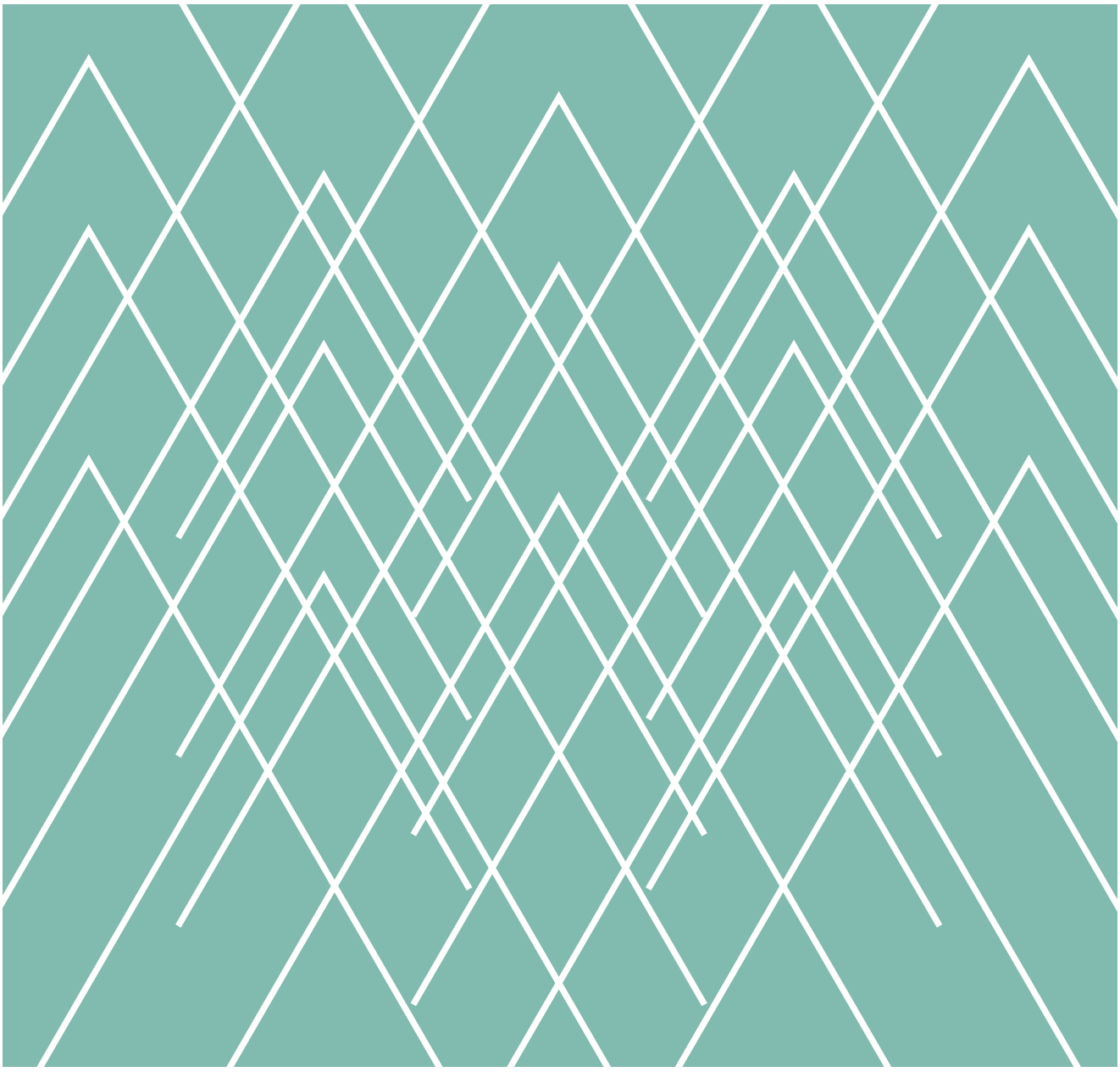
Klassifizierung von Personen und Detektion von Fahrrädern auf Bildern

Stefanie Krause



▲ Hochschule Harz
Hochschule für angewandte Wissenschaften

WAIT

Impressum

Inhaltlich verantwortlich

Autor/-in der Abschlussarbeit

Institution

Der Fachbereich Automatisierung und Informatik ist ein Fachbereich der Hochschule Harz. Die Hochschule Harz ist eine Körperschaft des öffentlichen Rechts. Sie wird durch den Rektor Prof. Dr. Folker Roland gesetzlich vertreten: info@hs-harz.de.

Umsatzsteuer-Identifikationsnummer

DE231052095

Adresse

Hochschule Harz
Fachbereich Automatisierung und Informatik
Friedrichstraße 57-59
38855 Wernigerode

Kontakt

Dekanin des Fachbereiches Automatisierung und Informatik
Prof. Dr. Andrea Heilmann
Tel.: +49 3943 659 300
Fax: +49 3943 659 399
E-Mail: dekanin-ai@hs-harz.de

Aufsichtsbehörde

Das Ministerium für Wirtschaft, Wissenschaft und Digitalisierung des Landes Sachsen-Anhalt (MW), Hasselbachstraße 4, 39104 Magdeburg, ist die zuständige Aufsichtsbehörde.

ISSN 2702-2293

Haftungsausschluss

Die Hochschule Harz weist auf Folgendes hin:

Die Hochschule Harz ist lediglich für die Veröffentlichung der einzelnen Werke zuständig, sie übernimmt keinerlei Haftung. Vielmehr gilt Folgendes:

- für den Inhalt der Publikation ist der/die Autor/-in verantwortlich
- mit der Erfassung in der Schriftenreihe Wernigeröder Automatisierungs- und Informatik-Texte verbleiben die Urheberrechte beim Autor/bei der Autorin
- die Einhaltung von Urheber- und Verwertungsrechten Dritter liegt in der Verantwortung des Autors/der Autorin

Vor Veröffentlichung bestätigte der/die Autor/-in,

- dass mit der Bereitstellung der Publikation und jedes Bestandteils (z.B. Abbildungen) nicht gegen gesetzliche Vorschriften verstoßen wird und Rechte Dritter nicht verletzt werden
- dass im Falle der Beteiligung mehrerer Autoren am Werk der/die unterzeichnende Autor/-in stellvertretend im Namen der übrigen Miturheber/-innen handelt
- im Falle der Verwendung personenbezogener Daten den Datenschutz (durch Einholen einer Einwilligung des Dritten zur Veröffentlichung und Verbreitung des Werks) zu beachten
- dass im Falle einer bereits erfolgten Veröffentlichung (z.B. bei einem Verlag) eine Zweitveröffentlichung dem Verlagsvertrag nicht entgegensteht
- dass die Hochschule Harz von etwaigen Ansprüchen Dritter (z.B. Mitautor/-in, Miturheber/-in, Verlage) freigestellt ist

Hochschule Harz – Hochschule für angewandte Wissenschaften
Fachbereich Automatisierung und Informatik

Hochschule Harz

Masterarbeit

Klassifizierung von Personen und Detektion von Fahrrädern auf Bildern

Stefanie Krause

Gutachter: Prof. Dr. Frieder Stolzenburg

Zweitgutachter: Marcel Altendeitering, M. Sc.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einführung

Dieses Kapitel vermittelt die grundlegende Motivation der Abschlussarbeit. Die Objekterkennung als wichtiger Teil der Bildanalyse hat schon seit einigen Jahren eine große Bedeutung und bildet einen komplexen Forschungszweig, in dem viele interessante Anwendungen möglich sind. Im ersten Abschnitt wird zunächst die Bedeutung sowie Anwendungsmöglichkeiten der Bildanalyse dargestellt. In Abschnitt 1.2 wird neben dem Kontext des Forschungsprojektes auch der Bilddatensatz erläutert, der im Folgenden für die Evaluation verschiedener Implementierungen verwendet wird. Dieses Kapitel zeigt außerdem die Rolle und Ziele der Masterarbeit. Die Einführung schließt mit den Aufbau der Arbeit ab.

1.1 Relevanz der Bildanalyse

Bereits im Jahr 2017 wurden 1,2 Billionen digitale Fotos geschossen.[1] Jedes Jahr steigt die Zahl an Bildern stark, sodass in Januar 2020 allein auf dem sozialem Netzwerk Instagram mehr als 50 Billionen Bilder hochgeladen wurden.[2] Durch die große Anzahl an Fotos ist die Verarbeitung von Bildern besonders relevant, da viele Informationen aus ihnen gewonnen werden können.

Die Objekterkennung ist einer der fundamentalsten und schwierigsten Herausforderungen in Bereich Computer Vision und hat deshalb in den letzten Jahren große Aufmerksamkeit erlangt [1]. Im Jahr 2001 wurde die erste Gesichtserkennung in Echtzeit ohne Einschränkungen (wie Gesichtsfarbenssegmentierung) entwickelt [2]. Seitdem haben sich die Verfahren sehr schnell weiterentwickelt und verbessert, sodass heute viele praktische Anwendungen genutzt werden. Die Analyse von Verkehrs- und Überwachungskameras wird von der Polizei zur Verbrechensbekämpfung verwendet [3]. Roboter sind in der Lage mithilfe der Objekterkennung visuelle Szenen zu verstehen [4]. Die für die Entwicklung des autonomen Fahrens erforderliche Detektion von Personen, Fahrrädern und Fahrzeugen ist neben viele weitere Nutzungsmöglichkeiten möglich [5].

1.2 Kontext und Projektziel

Diese Masterarbeit entsteht in dem Projekt *Metropole Ruhr: Digitale Modelldestination NRW* mit dem Fraunhofer Institut für Software und Systemtechnik und der Ruhr Tou-

[1] `https://blog.wiwo.de/look-at-it/2017/09/14/12-billionen-digitale-fotos-werden-allein-2017-geschossen-davon-85-prozent-per-smartphone/`

[2] `https://www.omnicoreagency.com/instagram-statistics/`

rismus GmbH (RTG). Der Fokus der Destinationsmanagementorganisation ist in diesem Projekt die Unterstützung des Tourismusmarketings. Der Praxispartner bietet diverse Touren im Bereich des boomenden Radtourismus an, die entsprechend auf verschiedenen Kanälen wie Facebook, Webseiten oder Flyern beworben werden. Wichtig ist es dabei, die unterschiedlichen Zielgruppen passgenau anzusprechen. Dazu wurden über die Jahre immer wieder professionelle Fotoshootings von verschiedenen Radtouren mit unterschiedlichen Personengruppen durchgeführt. Die Bilder dieser Fotoshootings werden in einer Cloud gespeichert und können von den Redakteuren abgerufen werden. Ein Ziel des Projektes ist es ein neues Datenbanksystem, zur Erleichterung der Arbeit der Redakteure, aufzubauen.

Die Bilder der Fotoshootings wurden in verschiedenen Ordnern sortiert. Dadurch entstand über die Jahre eine unübersichtliche Ordnerstruktur, die in Abbildung 1.1 dargestellt ist. Insgesamt verfügt die RTG über eine Auswahl von 1.852 Bildern, die drei verschiedene Radrouten bewerben sollen. Da die Fahrradrouten durch unterschiedliche Städte führen und verschiedene Zielgruppen zeigen entstand eine komplexe Ordnerstruktur mit 80 Ordnern.Besonders die Städte und Regionen sind bis ins Detail aufgegliedert, allerdings wurden nur in einigen wenigen Fällen die Fotos ebenfalls ihren Zielgruppen zugeordnet. Möchte ein Redakteur beispielsweise einen Beitrag auf einer Social-Media-Plattform veröffentlichen, muss dieser diverse Ordner nach Bildern seiner gewünschten Zielgruppe per Hand durchsuchen. Dies ist durch die Anzahl an Ordnern und Bildern sehr zeitaufwendig. Sollen beispielsweise junge Paare angesprochen werden, wird ein passendes Foto aus der Cloud ausgewählt auf dem ein junges Paar zu sehen ist. Die Suche nach einem geeigneten Bild kann sehr viel Zeit in Anspruch nehmen, da viele verschiedene Ordner nach Bildern mit jungen Paaren selektiert werden müssen. Besonders die Verschachtelung der Ordner ist unübersichtlich und undurchsichtig. So gibt es beispielsweise dreimal den Ordner *Muehlheim*. Das Aussuchen geeigneter Fotos soll für die Redakteure erleichtert werden, indem eine Datenbank gezielt nach bestimmten Suchbegriffen, wie der Zielgruppe junges Paar, durchsucht werden kann. Die vielschichtige Ordnerstruktur soll obsolet und eine einfache Schlagwortsuche nach Bildmerkmalen möglich werden. Das Ziel ist dabei eine automatisierte Zuordnung in die Unterscheidungsgruppen um den händischen Aufwand der Mitarbeiter der RTG zu minimieren. Dazu sollen zunächst Zielgruppen des Radtourismus anhand von Interviews ermittelt werden. Im nächsten Schritt wird geprüft, ob diese Zielgruppen automatisch auf den Bildern erkannt werden können.

Da die Bilder der Fotoshootings teilweise Personen mit oder ohne Räder sowie Panoramaaufnahmen enthalten, soll des Weiteren erkannt werden, ob Räder oder Personen auf einem Bild zu sehen sind. Beispielbilder des Datensatzes der RTG sind in Abbildung 1.2 zu sehen. Dabei ist die Diversität der Fotos klar erkennbar. Es sind verschiedene Personen unterschiedlich nah sowie mit und ohne Fahrrad zu sehen, die teilweise einen Helm und Sonnenbrille tragen. Außerdem sehen die Model selten direkt in die Kamera, da die Bilder möglichst natürlich und nicht gestellt wirken sollen.

