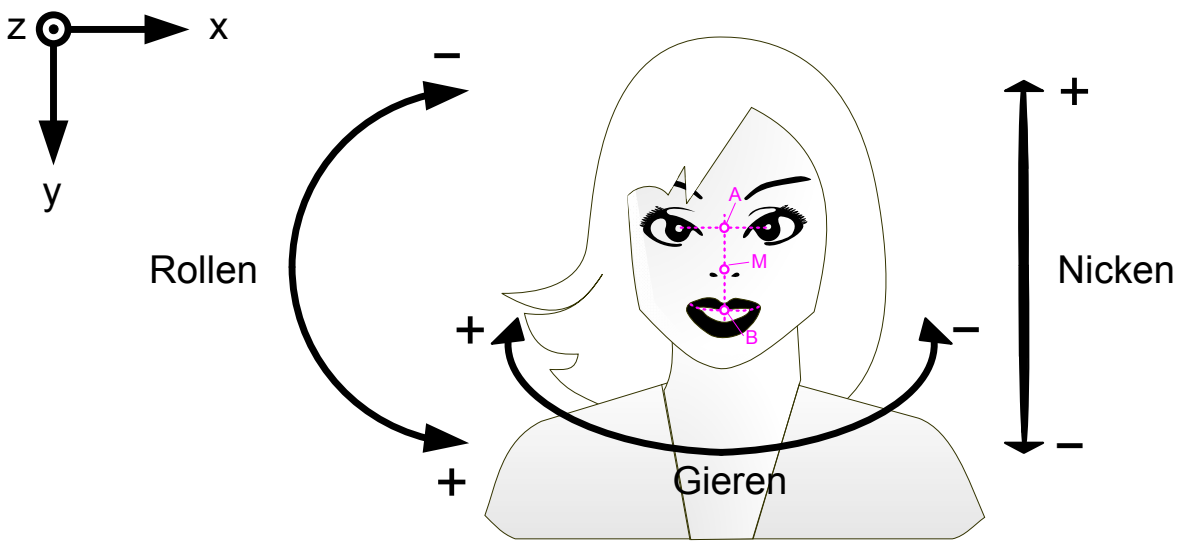

Abbildung 7.2: Konventionen zur Beschreibung der Metadaten.

Diese sind der Rollwinkel Φ (engl: roll), der Nickwinkel ρ (engl.: pitch) und der Gierwinkel Θ (engl.: yaw). Alle Rotationsachsen laufen durch den Punkt M. Im Ausgangszustand ($\Phi = \rho = \Theta = 0$) liegt die Nickachse parallel zu X-Achse, die Gierachse parallel zur Y-Achse und die Rollachse parallel zur Z-Achse. Letztere zeigt aus der Bildebene hinaus. Entsprechend der in Abbildung 7.2 eingezeichneten Vorzeichenkonventionen werden die Rotationen mit den Matrizen nach Gleichung 7.2 beschrieben. Die vollständige Pose Q ergibt sich aus der Multiplikation der Matrizen entsprechend Gleichung 7.3.

$$R = \begin{pmatrix} \cos\Phi & -\sin\Phi & 0 \\ \sin\Phi & \cos\Phi & 0 \\ 0 & 0 & 1 \end{pmatrix}; \; P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\rho & -\sin\rho \\ 0 & \sin\rho & \cos\rho \end{pmatrix}; \; Y = \begin{pmatrix} \cos\Theta & 0 & \sin\Theta \\ 0 & 1 & 0 \\ -\sin\Theta & 0 & \cos\Theta \end{pmatrix} \tag{7.2}$$

$$Q = RPY = \begin{pmatrix} \cos\Phi\cos\Theta - \sin\Phi\sin\rho\sin\Theta & -\sin\Phi\cos\rho & \cos\Phi\sin\Theta + \sin\Phi\sin\rho\cos\Theta \\ \sin\rho\cos\Theta + \cos\Phi\sin\rho\sin\Theta & \cos\Phi\cos\rho & \sin\rho\sin\Theta - \cos\Phi\sin\rho\cos\Theta \\ -\cos\rho\sin\Theta & \sin\rho & \cos\rho\cos\Theta \end{pmatrix} \tag{7.3}$$

7.1.2 Annotierung

Um die im letzten Abschnitt beschriebenen Metadaten zu den Gesichtsbildern zu erhalten, gibt es grundsätzlich zwei Vorgehensweisen. Die erste Möglichkeit besteht darin, die Gesichter in einer kontrollierten Umgebung aufzunehmen, bei der die Entfernung und Lage der Kamera zum aufgenommenen Gesicht bekannt sind. Ein Beispiel hierfür ist die PIE-Datenbank in [Sim u. a. 2003]. Hier wurde ein Raum mit 17 Kameras und 23 Blitzlichtern

ausgestattet. Der Kopf einer Versuchsperson wird in einer bestimmten Position fixiert und das Gesicht bei verschiedenen Beleuchtungssituationen aufgenommen. Der Nachteil dieser Vorgehensweise ist, dass die Anzahl der verschiedenen Posen von der Anzahl der verwendeten Kameras abhängt. Die zweite Möglichkeit besteht darin, nachträglich die Metadaten zu den Gesichtsbildern manuell hinzuzufügen. Hierzu wird ein geeignetes Hilfsprogramm benötigt. Der Vorteil dieser Vorgehensweise ist, dass Bilder verwendet werden können, die in einer unkontrollierten Umgebung aufgenommen wurden. Der Nachteil ist, das eine manuelle Annotierung zeitaufwändig ist. Für die hier durchgeführten Versuche wurden Bilder aus einer unkontrollierten Umgebung verwendet. Deshalb wurde ein Hilfsprogramm implementiert, wie es ähnlich in [Osadchy u. a. 2007] vorgeschlagen wird. Abbildung 7.3 illustriert die Vorgehensweise. Im ersten Schritt markiert der Anwender die im letzten Abschnitt definier-

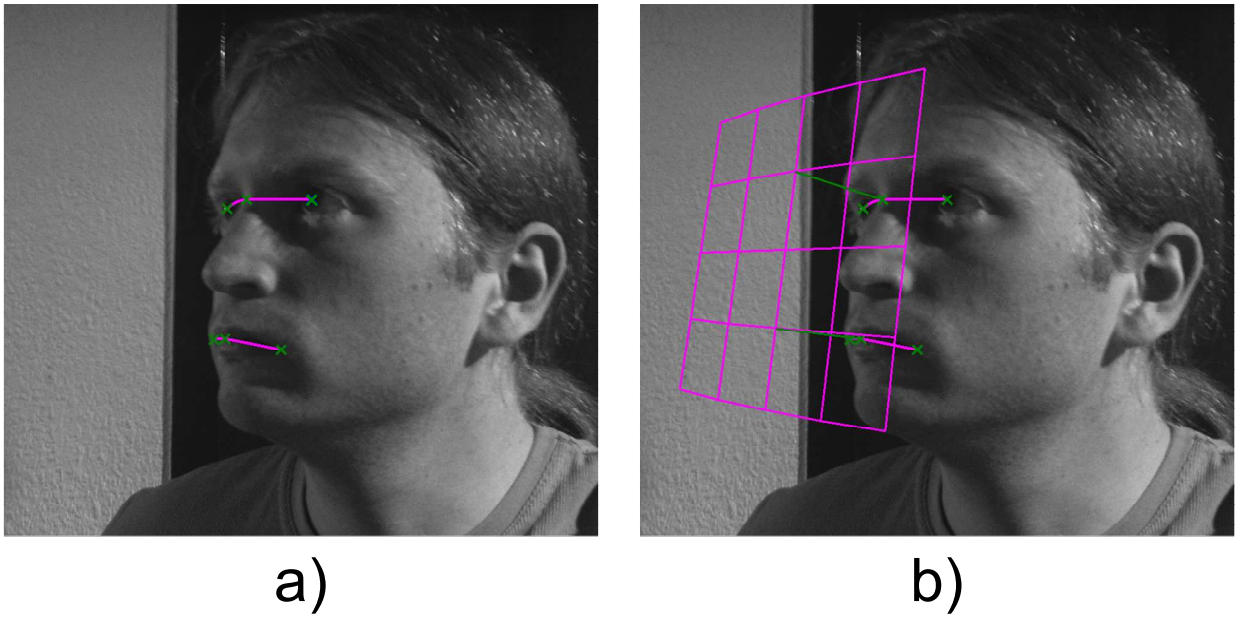

Abbildung 7.3: Funktionsweise des Annotierungsprogramms.

ten Punkte A und B, um so die Größe und den Mittelpunkt festzulegen (Teil a). Nun wird ein ebenes Gitternetz eingezeichnet, welches nach der Vorschrift in Gleichung 7.3 um den Mittelpunkt rotiert werden kann. Der Anwender muss alle drei Winkel so einstellen, dass die Ebene parallel zum Gesicht liegt (Teil b). Auf diese Weise lässt sich die Pose relativ genau angeben. Kleine Abweichungen werden dabei in Kauf genommen.

7.1.3 Vorverarbeitung

Während des Trainings wird die Menge der Beispielbilder mehrfach durchlaufen, um den Trainingserfolg weiter zu steigern. Ein solcher Durchlauf wird im Folgenden als Iteration bezeichnet. Werden jedoch zu viele Iterationen durchgeführt, erhöht sich die Gefahr der eingangs schon erwähnten Überanpassung. Um dies nach Möglichkeit zu vermeiden, wer-

den vor jeder Iteration einige Vorverarbeitungsschritte durchgeführt, um die Variation der Trainingsbilder künstlich zu steigern. Im Ausgangszustand liegen die Grauwerte in einem Bereich von $-0,5$ bis $0,5$. Die folgenden Maßnahmen werden bei positiven und negativen Trainingsbeispielen durchgeführt:

- Horizontale Spiegelung - Die Bilder werden vor jeder zweiten Iteration horizontal gespiegelt. Dies ist die einfachste und effektivste Methode die Variation der Trainingsmenge zu erhöhen. Im Falle der positiven Beispiele wird hierbei die Symmetrieeigenschaft des Gesichts ausgenutzt. Es gilt dann $\Phi_{neu} = -\Phi_{alt}$, $\rho_{neu} = \rho_{alt}$ und $\Theta_{neu} = -\Theta_{alt}$.
- Variation der Helligkeit - Es wird ein zufälliger Wert gleichverteilt aus dem Intervall zwischen $-0,1$ und $0,1$ ermittelt, der auf jeden Pixel des Bildes hinzuaddiert wird.
- Variation des Kontrasts - Es wird ein zufälliger Faktor gleichverteilt aus dem Intervall zwischen $0,9$ und $1,1$ ermittelt, der mit jedem Pixel des Bildes multipliziert wird.
- Additives Rauschen - Ein additives, normalverteiltes, weißes Rauschen wird hinzugefügt. Die Stärke des Rauschens wird bei jedem Bild zufällig über die Varianz der Normalverteilung eingestellt. Diese wird gleichverteilt aus einem Intervall zwischen 10^{-4} und 10^{-5} ermittelt.

Für die positiven Beispiele werden darüber hinaus noch zwei weitere Schritte durchgeführt. Der erste Schritt besteht darin, die Bilder um einen zufälligen Winkel zu drehen. Auf diese Weise wird die Posenvariation noch weiter erhöht. Mit α als Drehwinkel des Bildes ergeben sich die neuen Winkel entsprechend den Gleichungen 7.4 bis 7.6. Die Drehwinkel werden für jedes Bild zufällig gewählt. Hierbei wird jedoch darauf geachtet, dass über die gesamte Trainingsmenge eine Gleichverteilung des Rollwinkels erreicht wird.

$$\rho_{neu} = \arcsin\left(\sin\alpha\cos\rho_{alt}\sin\Theta_{alt} + cos\alpha\sin\rho_{alt}\right) \tag{7.4}$$

$$\Theta_{neu} = \arcsin\frac{\cos\rho_{alt}\sin\Theta_{alt}\cos\alpha - \sin\rho_{alt}\sin\alpha}{\cos\rho_{neu}} \tag{7.5}$$

$$\Phi_{neu} = \arcsin\frac{-\sin\alpha\cdot(\cos\Phi_{alt}\cos\Theta_{alt} - \sin\rho_{alt}\sin\Phi_{alt}\sin\Theta_{alt}) + \cos\alpha\cos\rho_{alt}\sin\Phi_{alt}}{\cos\rho_{neu}} \tag{7.6}$$

Als zweiter Schritt wird die Größe des Gesichts zufällig mit einem Faktor im Bereich von 1 bis $\sqrt{2}$ skaliert. Da sich die Größe des Trainingsbildes dabei nicht ändert, entspricht dies einem Hereinzoomen. Auf diese Weise wird eine Robustheit gegenüber Größenvariationen erreicht. Wie bereits in Kapitel 6.1 beschrieben wurde, ist dies für das Lokalisierungsverfahren von entscheidender Bedeutung.

7.2 Durchführung des Trainings

In diesem Abschnitt werden die für diese Arbeit durchgeführten Untersuchungen zum Training beschrieben und die dabei ermittelten Ergebnisse betreffend des Trainingsfortschritts ausgewertet. Für die durchgeführten Experimente wurden vier unterschiedliche Faltungsnetze, F_A, F_B, F_C und F_D eingesetzt. Die vollständige Beschreibung der Netzstrukturen befindet sich in Anhang A. Im Folgenden werden die wichtigsten Eckdaten hervorgehoben. Die Netze F_A, F_B und F_C sind für eine integrierte Schätzung von zwei Posenwinkel konzipiert worden und haben entsprechend neun Ausgabewerte (vgl. Kapitel 5.3). F_A bezeichnet das bereits in Kapitel 5.3 betrachtete Netz nach [Osadchy u. a. 2007] und verfügt über sechs Schichten sowie 63.493 Parameter. Das Eingabebild hat eine Größe von 32x32 px. F_B hat eine ähnliche Struktur wie F_A, verfügt aber nur über 42.725 Parameter. F_C hat lediglich fünf Schichten und 30.288 Parameter. Auch das Eingabebild ist hier mit einer Größe von 28x28 px etwas kleiner. Bei allen Versuchen wird für die Netze F_A, F_B und F_C die Trainingsfunktion nach Gleichung 3.12 verwendet. Für die Pose gilt $Z = (\Theta, \Phi)$, wobei Θ den Gier- und Φ den Rollwinkel bezeichnet. Der Nickwinkel wird hier (ebenso wie in [Osadchy u. a. 2007]) nicht explizit betrachtet. Der Grund hierfür ist, dass bei den Trainingsdaten der Winkelbereich bezüglich des Nickens relativ klein ist und die erforderliche Gleichverteilung nicht vorliegt. Letzteres wäre jedoch für ein ausgewogenes Posentraining wichtig. Da es äußerst aufwendig ist, ein Trainingsset zusammenzustellen, bei denen Gesichter in allen möglichen Posen ausgewogen und in ausreichender Zahl zur Verfügung stehen, wurde der Schwerpunkt hier lediglich auf den Gierwinkel gelegt. Eine Gleichverteilung des Rollwinkels wird, wie in Abschnitt 7.1.3 beschrieben wurde, durch Drehungen der Gesichtsbilder erreicht.

Das Netz F_D unterscheidet sich von F_C nur in der letzten Schicht. Anstatt neun hat es nur einen Ausgabewert. Es ist also nicht für eine integrierte Posenschätzung geeignet. Stattdessen wird der einzige Ausgabewert direkt einem Label zugeordnet. Für das Training werden zwei Richtwerte $L_{pos} = L$ und $L_{neg} = -L$ für positive und negative Trainingsbeispiele festgelegt. Die Trainingsfunktion ergibt sich dann einfach zu $\mathcal{L} = \sum_{P \in \mathcal{S}} \left((-1)^{Y_P} \cdot L - D_P\right)^2$, wobei Y_P das Label und D_P den Ausgabewert von F_D für das Trainingsbeispiel P bezeichnet. Dies entspricht wieder der Minimierung eines quadratischen Fehlers über alle Trainingsbeispiele. Es wird dabei eine reduzierte Trainingsmenge mit rund 3.000 Gesichtsbildern verwendet, bei denen alle Posewinkel im Bereich von $-45°$ bis $45°$ liegen. Das Netz wird später verwendet, um in einem direkten Vergleich mit F_C zu ermitteln, in wie weit die implizite Posenschätzung die Detektionsergebnisse auch bei nur kleinen Posenvariationen verbessern kann.

Bei allen Netzen erfolgt das Training durch Backpropagation entsprechend den Ausführungen aus Kapitel 4. Die Netzparameter werden entsprechend den Empfehlungen in [LeCun u. a. 1998a] mit gleichverteilten Zufallswerten im Intervall von $-1,2$ bis $1,2$ initialisiert, wobei die Werte jeweils durch den Fan-In des zugehörigen Neurons geteilt werden. Vor jeder Iteration werden die Trainingsbeispiele gemischt und entsprechend Abschnitt 7.1.3 vorverarbeitet.

Beim Training werden nun immer abwechselnd positive und negative Trainingsbeispiele an das Netz gelegt. Ist die Trainingsmenge abgearbeitet, werden vor einer neuen Iteration die aktuellen Netzparameter zwecks einer späteren Evaluierung gespeichert (siehe Kapitel 8) und die Lernfortschritte protokolliert.

Es werden nicht alle 13.400 Trainingsbeispiele für das Training verwendet. Eine Teilmenge mit 1.400 Beispielen, davon jeweils 700 Gesichter und Nich-Gesichter, wird separat als Testset zur Bewertung der Lernfortschritte eingesetzt. Für das Training verbleiben so 12.000 Trainingsbeispiele. Die Bilder des Testsets werden analog zu den eigentlichen Trainingsbeispielen vorverarbeitet und jeweils mit und ohne einer horizontalen Spiegelung verwendet, so dass insgesamt 2.800 Testmuster zur Verfügung stehen.

Nach der hier beschriebenen Vorgehensweise wurden verschiedene Versuche mit unterschiedlichen Einstellungen der Trainingsparameter durchgeführt. Dabei wurden zwei unterschiedliche Trainingsstrategien getestet. Die erste Strategie, die in Abschnitt 7.2.1 betrachtet wird, ist die Verwendung einer absteigenden globalen Lernrate. Bei den in Abschnitt 7.2.2 beschrieben Versuchen wird hingegen für jeden Netzparameter eine individuelle Lernrate bestimmt. Abschließend werden in Abschnitt 7.2.3 Vergleiche mit unterschiedlich großen Trainingsmengen angestellt. In diesem Zusammenhang werden auch die unterschiedlichen Laufzeiten des Trainings betrachtet.

7.2.1 Absteigende globale Lernrate

Einen großen Einfluss auf den Trainingserfolg hat die globale Lernrate η in Gleichung 4.9. Wird η zu klein gewählt, stellen sich Fortschritte nur langsam ein. Dagegen darf η auch nicht zu groß gewählt werden, da die Gefahr besteht, dass einzelne Netzparameter stark divergieren (vgl. Abbildung 4.4), was sich negativ auf den Trainingserfolg auswirkt. Sinnvolle Werte für η liegen im Bereich von 10^{-4} bis 10^{-3}. Bei den Versuchen hat sich jedoch gezeigt, dass mit eine konstanten Lernrate keine zufriedenstellenden Ergebnisse in angemessener Zeit erzielt werden können. Stattdessen hat sich folgende Strategie bewährt: Die Lernrate wird zunächst relativ groß gewählt (bspw. 0.004) und nach jeder Iteration mit einem konstanten Faktor (bspw. 0.75) reduziert. Abbildung 7.4 zeigt für alle vier Netze die Trainingsfortschritte anhand der durchschnittlichen Energiewerte über allen positiven (rot) und negativen (grün) Testmuster. Des Weiteren wurde nach jeder Iteration der optimale Schwellwert (T in Gl. 3.10) ermittelt, mit dem auf der Testmenge die meisten Labels korrekt zugeordnet werden (blau). Bei den Netzen F_A, F_B und F_C ergeben sich sehr ähnliche Verläufe: Zu Beginn befinden sich alle gemappten Punkte in der Nähe der Mannigfaltigkeit und erzeugen entsprechend kleine Energiewerte. Nach und nach entfernen sich dann die negativen Beispiele entsprechend Gleichung 3.18 immer weiter von der Mannigfaltigkeit und erzeugen immer größere Energiewerte. Die Streuung ist dabei jedoch recht hoch, d.h. vereinzelte Beispiele

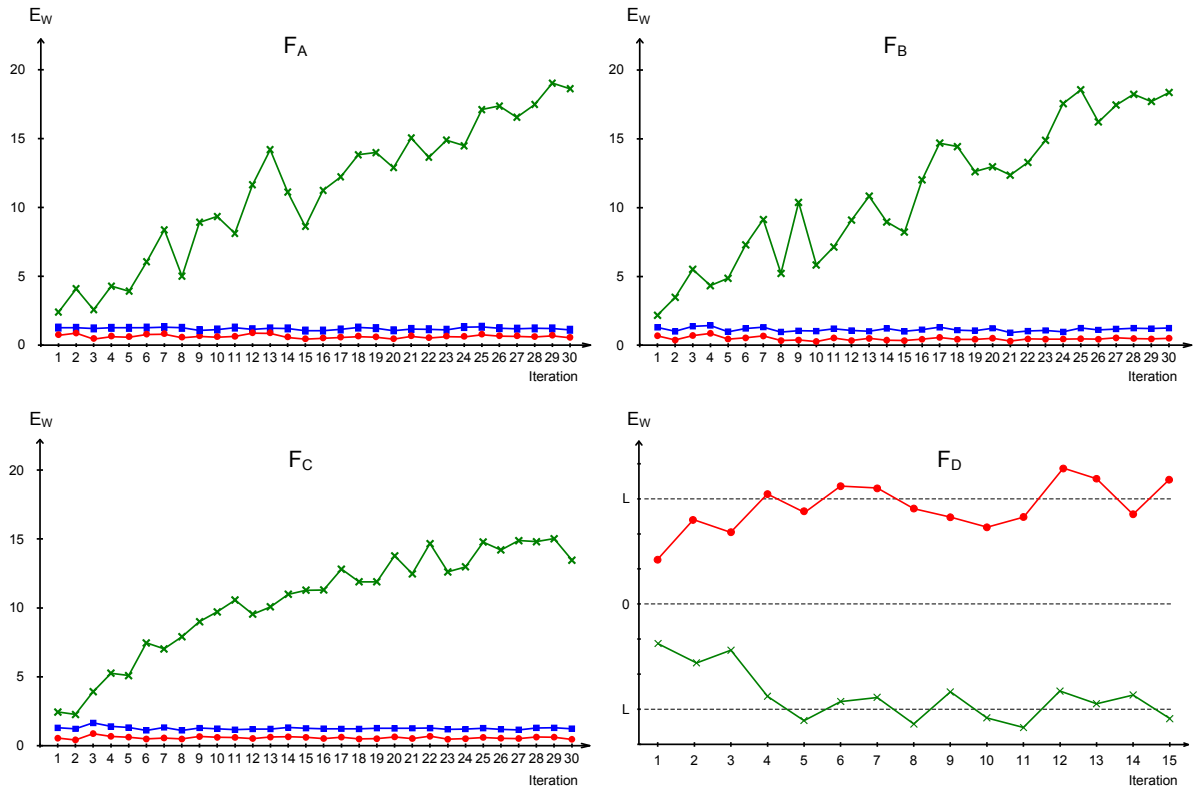

Abbildung 7.4: Durchschnittliche Energiewerte bei den positiven (rot) und negativen (grün) Testmustern sowie der optimale Schwellwert (blau) nach jeder Iteration.

können immer noch relativ niedrige Energiewerte erzeugen. Dies ist der Grund, warum der optimale Schwellwert relativ nah an dem durchschnittlichen Energiewert der positiven Beispiele liegt. Unterschiede zwischen den Netzen zeigen sich vor allem darin, wie schnell die Energiewerte für die negativen Eingaben steigen. Je größer das Netz ist, desto schneller ist der Anstieg. Zum Vergleich wurde auch der Verlauf für das Netz F_D dargestellt. Wie zu sehen ist, trennen sich die durchschnittlichen Energiewerte der positiven und negativen Eingaben schon nach wenigen Iterationen und bewegen sich in der Nähe der zuvor festgelegten Richtwerte L bzw. $-L$.

Abbildung 7.5 zeigt für die dargestellten Energieverläufe die zugehörigen Detektionsraten, d.h. die Anzahl der korrekt zugeordneten Labels auf der Testmenge. Für eine übersichtliche Darstellung wurde hier (anders als später in Kapitel 8) nur das Ergebnis für den optimalen Schwellwert eingetragen und es wurde nicht zwischen positiven und negativen Detektionen unterschieden. Bezüglich eines optimalen Schwellwertes wurden in der Regel jedoch in etwa gleich viele falsch-positive und falsch-negative Ergebnisse erzeugt. Auffällig bei den Verläufen ist, dass der beste Spitzenwert mit rund 87% nach der 9. Iteration beim etwas kleineren Netz F_B erzielt wurde. Diese Beobachtung spricht für die Überlegung, dass kleinere Netze bei kleineren Trainingsmengen vorteilhafter sind, da die Gefahr einer Überanpassung reduziert wird. Das größte Netz F_A erreicht seinen Spitzenwerte erst später mit rund 85,5% nach der 16. Iteration. Das kleinste Netz F_C erreicht den besten Wert nach der 12. Iteration mit

rund 83%. Für die nachfolgenden Iterationen werden die Ergebnisse bei allen Netzen nach erreichen der Bestmarke nach und nach schlechter.

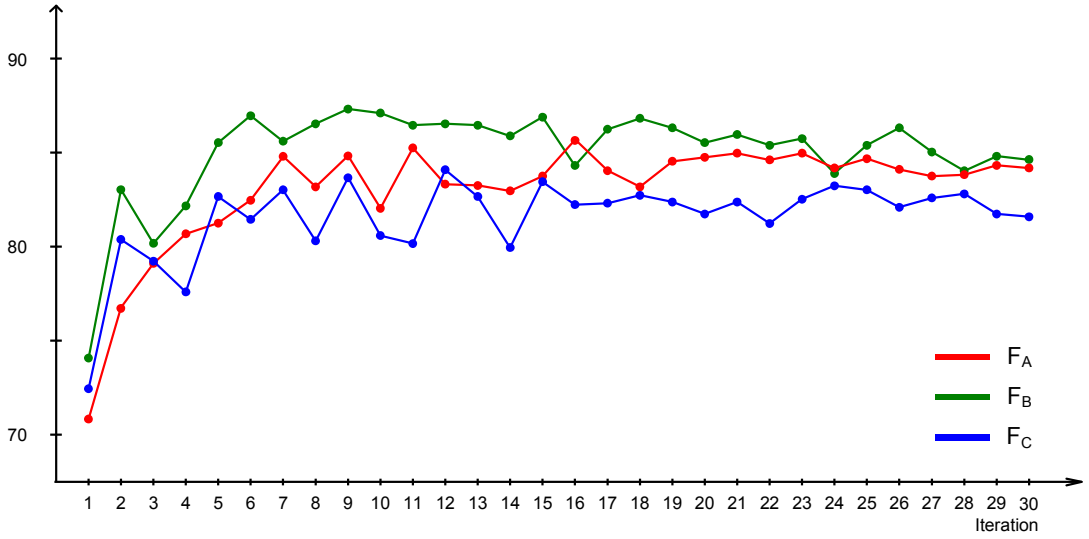

Abbildung 7.5: Detektionsrate auf der Testmenge bei optimalem Schwellwert. Angaben in Prozent.

Ein weiterer Parameter, der einen erheblichen Einfluss auf den Trainingsfortschritt hat, ist K in Gleichung 3.18. Durch ihn wird eingestellt, wie stark die Netzparameter variiert werden sollen, um die negativen Eingaben weiter von der Mannigfaltigkeit zu entfernen. Sinnvolle Werte für K liegen im Bereich von $n \cdot 10^{-1}$ bis 10^{-3} mit $n \approx 0,25$. Bei Versuchen mit deutlich größeren Werten als $0,25$ kam es zu starken Divergenzen, die einen Trainingserfolg unmöglich machten. Bei kleineren Werten als $0,001$ hingegen stellten sich Trainingsfortschritte nur sehr langsam ein. Bei den bisher betrachteten Versuchen galt stets $K = 0,1$. Für das Netz F_B zeigt Abbildung 7.6 die Energieverläufe der negativen Testeingaben (links) sowie die Trefferquoten (rechts) bei unterschiedlichen Werten für K. Erwartungsgemäß wächst die durchschnittliche

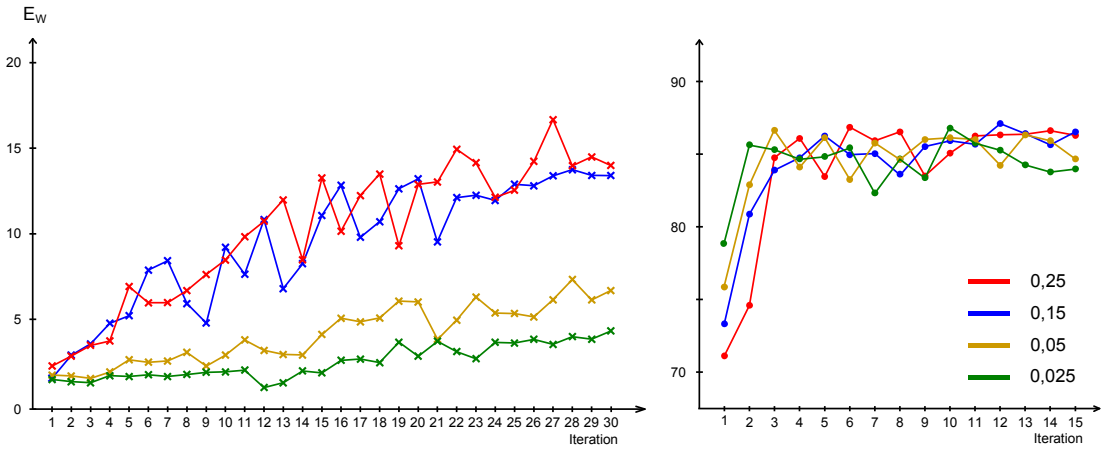

Abbildung 7.6: Training bei Netz F_B mit unterschiedlichen Werten für K. Rot: $0,25$; Blau: $0,15$; Braun: $0,05$; Grün: $0,025$.