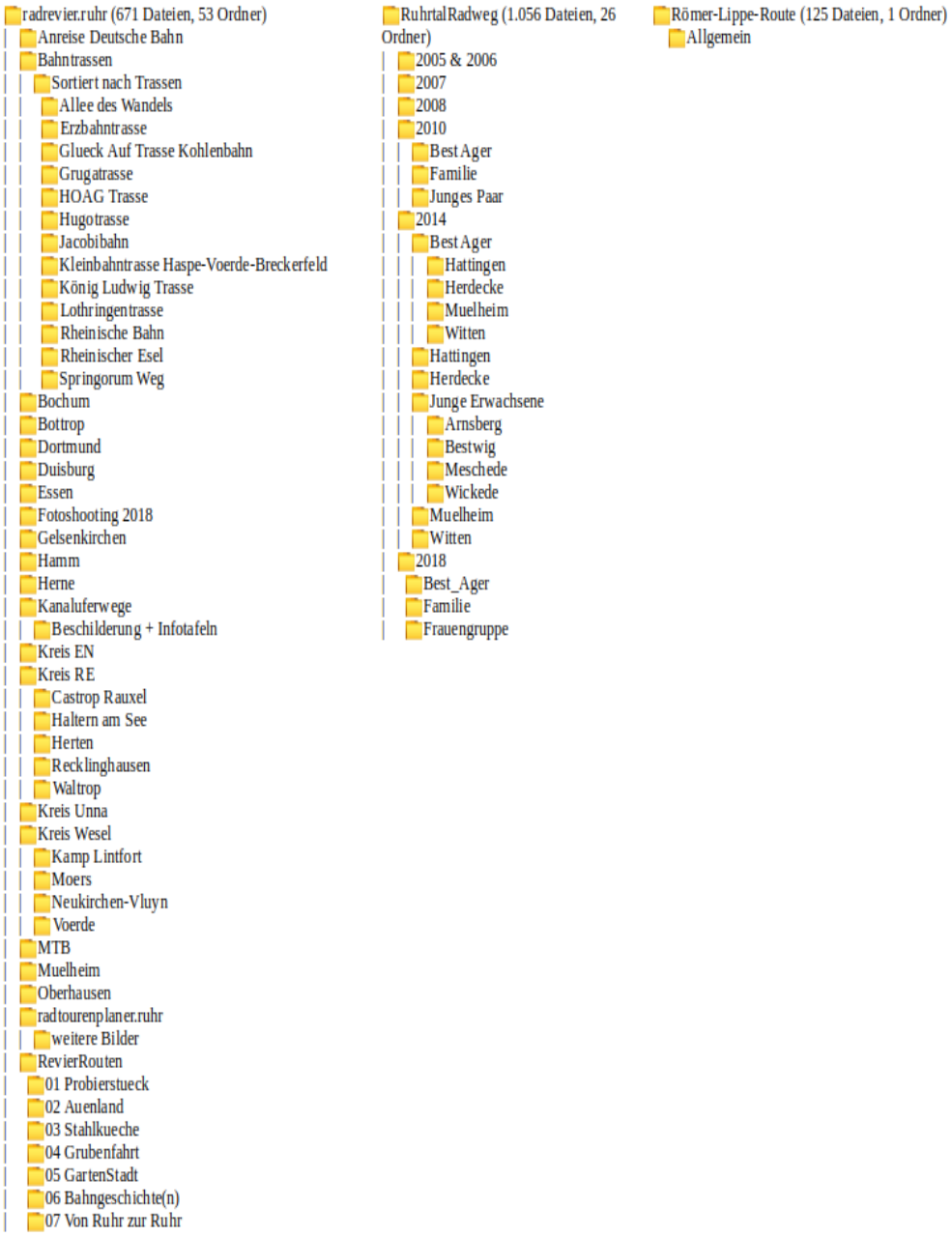


Abbildung 1.1: Ordnerstruktur des Datensatzes der RTG

Abbildung 1.2: Kollage verschiedener Beispielbilder aus dem Datensatz der RTG

1.3 Ziel dieser Arbeit

Ziel dieser Arbeit ist zum Einen die Identifikation relevanter Zielgruppen im Radtourismus. Die Merkmale der Zielgruppen sollen durch Interviews mit der Ruhr Tourismus GmbH ermittelt werden. Zum Anderen sollen anhand der ermittelten Merkmale die Zielgruppen auf Bildern vorhergesagt werden. Dazu sollen zunächst mit Hilfe vortrainierter neuronaler Netze Personen erkannt, lokalisiert und das Alter sowie Geschlecht eingeordnet werden. Anschließend soll eine Zuordnung der Personen zu Zielgruppen des Radtourismus anhand von Regeln erfolgen. Diese Regeln sollen vom Autor der Masterarbeit herausgearbeitet und in bestehende Implementierungen eingearbeitet werden. Des Weiteren sollen Fahrräder in den Bildern erkannt werden. Für die Detektion von Rädern soll eine geeignete bestehende Implementierungen auf dem Praxisdatensatz der RTG evaluiert werden. Dabei soll kritisch betrachtet werden ob, diese Implementierung praxistauglich ist. Abschließend soll ein Fazit über die verfügbaren Datensätze zur Objekterkennung gezogen werden, da bereits aus der Literatur bekannt ist, das oft Probleme durch die Datensatzvoreingenommenheit bestehen. Dies führt zu einer schlechten Generalisierung auf unbekannten Beispielen.

In der Arbeit soll die Beschreibung der Objekterkennung mit gefalteten neuronalen Netzen (CNNs), die Darstellung der Vorgehensweise und Evaluation der Ergebnisse im Fokus stehen. Es werden die verschiedenen Methoden der Objekterkennung erläutert und die verwendeten CNNs erklärt. Dabei wird besonders darauf geachtet auf welchen Datensätzen das neuronale Netz trainiert wurde. Zur Evaluation wird der Bilddatensatz der RTG verwendet. Die Ergebnisse werden anhand verschiedener Metriken ausführlich evaluiert und die Nützlichkeit der Implementierungen für das Projekt kritisch betrachtet. Zusätzlich wird eine generelle Eignung für andere Anwendungsbereiche untersucht. Diese Arbeit soll neben der Beantwortung der Forschungsfragen, Probleme und offene Forschungslücken im Bereich Objekterkennung im Praxiskontext aufdecken und einen Ausblick für weitere

Forschungsarbeiten bieten. Die zu beantworteten Forschungsfragen lauten konkret:

1. Welche Zielgruppe gibt es im Radtourismus?
2. Wie können Zielgruppen auf Bildern erkannt werden?
3. Sind bestehende Implementierungen zur Fahrraderkennung praxistauglich?
4. Sind vorhandene Datensätze ausreichend um eine Generalisierung der Objekterkennung auf Praxisdaten zu ermöglichen?

1.4 Aufbau der Arbeit

Diese Arbeit beginnt mit einem theoretischen Kapitel um die Grundlagen zu legen. Als Basis dieser Abschlussarbeit wird zunächst das Konzept des maschinellen Lernens erläutert. Anschließend werden künstliche neuronale Netze (kNN) und speziell CNNs eingeführt. Dabei wird besonders das Training und die Evaluation neuronaler Netze thematisiert. Außerdem werden verschiedene Methoden der Objekterkennung vorgestellt, sowie Probleme von Bilddatensätzen thematisiert. Das Kapitel 2 wird mit der verwendeten Soft- und Hardware abgeschlossen.

In Kapitel 3 wird die Methodik dargestellt sowie die verschiedenen Implementierungen getestet und evaluiert. Zunächst werden die relevanten Zielgruppen des Radtourismus erläutert, welche durch Interviews mit der RTG ermittelt wurden. Daraufhin wird beschrieben, wie Gesichter auf Bildern erkannt und lokalisiert, sowie deren Geschlecht und Alter eingestuft werden. Die Anzahl der Personen auf Bildern wird anschließend dargestellt und ausgewertet, bevor in Abschnitt 3.4 die Geschlechtserkennung anhand des gesamten Körpers analysiert wird. In Abschnitt 3.5 wird die Detektion von Fahrrädern beschrieben. Hierfür kann auf das gleiche vortrainierte Netze zurückgriffen werden wie bei der Erkennung der Personen.

Die Beantwortung der Forschungsfragen findet in Kapitel 4 statt. Die Masterarbeit schließt mit einer Zusammenfassung der wichtigsten Forschungsergebnisse, offener Forschungsfragen und einem Ausblick für zukünftige Arbeiten ab.

2 Theoretische Grundlagen

Bevor speziell die für die Objekterkennung verwendeten CNNs eingeführt werden, wird mit einer kurzen Erläuterung des maschinellen Lernens sowie einer Einführung in kNN begonnen. Im Anschluss wird das Training und die Evaluation neuronaler Netze erläutert. In Abschnitt 2.6 wird die Objekterkennung und Datensatzprobleme beschrieben, bevor zum Schluss des Grundlagenkapitels die verwendete Soft- und Hardware vorgestellt wird.

2.1 Maschinelles Lernen

Das maschinelle Lernen ist ein nützliches Hilfsmittel um Probleme zu modellieren, die anhand von Beispielen gelernt werden können. Mit Hilfe der Statistik wird ein mathematisches Modell aufgebaut mit der Hauptaufgabe, Schlussfolgerungen aus Beispielen zu ziehen [6]. Dabei wird zwischen zwei Modellen unterschieden. Zum einen, ist es möglich Vorhersagen für die Zukunft zu treffen und zum anderen Informationen aus Daten abzuleiten (oder beides) [6]. Die typische Verwendungsart von maschinellem Lernen ist das überwachte Lernen [7, S. 3]. Dabei wird ein Lernalgorithmus mit verschiedenen Beispielen trainiert, welche zuvor gelabelt wurden. Labeln bedeutet ein Objekt seinem Zielwert zuzuordnen. Für jedes Datenobjekt, das im Training als Eingabewert dient, muss bereits das gewünschte Ergebnis zugewiesen sein. Dazu muss im Vorfeld bestimmt werden welche Zielwerte gewünscht sind und die Trainingsdaten mit diesen gelabelt werden. Beispielsweise wird bei der Objekterkennung auf Bildern ein Bildobjekt z.B. ein Fahrrad mit einer Umrandung markiert und entsprechend einer Klasse von relevanten Objekten benannt. Für jede Klasse ist eine bestimmte Anzahl an Beispielen nötig, damit der Algorithmus lernen und die Labels von neuen Daten vorhersagen kann. Eine Problemstellung bei dir ein Algorithmus die Klasse von neuen, unbekannten Daten, aus einer Menge diskreter Kategorien basierend auf den Trainingsdaten vorhersagt, wird Klassifizierungsaufgabe genannt.

Da Trainingsdaten für einen maschinellen Lernalgorithmus nicht jede mögliche Eingabeinstanz beinhalten können (z.B. nicht alle möglichen Fahrradgrößen, -formen und -farben), ist es wichtig, dass der Lernalgorithmus generalisieren kann [7, S. 2]. Werden einfache Modelle mit zu wenig Beispielen trainiert, könnten wichtige Aspekte des "wahren" Modell nicht abgebildet werden. Gleichwohl ist eine Überanpassung des Modells auf unwichtige Details ebenfalls unerwünscht. Wichtig ist, dass die Beispiele nicht lediglich auswendig gelernt, sondern Muster und Gesetzmäßigkeiten in den Trainingsdaten erkannt werden. Die Güte eines Algorithmus des maschinellen Lernens kann an der Qualität und Quantität der Fehler gemessen werden. Mit einer Verlustfunktion, wie z.B. der mittleren quadratischen Abweichung, werden den Fehlern Kosten zugeordnet. Diese werden in der Trainingsphase

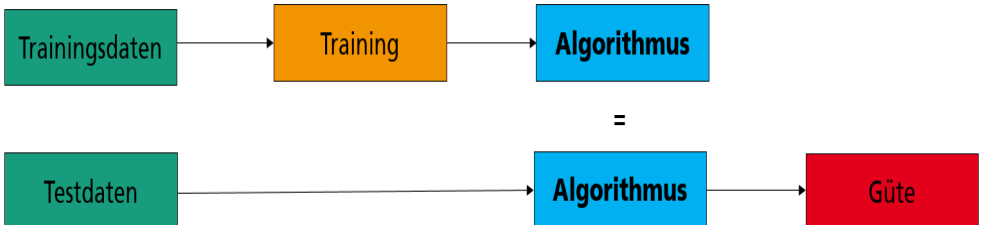


Abbildung 2.1: Schematische Darstellung der Schritte des Maschinellen Lernens. Die Gesamtdaten werden in Trainings- und Testdaten aufgeteilt. Mit den Trainingsdaten wird ein Algorithmus angelernt und später mit den Testdaten auf seine Güte überprüft.

des Algorithmus minimiert. Eine schematische Darstellung des Ablaufes eines maschinellen Lernverfahrens ist in Abbildung 2.1 zu sehen. Zu Beginn wird der Datensatz in Trainings- und Testdaten aufgeteilt. Dafür werden in der Regel 2/3 der Daten für das Training verwendet. Mit den gelabelten Trainingsdaten wird ein Algorithmus angelernt. Um die Güte dieses Algorithmus zu testen werden unbekannte Testdaten in den Algorithmus eingelesen und das vorhergesagte Ergebnis mit dem tatsächlichen Ergebnis verglichen. Dafür wird eine Verlustfunktion zur Berechnung des Abstandes der Ergebnisse verwendet.

2.2 Künstliche Neuronale Netze

KNN gehören zu der Klasse der maschinellen Lernalgorithmen. Sie sind besonders gut geeignet viele Probleme der Informatik zu lösen [8]. In dieser Arbeit liegt der Fokus auf CNNs, diese sind spezielle kNN. Zunächst wird allgemein in kNNs eingeführt, bevor CNNs beschrieben werden.

KNNs ahmen das menschliche Gehirn nach. Ab 1943 waren Warren McCulloch und Walter Pitts Pioniere in der Forschung. Sie präsentierten das erstes Modell mit künstlichen Neuronen [9, S. 3]. Ebenso federführend war Frank Roseblatt, der 1958 das Perceptron entwickelte, welches ein allgemeineres Modell als das von McCulloch-Pitts darstellt [9, S. 55].

In Abbildung 2.2 ist der Aufbau eines künstlichen Neurons k zu sehen. Es enthält m Eingabeparameter x_j. Jeder diese Eingabeparameter des Neurons k hat ein Gewicht w_{kj}. Die Gewichte beschreiben die Intensität des Informationsflusses entlang einer Verbindung. Die Eingaben und Gewichte werden linear kombiniert und anschließend die modifizierten Eingabesignale summiert. Zu diesem Ergebnis wird eine Konstante, der sog. Bias oder Verzerrung b, addiert. Diese Summe ist die Eingabe einer Aktivierungsfunktion ϕ, welche y_k als Ausgabe zurückgibt. Die Ausgabe y_k lässt sich wie folgt beschreiben:

$$y_k = \phi\left(\left(\sum_{j=0}^{m} w_{kj} \cdot x_j\right) + b\right) \tag{2.1}$$

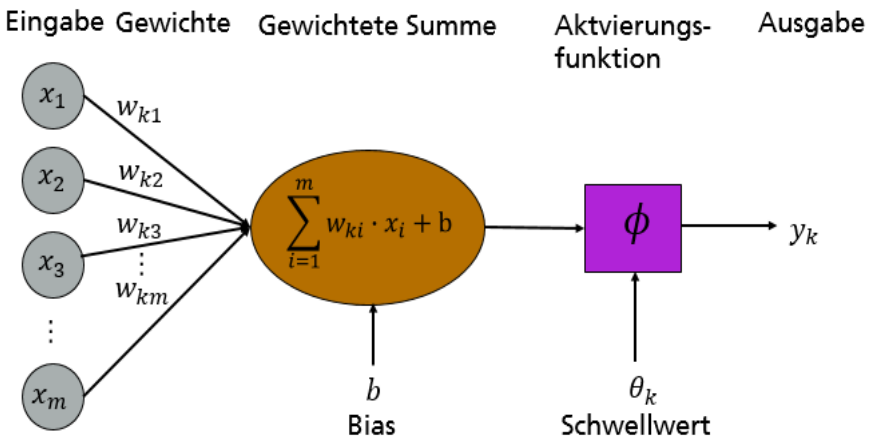


Abbildung 2.2: Künstliches Neuron k mit m Eingabeparametern.