Energie bei größeren Werten stärker an. Dies hat jedoch keinen signifikanten Einfluss auf die

Trefferquoten. Betrachtet man nur die ersten drei Iterationen scheinen kleinere Werte zwar vorteilhafter zu sein, jedoch erreichen alle Verläufe einen Spitzenwert von etwa 87% innerhalb der ersten 15 Iterationen. Der Vorteil bei großen Werten für K ist, dass im Endeffekt die Ausgaben des Netzes eine stärkere Streuung aufweisen. Hierdurch entstehen größerer Spielräume bei der Wahl des Schwellwertes. Somit ist eine feinere Einstellung möglich, wenn es darum geht, die Rate der falsch-negativen Ergebnisse auf Kosten einer höheren Zahl an falsch-positiven Treffern zu reduzieren (vgl. Kapitel 8).

7.2.2 Individuelle Lernraten

Die zweite Trainingsstrategie, die in dieser Arbeit untersucht wurde, ist die Verwendung individueller Lernraten durch die Berechnung der Hesse-Diagonalen entsprechend den Ausführungen in Kapitel 4.3.2. Hierbei wird nicht nur berücksichtigt, wie groß die Anteile der einzelnen Parameter am Fehler sind, sondern auch, wie stark sich die Änderungen eines bestimmten Parameters auf die Netzausgabe auswirkt. Auf diese Weise werden die Netzparameter zielgerechter angepasst. Die individuellen Lernraten h_{kk} werden vor jeder Iteration neu berechnet. Die Anpassung der Parameter erfolgt dann entsprechend Gleichung 4.9. In dieser Formel ist der globale Parameter η nun konstant. Aufgrund der vorgenommenen Näherungen bei der Bestimmung der einzelnen Lernraten (vgl. Kapitel 4.3.2), kann es vorkommen, dass sich vereinzelt sehr kleine Werte für h_{kk} ergeben. Der Parameter μ soll in etwaigen Fällen verhindern, dass der zugehöriger Parameter w_k dann divergiert. Wird μ zu klein gewählt, erhöht sich die Gefahr einer Divergenz. Bei einer zu großen Wahl wiederum wird der Einfluss der individuellen Lernraten zu stark geschmälert. Bei den Versuchen konnten gute Trainingserfolge mit $\eta = \mu = 0,001$ erzielt werden.

Für das Netz F_C stellt Abbildung 7.7 die Trainingsstrategien gegenüber. Eingetragen sind wieder die Detektionsraten auf der Testmenge für 30 Iterationen bei optimalem Schwellwert. Neben den Ergebnissen der beiden Strategien sind zum Vergleich auch die Resultate eingetragen worden, die bei der Wahl einer konstanten globalen Lernrate erzielt wurden. Letztere sind jedoch mit einer Trefferquote, die durchgängig unter 80% liegt, nicht zufriedenstellend. Dahingegen konnte mit einer absteigenden Lernrate eine Trefferquote von rund 87% nach der 9. Iteration und mit individuellen Lernraten von rund 87,5% nach der 7. Iteration erzielt werden (falsch-positive und falsch-negative Ergebnisse sind dabei wieder in etwa ausgeglichen). Letztendlich kann mit beiden Strategien ein zufriedenstellendes Trainingsergebnis erreicht werden. Der Nachteil bei einer absteigenden Lernrate ist, dass in Hinsicht auf unterschiedliche Trainingsszenarien jeweils ein günstiger Startwert und ein geeigneter Abstiegsgrad für die Lernrate gefunden werden muss. Hingegen ist der Nachteil bei den individuellen Lernraten die erhöhte Rechenzeit. Die Berechnung der Parameter h_{kk} nimmt genauso viel Zeit in Anspruch wir das eigentliche Training selbst. Die Rechenzeit lässt sich jedoch reduzieren, wenn die Berechnung nicht über die gesamte Trainingsmenge erfolgt, sondern nur über eine

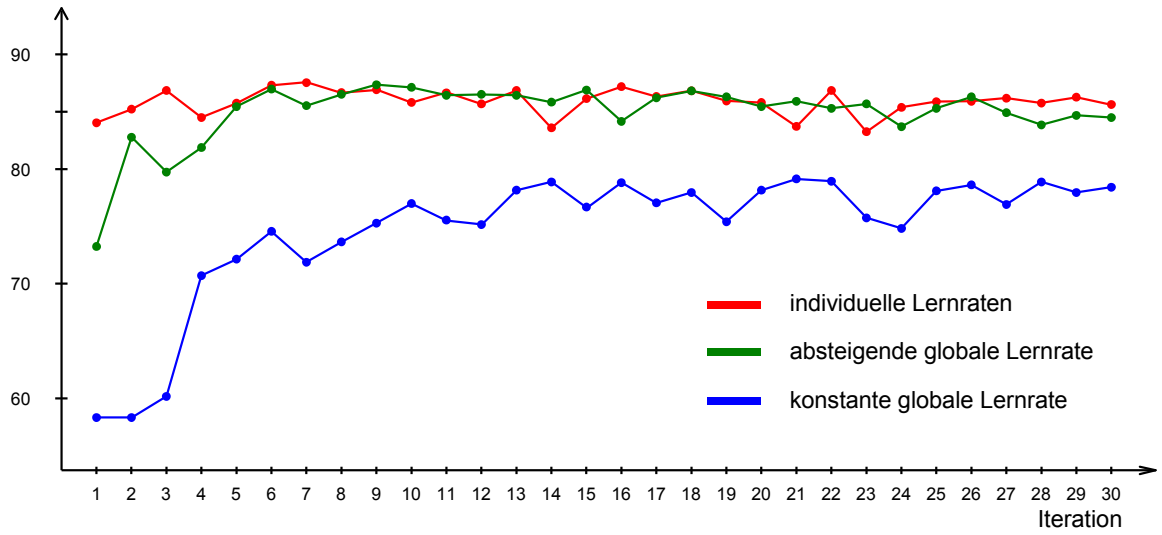

Abbildung 7.7: Vergleich der Trainingsstrategien bei Netz F_B. Angegeben sind die Trefferquoten auf der Testmenge in Prozent.

zufällig gewählte Teilmenge. Bei den Versuchen hat sich gezeigt, dass bereits ein Zehntel der Trainingsmenge ausreichend ist, um gute Trainingserfolge zu erzielen.

7.2.3 Größe der Trainingsmenge

Die Zusammenstellung einer geeigneten Trainingsdatenmenge für eine bestimmte Problemstellung ist ein aufwendiges Unterfangen. Deshalb ist die Frage nach der erforderlichen Größe der Menge interessant. Zu diesem Zweck wurden die hier betrachteten Netze für einen direkten Vergleich mit unterschiedlich großen Mengen trainiert. Dazu wurde aus dem Trainingsset mit rund 12.000 Beispielbildern (jeweils 6.000 Gesichter und Nicht-Gesichter) zufällige Teilmengen mit 10.000, 8.000, 6.000, 4.000 und 2.000 Bildern gebildet (mit jeweils gleich vielen Gesichtern und Nicht-Gesichtern). Für das Netz F_B zeigt Abbildung 7.8 den Verlauf für die unterschiedlichen Mengen bei gleicher Trainingskonfiguration. Angegeben ist hier wieder der Anteil der korrekt ermittelten Label in Prozent. Wie zu erkennen ist, besteht ein starker Zusammenhang zwischen der Trefferquote und der Größe der Trainingsmenge. Mit zunehmender Größer reduziert sich jedoch der erzielte Gewinn. So ist bspw. der Unterschied zwischen den Verläufen bei 10.000 und 12.000 Trainingbeispielen nicht mehr so stark wie bei 2.000 und 4.000 Beispielen. Wenn durch eine Weitere Vergrößerung der Trainingsmenge keine wesentlichen Verbesserungen mehr erzielt werden können, sollte auf eine größere Netzstruktur ausgewichen werden. Die Größe des Netzes F_B ist hier aber in jedem Fall noch ausreichend.

Tabelle 7.1 zeigt abschließend die Laufzeiten des Trainings für die verschiedenen Netzstrukturen bei unterschiedlich großen Datenmengen. Wie zu sehen ist, nimmt eine Iteration bei allen Fällen weniger als zehn Minuten in Anspruch. Das gesamte Training dauert dement-

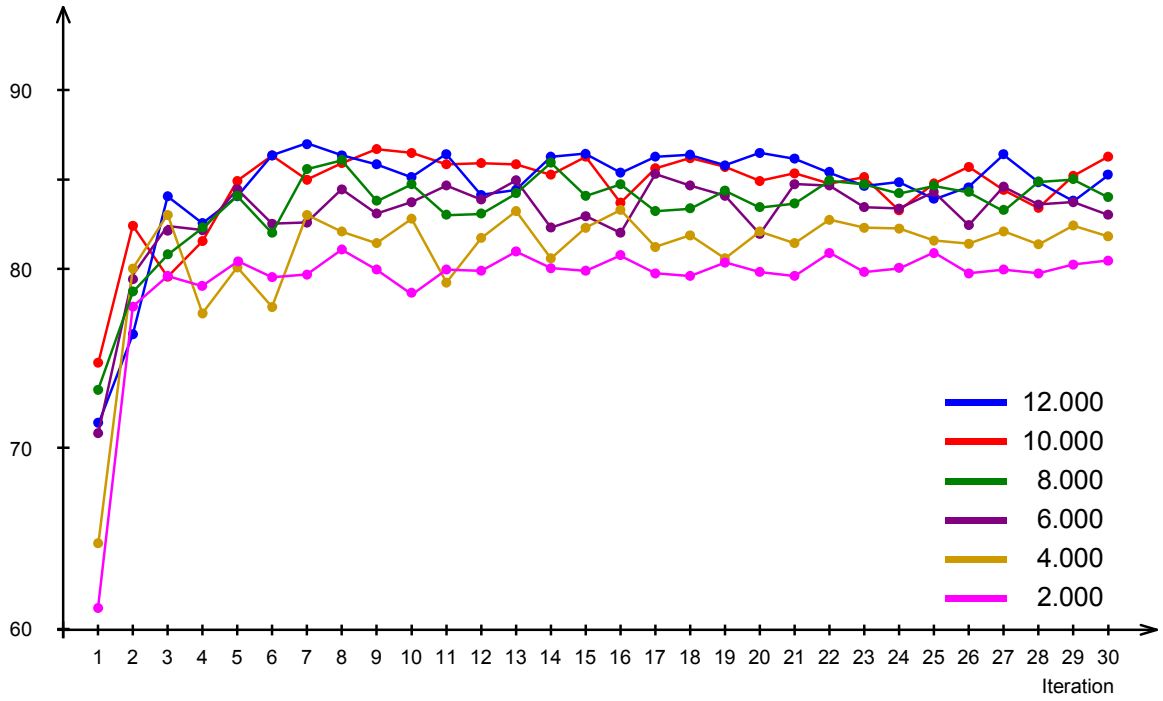

Abbildung 7.8: Trainingsfortschritte bei unterschiedlich großen Trainingsmengen. Angegeben ist die Trefferquote auf der Testdatenmenge in Prozent.

sprechend nur wenige Stunden.

7.3 Zusammenfassung

In diesem Kapitel wurde ausführliche erläutert, wie das Training für diese Arbeit konkret umgesetzt wurde. Zunächst wurde der Aufbau der verwendeten Trainingsdatenmenge betrachtet. Die Gesichter wurden aus einem Satz von Bildern gesammelt, die aus einer unkontrollierten Umgebung aufgenommen wurden. Mit einem geeigneten Hilfsprogramm wurden Position, Größe und Pose der Gesichter nach zuvor festgelegten Konventionen ermittelt. Es wurde gezeigt, wie eine hohe Posenvielfalt und eine geeignete Vorverarbeitung für ein abwechslungsreiches Training sorgen, um eine Überanpassung des Faltungsnetzes möglichst zu vermeiden. Im zweiten Teil dieses Kapitels wurden die verschiedenen Trainingsexperimente

	2.000	4.000	6.000	8.000	10.000	12.000
F_A	64,38	128,77	193,15	257,53	321,92	386,30
F_B	41,38	82,76	124,14	165,52	206,90	248,28
F_C	35,65	71,30	106,94	142,58	178,23	213,88

Tabelle 7.1: Laufzeiten des Trainings für die verschiedenen Netzstrukturen (Zeilen) bei unter unterschiedlich großen Datenmengen (Spalten). Angegeben ist die durchschnittliche Laufzeit pro Iteration in Sekunden.

beschrieben. Es wurde die Verwendung einer globalen, absteigenden Lernrate sowie individueller Lernraten für die einzelnen Netzparameter untersucht. Mit beiden Strategien konnten gute Trainingserfolge erzielt werden, wobei eine absteigende, globale Lernrate jedoch mehr Sorgfalt erfordert, wenn es um die geeignete Wahl einer Anfangsgröße und einer Abstiegsrate geht. Bei den Experimenten stellten sich wichtige Zusammenhänge heraus: Es wurde gezeigt, dass sich eine größere Trainingsdatenmenge generell positiv auf den Trainingserfolg auswirkt. Die Größe des Faltungsnetzes sollte jedoch auf die Größe der Trainingsmenge abgestimmt sein. Bei zu vielen Netzparametern besteht die Gefahr einer Überanpassung, während bei zu wenigen Parametern das Potential der Trainingsdatenmenge nicht voll ausgeschöpft werden kann.