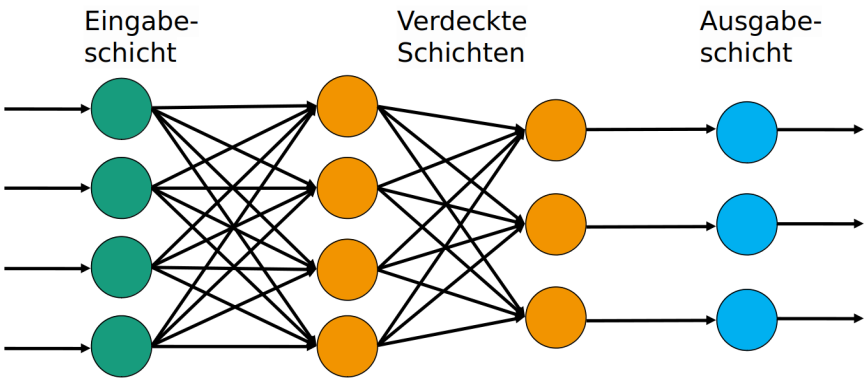


Abbildung 2.3: Vorwärtsgerichtetes, vollvernüpftes neuronales Netz mit 4 Eingabeneuronen in der Eingabeschicht, 2 verdeckten Schichten (orange) und 3 Ausgabeneuronen in der Ausgabeschicht.

Ein neuronales Netz ist ein gerichteter Graph, dessen Knoten als Neuronen und dessen Kanten als Verbindungen bezeichnet werden [10]. Jeder Verbindung ist ein Gewicht zugeordnet. Die Neuronen sind in verschiedenen Schichten angeordnet. Es gibt drei verschiedene Arten von Schichten: die Eingabeschicht, die verdeckte Schicht und die Ausgabeschicht [11]. In Abbildung 2.3 ist ein vollverknüpftes, vorwärtsgerichtetes neuronales Netz zu sehen. Man spricht von einem vorwärtsgerichteten neuronalen Netz, falls die Verbindungspfeile ausschließlich in Richtung der Ausgabeschicht zeigen. Das Netz ist vollverknüpft, da jede Ausgabe der Neuronen einer Schicht die Eingabe jedes Neurons der nächsten Schicht ist. Die erste Schicht bekommt den Input aus den Eingabedaten und die nächsten Schichten den verarbeiteten Input aus der vorherigen Schicht. Die Ausgabeschicht konvertiert die Ausgabe der letzten verdeckten Schicht zur gewünschten Ausgabe des neuronalen Netzes, wie beispielsweise eine Klassifizierung zum Objekt Fahrrad mit einer bestimmten Wahrscheinlichkeit.

Es gibt verschiedene Arten von kNNs, die sich im Aufbau unterscheiden. CNNs, die sich Parameter teilen und eine beschränkte Anzahl an Verknüpfungen zwischen Neuronen haben, sind sehr gut zur Objekterkennung geeignet [12, S. 335]. Außerdem gibt es einige weitere Varianten von kNNs, z.B. rekurrente neuronale Netze (RNN). Diese werden verwendet, um zeitliche Abhängigkeiten zu kreieren, beispielsweise in sequenziellen Aufgabenstellungen, wie bei der Spracherkennung [13]. So gibt es für verschiedene Aufgabestellungen spezielle Netze. Diese werden nicht alle in dieser Arbeit betrachtet. Im Folgenden werden die, in der Abschlussarbeit verwendeten, CNNs vorgestellt.

2.3 Gefaltetes Neuronales Netz

Die Idee für CNNs wurde von einem Konzept der Biologie inspiriert. Hubel und Wiesel haben 1962 erstmals bei einem Experiment mit Katzen herausgefunden, dass Neuronen im visuellen Cortex (auch Sehrinde) sensitiv auf bestimmte Stimuli, wie Kanten, reagieren [14]. Im Jahr 1982 stellten Fukushima und Miyake das mehrschichtige kNN Neocognitron basierend auf rezeptiven Feldern vor, welches als die erste Implementierung von CNNs gesehen werden kann [15]. Die biologische Funktion kann durch die mathematische Operation der Faltung beschrieben werden [16]. Als CNN werden kNNs bezeichnet, die anstelle der gewöhnlichen Matrixmultiplikation eine Faltung in mindestens einer Schicht verwenden. Eine Faltung im CNN ist ein Schieben eines Filters (oft auch Kernel genannt) über Teilbilder mit Anwendung des Skalarproduktes. Filter sind Matrizen mit fester Größe z.B. 2×2 oder 3×3 und beschreiben eine bestimmte Form die im Bild gesucht wird. Sie sind ein Set trainierbarer Gewichte. Mit Filter wird versucht das Eingabebild auf gewisse Eigenschaften zu reduzieren. Die diskrete Faltungsoperation zwischen einem Bild f und einer Filtermatrix g ist definiert als:

$$h[x, y] = f[x, y] * g[x, y] = \sum_n \sum_m f[n, m] \cdot g[x - n, y - m] \tag{2.2}$$

Dabei ist das Symbol * der Faltungsoperator. In der Gleichung (2.2) wird das Skalarprodukt [17] eines Teilbildes von f, welches die gleiche Größe wie der Filter g hat, mit g gebildet. Dies kann als eine Berechnung des Ähnlichkeitsgrades interpretiert werden, da

das Skalarprodukt zweier Vektoren den Grad der Überlappung zwischen ihnen beschreibt [11, S. 186]. Es werden verschiedene Bildausschnitt betrachtet und jedes Element der Filtermatrix elementweise mit der zugehörigen Position im Bild multipliziert. Für einen n × n Filter werden dann n × n Multiplikationen durchgeführt. Die Ergebnisse der Multiplikationen werden addiert und repräsentieren wie stark der Filter mit dem Bildausschnitt übereinstimmt. Wichtig ist dabei, dass das Skalarprodukt eine andere Operation als die Matrixmultiplikation ist. Der Filter wandert von links nach rechts mit einer bestimmten Schrittweite über die Eingabematrix und springt nach jedem Durchlauf in die nächst tiefere Zeile. Eine Schrittweite von zwei bei einer Filtergröße von 2 × 2 führt beispielsweise pro Filter zu einer Halbierung der Größe der Ergebnismatrix im Vergleich zur Eingabematrix. Bei CNNs wird der Filter oft kleiner als das Eingabebild gewählt, was zur einer Reduktion der Ausgabedimension führt. Dieses Prinzip wird als spärliche Interaktionen oder spärliche Gewichte bezeichnet [12, S. 330]. Aus diesem Grund müssen weniger Parameter gespeichert werden.

Ein CNN kann mit Filtern ortsunabhängige Strukturen in den Eingabebildern erkennen. Die Art der Filter wird dabei nicht vorgegeben, sondern wird vom Netz gelernt. Auf der ersten Ebene werden die Filter dabei von einfachen Strukturen, wie Linien und Kanten, aktiviert. In der nächsten Ebene werden Strukturen gelernt die aus der Kombination dieser Basisstrukturen bestehen z.B. Kurven, einfache Formen etc. [18]. Wie in Abschnitt 2.2 bereits erklärt bestehen kNNs, also auch CNNs, aus einer Eingabe- und Ausgabeschicht sowie verdeckten Schichten. Die verdeckten Schichten sind im CNN typischerweise die Faltungsschichten (engl. Convolutional Layers) mit nicht-linearer Aktivierungsfunktion und sogenannten Pooling Schichten. Am Ende eines CNNs befindet sich typischerweise mindestens eine vollvernetzte Schicht [7, S. 269]. Im Folgenden betrachten wir diese Schichten im Detail. Die Ein- und Ausgaben der Schichten sind Felder, welche Merkmalskarten (engl. Feature Maps) genannt werden [19].

Um die Funktionsweise und Schichten von CNNs zu verstehen ist es wichtig, sich klar zu machen, wie ein Bild im Computer verarbeitet wird. Ein Schwarz-Weiß-Bild ist ein zweidimensionales Array aus Pixeln. Ein Pixel ist die kleinste Einheit in digitalen Bildern. Pixel ergeben kombiniert das gesamte Bild. Jeder Pixel wird im Computer mit einem Zahlenwert dargestellt, dieser stellt die Intensität des Bildausschnittes dar. Ein Computer verarbeitet, wie in Abbildung 2.4 zu sehen, Bilder als Zahlenmatrizen. Die Werte der Pixel sind mit der 8-Bit Repräsentation Zahlen zwischen 0 und 255. Im Schwaz-Weiß-Bild bedeutet der Wert 0 ein schwarzes Pixel. Der Pixelwert ist 255 für einen weißen Bildabschnitt. Die Pixelwerte zwischen 0 und 255 sind unterschiedliche Graustufen.

Die Bildauflösung bezeichnet die Anzahl der Pixel im Bild. Je mehr Pixel ein Bild enthält, desto besser ist die Qualität. Bildauflösungen werden z.B als 320 × 240, 640 × 480, 800 × 600, 1024 × 768 Pixel beschrieben. Das bedeutet zum Beispiel, dass ein Bild 1024 Pixelspalten und 768 Pixelzeilen hat. Die Gesamtzahl der Pixel erhält man durch multiplizieren der Anzahl der Pixelspalten und -zeilen. Im Gegensatz zu Schwarz-Weiß-Bildern besitzen digitale Farbbilder drei Kanäle. In der Regel wird der RGB Farbraum verwendet. In diesem hat ein Bild dreidimensionale Werte. Abhängig von der Größe und Auflösung des Bildes sieht der Computer ein Farbbild als z.B. ein 640 × 480 × 3 Feld aus Zahlen, wobei die ersten zwei Dimensionen die Standarddimensionen der Ebene sind und

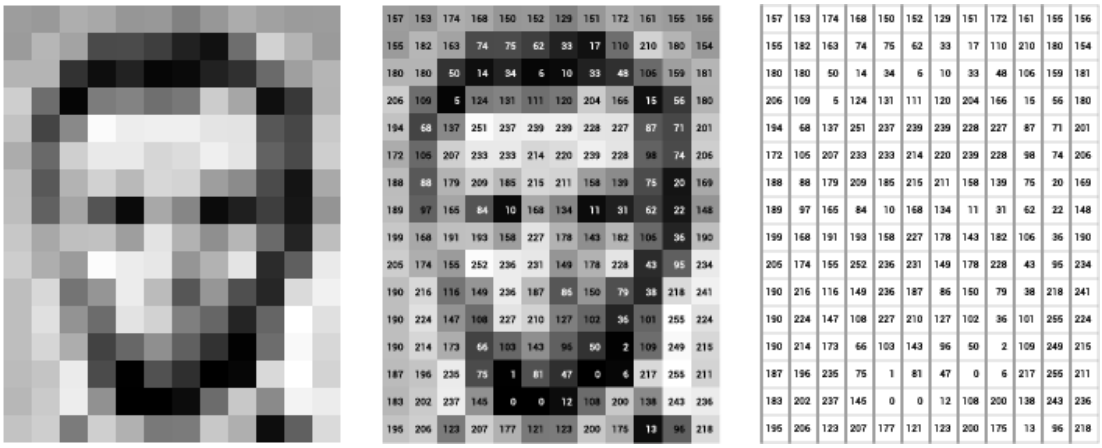


Abbildung 2.4: Bild eines Gesichtes mit 12 × 16 Pixel (links) und mit den zugehörigen Intensitäten als Pixelwerte (mittig). Im Bild rechts ist dargestellt wie ein Computer das Bild sieht.

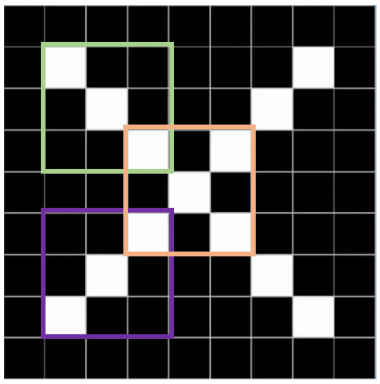


Abbildung 2.5: Drei farbig umrandeten Filter beschreiben die wichtigsten Merkmale im Bild. Die grün und lila umrandeten Merkmale kommen dabei zwei mal an verschiedenen Bildstellen vor. Sie werden beide von den über das Bild gleitenden Filtern erkannt.

die dritte, zusätzliche Dimension für jede der Farben Rot, Grün, Blau steht.

Möchte man ein Objekt auf einem Bild erkennen werden Filter verwendet, welche die wichtigsten Merkmale des Bildes beschreiben. Im Beispiel des Buchstaben X genügen drei Filter um alle relevanten Bildmerkmale zu illustrieren. Dies wird in Abbildung 2.5 veranschaulicht. Durch die Faltungen kann die Ähnlichkeit eines Filters mit allen Bildabschnitten ermittelt werden. Dadurch ist eine Aussage möglich, ob sich auf dem Bild tatsächlich ein bestimmtes Objekt befindet.

Im Bereich Computer Vision gibt es unterschiedliche Filter. Nach [20] lässt sich die Mehrheit der verschiedenen Methoden in einer der folgenden zwei Kategorien einteilen:

1. **Gradienten Methode**: Die Gradienten Methode erkennt Kanten, indem das Maximum und Minimum der ersten Ableitung des Bildes betrachtet wird. Ein Beispiel

Abbildung 2.6: Beispielhafte Anwendung zweier verschiedener Kantenfilter auf ein Eingabebild. In der Mitte ist ein Beispiel für die Gradientenmethode zu sehen, Rechts die Laplacemethode.

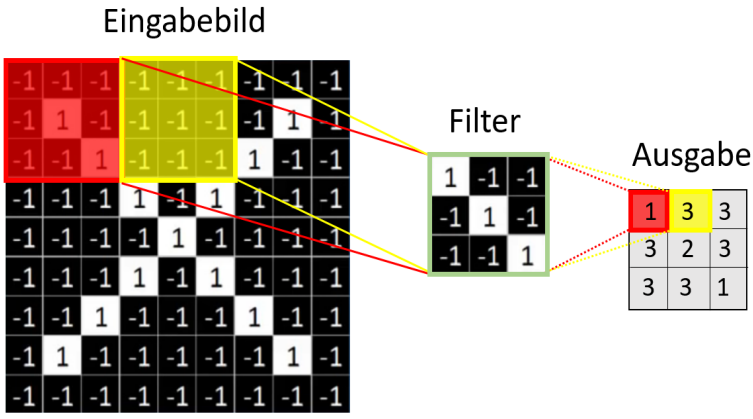


Abbildung 2.7: Funktionsweise der Faltung am Beispiel eines ausgewählten Filters (Schrittweite 3).

hierfür ist der Sobel Filter, bei dem gefundene Merkmale sehr scharfe Karten aufweisen (vergleiche Abbildung 2.6 Bild mittig).

2. **Laplace Methode**: Die Laplace Methode sucht nach Nullstellen mit Vorzeichenwechsel in der zweiten Ableitung des Bildes um Karten zu erkennen. Ein Beispiel hierfür ist Laplacian of Gaussian (vergleiche Abbildung 2.6 Bild rechts).

Beschreibung der Schichten

In diesem Abschnitt werden die verschiedenen Schichten des CNNs beschrieben. In CNNs enthalten die verdeckten Schichten mindestens eine Faltungsschicht. In der Faltungsschicht wird die mathematische Operation der Faltung mit jedem Filter durchgeführt. In der Abbildung 2.7 ist exemplarisch die Faltung eines Filters mit dem Eingabebild veranschaulicht. Der Filter gleitet mit einer bestimmten Schrittweite (hier 3) über das Bild und für jeden Bildabschnitt wird eine Faltung mit der Filtermatrix berechnet. In einer Faltungsschicht wird diese mathematische Operation für alle Filter durchgeführt.

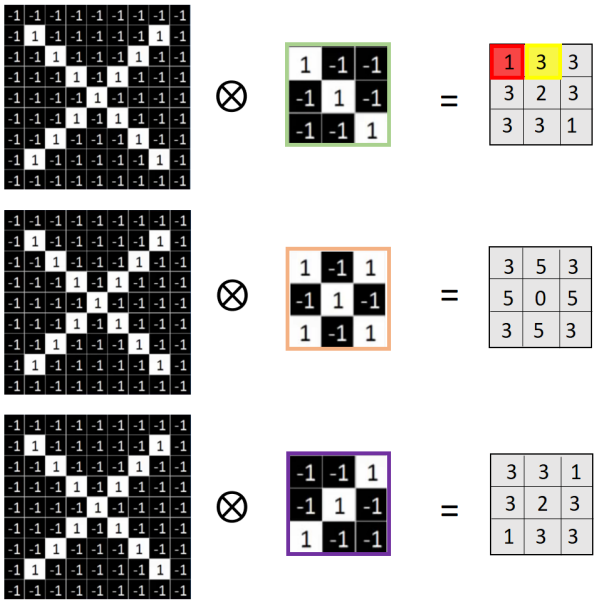


Abbildung 2.8: Operationen in der Faltungsschicht. Mit allen 3 Filtern des Buchstaben X wird die Faltung berechnet.

In unserem Beispiel, vergleiche Abbildung 2.8, werden alle drei Filtermatrizen auf das Eingabebild angewendet. Jeder Filter findet ein bestimmtes Merkmal an allen Orten in der Eingabe [19].

Da die gleichen Filter auf alle Teile des Bildes angewendet werden, ist die Anzahl freier Parameter viel kleiner als bei vollverknüpften neuronalen Schichten [21]. Die Neuronen der gefalteten Schicht teilen sich oft die gleichen Parameter und sind nur zu einer lokalen Eingaberegion verknüpft. Das sogenannte Parameter Sharing bezieht sich auf die Verwendung desselben Parameters für mehr als eine Funktion in einem Modell [12, S. 331]. In einem traditionellen neuronalen Netz wird jedes Gewicht nur einmal genutzt um die Ausgabe einer Schicht zu berechnen. Im Gegensatz dazu wird jedes Filtergewicht auf jede Position in der Eingabe angewendet. Die Parameterteilung resultiert aus der Faltung und stellt die Translationsinvarianz sicher. Diese garantiert eine Toleranz gegenüber gewissen Verschiebungen in den Eingabedaten. Die exakte Position eines Merkmales ist in einem Bild nicht entscheidend, da verschiedene Bilder eines Objektes ohnehin leichte Abweichungen durch beispielsweise verschiedene Positionen beim Fotografieren haben. Die Merkmale aus der Faltungsschicht können positive oder negative Werte annehmen (oder 0). Ein Wert kann z.B. positiv sein, wenn in einer bestimmten Region der Eingabe ein bestimmtes Muster, wie ein Fahrradreifen, zu sehen ist und negativ oder null, wenn in dieser Region das Muster nicht vorhanden ist.

Die Faltungsschicht hat typischerweise eine nicht-lineare Aktivierungsfunktion ϕ, welche manchmal als einzelne Schicht beschrieben wird. Da die semantischen Informationen in einem Bild (z.B. eine Familie fährt gemeinsam Fahrrad) offensichtlich ein nicht-lineares Überführen vom Pixelwerten in der Eingabe ist, soll das Verarbeiten der Eingabe des CNNs zu seiner Ausgabe ebenfalls nicht-linear sein. Traditionelle Aktivierungsfunktionen sind hierfür $\phi(x) = \tanh(x)$ oder die Sigmoid-Funktion $\phi(x) = (1 + e^{-x})^{-1}$ [22].

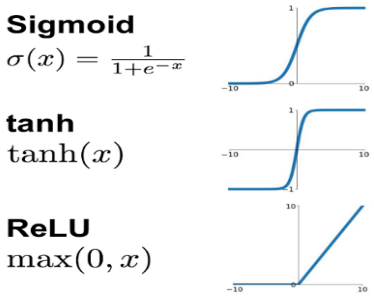


Abbildung 2.9: Funktionen und Graphen der wichtigsten Aktivierungsfunktionen. Abbildung entnommen aus `http://www.ai-united.de/arbeitsweise-eines-neuronalen-netzwerkes-algorithmen-training-aktivierungs-und-verlustfunktionen/`.

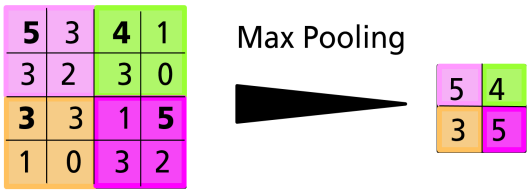


Abbildung 2.10: Beispiel des Maxpooling mit Schrittweite von 2 und Fenstergröße 2×2. Für jedes Fenster wird der Maximalwert (fett hervorgehoben) extrahiert. Dadurch verkleinert sich die Merkmalskarte.

In der aktuellen Forschung wird oft Rectified Linear Unit (ReLU) [23] verwendet. Diese Aktivierungsfunktion lautet $\phi(x) = \max(0, x)$. Alle negativen Werte werden bei der Funktion null. Dies führt dazu, dass die Ausgabe nur aktiviert wird wenn ein Muster in einer bestimmten Bildregion gefunden wurde. Dies ist der Fall wenn ein positiver Wert in der Merkmalskarte vorhanden ist. Die monoton wachsende Funktion beschleunigt die Konvergenz des Gradientenverfahren verglichen mit der *Sigmoid* und *Tanh* Aktivierungsfunktion um die sechsfache Geschwindigkeit [22], da die Gewichte eines Neurons während des gesamten Trainingsprozesses mit der gleichen Geschwindigkeit weiter wachsen können (vgl. Abbildung 2.9). Außerdem wird die Funktion gerne verwendet, da es einfacher und somit schneller ist sie zu berechnen [23].

Eine weitere Schicht ist die sogenannte Pooling Schicht. In dieser werden benachbarte Pixel zu einem Wert zusammengefasst. Dazu wird zunächst eine Fenstergröße gewählt (meist zwei oder drei) und eine Schrittweite (oft zwei). Das Fenster wird mit der gewählten Schrittweite über das Bild geschoben und für jedes Fenster werden die darin enthaltenen Werte aggregiert [18]. Für gewöhnlich wird der Durchschnitts- oder Maximalwert des Fensters berechnet [18]. In Abbildung 2.10 ist eine Bündelung zum Maximalwert eines Fensters abgebildet. Diese Operation verkleinert die Merkmalskarten und sorgt dafür, dass der exakte Standort des Merkmales "verwischt" wird. Dadurch wir die Objekterkennung robust gegenüber kleinen Änderungen der Position im Bild. Dieses Phänomen wird als Translationsinvarianz bezeichnet. Diese Aggregierung wird für alle Merkmalskarten durchgeführt.

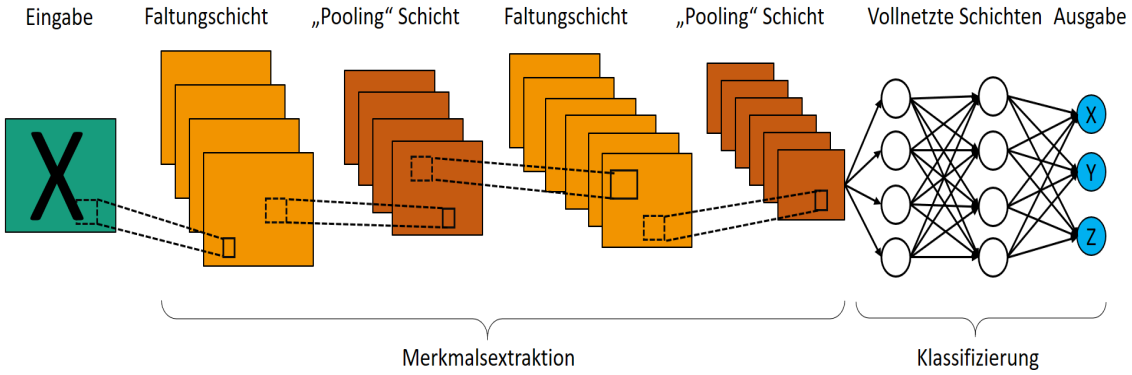


Abbildung 2.11: Beispiel einer gefalteten neuronalen Netzarchitektur.

Nach einer Serie von Faltungs- und Pooling Schichten wird die Merkmalskarte des Bildes extrahiert und alle Neuronen in eine vollverknüpfte Schicht transformiert [18]. Mindestens die letzte Schicht vor der Ausgabe in einem CNNs ist eine vollverknüpfte Schicht. Es können allerdings auch mehrere dieser Schichten hintereinander gelagert sein. In einer vollverknüpften Schicht ist jede Ausgabe mit jeder Eingabe der vorherigen Schicht direkt verbunden. Dies ist analog zu der Neuronenverknüpfung in einem traditionellen kNN (vgl. Abbildung 2.3). Nachdem die Faltungs- und Pooling Schichten durchlaufen sind, enthält die Ausgabe der aktuellen Schicht eine verteilte Repräsentation des Eingabebildes. Alle diese Merkmale der aktuellen Schicht werden genutzt, um für jede Klasse eine Wahrscheinlichkeit zu berechnen, dass diese im Bild zu sehen ist. Die Klassifizierung der Eingabebilder erfolgt anhand der zuvor identifizierten Merkmale. Im Fall einer Mehrklassenklassifizierung wird eine vollverknüpfte Schicht mit einer Softmaxfunktion verwendet [7, S. 269]. Die Softmax Funktion ist auch als normalisierte Exponentialfunktion bekannt und sorgt dafür das sich die Wahrscheinlichkeiten, dass ein Objekt zu einer bestimmten Klasse gehört, zu eins addieren.

Eine schematische Darstellung eines CNNs mit einer Kombination aus den verschieden Schichten ist in Abbildung 2.11 zu sehen. In Netzarchitekturen aus der Praxis gibt es mehrere Faltungs- und Pooling Schichten. In jeder Phase ist die Invarianz der Eingabetransformationen im Vergleich zur vorherigen Schicht größer. Die allmähliche Verringerung der räumlichen Auflösung wird allerdings durch eine zunehmende Anzahl von Merkmalen kompensiert [7, S. 269].

2.4 Training

Ein neuronales Netz wird trainiert indem alle Gewichte und Bias kalkuliert werden. Dabei lernt das Netz die Zielausgabe von bekannten Eingaben zu approximieren. Es wird die vorhergesagte Ausgabe mit dem wahren Ergebnis verglichen und eine Fehlerfunktion oder auch Verlustfunktion genannt berechnet. Da es schwierig ist, alle Gewichte eines mehrschichtigen Netzes analytisch zu berechnen wird der Fehlerrückführungsalgorithmus (engl. Backpropagation) verwendet. Dieser ist einfach, effizient und berechnet die Gewichte iterativ [12, S. 204]. Zum Lernen der Gewichte wird das Gradientenverfahren verwendet.

Damit werden die trainierbaren Parameter modifiziert, mit dem Ziel, dass die Fehlerfunktion des gesamten Trainingsdatensatz sinkt.

Der Ablauf des Training ist wie folgt: Zunächst werden die Gewichte initialisiert, z.B. mit kleinen zufälligen Werten. Dann wird ein Eingabevektor vorwärts ins Netz gegeben. Die erhaltene Netzausgabe wird mit der Zielausgabe mit Hilfe einer Fehlerfunktion verglichen. Wird die mittlere quadratische Abweichung als Fehlerfunktion verwendet, ist der Ausgabefehler einfach die Differenz der aktuellen und gewünschten Ausgabe. Das Ergebnis der Verlustfunktion wird dann ins Netz zurückpropagiert um die Fehler der Neuronen in den verdeckten Schichten zu berechnen. Dieser kann mit der Kettenregel der Infinitesimalrechnung berechnet werden.

Sei E_n die Fehlerfunktion für die Eingabe n, w_{ki} die Gewichte des Neurons k $(i = 1, ..., m)$ und die Aktivierung a_k des Neurons k, dann lautet die Kettenregel wie folgt:

$$\frac{\partial E_n}{\partial w_{ki}} = \frac{\partial E_n}{\partial a_k} \frac{\partial a_k}{\partial w_{ki}} \tag{2.3}$$

Durch rekursive Anwendung der Kettenregel wird die partielle Ableitung $\frac{\partial E_n}{\partial w_{ki}}$ der Kostenfunktion für alle Gewichte w_{ki} und Bias b im Netz berechnet. Der Ausdruck gibt uns Auskunft darüber, wie schnell sich die Kosten ändern wenn die Gewichte und Bias angepasst werden. Der Fehlerrückführungsalgorithmus berechnet die Kettenregel in einer speziellen Reihenfolge die sehr effektiv ist [12, S. 205]. Der Gradient wird berechnet, indem die Fehlerfunktion am Ende verwendet und der Fehler rückwärts durch die Schichten verarbeitet wird. Die Neuronengewichte können mit dem Gradientenverfahren, abhängig von ihrem Einfluss auf den Fehler, angepasst werden. Der Gradient zeigt dabei entgegen der Richtung des stärksten Wertverlustes der Fehlerfunktion, idealerweise in Richtung des globalen Minimums. Neue Gewichte errechnen sich folgendermaßen aus den alten Gewichten:

$$\text{neue Gewichte} = \text{alte Gewichte} - \text{Lernrate} \cdot \text{Gradient} \tag{2.4}$$

Die Lernrate ist dabei die Schrittgröße mit der das (lokale) Minimum erreicht wird. Sie ist größer 0 und wird zunächst gewählt und durch Probieren angepasst. Es ist nicht einfach eine gute Lernrate zu wählen. Wenn die sie zu klein ist dauert es sehr lange bis zu einer Kovergenz, ist die Lernrate allerdings zu groß kann die Konvergenz behindern werden und dazu führen, dass die Verlustfunktion um das Minimum schwankt oder sogar divergiert [24]. Nach [25] ist der beste Weg die richtige Lernrate zu finden indem Experimente mit kleinen aber repräsentativen Stichproben aus dem Trainingsset durchgeführt werden. Ist die Steigung der Kostenfunktion groß, so sind die Kosten weit vom Minimum entfernt und die Gewichte müssen stärker angepasst werden. Falls der Gradient bereits klein ist, sind die Kosten fast optimal und die Gewichte werden nur minimal anpasst. Ist die Steigung sehr nahe an der Null muss nicht weiter anpassen werden. In Abbildung 2.12 ist eine Anpassung der Gewichte mit Hilfe des Gradientenverfahren dargestellt.