Kapitel 8

Evaluierung

Dieses Kapitel beschreibt die für diese Arbeit durchgeführte Evaluierung des Lokalisierungsverfahrens. Im ersten Teil in Abschnitt 8.1 werden die Ergebnisse aufgezeigt, die auf Standardtestsets erzielt wurden, die bereits in anderen Arbeiten verwendet wurden. Im zweiten Teil in Abschnitt 8.2 wird das Lokalisierungsverfahren in dem eingangs in Kapitel 1 schon angesprochenen Konferenzraums erprobt. Abschließend werden in Abschnitt 8.3 die Laufzeiten der Lokalisierug betrachtet.

Bei allen Tests werden zur Bewertung der Treffer die gleichen Kriterien verwendet. Diese wurden aus Gründen der Einheitlichkeit aus [Osadchy u. a. 2007] übernommen. Ein Gesicht F gilt demnach als detektiert, wenn es einen Treffer H gibt, der die Bedingungen nach Gleichung 8.1 und Gleichung 8.2 erfüllt.

$$\frac{1}{2} \cdot G_F \leq G_H \leq 2 \cdot G_F \tag{8.1}$$

$$||M_F - M_H|| \leq 1,2 \cdot G_F \tag{8.2}$$

Hierbei bezeichnen G_F und M_F bzw. G_H und M_H die Größe und den Mittelpunkt des Gesichts F bzw. des Treffers H nach Gleichung 7.1. Alle Treffer, denen nach diesen Kriterien kein Gesicht zugeordnet werden kann, werden als falsch-positiv gewertet.

8.1 Standardtestsets

In diesem Abschnitt werden die Ergebnisse betrachtet, die das Lokalisierungsverfahren auf drei verschiedenen Testsets erzielt hat. Die Eigenschaften dieser Sets werden im Folgenden beschrieben:

- FRONTAL - Dieses Set beinhaltet Gesichter aus einer Frontalansicht. Es wurde ursprünglich in [Sung und Poggio 1998] eingesetzt und besteht aus 130 Bildern mit insgesamt 527 Gesichtern. Die Rollwinkel aller Gesichter liegen in einem Bereich von $\pm 20°$. Die Gierwinkel gehen nicht über $\pm 30°$ und die Nickwinkel nicht über $\pm 15°$ hinaus.
- INPLANE - Ausgehend von einer Frontalansicht beinhaltet dieses Set Gesichter, die in der Bildebene rotiert sind. Es wurde ursprünglich in [Rowley u. a. 1998] verwendet und besteht aus 50 Bildern mit insgesamt 225 Gesichtern. Die Gier- und Nickwinkel der Gesichter liegen nur in wenigen Ausnahmefällen außerhalb eines Bereiches von $\pm 5°$.
- PROFILE - Die 208 Bilder dieses Sets beinhalten insgesamt 255 Gesichter aus einer Halb- oder Vollprofilansicht. Das bedeutet, die Gierwinkel sind größer als $45°$ bzw. kleiner als $-45°$. Die Nickwinkel der Gesichter liegen in einem Bereich von $\pm 20°$, während die Rollwinkel nur vereinzelt außerhalb eines Bereiches von $\pm 5°$ liegen. Das Set wurde ursprünglich in [Schneiderman und Kanade 2000] benutzt.

Die Ergebnisse für die Testsets werden im Folgenden mit so genannten ROC-Kurven (Receiver Operating Characteristic) dargestellt. Die vertikale Achse steht hierbei für den Anteil der erfolgreich detektierten Gesichter in Prozent, während die horizontale Achse die durchschnittliche Anzahl der falsch-positiven Treffer pro Bild angibt. Die Werte der Kurve werden ermittelt, indem die Detektionsergebnisse wiederholt für unterschiedliche Schwellwerte (T in Gl. 3.10) ermittelt werden. Auf eine ähnliche Weise wird auch die Posenschätzung bewertet. In diesem Fall steht die vertikale Achse für den Anteil der korrekt ermittelten Posen, während die horizontalen Achse die Winkeltoleranz in Grad angibt.

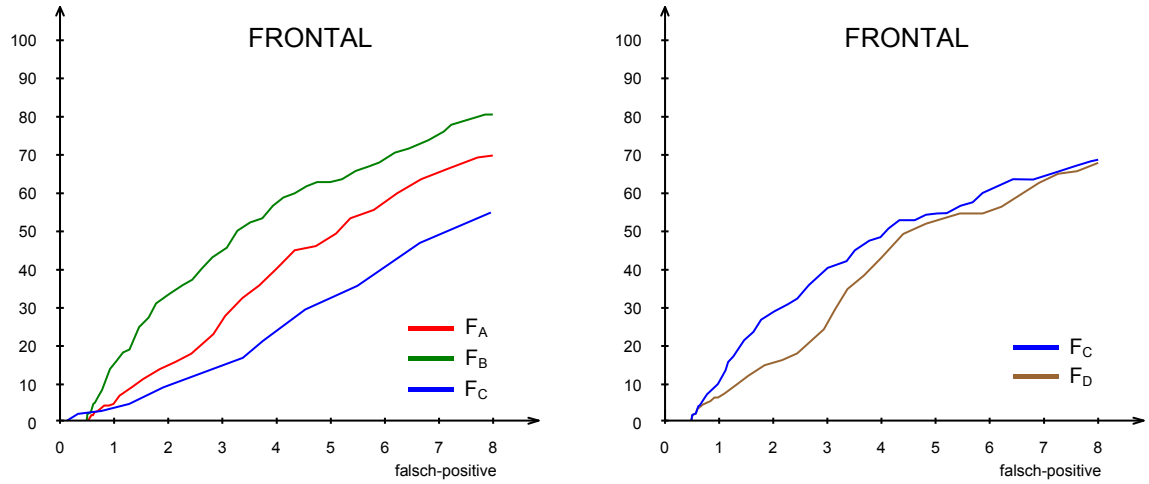

Abbildung 8.1: ROC-Kurven für das Testset FRONTAL. Links: Unterschiedliche Netzgrößen; Rechts: Mit und ohne implizite Posenschätzung.

Abbildung 8.1 zeigt die Kurven für das Testset FRONTAL. Das Diagram links zeigt die Kurvenverläufe für die Netze F_A, F_B und F_C. Wie schon im Training beobachtet wurde, werden

die besten Ergebnisse mit dem Netz F_B erzielt. Dies untermauert die dort gemachte Vermutung, dass es bei Netz F_A aufgrund seiner hohen Anzahl an Parametern (relativ zur Größe der Trainingsmenge) zu einer Überanpassung gekommen ist. Netz F_C hat hingegen eine zu kleine Parameterzahl und liefert in Folge dessen die schlechtesten Resultate. Zu den Ergebnissen

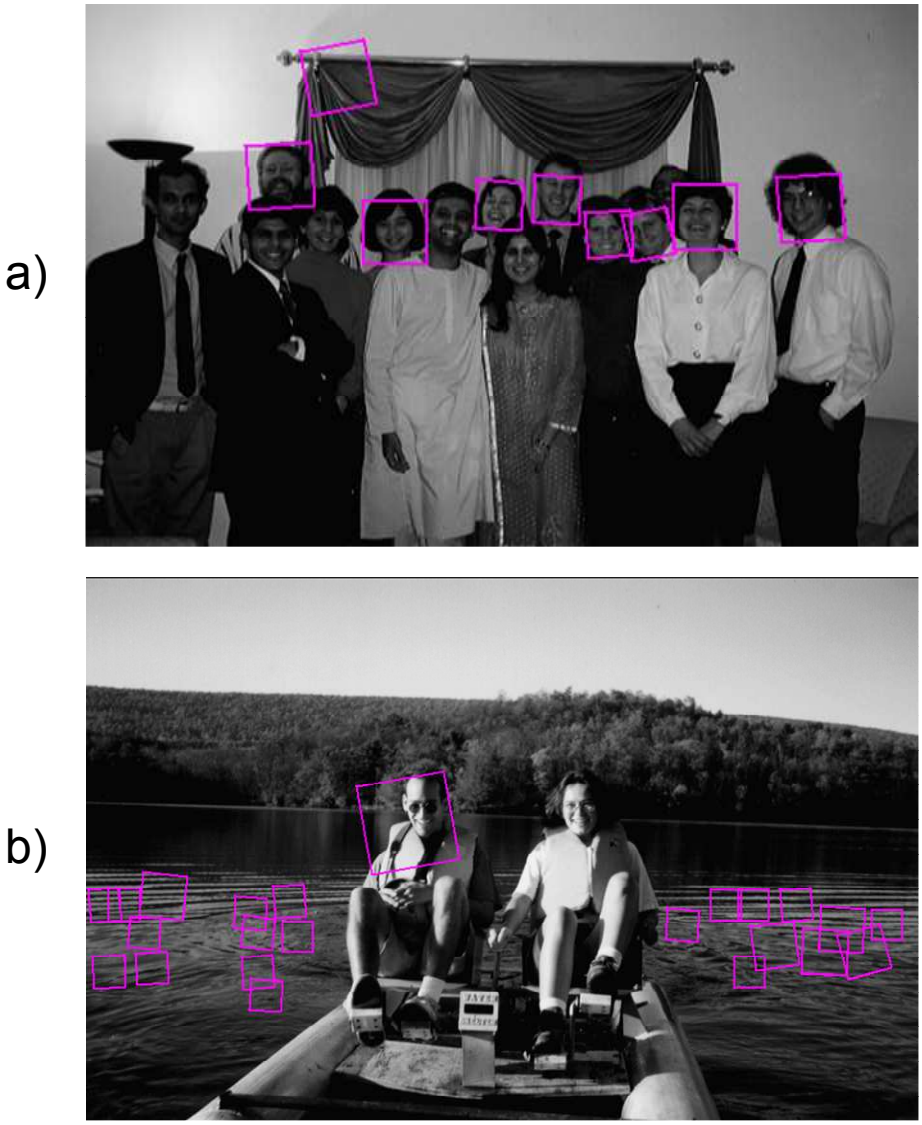

Abbildung 8.2: Zwei Bilder aus dem Set FRONTAL mit Ergebnissen bei einem mittelgroßen Schwellwert. Während bei a) nur ein falsch-positiver Treffer vorkommt, kommen bei b) Fehldetetkionen in großer Zahl vor.

die in [Osadchy u. a. 2007] für dieses Testset berichtet wurden, ist hier ein qualitativer Unterschiede zu verzeichnen. Dort wird eine Detektionsrate von 88% bei durchschnittlich 1,28 falsch-positiven Treffern pro Bild erreicht, während hier bei einer Detektionsrate von 81% im Schnitt bereits 8 falsch-positive Treffer auftreten. Ein Qualitätsunterschied war zu erwarten, bedenkt man, dass für diese Arbeit eine um den Faktor 5 kleinere Trainingsmenge verwendet wurde als in [Osadchy u. a. 2007]. In Abschnitt 7.2.3 wurde gezeigt, dass ein evidenter Zusammenhang zwischen der Größe der Trainingsmenge und dem Trainingserfolg besteht. Darüber

hinaus erklärt sich die relativ hohe Zahl der falsch-positiven Treffer wie folgt: Während bei vielen Testbildern keine oder nur sehr wenige falsch-positive Treffer vorkommen, gibt es wiederum andere Bilder, bei denen diese bereits bei mittelgroßen Schwellwerten sehr gehäuft auftreten. Abbildung 8.2 zeigt für beide Fälle ein Beispielbild aus dem Testset FRONTAL. Die Fehldetektionen in hoher Zahl auf der Wasseroberfläche bei Teil b wirken sich äußerst negativ auf die Gesamtstatistik aus. In Kapitel 7.1 wurde bereits beschrieben, wie bei der Zusammenstellung der negativen Trainingsbeispiele diesbezüglich gezielt einige Problemfälle berücksichtigt worden sind (Streifenmuster etc.). In der Praxis können aber sehr viele solcher Fälle auftreten. Für die Testsets gilt dies insbesondere, da hier eine sehr hohe Vielefalt an unterschiedlichen Hintergründen vorliegt. Es besteht aber prinzipiell die Möglichkeit, mit einem entsprechenden Arbeitsaufwand noch mehr Problemfälle abzudecken, um auf diese Weise einen steileren Kurvenanstieg zu erreichen.

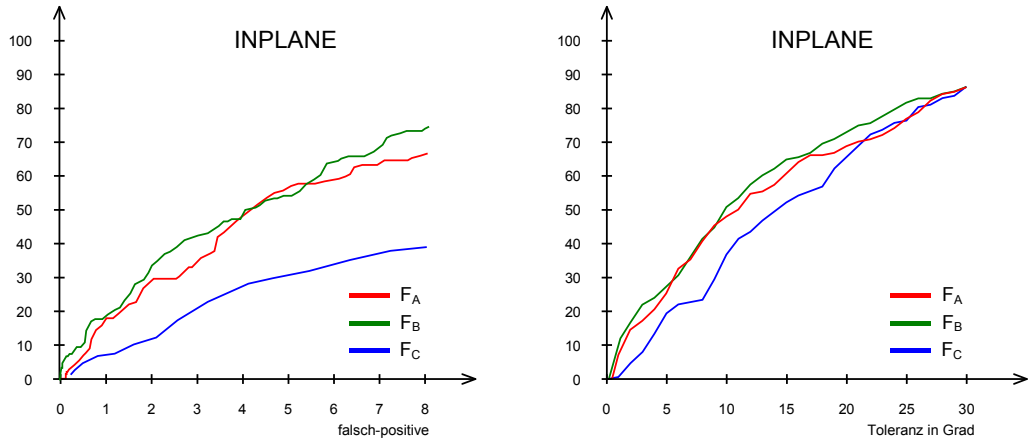

Abbildung 8.3: ROC-Kurven für das Testset INPLANE. Links: Detektionsraten; Rechts: Schätzung des Rollwinkels.

Das Diagramm rechts in Abbildung 8.1 demonstriert, dass durch das implizite Erlernen der Pose ein echter Gewinn erzielt wird. Wie bereits im letzten Kapitel beschrieben wurde, sind die Netze F_C und F_D mit Ausnahme der letzten Schicht identisch aufgebaut. Der Unterschied ist, das Netz F_C implizit die Pose schätzt, während Netz F_D nur ein Labeling durchführt. Für den Vergleich wurden beide Netze mir einer Trainingsmenge mit rund 3.000 Gesichtern trainiert. Alle Posenwinkel sind dabei kleiner als 45°. Zunächst fällt auf, dass das Netz F_C durch die Spezialisierung auf diesem Testset bessere Resultate liefert, als zuvor. Die wichtigere Erkenntnis ist jedoch, dass es ebenfalls bessere Ergebnisse erzielt als das Netz F_D. Der Kurvenverlauf ist insbesondere im vorderen Bereich deutlich steiler.