Der Algorithmus beginnt von vorne mit neuen Eingaben bis die Gewichte konvergieren. Das Ziel ist es das globale Minimum zu erreichen. Allerdings ist es möglich, dass das Netz nur zu einem lokalen Minimum konvergiert. KNNs sind oft komplexer und haben keine so simple, konvexe Fehlerfunktion wie in Abbildung 2.12 dargestellt. Oft gibt es mehrere lokale Minima in der Fehlerfunktion und es besteht die Gefahr in einem lokalen und nicht

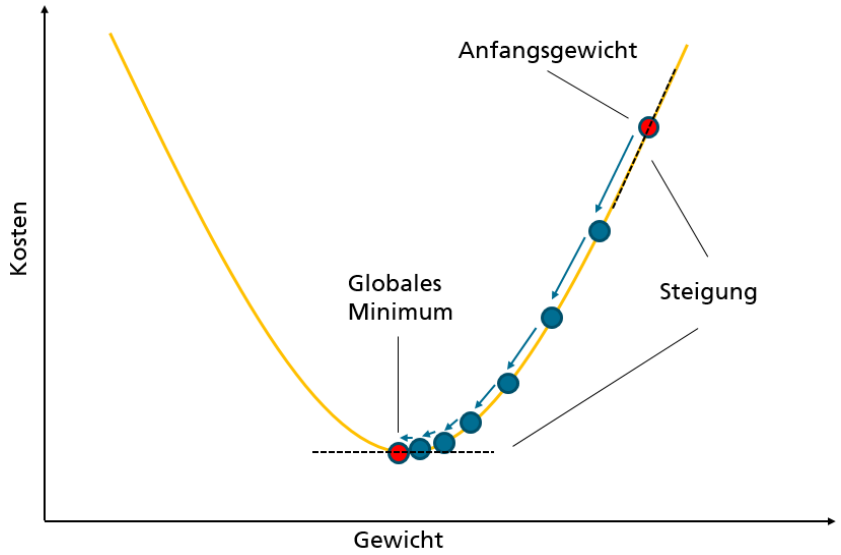


Abbildung 2.12: Gradientenverfahren: Mit Hilfe der Steigung der Kostenfunktion (schwarz gestrichelt) werden neue Gewichte in Richtung des Minimums der Kostenfunktion berechnet. Dabei wird bei einem Anfangsgewicht begonnen und abgebrochen wenn der Gradient der Kostenfunktion sehr nahe an der Null ist.

globalen Minimum zu landen. Allerdings wird mit dem Gradientenverfahren sehr schnell ein Minimum der Fehlerfunktion erreicht, weshalb die Methode gerne verwendet wird [7, S. 239].

Es ist wichtig anzumerken, dass die beschriebenen Verfahren Fehlerrückführung und Gradientenverfahren als zwei unabhängige Algorithmen verstanden werden. Die Fehlerrückführung führt den Fehler rückwärts durch das Netz um die Ableitung zu evaluieren. Dies ist auch bei anderen Netzen möglich und kann auf verschiedene Fehlerfunktionen angewendet werden [7, S. 241]. Das Gradientenverfahren wird zum Lernen der Gewichte angewendet. Es gibt drei verschiedene Varianten des Gradientenverfahren, dass sogenannte Batch, Mini Batch und stochastische Gradienteverfahren [24]. Beim Batch Gradientenverfahren werden alle Trainingsdaten auf einmal in Betracht gezogen. Es wird dann der Durchschnitt des Gradienten aller Trainingsbeispiele berechnet und für die neuen Parameter verwendet. Wenn der Datensatz groß ist, wird eher das stochastische Gradientenverfahren verwendet, wobei nur ein Trainingsbeispiel betrachtet wird. Eine Mischung des Batch Gradientenverfahren und des stochastische Gradientenverfahren ist das Mini Batch Gradientenverfahren. Dabei wird eine feste Anzahl an Trainingsbeispielen verwendet, die kleiner ist als der gesamte Datensatz. Gängige Mini Batch Größen sind zwischen 50 und 256 Trainingsbeispiele [24].

Um eine Überanpassung des Netzes zu vermeiden kann Dropout Lernen [26, 27, 28] angewendet werden. In einer einer sog. Dropout Schicht werden dabei zufällige Ausgabewerte von Neuronen nicht berücksichtigt [28]. Genauer wird auf Schichten, auf die der Dropout beim Training angewendet wird, mit einer bestimmten Wahrscheinlichkeit ein Neuron aktiv gehalten und anderenfalls auf Null gesetzt. Ein Neuron wird dabei temporär aus

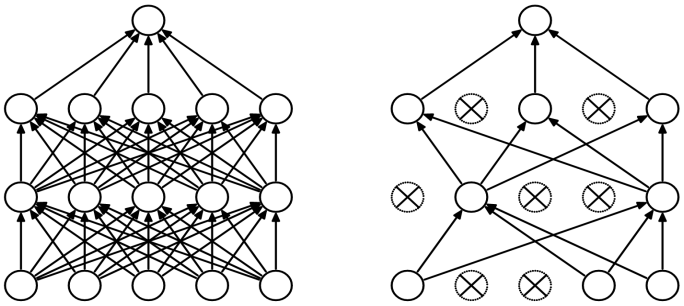


Abbildung 2.13: Darstellung des Dropout Trainings. Das linke Bild zeigt ein gängiges kNN mit 2 verdeckten Schichten, rechts ist ein Beispiel eines Netzes, auf das Dropout Lernen angewendet wird dargestellt. Die Abbildung ist aus [27] entnommen.

dem Netz mit allen Ein- und Ausgangsverknüpfungen entfernt [27]. Dieses Prinzip ist in Abbildung 2.13 veranschaulicht.

2.5 Evaluation Neuronaler Netze

Bei einer Klassifizierung werden Objekte verschiedenen Klassen zugeordnet. Dabei kommt es im Allgemeinen zu Fehlern, da in manchen Fällen ein Objekt einer falschen Klasse zugewiesen wird. Für diese Fehler können relative Häufigkeiten ermittelt werden, die quantitative Maße zur Beurteilung eines Klassifikators liefern. Es gibt verschiedene Evaluationsmöglichkeiten um die Güte eines neuronalen Netzes zu bewerten.

Eine Konfusionsmatrix oder auch Wahrheitsmatrix genannt, kann zur Beurteilung eines Klassifikators dienen. Häufig gibt es zwei mögliche Klassen, d.h., die Klassifikation ist binärer Natur. Im Falle einer solchen Klassifikation gibt es zwei Arten von Fehlern. Ein Objekt wird der ersten Klasse zugeordnet, obwohl es der zweiten angehört oder ein Beispiel wird der zweiten Klasse zugeordnet, obwohl es der ersten angehört. In einer quadratischen Tabelle werden die Häufigkeiten des Auftretens für alle möglichen Kombinationen von ermittelter und tatsächlicher Klasse eingetragen. Eine Konfusionsmatrix sieht für den Fall das es zwei Klassen gibt allgemein wie in Tabelle 2.1 abgebildet aus [29]. Es werden vier verschiedene Merkmalskombinationen unterschieden:

1. richtig positiv: Ein Beispiel ist positiv und wird als positiv erkannt.
2. falsch positiv: Ein Beispiel ist negativ und wird als positiv erkannt.
3. falsch negativ: Ein Beispiel ist positiv und wird als negativ erkannt.
4. richtig negativ: Ein Beispiel ist negativ und wird als negativ erkannt.

In die Konfusionstabelle werden jeweils die Auftrittshäufigkeiten der vier Merkmalskombinationen als ganze Zahlen eingetragen. Die richtig vorhergesagten Beispiele befinden sich auf der Diagonalen, die falsch vorhergesagten Objekte befinden sich in den übrigen Zellen der Matrix. Im Fall eines N-Klassen-Problems besteht die Konfusionsmatrix aus

Tabelle 2.1: Konfusionsmatrix für zwei mögliche Klassen

		tatsächliche Klasse	
		positiv	negativ
ermittelte Klasse	positiv	richtig positiv (TP)	falsch positiv (FP)
	negativ	falsch negativ (FN)	richtig negativ (TN)

einer N × N Matrix. Auch dann befinden sich die richtig klassifizierten Werte auf der Diagonalen.

Wie viele falsch positive oder falsch negative Werte in der Praxis tragbar sind, ist abhängig vom konkreten Anwendungsfall. In sicherheitsrelevanten Anwendungen, wie bei Krankheitstests oder bei Sicherheitsalarmen, ist es wichtig, dass kein positiver Fall unerkannt bleibt. Eine falsch positive Fehleinschätzung ist weniger kritisch als ein unentdecktes positives Ereignis. In anderen Anwendungsfällen kann dies genau andersherum sein. Bei einer Suchanfrage im Internet ist es beispielsweise nicht kritisch, wenn nicht alle korrekten Ergebnisse geliefert werden. Allerdings ist es störend, wenn falsche Objekte angezeigt werden. Aus diesem Grund ist der konkrete Anwendungsfall bei der Bewertung der Werte der Konfusionsmatrix zu beachten.

Mit den Werten der Konfusionsmatrix können verschiedene Maßzahlen für die Güte der Klassifikation berechnet werden. Eine wichtige Kennzahl ist die Genauigkeit (eng. Precision) [29]. Sie gibt den Anteil der korrekt positiv eingestuften Objekte an, bezogen auf die Menge aller positiv vorhergesagten Objekte (dies beinhaltet auch Objekte die falsch als positiv eingestuft wurden). Die Formel für die Genauigkeit ist eine bedingte Wahrscheinlichkeit:

$$\begin{aligned}\text{Genauigkeit} &= P(\text{richtig positiv erkannt} \mid \text{positiv erkannt}) \\ &= \frac{\text{Anzahl der richtig positiven}}{\text{Anzahl der richtig positiven} + \text{Anzahl der falsch positiven}} \\ &= \frac{\text{TP}}{\text{TP} + \text{FP}}\end{aligned}$$

Die Trefferquote (eng. Recall) bestimmt den Anteil der richtig positiv klassifizierten Objekte an der Gesamtheit aller tatsächlich positiven Beispiele [30]. Die Kennzahl stellt ebenfalls eine bedingte Wahrscheinlichkeit dar, die in Formeln wie folgt formuliert werden kann:

$$\begin{aligned}\text{Trefferquote} &= P(\text{positiv erkannt} \mid \text{tatsächlich positiv}) \\ &= \frac{\text{Anzahl der richtig positiven}}{\text{Anzahl der richtig positiven} + \text{Anzahl der falsch negativen}} \\ &= \frac{\text{TP}}{\text{TP} + \text{FN}}\end{aligned}$$

Die Werte für die Genauigkeit und Trefferquote liegen jeweils zwischen null und eins. Je näher die Kennzahlen an eins sind, desto besser. Da es schwierig ist, Modelle zu vergleichen, die eine hohe Trefferquote und eine niedrige Genauigkeit oder andersherum haben,

wurde ein Maß entwickelt, das beide Zahlen zugleich misst. Mit dem sogenannten F1-Kennwert wird mithilfe des harmonischen Mittels eine Kombination aus Trefferquote und Genauigkeit berechnet [30]. Dies ist wichtig, da es nicht möglich ist beide Gütekriterien unabhängig voneinander zu optimieren. Verbessert man die Genauigkeit, verschlechtert sich meist die Trefferquote und umgekehrt. Die Formel für den F1-Kennwert lautet:

$$\text{F1-Kennwert} = \frac{2 \cdot \text{Genauigkeit} \cdot \text{Trefferquote}}{\text{Genauigkeit} + \text{Trefferquote}}$$

Der F1-Wert kann Werte zwischen 0 und 1 annehmen. Das Minimum 0 wird erreicht wenn alle positiven Werte falsch klassifiziert wurden. Dann ist $TP = 0$. Das Maximum 1 wird erhalten, wenn Genauigkeit und Trefferquote ebenfalls ihren Maximalwert 1 erreichen. Dies ist genau dann der Fall wenn $FN = FP = 0$, d.h. es keine falschen Vorhersagen gibt. Probleme des F1-Wertes sind, dass er unabhängig vom TN-Wert ist und nicht symmetrisch ist, wenn die Klassen vertauscht werden.

Neben diesen Kennzahlen ist es ebenfalls üblich die Richtigkeit (eng. Accuracy) der Klassifikation zu errechnen [31]. Diese lässt sich mit den Werten der Konfusionsmatrix berechnen [29]:

$$\text{Richtigkeit} = \frac{\text{Anzahl der richtigen Vorhersagen}}{\text{Anzahl aller Vorhersagen}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Es werden alle richtigen Klassifizierungen durch die Gesamtheit aller Klassifizierungen geteilt, um die Güte eines Netzes zu bewerten. Dies ist ein sehr einfaches und oft genutztes Mittel zur Bewertung von kNNs. Es können Werte aus dem Intervall $[0, 1]$ angenommen werden, wobei eins eine perfekte Klassifikation bedeutet.

Nach [32] sind der F1-Kennwert und die Richtigkeit die populärsten Metriken zur Beurteilung von Klassifizierungsaufgaben. Allerdings sind beide eher optimistische statistische Maße, vor allem bei unausgeglichenen Datensätzen [32]. Wird ein trivialer Algorithmus, der lediglich lernt welche Klasse am häufigsten vorkommt und immer diese Klasse vorhersagt, auf einem unbalancierten Datensatz mit den eben beschriebenen Maßen angewendet, so wird die Güte der Vorhersage nicht realistisch gemessen. Besteht beispielsweise das Datenset zu 95% aus positiven Werten und nur zu 5% aus negativen Werten kann ein Klassifikator der immer alle Beispiele positiv einstuft eine extrem hohe Richtigkeit erreichen. Allerdings würde ein solcher Klassifikator auf einem anderen Datensatz, der besser balanciert ist, erheblich schlechter abschneiden.

Ein zuverlässigeres Maß besonders bei unbalacierten Daten ist der Matthews Correlation Coefficient (MCC). Dieses gibt nur dann einen hohes Ergebnis zurück, wenn alle vier Werte in der Konfusionsmatrix gute Werte proportional zur Anzahl der positiven und negativen Objekte im Datenset haben. Nach [33] ist der MCC das einzige Maß, welcher nur einen hohen Wert zurückgibt, wenn sowohl die Mehrheit der positiven als auch der negativen Objekte richtig vorhergesagt wurde. Der MCC verwendet die Werte der Konfusionsmatrix und berechnet den Pearson Korrelationskoeffizient [30] zwischen den tatsächlichen und vorhergesagten Werten:

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP}) \cdot (\text{TP} + \text{FN}) \cdot (\text{TN} + \text{FP}) \cdot (\text{TN} + \text{FN})}}$$

Abbildung 2.14: Bilder zur Unterscheidung zwischen Klassifizierung (Objekt wird auf dem Bild erkannt), Lokalisierung (Ort des Objektes wird erkannt) und Objekterkennung (Objekt und Ort werden erkannt). Die Fotos wurden aus dem Datensatz der RTG entnommen.

Der Wertebereich liegt im Intervall von $[-1, 1]$, wobei im schlechtesten Fall der MCC-Wert -1 und im besten Fall 1 ist.

2.6 Objekterkennung

Zunächst ist es wichtig die Objekterkennung abzugrenzen und verschiedene Begriffe zu unterscheiden. Bei der Bildklassifizierung wird ein Eingabebild mit einem Klassenlabel versehen, wenn sich ein Objekt im Bild befindet. Ist z.B. ein Rad auf einem Bild abgebildet, kann dieses Foto der Klasse *Fahrrad* zugeordnet werden. Ist die Position auf der sich ein Objekt im Bild befindet durch ein minimal umgebendes Rechteck, eine sogenannte *Bounding Box*, eingerahmt, so wird von einer Objektlokalisierung gesprochen. Die Objekterkennung beinhalten beides, die Bildklassifizierung und die Objektlokalisierung. Für ein Eingabebild werden alle Objekte mit einem minimalen Rechteck umschlossen. Dabei können mehrere gleiche Objekte als auch unterschiedliche Objekte auf einem Bild erkannt und lokalisiert werden. Vergleiche dazu Abbildung 2.14.

Zu beachten ist, dass es mithilfe des minimal umgebenden Rechtecks nicht möglich ist die Form des Objektes festzustellen. Damit kann keine Aussage getroffen werden, ob es kurvige Abschnitte in der Objektumrandung gibt. Für diese Problemstellung gibt es die Bildsegmentierung [34], welche eine Erweiterung der Objekterkennung darstellt. Hier wird ein Objekt pixelweise markiert, wodurch die Form des Objektes bestimmbar wird. Dies ist für verschiedene Anwendungsfälle, wie z.B. in der Medizin interessant, allerdings in unserem Kontext nicht relevant.

Eine Objekterkennung ist mit Methoden des Maschinellen Lernens ohne die Verwendung von neuronalen Netzen möglich. Dafür können *Histograms of Oriented Gradients* [35] und *Support Vector Machine* [36] verwendet werden oder der *Viola-Jones Algorithmus* [37]. Allerdings wird in der aktuellen Forschung zumeist auf CNNs zurückgriffen, da diese bessere Erkennungsraten erreichen. Heutzutage ist dies der populärste Weg zur Objekterkennung und wird deshalb in der Masterarbeit verwendet. Die neuesten Objekterkennungmethoden

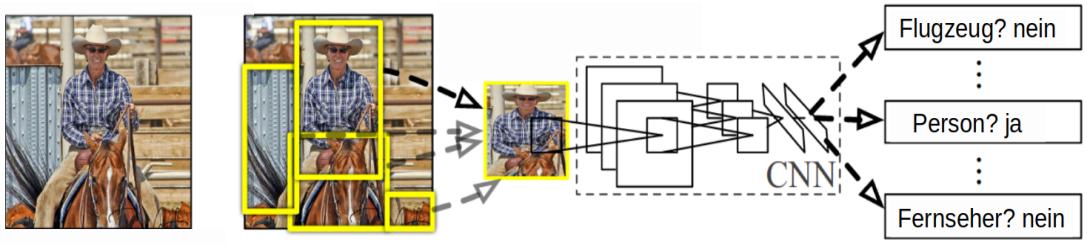


Abbildung 2.15: Vorgehen bei der Objekterkennung mit R-CNNs: (1) Das System bekommt das Eingabebild, (2) es werden ca. 2000 Regionsvorschläge ermittelt, (3) für jeden Vorschlag werden die Merkmale mit einem CNN bestimmt, (4) die Regionen werden mit einer linearen SVM klassifiziert. Das Schaubild wurde entnommen aus [38].

mit CNNs können in zwei Klassen unterschieden werden. Es gibt zweistufige regionsbasierde und einstufige Detektoren.

Zweistufige Detektoren

Im Bereich der zweistufigen Objektdetektoren gibt es verschiedene aufeinander aufbauende Lösungen, die im Folgenden kurz vorgestellt werden. Dabei ist die Idee, dass für ein Bild eine Menge von Regionen, sogenannte Regions of Interest (RoI) bestimmt werden in denen möglicherweise ein Objekt zu sehen ist. Diese Regionen werden daraufhin klassifiziert. Sogenannte Regional CNN (R-CNN) [38] nutzen diese grundlegende Vorgehensweise erstmals. In Abbildung 2.15 ist der Ablauf der Objekterkennung mit R-CNNs dargestellt. Mit einem *Selectiv Search Algorithmus* [39] werden ca. 2.000 Bildregionen vorgeschlagen auf denen sich ein Objekt befinden könnte. Für jede RoI wird dann mithilfe eines CNNs ein Merkmalsvektor ausgegeben. Anschließend können die Regionen mit einer klassenspezifischen linearen Support Vector Maschine (SVM) [40, 41] klassifiziert werden. Wenn ein Objekt erkannt wurde kann dann die Bounding Box verfeinert und die Objekte mit hoher Klassifikationsgenauigkeit ausgegeben werden.

Fast R-CNN [42] verbessern R-CNNs indem das gesamte Bild in ein einziges CNNs eingelesen wird. Dadurch ist die Architektur mehr als 200 fach schneller, mit einer Testzeit von 0,3 Sekunden pro Bild. Die RoIs werden wie bei R-CNNs durch einen *Selectiv Search Algorithmus* ermittelt. Fast R-CNN berechnet die Merkmalskarten aus dem gesamten Bild. Anschließend werden die ROIs direkt aus ihnen abgeleitet. Die Klassifizierung erfolgt nicht wie beim R-CNN durch eine SVM, sondern in einer Softmaxschicht.

Bei Faster R-CNNs [43] werden die Regionsvorschläge nicht mehr mit dem *Selectiv Search Algorithmus* gesucht, sondern ein *Region Proposal Network* [43] verwendet, dass auf der letzten Faltungsschicht des CNNs aufsetzt. Faster R-CNNs benötigen beim Testen nur 0,2 Sekunden pro Bild.

Im Gegensatz zu den vorherigen regionsbasierten Detektoren ist das Region-based Fully Convolutional Network (R-FCN) [44] mit ähnlicher Genauigkeit schneller, da alle lernbaren Schichten Faltungsschichten sind. Das Netz enthält keine vollverknüpften Schichten [45]. In der letzten Schicht eines R-FCN findet die einzige regionsbasierte Operation statt.

Diese Schicht ist eine positionssensitve RoI-Pooling-Schicht [44]. Diese aggregiert die Ausgabe der letzten Faltungsschicht und generiert Ergebniswerte für jede RoI.

Einstufige Detektoren

Es gibt zwei prominente einstufige Detektoren: Single Shot Detector (SSD) [46] und You Only Look Once (YOLO) [47]. Sie haben das gleiches Algorithmuskonzept. Beide benötigen den ersten Hauptschritt von R-CNN, die Bestimmung der RoIs, nicht. Aus diesem Grund sind SSD und YOLO schneller und dadurch für Echtzeitprozesse anwendbar. Der Single Shot Detector [46] verwendet standard Bounding Boxen mit verschiedenen Radien und Skalierungen. Der Algorithmus nutzt unterschiedliche Skalierungen, indem kleinere und größere Merkmalskarten verschiedener Faltungsschichten verwendet werden. Das Netzwerk generiert Werte wie stark ein Objekt aus jeder Objektkategorie in jeder standard Bounding Box zu sehen ist und passt die Boxen an, sodass sie besser das Objekt umschließen. Da die Methode eine große Anzahl an Boundig Boxen generiert, wird eine Non-Maximum Suppression (NMS) [48] verwendet, welche die Boxen unterhalb eines bestimmten Konfidenzschwellwerts verwirft und die Box mit dem größten Wert auswählt. Nach [48] kann eine NMS als eine lokale Maximumsuche formuliert werden, da lokal ein Wert gesucht wird, der größer ist als seine Nachbarn.

Beim Detektor YOLO [47] wird ein gegebenes Bild in ein Gitter eingeteilt. Jede Zelle des Gitters berechnet eine festgelegte Anzahl an Bounding Boxes, sog. Anchor Boxes. Außerdem wird ein Konfidenzfaktor berechnet. Dieser besteht aus der Wahrscheinlichkeit, dass die jeweilige Zelle ein Objekt enthält, sowie der Genauigkeit der vorhergesagten Box. Neben den Vorhersagen für die Lokalisierung des Objektes wird auch die Klassifizierung pro Zelle erledigt. Jede Zelle ist dabei für die Vorhersage der Klasse des Objekts, das in der Zelle enthalten ist, zuständig. Beide Information werden dann am Ende zur Objekterkennung kombiniert. Die Architektur ist sehr schnell, allerdings ist die Lokalisierung von Objekten nicht so genau.

Bilddatensätze

Die Güte der Objekterkennung ist abhängig von dem Bilddatensatz mit dem das CNN trainiert wurde. Dabei ist nicht nur die Größe eines Trainingsdatensatzes entscheidend, sondern auch wie vielfältig die enthaltenen Fotos sind. Bilder werden oft aus einem kanonischen Blickwinkel [49] aufgenommen. Dies ist ein typischer Blickwinkel aus dem das Objekt besonders leicht erkannt wird. Abhängig davon ob Fotos aus einem kanonischen Blickwinkel geschossen wurden können drei Arten von Bildern unterschieden werden [50]:

1. ikonische Bilder: Bilder enthalten ein einziges großes, zentriertes Objekt in kanonischer Perspektive.
2. ikonische Szenen: Bilder mit mehreren Objekten aus kanonischem Blickwinkel, meist ohne Personen.
3. nicht-ikonische Bilder: Bilder beinhalten eine Vielzahl an Kontextinformationen und viele verschiedenen Objekte pro Bild mit variierendem Blickwinkel.

Abbildung 2.16: Je vier Beispiele ikonischer Bilder (links), ikonischer Szenen (mitte) und nicht-ikonischer Bilder (rechts). Abbildung entnommen aus [50].

Beispiele für die drei verschiedenen Bildarten sind in Abbildung 2.16 abgebildet. Es wurde in [51] gezeigt, dass Netze die mit nicht-ikonischen Bildern trainiert wurden besser generalisieren. Oft sind die Erkennungssysteme mit ikonischen Bildern gut, wenn sie auch auf solchen Datensätzen getestet werden. Allerdings haben sie größere Schwierigkeiten auf Realweltbildern Objekte zu erkennen, da hier in der Regel kein ikonischer Blickwinkel und oft ein komplexer Hintergrund vorliegt. Aus diesem Grund ist es sinnvoll bereits für das Training nicht-ikonische Bilder zu verwenden. Diese können beispielsweise auf der Webseite von Flickr gefunden werden. Flickr enthält Fotos von Amateurfotografen, die mit Metadaten und Schlagwörtern verstehen wurden. Damit ist es möglich nach Bildern mit bestimmten Objekten zu suchen. Dabei ist es nutzbringend nach verschiedenen Synonymen oder mehreren Begriffen gleichzeitig zu filtern, um eine große Diversität an Bildern zu erhalten. Dieses Vorgehen wurde beispielsweise bei der Erstellung des Common Objects in Context (COCO) Datensatzes [50] verwendet. Es ist zu beachten, dass nicht alle Bilddatensätze nicht-ikonische Bilder enthalten. Deshalb muss bei der Auswahl eines Bilddatensatz für das Training des Netzes nicht nur ein Datensatz nach der Anzahl der Bilder, sondern auch anhand der Diversität ausgewählt werden. Eine der bekanntesten Verzerrungen ist das sich Objekte meinst im Zentrum von Bildern befinden [51]. Dadurch kann es passieren das Algorithmen lernen, dass sich in der Mitte eines Bildes immer ein Objekt befindet und dadurch Objekte am Rand eines Bildes nicht erkannt werden.

Datensätze sind nicht nur Quelle zum Training sondern auch ein Mittel um Algorithmen zu vergleichen. Kritisch ist zu sehen, dass der Maßstab oftmals darauf liegt, die besten Ergebnisses auf dem neusten Datensatz zu erreichen [51]. Ein großes Warnzeichen ist, dass es kaum Forschungsliteratur gibt, in der auf einem Datensatz trainiert und auf einem Anderem getestet wurde [51]. Wenn die Datensätze wirklich die Realwelt abbilden würden, wäre es kein Problem auf verschiedenen Datensätzen zu trainieren und testen. Dadurch wäre mehr Zugang zu dringend benötigten verschlagworteten Daten möglich. Mit einer Kreuz-Datensatz-Validierung kann gezeigt werden, wie stark befangen Datensätze sind, indem auf einem Datensatz trainiert wird und auf einem Anderen getestet wird [51]. In [51] wurden sechs Datensätze verglichen: Scene Understanding (SUN) [52], LabelMe [53], PASCAL Visual Object Classes 2007 (VOC 2007) [54], ImageNet [55], Caltech-101 [56], und MSRC [57].

Tabelle 2.2: Die Datensatzvoreingenommenheit wird gezeigt durch Kreuzvalidierung sechs verschiedener Datensätze. Es sollen Personen erkannt werden. In den Spalten der Tabelle ist die Güte abzulesen, wenn auf einem Datensatz getestet und auf den Anderen trainiert wurde. In den Zeilen ist dargestellt, dass auf einem Datensatz trainiert und auf alle Anderen getestet wurde. Die Diagonalelemente liefern die Richtigkeit wenn auf einem Datensatz trainiert und getestet wurde. Zum Vergleich wie stark sich die Vorhersagegenauigkeit ändert, wenn auf einem anderen Datensatz getestet wird, ist am Ende der Tabelle gegenübergestellt wie sich das Testen auf dem selben Datensatz unterscheidet, verglichen mit dem Testen auf allen anderen Datensätzen im Durchschnitt. In der letzten Spalte ist dieser Unterschied in Prozent dargestellt. Die Tabelle wurde entnommen aus [51].