Abbildung 8.3 zeigt die Ergebnisse für das Testset INPLANE. Für die unterschiedlichen Netze sind im Diagramm links die Detektionsraten angegeben, während das Diagramm rechts die Ergebnisse zur Schätzung des Rollwinkels darstellt. Analog zeigt Abbildung 8.4 die Ergebnisse für das Testset PROFILE. Das Diagramm links zeigt wieder die Detektionsraten, während rechts die Ergebnisse der Posenschätzung bezüglich des Gierwinkels dargestellt sind.

Bei allen Testsets ergeben sich recht ähnliche Verläufe. Das Netz F_B erreicht insgesamt die besten Ergebnisse. Beim FRONTAL- und PROFILE-Set werden die Hälfte der Gesichter mit durchschnittliche drei falsch-positiven Treffern pro Bild gefunden. Für das INPLANE-Set gilt das gleiche mit durchschnittlich vier falsch-positiven Treffern. Bei durchschnittlich acht falsch-positiven Treffern wird beim FRONTAL-Set 81%, beim INPLANE-Set 75% und beim PROFILE-Set 83% der Gesichter detektiert. Zum Vergleich sei hier erwähnt, dass sowohl in [Osadchy u. a. 2007] als auch in [Viola und Jones 2003] für das INPLANE-Set eine Detektionsrate von 90% bei 4,25 und für das PROFILE-Set von 83% bei 3,25 falsch-positiven Treffern pro Bild berichtet wurde.

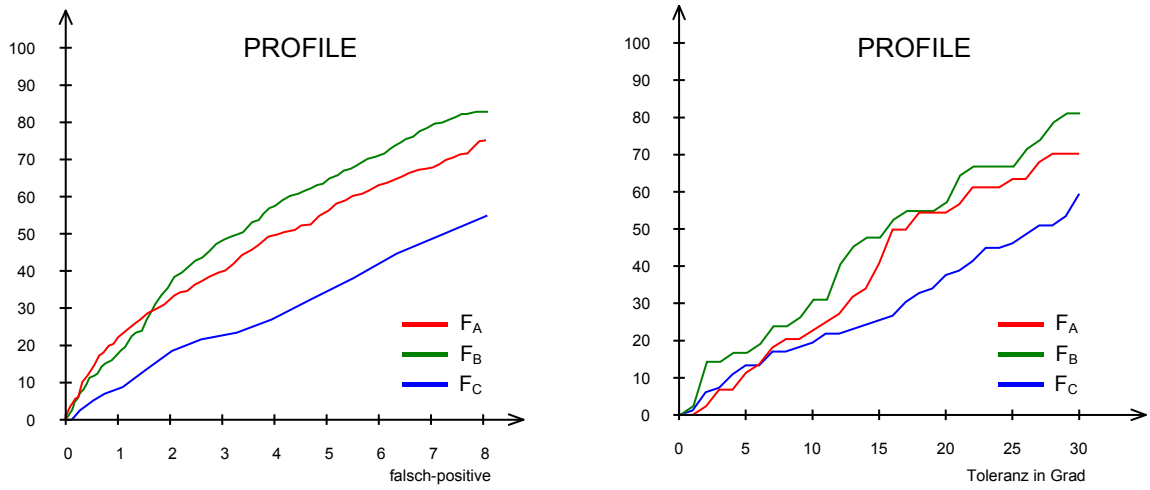

Abbildung 8.4: ROC-Kurven für das Testset PROFILE. Links: Detektionsraten; Rechts: Schätzung des Gierwinkels.

Die Kurven für die Posenwinkel zeigen, dass bei kleinen Winkeltoleranzen nur sehr wenig Posen richtig geschätzt werden. Bis zu einem gewissen Grad ist dies auch der Ungenauigkeit bei der manuellen Annotierung zuzuschreiben. Jedoch werden bei einer etwas größeren Toleranz von ca. 15° die Rollwinkel und bei einer Toleranz von ca. 20° auch die Gierwinkel zu einem Anteil von rund 70% richtig geschätzt. Des Weiteren ist zu erkennen, dass die Kurve beim Rollwinkel steiler als beim Gierwinkel verläuft. Die Ergebnisse sind hier also insgesamt etwas besser.

8.2 Tests im Konferenzraum

Der „intelligente Raum“ des Instituts für Roboterforschung der technischen Universität Dortmund dient als Testumgebung für verschiedene Projekte im Bereich der Mensch-Maschinen-Interaktion. Er ist eingerichtet wie ein Konferenzraum und verfügt über mehrere Deckenkameras, die an den verschiedenen Raumecken angebracht sind. Die Vorgabe für die durchgeführten Tests ist, dass sich die Personen, die sich in diesem Raum aufhalten, natürlich

verhalten. Das bedeutet insbesondere, dass sie nicht bewusst in eine der Kameras schauen, um den Detektionsprozess zu begünstigen. Hieraus ergibt sich, dass Gesichter aus den verschiedensten Posen aufgenommen werden. Dies sind ideale Vorraussetzungen, um den hier betrachteten Gesichtsdetektor zu testen. Abbildung 8.6 illustriert dies anhand einiger Beispielaufnahmen.

Abbildung 8.5: Beispielaufnahmen des Konferenzraumes. Das Verfahren ist in der Lage Gesichter mit sehr unterschiedlichen Posen zu detektieren.

Bei den Tests wurde das Netz F_B verwendet, weil es bei den Standardtestsets die besten Resultate erzielt hat. Es wurden zwei Maßnahmen getroffen, um das Detektionsverfahren speziell auf den Einsatz im Konferenzraum abzustimmen. Zunächst wurde ausgehend von einer Bildgröße von 378x278 px der Suchbereich auf Gesichter im Größenbereich von $10px < G < 40px$ eingegrenzt (entspricht fünf Skalierungsstufen). Hierbei wird angenommen, dass ein gewisser Mindestabstand zwischen den Personen im Raum und der jeweiligen Kamera besteht. Auf diese Weise wird zum Einen die Anzahl der falsch-positiven Treffer reduziert und zum Anderen die Ausführungsgeschwindigkeit erhöht. Mit dem in Kapitel 6.3 beschriebenen Verfahren lassen sich so im Schnitt 5,4 Einzelbilder pro Sekunde bearbeiten (vgl. Abschnitt 8.3). Als zweite Maßnahme wurde für das Training die Menge der negativen Trainingsbeispiele dahingehend modifiziert, dass sie zu 10% Beispielbilder enthält, die aus Aufnahmen des Konferenzraumes entnommen wurden. Hierdurch reduziert sich die Rate der

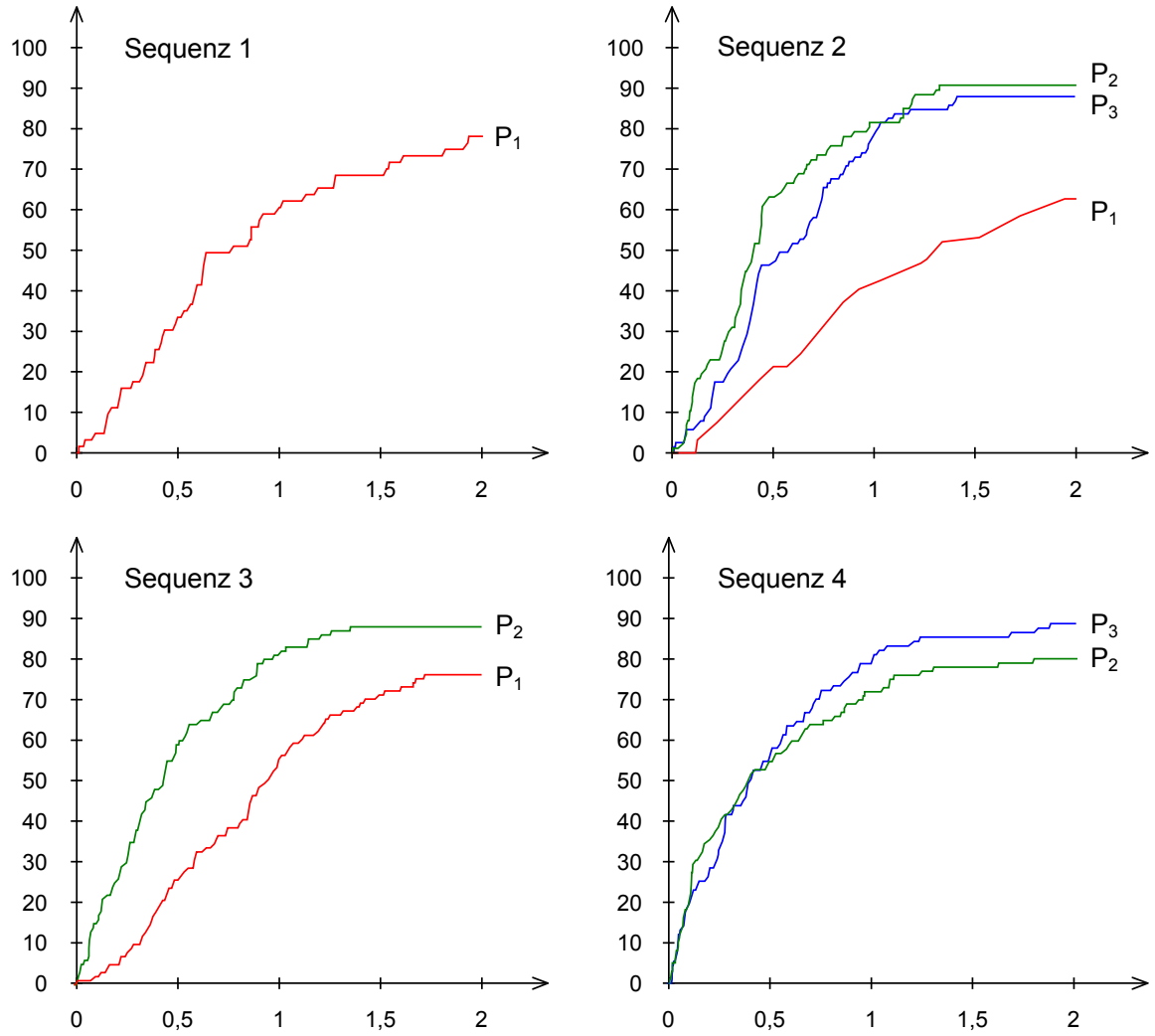

Abbildung 8.6: Ergebnisse für die vier Testsequenzen.

falsch-positiven Treffer noch weiter. Mit diesen Vorraussetzungen wurden vier Testsequenzen untersucht. Jede Sequenz besteht aus 60 bis 83 Einzelbildern und wurde jeweils von einer anderen Kamera aufgenommen. Es sind Szenen zu sehen, bei denen bis zu drei Personen im Konferenzraum agieren. Eine qualitative Beschreibung der Szenen findet sich in Anhang B. Abbildung 8.6 zeigt die Ergebnisse der Untersuchung. Bei der Auswertung wurden die Ergebnisse auf die einzelnen Personen (P_1, P_2 und P_3) bezogen.

Person P_1 neigt bei allen Sequenzen dazu, auf den Boden zu sehen. Hierdurch ergeben sich große Nickwinkel (größer als 45°), wodurch die Detektion etwas unzuverlässiger wird. Hierdurch sind die Detektionsraten bei Person P_1 etwas niedriger, als bei den anderen Personen. Bei durchschnittlich einer Fehldetektion pro Bild wird Person P_1 in rund 60% der Fälle bei Sequenz 1, in 40% der Fälle bei Sequenz 2 und in 55% aller Fälle bei Sequenz 3 detektiert. Bei 2 falsch-positiven Treffern gelten in dieser Reihenfolge Detetktionsraten von 78%, 63% bzw. 75%.

Für Person P_2 konnten bei der zweiten und dritten Sequenz gute Ergebnisse erzielt wer-

den. Bei beiden Sequenzen dreht die Person zwischenzeitlich ihren Kopf, so dass hier sowohl Frontal- als auch Halb- und Vollprofilansichten vorliegen. Etwas schlechter sind die Ergebnisse in der vierten Sequenz, da der Gierwinkel hier stellenweise größer als 90° wird, was im Trainingsset nicht vorgesehen war. Bei durchschnittlich einer Fehldetektion liegen die Detektionsraten bei rund 80% für Sequenz 2, 81% für Sequenz 3 und 70% für Sequenz 4. Bei zwei falsch-positiven Treffern wird bei den jeweiligen Sequenzen das Gesicht in 91%, 89% bzw. 80% der Fälle gefunden.

Person P_3 ist in der zweiten Sequenz vorwiegend im Vollprofil und in der vierten Sequenz überwiegend aus einer Frontalansicht zu sehen. Mit beiden Einstellungen kommt der Detektor gut zurecht. So konnten das Gesicht bei den Sequenzen 2 und 4 in 76% bzw. 80% der Fälle bei durchschnittlich einer Fehldetektion und in 88% bzw. 89% der Fälle bei zwei Fehldetektionen gefunden werden.

Insgesamt zeigen die Untersuchungen, dass durch die Spezialisierung des Detektors auf eine bestimmte Umgebung, gute Resultate erzielt werden können. Der Versuch, mit einer relativ kleinen Trainingsmenge ein funktionierenden Gesichtsdetektor zu realisieren, hat sich somit als durchführbar erwiesen.

8.3 Laufzeiten

In diesem Abschnitt werden die Laufzeiten des Gesichtsdetektors für unterschiedliche Bildgrößen betrachtet. Die Zeiten wurden für einen Graphikchip des Modells „nVidia GeForce 8800 GT“ gemessen, wobei das in Kapitel 6.3 beschriebenen Parallelisierungsverfahren verwendet wurde. Zusätzlich werden zum Vergleich die Laufzeiten für eine CPU des Modells „AMD Athlon XP 2400+“ angegeben (ohne besondere Hardwareoptimierung). Die Ergebnisse der durchgeführten Messungen zeigt Tabelle 8.1, wobei alle Angaben in Millisekunden pro Bild erfolgen. Die Spalten geben an, auf wie vielen Skalierungsstufen die Detektion ausgehend von der Originalgröße durchgeführt wurden. (Die Anzahl der Skalierungsstufen entspricht dem Suchbereich bzgl. der Gesichtsgröße). Es wurde bei allen Messungen das Fal-

		1	2	3	4	5	6	7	8	9
800x600	GPU	318	488	581	619	647	672	695	715	733
	CPU	6826	10176	11625	12462	12799	12970	12996	13003	13015
640x480	GPU	209	312	373	407	434	456	477	497	X
	CPU	4355	6470	7408	7936	8102	8173	8246	8318	X
378x278	GPU	73	109	137	162	184	203	222	X	X
	CPU	1402	2100	2422	2539	2564	2596	2625	X	X

Tabelle 8.1: Laufzeiten der Detektion für einen Graphikchip (GPU) und einer CPU. Zeilen: Größe des Eingabebildes; Spalten: Anzahl der Skalierungsstufen. Angaben in Millisekunden pro Bild.

tungsnetz F_B verwendet. Die Ergebnisse lassen sich aber näherungsweise anhand der Anzahl der Netzparameter auch auf andere Netzgrößen umrechnen, da der Parallelisierungsgrad weniger von der Netzstruktur, als viel mehr von den Hardwarebeschränkungen des Graphikchips abhängt. Die Zahlen zeigen, dass auf der Graphikkarte gegenüber der seriellen Ausführung auf der Vergleichs-CPU eine Steigerung um einen Faktor von annähernd 20 erreicht werden konnte.

8.4 Zusammenfassung

In diesem Kapitel wurde das hier betrachtete Detektionsverfahren ausführlich evaluiert. Zunächst wurden die Ergebnisse für drei Standardtestsets für unterschiedliche Posenbereiche betrachtet. Es hat sich gezeigt, dass der Detektor für alle Sets ähnliche Ergebnisse erzielt. Er weist also unabhängig von der Pose eine gleich bleibende Qualität auf. Des Weiteren wurde in einem direkten Vergleich gezeigt, dass die implizite Posenschätzung einen echten Vorteil gegenüber einem einfachen Labeling bietet. Insgesamt unbefriedigend war die relativ hohe Rate an falsch-positiven Treffern bei hohen Detektionsraten. Im zweiten Teil dieses Kapitels wurde jedoch gezeigt, wie durch die Anpassung des Detektors an seine Einsatzumgebung diese Rate merklich reduziert werden konnte. Bei den Testsequenzen im Konferenzraum konnten für die Gesichter der einzelnen Personen hohe Detektionsraten mit einer geringen Fehlerquote erzielt werden. Abschließend wurde in diesem Kapitel die Geschwindigkeitssteigerung (bis zu Faktor 20) aufgezeigt, die durch eine Parallelisierung des Detektionsverfahrens mit Hilfe einer Grafikkarte erbracht werden konnte.

Kapitel 9

Zusammenfassung und Ausblick

Im Folgenden werden die wichtigen Punkte dieser Arbeit noch einmal zusammengefasst. Im Anschluss daran folgt ein Ausblick.

Einleitend wurde in Kapitel 1 die besondere Bedeutung der Gesichtsdetektion im Bereich der Robotik und Mensch-Maschinen-Interaktion hervorgehoben. Des Weiteren wurden hier die Ziele dieser Arbeit genannt. Das in [Osadchy u. a. 2007] beschriebene, poseninvariante Gesichtsdetektionsverfahren sollte mit einem angemessenem Aufwand, d.h. einer moderaten Größe der Trainingsdatenmenge und der Faltungsnetzstruktur, realisiert werden. Um dabei eine hohe Ausführungsgeschwindigkeit zu erreichen, ist die Verwendung eines Grafikkarte vorgesehen worden. Ein besonderes Augenmerk sollte auch auf die Fortschritte beim Training von Faltungsnetzen für unterschiedliche Parametereinstellungen gelegt werden. Als Testumgebung für den fertigen Detektor wurde ein Konferenzraum mit Deckenkameras vorgesehen.

In Kapitel 2 wurde die Problemstellung der Gesichtsdetektion im Allgemeinen betrachtet. Als besondere Herausforderung wurde dabei das Problem der Posenvariation hervorgehoben und der Begriff der Posenschätzung definiert. Des Weiteren wurde ein Querschnitt zu verschiedenen Herangehensweisen zur Entwicklung eines Gesichtsdetektors aufgezeigt. Darüber hinaus wurden verwandte Verfahren betrachtet, die entweder einen Schwerpunkt auf das Posenproblem legen oder aber neuronale Netze verwenden. Dabei wurden die beiden wichtigsten Vorteile des hier betrachteten Verfahrens hervorgehoben. Der erste Vorteil ist die Integration der Posenschätzung in den Detektionsprozess. Hierdurch wird eine Verbesserung der Detektion erreicht und eine einheitliche Beschreibung des Gesamtprozesses ermöglicht. Der zweite Vorteil ist die Geschwindigkeitssteigerung, die durch die besondere Struktur der Faltungsnetze im Vergleich zu herkömmlichen neuronalen Netzen erreicht werden kann.

Eine Übersicht zu dem Gesamtverfahren wurde in Kapitel 3 gegeben. Dabei wurde erläutert, wie die Posenschätzung in den Detektionsprozess integriert wird. Hierzu wird die Mannigfaltigkeit der Gesichtsposen im euklidischen Raum abgebildet, so dass die Pose und das

Label eines Gesichts einheitlich beschrieben und mit Hilfe eines Mapping-Moduls ermittelt werden können. Das Mapping-Modul wird durch ein Faltungsnetz realisiert. Faltungsnetze stellen eine Spezialform vorwärtsgerichteter, mehrschichtiger Perzeptrons dar, deren Aufbau in Kapitel 4 im Rahmen der Grundlagen neuronaler Netze beschrieben wurde. Des Weiteren wurde hier auch die Funktionsweise des Backpropagation-Verfahrens erläutert und als relevante Erweiterung hierzu wurde das Prinzip der geteilten Gewichte erklärt. In Kapitel 5 wurden schließlich die Struktur und die Funktionsweise von Faltungsnetzen erläutert. Sie bestehen aus mehreren Schichten von Faltungsoperationen, Unterabtastungen und vollen Verbindungen, die durch eine Vernetzung von Neuronen modelliert werden. Ein wichtiger Aspekt, der anhand einiger Beispiele verdeutlicht wurde, ist die universelle Verwendbarkeit von Faltungsnetzen für die verschiedensten Problemstellungen.

In Kapitel 6 ging es um eine effiziente Implementierung des Lokalisierungsverfahrens. Durch das von Faltungsnetzen verwendete Prinzip der geteilten Gewichte, entstehen durch eine wiederholte Anwendung des Faltungsnetzes auf überlappende Bildbereiche Rechenredundanzen. Um diese vollständig einzusparen, wurde der Ansatz gewählt, alle Faltungsnetzoperationen stets auf das gesamte Eingabebild anzuwenden. Um schließlich Gesichter unterschiedlicher Größe finden zu können, wird dieses Vorgehen wiederholt auf unterschiedlichen Skalierungsstufen angewendet. Um einen weiteren Geschwindigkeitszuwachs zu erreichen, wurden Möglichkeiten der Parallelisierung aufgezeigt. Dabei wurde die Implementierung des Verfahrens für einen Graphikchip beschrieben, die im wesentlichen darauf beruht, das Eingabebild in Rechtecke zu zerlegen, die von mehreren Threads parallel bearbeitet werden.

Die konkrete Umsetzung des Trainings sowie eine ausführliche Untersuchung der Trainingsfortschritte erfolgte in Kapitel 7. Es wurde gezeigt, dass die verwendete Menge der Gesichtsbilder eine hohe Posenvielfalt aufweist. Für ein ausgewogenes Training bzgl. der Posenschätzung ist eine Gleichverteilung der Posenwinkel erforderlich, die im Falle des Gierwinkels durch die geeignete Zusammenstellung der Trainingsbilder und im Falle des Rollwinkels durch eine entsprechende Vorverarbeitung (Rotation) dieser Bilder erreicht wurde. Es konnten also eine integrierte Posenschätzung für zwei Winkel realisiert werden. Mit zwei verschiedenen Trainingsstrategien konnten gute Erfolge erzielt werden. Bei der einen Strategie wurde eine absteigende globale Lernrate verwendet, bei der Anderen hingegen wurden individuelle Lernraten für jeden Netzparameter bestimmt. Bei den Untersuchungen zeigte sich, dass generell eine möglichst große Trainingsdatenmenge vorteilhaft ist. Es stellte sich dabei jedoch auch heraus, dass die Größe des Faltungsnetzes auf die Trainingsmenge abgestimmt sein sollte. Abschließend wurde noch auf die geringen Laufzeiten des Trainings von nur wenigen Stunden hingewiesen.

Bei der Evaluierung in Kapitel 8 wurde anhand von drei Standardtestsets gezeigt, dass die Qualität der Detektion bei verschiedenen Winkelbereichen annähernd gleich bleibend ist, und dass durch die Integration der Posenschätzung im Vergleich zu einem einfachen

Labeling tatsächlich ein Gewinn erzielt wird. Die durchschnittlich relativ hohe Anzahl an falsch-positiven Treffern konnte für den Einsatz des Detektors im Konferenzraum durch eine geeignete Spezialisierung merklich reduziert werden. Für die untersuchten Testsequenzen konnten so gute Ergebnisse erzielt werden. Die geringen Laufzeiten des Verfahrens, die mit Hilfe der Graphikkarte erzielt werden konnten, waren ebenfalls zufriedenstellend. Die Eingangs in dieser Arbeit aufgestellten Ziele konnten somit erfüllt werden.

Insgesamt betrachtet konnte das hier untersuchte Gesichtsdetektionsverfahren überzeugen. Dies gilt insbesondere deshalb, weil eine stetige Weiterentwicklung des Verfahrens möglich ist und vielversprechend erscheint. Bei der Steigerung der Parallelisierung sind kaum physikalische Grenzen gesetzt. Dank der Computerspielindustrie wird entsprechende Hardware für den Endverbrauchermarkt laufend verbessert. So ist es wahrscheinlich, dass schon in wenigen Jahren kostengünstig echtzeitfähige neuronale Netze realisiert werden können, die um ein vielfaches Größer sind, als die, die in dieser Arbeit verwendet wurden. Es ist zu erwarten, dass dann auch die Qualität der Detetktion in einem entsprechenden Umfang steigt. Neben der Posenvariation könnten dann auch andere Probleme in den Detektionsprozess miteinbezogen werden. Beispielsweise könnte der Detektor implizit das Geschlecht einer Person erkennen oder prüfen, ob sie eine Brille trägt. Das Hauptproblem dabei ist jedoch Folgendes: Je mehr Wissen sich das Faltungsnetz aneignen soll, desto größer sind auch die Anforderungen an die Trainingsdatenmenge. Dies gilt auch deswegen, weil mit der Größe des Netzes auch die Anzahl der Trainingsbeispiele steigen sollte. Müssen dabei noch bestimmte Annotierungsdaten hinzugefügt werden, wie im Falle der Pose, erhöht sich der Aufwand entsprechend noch mehr. Eine sinnvolle Verbesserungsmöglichkeit wäre es deshalb, mehr Informationen aus den Trainingsdaten zu beziehen. Dies kann darin bestehen, Farbinformationen zu berücksichtigen. Hierzu müsste das Faltungsnetz um zusätzliche Eingänge und eigene Felder für die unterschiedlichen Farbkanäle erweitert werden.

Die hier gegebenen Anregungen zeigen, dass zukünftig hinreichend Möglichkeiten für weitere Entwicklungen und Untersuchungen bezüglich des Themas „Faltungsnetze zur Gesichtsdetektion“ bestehen. Darüber hinaus lässt sich die im Rahmen dieser Arbeit gelieferte Implementierung auch hervorragend nutzen, um Versuche nicht nur mit Gesichtern, sondern auch mit anderen Objektklassen durchzuführen.

Anhang A

Faltungsnetzstrukturen

Im Folgenden werden die Strukturen der vier in dieser Arbeit verwendeten Faltungsnetze (F_A, F_B, F_C und F_D) in tabellarischer Form angegeben. Die Spalte „Typ“ gibt die Verbindungsart zur jeweils vorangegangenen Schicht bzw. zum Eingabebild an. Bei Faltungen ist in Klammern die Größe der Filtermaske angegeben. Unterabtastungen halbieren stets die Dimensionen. Das Symbol „‖“ gibt an, dass jedes Feld mit genau einem anderen Feld der vorangegangenen Schicht verbunden ist, und dass alle Verbindungen parallel verlaufen. Das Symbol „*“ kennzeichnet eine Symmetriebrechung. Eine zusätzliche Tabelle zeigt in dem Fall an, welche Felder miteinander verbunden sind. Dort sind die Felder der mit „*“ markierten Schicht horizontal und die der vorangegangenen Schicht vertikal eingetragen. Ein „X“ markiert eine Verbindung. Es sind nur drei Tabellen angegeben, da die Netze F_C und F_D die gleiche Symmetriebrechung haben. Die Spalten „Größe“ und „Felder“ beziehen sich auf die Größe und Anzahl der Felder. Die letzten drei Spalten beziehen sich auf die Neuronen und deren Parameter. Die Anzahl der Kanten und Gewichte können aufgrund geteilter Gewichte voneinander abweichen (siehe Kapitel 4.3.3).

Faltungsnetz F_A (nach [Osadchy u. a. 2007])

Schicht	Typ	Größe (px)	Felder	Neuronen	Kanten	Gewichte
S_1	Faltungen (5x5)	28x28	8	6272	163072	208
S_2	Unterabtastungen ‖	14x14	8	1568	7840	16
S_3	Faltungen (5x5) *	10x10	20	2000	202000	2020
S_4	Unterabtastungen ‖	5x5	20	500	2500	40
S_5	volle Verbindung	1x1	120	120	60120	60120
S_6	volle Verbindung	1x1	9	9	1089	1089
$\sum$			185	10469	436621	63493

Tabelle A.1: Struktur von Faltungsnetz F_A. Das Eingabebild hat eine Größe von 32x32 px.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	X						X	X	X					X	X	X	X		X	X
1	X	X						X	X	X					X	X	X	X		X
2	X	X	X						X	X	X					X	X	X	X	X
3		X	X	X					X	X	X	X						X	X	X
4			X	X	X					X	X	X	X						X	X
5				X	X	X					X	X	X	X			X			X
6					X	X	X					X	X	X	X		X	X	X	X
7						X	X	X					X	X	X	X		X	X	X

Tabelle A.2: Symmetriebrechung in Faltungsnetz F_A zwischen den Schichten S_2 und S_3.