Training \ Test	SUN09	LableMe	VOC 2007	ImageNet	Caltech101	MSRC	selbst	Durchschnitt andere	Unterschied in Prozent
SUN09	**69,6**	56,8	37,9	45,7	52,1	72,7	69,6	53,0	24%
LableME	58,9	**66,6**	38,4	43,1	57,9	68,9	66,6	53,4	20%
VOC 2007	56,0	55,6	**56,3**	55,6	56,8	74,8	56,3	59,8	-6%
ImageNet	48,8	39,0	40,1	**59,6**	53,2	70,7	59,6	50,4	15%
Caltech101	24,6	18,1	12,4	26,6	**100**	31,6	100	22,7	77%
MSRC	33,8	18,2	30,9	20,8	69,5	**74,7**	74,7	34,6	54%
Durchschnitt andere	44,4	37,5	31,9	38,4	57,9	63,7	71,1	45,6	36%

In Tabelle 2.2 wurden die Ergebnisse der Kreuzvalidierung verschiedener Datensätze zur Erkennung von Personen aus [51] dargestellt. Die Validierung der Klassifizierung und Objekterkennung von Autos, sowie die Klassifizierung von Personen ist ebenfalls in [51] detailliert dargestellt. Auf die Einbindung dieser wurde aufgrund des erheblichen Umfangs verzichtet. Die Problematik sollte an den Ergebnissen der Kreuzvalidierung der Personenerkennung ausreichend erkennbar sein.

In den Spalten der Tabelle ist die Güte abzulesen, wenn auf einem Datensatz getestet und auf den Anderen trainiert wird. Hingegen ist in den Zeilen dargestellt, dass das Training auf einem Datensatz durchgeführt und auf den Anderen getestet wurde. Es ist zu beachten, dass die Training- und Testverfahren nicht die gleichen sind, die ursprünglich auf dem Datensatz verwendet wurden. Die Güte ist daher nicht entscheidend, vielmehr die Unterschiede der Leistungen der Datensätze. Die besten Vorhersageergebnisse sind auf der Diagonalen der Tabelle zu finden. Dies war zu erwarten, da in diesem Fall auf dem gleichen Datensatz trainiert und getestet wurde. Generell haben fast alle Datensätze schlechtere Ergebnisse, wenn sie auf unterschiedlichen Datensätzen trainiert und getestet werden. Betrachtet man die Zeilen der Tabelle, kann evaluieren werden wie gut ein Datensatz auf Anderen generalisiert. In den Spalten kann analysiert werden wie einfach ein Datensatz für die Anderen ist.

Caltech 101 und MSRC sind die einfachsten Datensätze, da sie bei bei der Personenerkennung die größten Spaltendurchschnitte und die kleinsten Zeilendurchschnitte haben. Sie sind für andere Datensätze beim Testen sehr leicht und können schlecht auf Anderen generalisieren. Auffällig ist, dass beide ikonischen Datensätzen Caltech 101 und MSRC die besten Ergebnisse liefern, wenn sie auf ihrem Datensatz trainiert und getestet werden und den stärksten Abfall haben, wenn sie auf anderen Datensätzen getestet werden. Dies ist am hohen Unterschied in der letzten Spalte abzulesen, bei dem der Durchschnitt des Testen auf anderen Datensätzen mit dem Testen auf dem eigenen Datensatz ins Verhältnis gesetzt wird. VOC 2007 ist der Datensatz, der am besten generalisiert. Dies ist am hohen Zeilendurchschnitt zu sehen. Der Datensatz hat den niedrigsten Spaltendurchschnitt was bedeutet, dass es für andere Datensätze schwierig ist, wenn sie auf VOC 2007 getestet werden. Damit konnten feststellen werden, dass die ikonischen Datensätze Caltech 101 und MSRC nicht zur Generalisierung geeignet sind, hingegen hat VOC 2007 die besten Ergebnisse geliefert. Dieser hat als einziger Datensatz besser auf anderen Datensätzen generalisieren können, wie auf seinem Eigenen. Aus diesem Grund hat er mit -6% einen negativen Unterschied in der letzten Spalte der Tabelle. Alle anderen Datensätze konnten erheblich schlechter auf unbekannten Datensätzen generalisieren.

Daraus lässt sich schließen, dass es eine Voreingenommenheit bei Datensätzen geben muss. Nach [58] ist die visuelle Welt so komplex, dass jede endliche Menge an Beispielen nur einzelne Aspekte beschreiben kann. Außerdem werden Beispielbilder zu einem bestimmten Zweck gesammelt, sodass unvermeidbar nur eine spezielle visuelle Region abgebildet wird. In [51] werden verschiedene Arten von Datensatz Problemen detailliert beschrieben, die zu einer Voreingenommenheit führen. Diese werden kurz zusammengefasst:

1. Auswahlvoreingenommenheit: Datensätze beinhalten meist nur eine bestimmte Art von Bildern z.B. Straßenszenen oder Bilder aus einer Internetschlagwortsuche.

2. Erfassungsvoreingenommenheit: Fotografen schießen ihre Bilder auf eine ähnliche Art und Weise, sodass das Foto ästhetisch aussieht. Dieses Problem wird auch über verschiedene Datensätze ähnlich sein.
3. Benennungsvoreingenommenheit: Da semantische Kategorien oft nicht detailliert genug definiert werden und verschiedene Personen, welche Objekten ihren Klassen zuweisen, unterschiedliche Klassenbenennungen für ein Objekt verwenden.
4. Negativdatensatz-Voreingenommenheit: Der Negativdatensatz wird als Rest der Welt angesehen. Falls dieser nicht repräsentativ oder unbalanciert ist kann dies Klassifikatoren erzeugen, die zu zuversichtlich sind.

Wie können wir mit der Datensatzvoreingenommenheit umgehen? In [59] wird ein Rahmenwerk beschrieben wie Schäden verursacht durch Datensatzvoreingenommenheit rückgängig gemacht werden können indem mehrere Datensätze zum Training verwendet werden. Es werden dabei zwei verschiedene Gewichte gelernt: (1) Voreingenommenheitsvektoren für jeden individuellen Datensatz und (2) Gewichte der visuellen Welt die alle Datensätze gemeinsam haben. Sie werden gelernt indem die Voreingenommenheit jedes Datensatzes rückgängig gemacht wird. Einen anderen Ansatz schlägt [58] vor. Hier werden Deep Convolutional Activation Feature (DeCAF) [60] Merkmale genutzt. Damit kann das Problem der Datensatzvoreingenommenheit jedoch nicht ganz gelöst werden, da das Negativdatensatz Problem immer noch besteht. In [61] wird gezeigt wie 12 verschiedene Datensätze verknüpft werden können und welche Probleme dabei auftreten.

2.7 Soft- und Hardware

Diese Sektion soll eine knappen Überblick über die in der Arbeit verwendete Soft- und Hardware geben. Als Programmierumgebung wurde Python[1] verwendet. Python ist eine objektorientierte Programmiersprache. Sie hat eine einfache, leicht zu erlernende Syntax, welche die Lesbarkeit erhöht und daher die Kosten für die Programmwartung reduziert. Ein großer Vorteil von Python ist die umfangreiche Standardbibliothek, die immer weiter wächst. Python bietet eine Vielzahl von Bibliotheken und Werkzeugen im Bereich Maschinellen Lernen, wie z.B. Pandas[2], Matplotlib[3], Scikit-learn[4] und Numpy[5], wodurch sich eine umfangreiche Data Science Community gebildet hat. Mithilfe dieser können sogar neue Entwickler mit geringem Vorkenntnissen und wenig Zeilen Code ein neuronales Netz nutzen [62]. Es werden in der Arbeit für die Objekterkennung verwendeten Bibliotheken im Folgenden kurz vorgestellt.

Open Source Computer Vision (OpenCV)[6] ist eine öffentliche Biblitothek zur Bild- und Videoanalyse. Die Bibliothek wurde anfangs von Intel entwickelt, ist in C und C++ geschrieben und läuft unter Linux, Windows und Mac OS [63]. Sie ist für Echtzeitanwen-

[1] `https://www.python.org/doc/essays/blurb/`
[2] `https://pandas.pydata.org/`
[3] `https://matplotlib.org/`
[4] `https://scikit-learn.org/stable/`
[5] `https://numpy.org/`
[6] `https://opencv.org/`

dungen entwickelt worden und hat Schnittstellen für Python, Ruby, Matlab und andere Programmiersprachen. Heutzutage besitzt die Bibliothek mehr als 2.500 optimierte Algorithmen und wurde weltweit mehr als 2,5Millionen mal heruntergeladen [64].

TensorFlow[7] ist eine öffentliche Programmbibliothek, die vom Google Brain Team zur einfachen Anwendung des maschinellen Lernen entwickelt wurde. In TensorFlow werden Datenflussgraphen verwendet [65]. Berechnungen können auf einer Vielzahl heterogener Systeme, wie mobile Geräte (Tablets, Smartphones) bis zu verteilten Großsystemen mit Hunderten von Maschinen und Tausenden von Rechengeräten, wie GPU-Karten, mit nur geringen Änderungen ausgeführt werden [65]. Das System ist flexibel und kann angewendet werden um eine Vielzahl von Algorithmen auszudrücken, einschließlich Training und Testen von tiefen neuronalen Netzen [65].

Convolutional Architecture for Fast Feature Embedding (Caffe) [66] ist eine öffentliche Bibliothek mit Methoden für tiefe Netzarchitekturen. Die Bibliothek ist in C++ geschrieben und ebenfalls in Matlab und Python implementiert. Der große Vorteil von Caffe ist die Schnelligkeit von Operationen. Es wird Compute Unified Device Architecture (CUDA)[8] unterstützt und die Bibliothek kann falls nötig zwischen dem Prozessor und der Grafikkarte wechseln [67]. Aufgrund der Schnelligkeit ist Caffe sehr nützlich für die Objekterkennung, da diese eine große Anzahl an Bildern zum Trainieren benötigt [68].

Gute Grafikkarten können die Leistung enorm beschleunigen und dadurch die Trainingszeiten stark reduzieren. Heutzutage können GPU Anwendungen, die aufgrund langer Ausführungszeiten für unmöglich gehalten wurden, ausführt werden [69]. Die moderne GPU ist nicht nur eine leistungsstarke Grafikmaschine, sondern auch ein hoch parallel programmierbarer Prozessor mit maximaler Rechen- und Speicherbandbreite der das CPU-Gegenstück deutlich übertrifft [70]. Das GPU-Design widmet sich der Optimierung von Grafikoperationen die von Natur aus parallel sind. Hingegen ist die CPU-Hardware für generische, sequentielle Codes optimiert und hauptsächlich für nicht rechnerische Aufgaben wie Verzweigungen vorgesehen [71]. Durch die Veröffentlichung der CUDA, einer leistungsstarken und einfach zu verwendenden Programmierumgebung für Nvidia-Karten, ist GPU Rechnen auch für Entwickler zugänglich die keine Experten für Computergrafik sind [71].

Für die Masterarbeit ist ausschließlich Python als Entwicklungsumgebung herangezogen worden. Es wurde die Python Version 3.6 genutzt. Zur Verfügung stand Fernzugriff auf eine Nvidia Titan X Grafikkarte. CUDA 10.1 wurde verwendet. Für die Gesichtserkennung wurde mit den Bibliotheken OpenCV und Caffe gearbeitet. OpenCV ist ebenfalls für die Erkennung von Personen und Fahrrändern eingesetzt worden. Bei der Geschlechtserkennung anhand des gesamten Körpers wurde sich TensorFlow bedient.

[7] `https://www.tensorflow.org/`

[8] CUDA ist eine von Nvidia entwickelte Programmier-Technik, mit der Programmteile durch den Grafikprozessor abgearbeitet werden können

3 Methodik und deren Evaluation

In diesem Kapitel werden zuerst die Zielgruppen des Radtourismus beschrieben. Anschließend wird versucht, diese auf Bildern zu erkennen. Mit einer Gesichtserkennung wird zunächst das Geschlecht- und Alters von Personen vorhergesagt. Daraufhin wird in Abschnitt 3.3.1 beschrieben wie Personen auf Bildern erkannt werden können. Da der Praxisdatensatz viele Ganzkörperbilder enthält wird das Geschlecht in Abschnitt 3.4 anhand des gesamten Körpers vorhergesagt. Das Kaptiel wird mit der Beschreibung und Analyse einer Fahrraderkennung abgeschlossen.

3.1 Analyse relevanter Zielgruppen und deren Charakteristika im Radtourismus

Der Radtourismus nimmt eine immer größere Bedeutung ein, da bereits jeder dritte Deutsche Ausflüge mit dem Rad unternimmt [72]. Um die Angebote im Radtourismus gut annoncieren zu können und bestmöglichst Kunden anzuwerben ist es wichtig, diese zielgerichtet zu adressieren. Dafür muss zunächst herausgefunden werden welche Zielgruppen im Radtourismus unterschieden werden können.

Mithilfe von Interviews mit der Ruhr Tourismus GmbH wurden die Zielgruppen im Radtourismus ermittelt. Zunächst wurden in zwei Interviews mit je einem Experten die relevanten Personenklassen und die daraus aufbauenden Zielgruppen identifiziert. Folgende Klassen sind für den Radtourismus herausgearbeitet worden: Frau, Mann, junge Frau, junger Mann, Best Ager, weiblicher und männlicher Best Ager sowie Kind und Jugendlicher. Diese Klassen werden in Tabelle 3.1 weiter spezifiziert. Prinzipiell ist es für den Radtourismus interessant, das Geschlecht zu unterschieden. Bei Kindern und Jugendlichen ist dies jedoch irrelevant. Die wichtigsten Zielgruppen sind: Junges Paar, Best Ager Paar, Familie, Frauengruppe und gemischtgeschlechtliche Gruppe. Wie sich diese aus den Personenklassen ergeben ist der Tabelle 3.2 zu entnehmen.

Um die Personenklassen auf Bildern zu erkennen, ist es nötig, dass Alter und Geschlecht der Personen zu ermitteln. Außerdem sollten Regeln festgelegt werden, welches Alter welche Personenklassen definiert. Mit den Personenklassen können Zielgruppen festgelegt werden. Hier ist zu definieren, wann welche Personen zu einer Zielgruppe gehören. Dabei wurde unter anderem diskutiert, ab welcher Personenzahl von einer Gruppe gesprochen wird und welche Personen eine Familie charakterisieren. Dafür fand eine Rücksprache mit drei weiteren Experten der RTG statt, um Altersgrenzen und genaue Zielgruppenbeschreibungen zu prüfen. Anhand der Interviews konnten die in Tabelle 3.1 und Tabelle 3.2 relevanten Personenklassen und Zielgruppen charakterisiert werden.