Faltungsnetz F_B

Schicht	Typ	Größe (px)	Felder	Neuronen	Kanten	Gewichte
S_1	Faltungen (5x5)	28x28	6	4704	122304	156
S_2	Unterabtastungen ‖	14x14	6	1176	5880	12
S_3	Faltungen (5x5) *	10x10	16	1600	151600	1516
S_4	Unterabtastungen ‖	5x5	16	400	2000	32
S_5	volle Verbindung	1x1	100	100	40100	40100
S_6	volle Verbindung	1x1	9	9	909	909
$\sum$			153	7989	322793	42725

Tabelle A.3: Struktur von Faltungsnetz F_B. Das Eingabebild hat eine Größe von 32x32 px.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	X				X	X	X			X	X	X	X	X		X
1	X	X				X	X	X			X	X		X	X	X
2	X	X	X				X	X	X			X	X		X	X
3		X	X	X			X	X	X	X			X	X		X
4			X	X	X			X	X	X	X			X	X	X
5				X	X	X			X	X	X	X	X		X	X

Tabelle A.4: Symmetriebrechung in Faltungsnetz F_B zwischen den Schichten S_2 und S_3.

Faltungsnetz F_C

Schicht	Typ	Größe (px)	Felder	Neuronen	Kanten	Gewichte
S_1	Faltungen (7x7)	22x22	5	2420	121000	250
S_2	Unterabtastungen ‖	11x11	5	605	3025	10
S_3	Faltungen (7x7) *	5x5	12	300	52975	2119
S_4	volle Verbindung	1x1	90	90	27090	27090
S_5	volle Verbindung	1x1	9	9	819	819
$\sum$			121	3424	204909	30288

Tabelle A.5: Struktur von Faltungsnetz F_C. Das Eingabebild hat eine Größe von 28x28 px.

	0	1	2	3	4	5	6	7	8	9	10	11
0	X			X	X	X		X	X	X	X	X
1	X	X			X	X	X		X	X		X
2	X	X	X			X	X	X		X	X	X
3		X	X	X		X	X	X	X			X
4			X	X	X		X	X	X	X	X	X

Tabelle A.6: Symmetriebrechung in den Netzen F_C und F_D zwischen den Schichten S_2 und S_3.

Faltungsnetz F_D

Schicht	Typ	Größe (px)	Felder	Neuronen	Kanten	Gewichte
S_1	Faltungen (7x7)	22x22	5	2420	121000	250
S_2	Unterabtastungen ‖	11x11	5	605	3025	10
S_3	Faltungen (7x7) *	5x5	12	300	52975	2119
S_4	volle Verbindung	1x1	90	90	27090	27090
S_5	volle Verbindung	1x1	1	1	91	91
$\sum$			113	3416	204181	29560

Tabelle A.7: Struktur von Faltungsnetz F_D. Es unterscheidet sich von Netz F_C nur in der letzten Schicht.

Anhang B

Testsequenzen

Im Folgenden wird eine qualitative Beschreibung der vier Testsequenzen des Konferenzraumes gegeben, die in Kapitel 8 untersucht wurden.

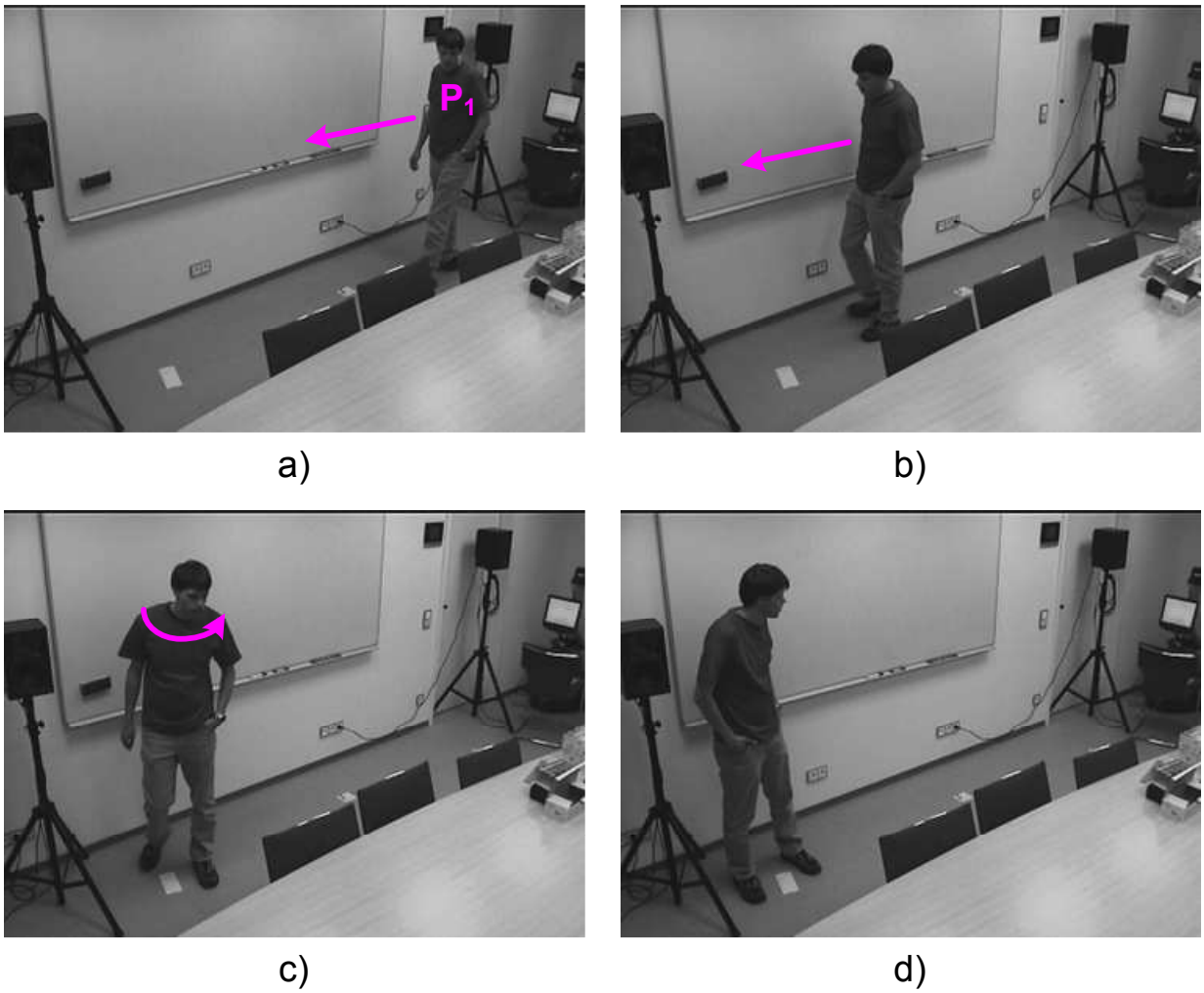

Abbildung B.1: Sequenz 1; Bildnummern: 3, 29, 38 und 75

Sequenz 1

Die erste Sequenz besteht aus 75 Einzelbildern und zeigt eine Person (P_1). Wie in Abbildung B.1 zu sehen ist, läuft die Person von rechts-oben in die Szene hinein. Zunächst ist sie im Halbprofil (Teil a) und später im Vollprofil (Teil b) zu sehen. Danach dreht sie ihren Kopf nach links (Teil c) und ist gegen Ende wieder im Profil zu sehen (Teil d). Da die Person die ganze Zeit über zum Boden sieht, ist der Nickwinkel entsprechend groß.

Sequenz 2

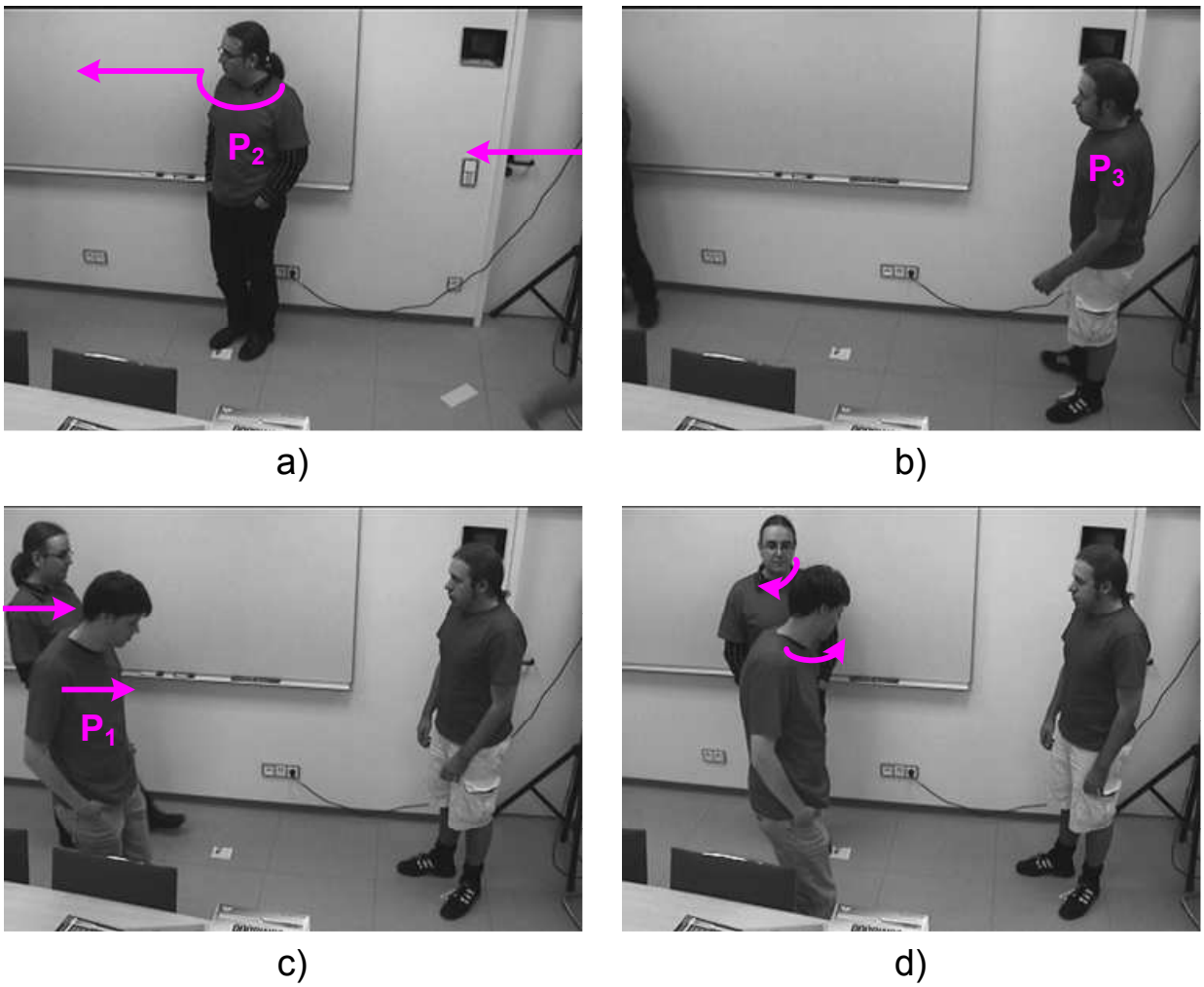

Abbildung B.2: Sequenz 2; Bildnummern: 2, 36, 75 und 82

Abbildung B.2 zeigt Ausschnitte aus der zweiten Sequenz. Sie besteht aus 83 Einzelbildern und zeigt drei Personen (P_1, P_2 und P_3). Zu Beginn dreht Person P_2 ihren Kopf nach rechts (Teil a) und läuft aus der Szene heraus. Gleichzeitig betritt Person P_3 die Szene (Teil b) und ist dabei im Profil zu sehen. Später betreten die Personen P_1 und P_2 die Szene (Teil c) und sind dabei ebenfalls im Profil zusehen. Gegen Ende dreht Person P_2 ihren Kopf nach

rechts und ist von Vorne zu sehen (Teil d). Gleichzeitig dreht Person P_1 ihren Kopf von der Kamera weg.

Sequenz 3

Die dritte Sequenz (Abbildung B.3) besteht aus 76 Einzelbildern und zeigt zwei Personen (P_1 und P_2). Person P_2 ist zunächst weit vorne im Bild im Profil zusehen (Teil a), dreht sich dann von der Kamera weg und läuft nach hinten. Hier dreht sie sich um und ist wieder im Profil zusehen (Teil b). Nun betritt Person P_1 die Szene und beide Personen laufen nach vorne, wobei die Gesichter teilweise frontal und teilweise im Halbprofil zu sehen sind (Teil c). Schließlich drehen sie die Köpfe zueinander hin und sind wieder im Profil zu sehen (Teil d). Person P_1 schaut auch hier wieder durchgängig nach unten.

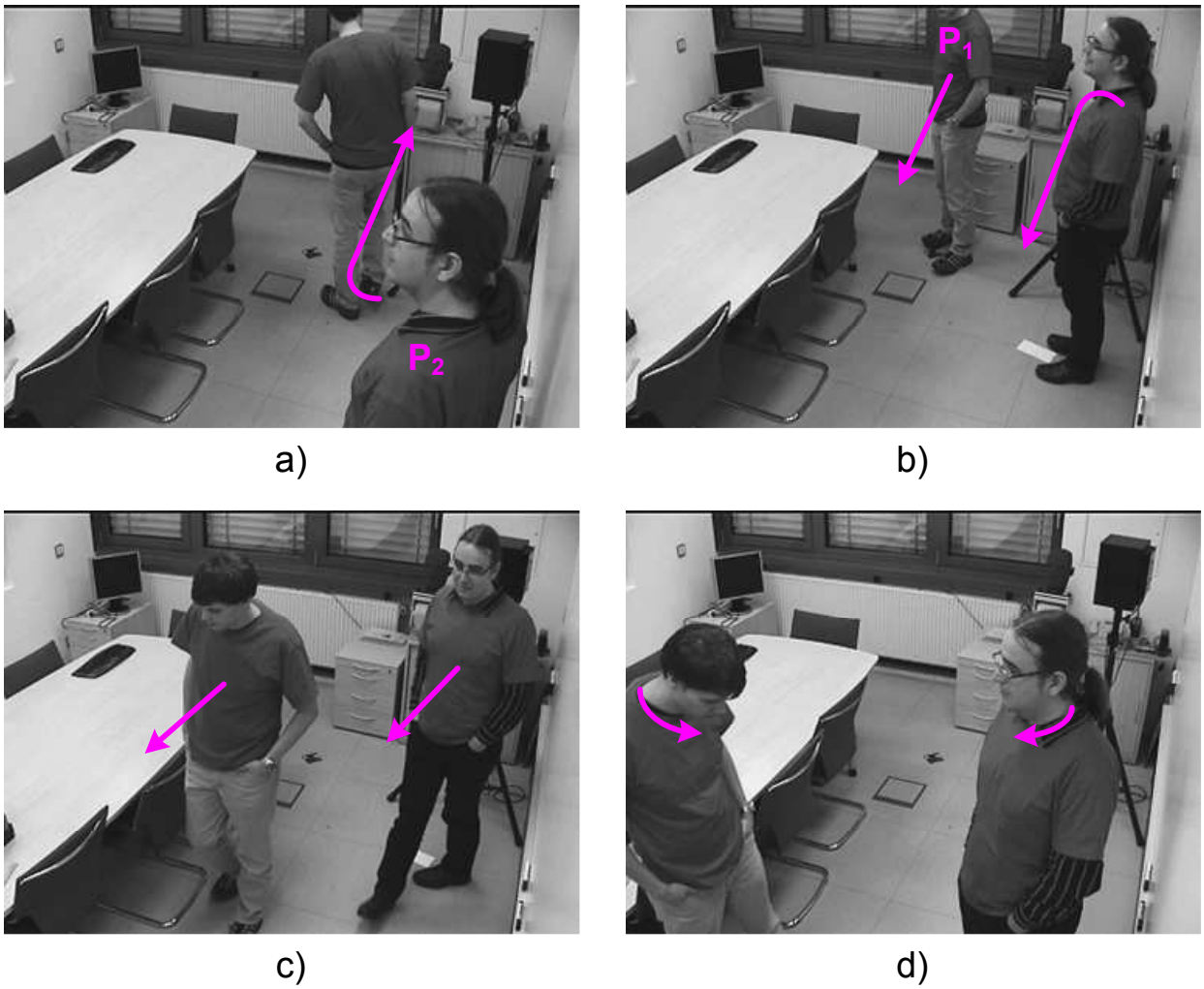

Abbildung B.3: Sequenz 3; Bildnummern: 3, 32, 51 und 72

Sequenz 4

Abbildung B.4 zeigt Ausschnitte der vierten Sequenz. Sie besteht aus 60 Einzelbildern und zeigt die Personen P_2 und P_3. Person P_3 ist zu Begin oben im Bild zu sehen (Teil a). Sein Gesicht bewegt sich vorübergehend von der Kamera weg und aus der Szene heraus. Person P_2 bewegt sich links auf die Wand zu und dreht sich danach so, dass sein Gesicht im Profil zu sehen ist (Teil b). Danach dreht er sich nach links, so dass nur noch sein Hinterkopf zu sehen ist (Teil c). Gleichzeitig bewegt sich Person P_3 auf die Kamera zu, so das sein Gesicht wieder erkennbar ist. Zunächst ist sein Gesicht frontal (Teil c), später im Halbprofil (Teil d) zu sehen.

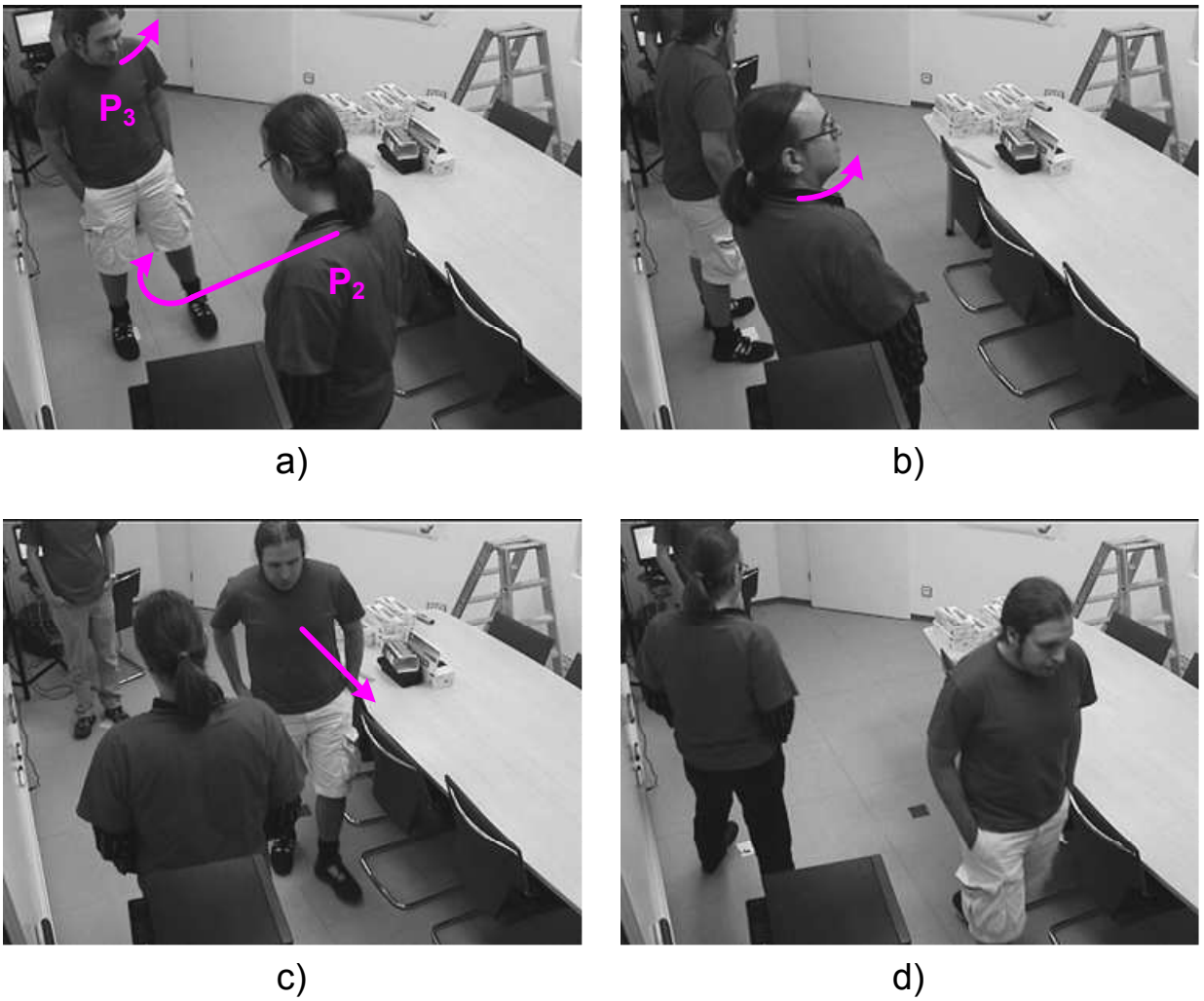

Abbildung B.4: Sequenz 4; Bildnummern: 1, 14, 47 und 66

Literaturverzeichnis

[Bishop 1995] Bishop, Christopher M.: *Neural Networks for Pattern Recognition*. Oxford University Press, 1995

[Chopra u. a. 2005] Chopra, Sumit ; Hadsell, Raia ; LeCun, Yann: Learning a Similarity Metric Discriminatively, with Application to Face Verification. In: *Proc. of Computer Vision and Pattern Recognition Conference*, IEEE Press, 2005

[Dass und Jain 2001] Dass, Sarat C. ; Jain, Anil K.: Markov Face Models, 2001, S. 680

[El-Bakry und Stoyan 2004] El-Bakry, H.M. ; Stoyan, H.: Fast neural networks for submatrix (object/face) detection. In: *Circuits and Systems, 2004. ISCAS '04. Proceedings of the 2004 International Symposium on* 5 (2004), May, S. V–764–V–767

[Gundimada und Asari 2004] Gundimada, Satyanadh ; Asari, Vijayan: Face detection technique based on rotation invariant wavelet features, 2004, S. 157–158

[Hsu u. a. 2002] Hsu, R.L. ; Abdel-Mottaleb, M. ; Jain, A.K.: Face Detection in Color Images, 2002, S. 696–706

[Kobayashi und Zhao 2007] Kobayashi, H. ; Zhao, Qiangfu. *Face detection with clustering, lda and NN*. October 2007

[LeCun u. a. 1998a] LeCun, Y. ; Bottou, L. ; Bengio, Y. ; Haffner, P.: Gradient-Based Learning Applied to Document Recognition. In: *Proceedings of the IEEE* 86 (1998), November, Nr. 11, S. 2278–2324

[LeCun u. a. 1998b] LeCun, Y. ; Bottou, L. ; Orr, G. ; Muller, K.: Efficient BackProp. In: Orr, G. (Hrsg.) ; K., Muller (Hrsg.): *Neural Networks: Tricks of the trade*, Springer, 1998b

[LeCun u. a. 2006] LeCun, Yann ; Chopra, Sumit ; Hadsell, Raia ; Marc'Aurelio, Ranzato ; Huang, Fu-Jie: A Tutorial on Energy-Based Learning. In: Bakir, G. (Hrsg.) ; Hofman, T. (Hrsg.) ; Schölkopf, B. (Hrsg.) ; Smola, A. (Hrsg.) ; Taskar, B. (Hrsg.): *Predicting Structured Data*, MIT Press, 2006

[LeCun u. a. 2004] LeCun, Yann ; Huang, Fu-Jie ; Bottou, Leon: Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting. In: *Proceedings of CVPR'04*, IEEE Press, 2004

[Li u. a. 2000] Li, Yongmin ; Gong, Shaogang ; Liddell, Heather: Support vector regression and classification based multi-view face detection and recognition. In: *In IEEE International Conference on Automatic Face & Gesture Recognition*, 2000

[Osadchy u. a. 2007] Osadchy, M. ; LeCun, Y. ; Miller, M.: Synergistic Face Detection and Pose Estimation with Energy-Based Models. In: *Journal of Machine Learning Research 8(2007)* 8 (2007), May, S. 1197–1215

[Osadchy u. a. 2005] Osadchy, R. ; Miller, M. ; LeCun, Y.: Synergistic Face Detection and Pose Estimation with Energy-Based Models. In: *Advances in Neural Information Processing Systems (NIPS 2004)*, MIT Press, 2005, S. 1197–1215

[Propp und Samal 1992] Propp, M. ; Samal, A.: Artificial Neural network Architectures for Human Face Detection. In: *Intelligent Eng. Systems through Artificial Nerual Networks* 2 (1992)

[Rosenblatt 1958] Rosenblatt, Frank: The perceptron: a probabilistic model for information storage and organization in the brain. In: *Psychological Review* 65 (1958), November, Nr. 6, S. 386–408

[Rowley u. a. 1998] Rowley, H.A. ; Baluja, S. ; Kanade, T.: Rotation Invariant Neural Network-Based Face Detection. In: *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on* (1998), Jun, S. 963–963. – ISSN 1063–6919

[Schneiderman und Kanade 2000] Schneiderman, H. ; Kanade, T.: A statistical method for 3D object detection applied to faces and cars. In: *IEEE Conference on Computer Vision and Pattern Recognition, 2000. Proceedings.* 1 (2000), S. 746–751

[Seow u. a. 2003] Seow, Ming-Jung ; Valaparla, D. ; Asari, V.K.: Neural network based skin color model for face detection. In: *Applied Imagery Pattern Recognition Workshop, 2003. Proceedings. 32nd* (2003), October, S. 141–145

[Seshadrinathan und Ben-Arie 2003] Seshadrinathan, Manoj ; Ben-Arie, J.: Pose invariant face detection. In: *Video/Image Processing and Multimedia Communications, 2003. 4th EURASIP Conference focused on* 1 (2003), July, S. 405–410

[Sim u. a. 2003] Sim, Terence ; Baker, Simon ; Bsat, Maan: The CMU Pose, Illumination, and Expression Database. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (2003), december, Nr. 12, S. 1615–1618

[Sung und Poggio 1998] Sung, K.-K. ; Poggio, T.: Example-based learning for view-based human face detection. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 20 (1998), Jan, Nr. 1, S. 39–51. – ISSN 0162–8828

[Tivive und Bouzerdoum 2003] Tivive, F.H.C. ; Bouzerdoum, A.: A new class of convolutional neural networks (SICoNNets) and their application of face detection. In: *Neural Networks, 2003. Proceedings of the International Joint Conference on* 3 (2003), July

[Vaillant u. a. 1993] Vaillant, R. ; Monrocq, C. ; LeCun, Y.: An Original approach for the localisation of objects in images. In: *International Conference on Artificial Neural Networks*, 1993, S. 26–30

[Viola und Jones 2002] Viola, Paul ; Jones, Michael: Robust Real-time Object Detection. In: *International Journal of Computer Vision* (2002)

[Viola und Jones 2003] Viola, Paul ; Jones, Michael: Fast multi-view face detection. In: *Technical Report TR2003-96, MERL* (2003), june

[Wang und Yang 2008] Wang, Jizeng ; Yang, Hongmei: Face Detection Based on Template Matching and 2DPCA Algorithm, 2008, S. 575–579

[Waring und Liu 2005] Waring, C.A. ; Liu, Xiuwen: Face detection using spectral histograms and SVMs, 2005, S. 467–476

[Yang und Huang 1994] Yang, Guangzheng ; Huang, Thomas S.: Human face detection in a complex background, 1994, S. 53–63

[Yang u. a. 2008] Yang, Jie ; Ling, C ; Zhu, Yitan ; Zheng, Zhonglong: A face detection and recognition system in color image series, 2008, S. 531–539

[Yang u. a. 2000] Yang, Ming-Hsuan ; Abuja, N. ; Kriegman, D.: Face detection using mixtures of linear subspaces, 2000, S. 70–76

[Yang u. a. 2002] Yang, Ming-Hsuan ; Kriegman, D. J. ; Ahuja, N.: Detecting faces in images: a survey. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 24 (2002), Nr. 1, S. 34–58

[Zhang und Izquierdo 2006] Zhang, Q. ; Izquierdo, E.: Multi-Feature based Face Detection, 2006, S. 572–576

[Zhong u. a. 2007] Zhong, Jin ; Zhen, Lou ; Jingyu, Yang ; Quansen, Sun: Face detection using template matching and skin-color information, 2007, S. 794–800

Erklärung

Ich versichere, dass ich diese wissenschaftliche Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Die Stellen der Arbeit, die anderen Werken dem Wortlaut oder Sinn nach entnommen sind, wurden in jedem einzelnen Fall durch Angabe der Quelle als Entlehnung kenntlich gemacht. Das Gleiche gilt auch für beigegebene Skizzen und Darstellungen. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Dortmund, den 18. Januar 2009

Einwilligung

Hiermit erkläre ich mich damit einverstanden, dass diese wissenschaftliche Arbeit nach den Bestimmungen des §6 Absatz 1 des Gesetzes über Urheberrecht vom 9.9.1965 in die Bereichsbibliothek aufgenommen und damit für Leser der Bibliothek öffentlich zugänglich gemacht wird.

Ferner bin ich damit einverstanden, dass gemäß §54 Absatz 1 Satz 1 dieses Gesetzes Leser zu persönlichen wissenschaftlichen Zwecken Kopien aus der Arbeit anfertigen dürfen.

Dortmund, den 18. Januar 2009