Tabelle 3.1: Personenklassen

Personenklassen	Charakteristikum
Frau	weiblich, 25-100 Jahre alt
Mann	männlich, 25-100 Jahre alte
junge Frau	weiblich, 25-32 Jahre alt
junger Mann	männlich, 25-32 Jahre alt
männlicher Best Ager	männlich, 48-100 Jahre alt
weiblicher Best Ager	weiblich, 48-100 Jahre alt
Kind	0-12 Jahre alt
Jugendlicher	15-20 Jahre alt

Tabelle 3.2: Zielgruppendefinition

Zielgruppen	Charakteristikum
Junges Paar	1 junge Frau und 1 junger Mann und keine andere Personengruppe
Best Ager Paar	1 weiblicher Best Ager und 1 männlicher Best Ager und keine andere Personengruppe
Familie	(mind. 1 Frau oder mind. 1 Mann) und (mind. 1 Kind oder mind. 1 Jugendlicher)
Frauengruppe	mind. 3 Frauen und keine andere Personengruppe
gemischtgeschlechtliche Gruppe	(mind. 3 Personen, die Frauen oder Männer sind) und (mind. 1 Frau) und (mind. 1 Mann)

Zu beachten ist, dass nicht bestimmt werden kann, ob zwei Personen ein Liebespaar oder befreundet sind. Genauso ist es über die Bilder der Zielgruppe Familie nicht möglich zu sagen, ob die abgebildeten Personen in einem Verwandtschaftsverhältnis zueinander stehen.

Die Best Ager sind einer der wichtigsten Zielgruppen im Radtourismus. Dies ist auch aus der Statistik des ADFC zu entnehmen [73]. Demnach sind die meisten Radreisenden zwischen 45 und 64 Jahren. Sie haben einen Anteil von über 50%. In der Regel spricht man von Best Agern bei einem Alter zwischen 50 und 65 Jahren, jedoch ändert sich das Verhältnis vom tatsächlichen zum gefühlten Alter immer weiter[1]. Laut Duden sind Best Ager Personen, die zur anspruchsvollen, konsumfreudigen Kundengruppe der über 40- bzw. 50-Jährigen gehören. Wer heute mit 70 noch aktiv, gesund und finanzkräftig ist, gehört ebenfalls zur Zielgruppe. Da für die Altersgruppen ein vortrainiertes Netz verwendet wurde, beginnt die Altersgruppe der Best Ager hier bei 48 Jahren und endet mit 100 Jahren. Wobei anzumerken ist, dass es kaum Radfahrfotos von 100-Jährigen gibt. Im vortrainierten neuronalen Netz ist diese Altersobergrenze gesetzt und keine weitere Altersunterscheidungen werden für Personen über 60 vorgenommen. Besonders durch die technische Entwicklung der Elektrofahrräder wird die Zielgruppe der Best Ager ver-

[1] `https://www.best-ager-50plus.de/jobclubs`

größert[2]. Im Bericht des ADFC aus dem Jahr 2020 nutzen bereits 30% der Ausflügler Elektroräder [72].

Des Weiteren sind junge Leute über 25 ebenfalls aktiv im Radtourismus. Der ADFC Statistik 2020 ist zu entnehmen, dass viele Paare gemeinsame Radreisen unternehmen. Über 55% reisen demnach mit ihrem (Ehe-)Partner. Dies stimmt mit den Aussagen der Experten überein, welche Best Ager Paare als größte Zielgruppe, gefolgt von jungen Paaren, einstufen. Männer unternehmen Radreisen in der Regel mit ihren Partnerinnen, ihrer Familie oder in Gruppen mit gemischten Geschlechtern, allerdings weniger Männer unter sich. Deshalb bildet eine Männergruppe keine relevante Zielgruppe im Radtourismus.

Es gibt kaum Literatur um die ermittelten Zielgruppen zu überprüfen. Im Bericht des Bundesministerium für Wirtschaft und Technologie zur Grundlagenuntersuchung Fahrradtourismus in Deutschland aus dem Jahr 2009 werden die drei Zielgruppen Rad fahren mit Tourenrad, Mountainbiken und Rennrad fahren unterschieden [74]. Diese berücksichtigen nicht bestimmte Alters- oder Geschlechtergruppen, sondern legidlich wie aktiv mit welcher Art Rad gefahren wird.

Die in Tabelle 3.1 und Tabelle 3.2 beschriebenen Personenklassen und Zielgruppen sollen auf dem vom RTG zur Verfügung gestellten Bilddatensatz erkannt werden. Dafür muss zunächst das Alter und Geschlecht der Personen in den Bildern bestimmt werden um daraufhin mittels Regeln die Personenklassen und Zielgruppen zu ermitteln. Zur Schätzung des Geschlechts und Alters wird auf ein vortrainiertes Netz zurückgegriffen, da für eine derart komplexe Aufgabe ein großes Trainingsset und hoher Trainingsaufwand nötig ist.

3.2 Alters und Geschlechtserkennung anhand des Gesichts

Um eine Zielgruppenzuordnung zu ermöglichen sind Informationen über das Geschlecht und Alter der Personen auf Bildern erforderlich. Aus diesem Grund wird ein vortrainiertes Netz zur Gesichtserkennung betrachtet, welches das Geschlecht sowie die Altersgruppen von Personen vorhersagen kann. In diesem Abschnitt wird zunächst das verwendete CNN beschrieben und anschließend auf dem Praxisdatensatz der RTG getestet.

3.2.1 Beschreibung des vortrainiertes Netzes zur Gesichtserkennung

Um das Geschlecht und Alter vorherzusagen, muss zunächst das Gesicht auf einem Bild erkannt werden. Eine Klassifizierung auf ganzen Fotos ist nicht möglich, da verschiedene Gesichter auf einem Bild zu sehen sein können. Deshalb werden zunächst die Gesichter auf Bildern erkannt, lokalisiert und dementsprechend das Bild auf die Bereiche, in denen ein Gesicht zu sehen ist, zugeschnitten. Ein Gesicht wird dabei mit einer gewissen Sicherheit

[2] `https://www.adfc.de/fileadmin/user_upload/Expertenbereich/Touristik_und_Hotellerie/Positionspapiere/ADFC__Tourismuspolitische_Positionen.pdf`

erkannt, diese muss größer einem festgelegten Schwellwert sein, damit es nicht verworfen wird. Für die erkannten Gesichter werden seine Merkmale extrahiert und damit für jedes Gesicht das Geschlecht und Alter vorhergesagt. Mit diesen Informationen kann eine Person einer Personengruppe zugeordnet werden.

Die relevanten Personengruppen im Radtourismus wurden in Tabelle 3.1 dargestellt. In der Implementierung der Gesichtserkennung wurden die Personengruppen ergänzt und die Anzahl der Personen in den einzelnen Personengruppen auf zunächst auf 0 gesetzt. Mit Regeln, die vorwiegend aus wenn-Bedingungen bestehen, kann dann bei bekanntem Geschlecht und Alter eine Person einer Personengruppe zugeordnet werden. Es wird dabei beispielsweise betrachtet, ob es sich um die Personengruppe eines weiblichen Best Agers handelt, in dem geprüft wird, ob für das Geschlecht weiblich und das Alter zwischen 48 und 100 Jahren vorgesagt wurde. Trifft das zu, wird die Anzahl der Personen in dieser Personengruppe um eins erhöht. Ist die Person kein weiblich Best Ager werden die anderen Personengruppen überprüft. Eine Person kann genau einer Personengruppe zugeordnet werden, da jede mögliche Kombination aus Geschlecht und Alter einer Personengruppe zugehörig ist. Wurde das Gesicht ihrer Personengruppe zugeordnet wird das nächste Gesicht betrachtet und mit dem CNN das Alter und Geschlecht bestimmt. Dann werden erneut die Regeln verwendet um die Personengruppe zu ermittelt. Nachdem alle Gesichter einer Personengruppe zugeordnet wurden, wird mit der Anzahl der Personen in den verschiedenen Personengruppen dem Bild eine Zielgruppe zugewiesen. Sind im Foto mindestens drei Frauen, keine Männer, keine Kinder oder Teenager zu sehen, dann ist auf dem Bild eine Frauengruppe dargestellt. So werden die verschiedenen Zielgruppen nacheinander überprüft. Falls das Bild keiner Personengruppe zugeordnet werden kann wird in jedem Fall die Anzahl an Personen in der jeweiligen Personengruppe angegeben. Ist beispielsweise nur eine ältere Frau auf dem Bild dann gehört dieses Bild keiner Zielgruppe an und es wird die Personengruppe mit Anzahl: *1 weiblicher Best Ager* ausgegeben. Die ausführliche Codeerweiterung für die Zielgruppen ist im Anhang nachzulesen. Der Ablauf der Zielgruppenzuordnung ist in Abbildung 3.1 schematisch dargestellt.

Die Gesichtserkennung ist ein wichtiger Forschungsbereich, vor allem seit des Booms der sozialen Medien Plattformen [75]. Es gibt einige öffentlich zugängliche Datensätze und Implementierungen zur Gesichtserkennung, sowie zur Bestimmung des Geschlechts und Alters von Personen auf Bildern und auch in Echtzeit über Kameras. Da der für die Masterarbeit zugrunde liegende Bilddatensatz der RTG sehr divers ist, weil oft mehrere Personen auf den Fotos sind, die meist nicht direkt in die Kamera sehen und Helme sowie Sonnenbrillen tragen, wurde ein vortrainiertes Netz ausgewählt, welches auf Realwertbildern von Smartphones trainiert wurde. Das verwendete Modell wurde von Tal Hassner und Gil Levi vortrainiert und die Implementierung[3] sowie der Datensatz[4] öffentlich gemacht. Mit der Implementierung kann das Geschlecht männlich/ weiblich erkannt werden sowie acht Altersgruppen von 0-2, 4-6, 8-12, 15-20, 25-32, 38-43, 48-53 und 60-100 Jahren. Zwischen den Intervallen der Altersgruppen wird kein Alter vorhergesagt. Ist seine Person 23 Jahre alt, liegt das zwischen den Intervallen 15-20 und 25-32. Da 23 näher an 25 Jahren als an 20 Jahren ist, müsste diese Person der Altersgruppe 25-32 Jahre korrekt zugeordnet werden. Das zum Training verwendete sogenannte Adience Datensatz besteht aus 26.580

[3] `https://github.com/smahesh29/Gender-and-Age-Detection`
[4] `https://www.kaggle.com/ttungl/adience-benchmark-gender-and-age-classification`

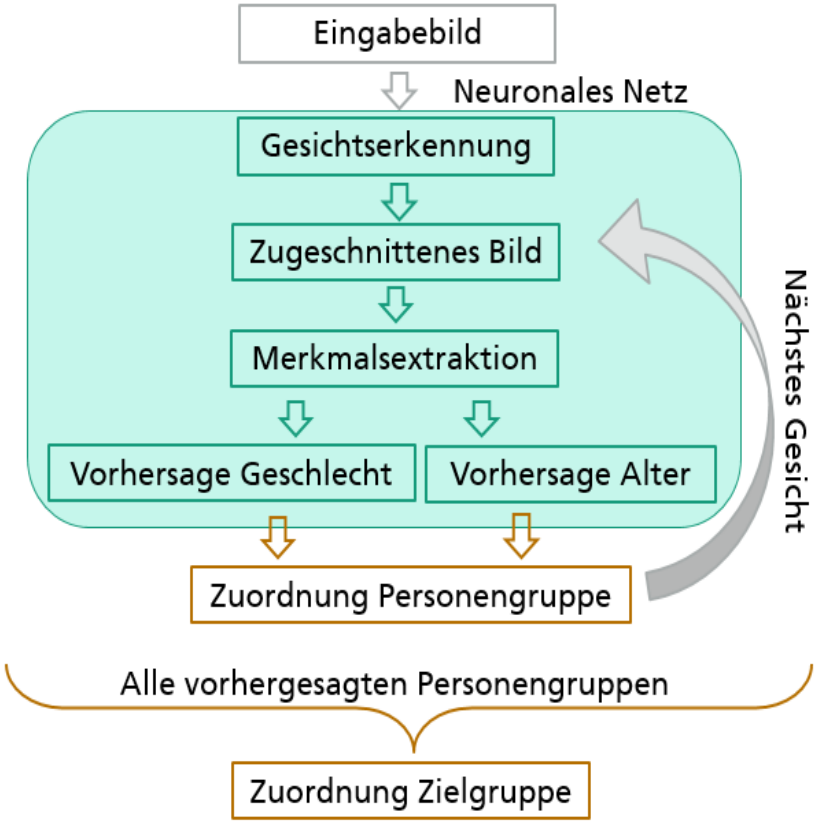


Abbildung 3.1: Schematischer Ablauf der Zielgruppenzuordnung mit Geschlechts- und Altersinformationen. Ein Bild wird in das neuronale Netz eingelesen. Die grün hinterlegten Schritte Gesichtserkennung, zugeschnittenes Bild, Merkmalsextraktion und daraus abgeleitete Geschlechts- und Altersvorhersage finden im neuronalen Netz statt. An die Geschlechts- und Altersvorhersage wird mit Regeln angeknüpft um die Personengruppe zu ermitteln. Die Bereiche in denen Regeln zum Tragen kommen sind in Braun dargestellt. Wurde die erste Personengruppe zugeordnet wird zurück ins Netz gesprungen um die Merkmale des nächsten Gesichtes zu erhalten. Damit wird das Geschlecht und Alter des zweiten Gesichtes vorhergesagt. Nach der Zuordnung der Personengruppe wiederholt sich der Prozess bis allen Gesichter eine Personengruppe zugeordnet wurden. Anschließend wird unter Berücksichtigung aller Personengruppen das Bild mit Regeln einer Zielgruppe zugeordnet.

Tabelle 3.3: Anzahl Bilder im Adience Datensatz nach Altersgruppen und Geschlecht. Die Tabelle wurde aus [75] entnommen.

	0-2	4-6	8-13	15-20	25-32	38-43	48-53	60-100	Gesamt
Männlich	745	928	934	734	2308	1294	392	442	8192
Weiblich	682	1234	1360	919	2589	1056	433	427	9411
Beide	1427	2162	2294	1653	4897	2350	825	869	19487

Fotos von 2.284 Personen. Der Bilddatensatz ist in der Forschungsgemeinschaft angesehen und gilt als herausfordernd [76].

In [77] wird beschrieben, wie die Bilddaten des Adience Datensatz gesammelt wurden. Dafür wurden iPhone Bilder, welche auf `flickr.com` hochgeladen und durch die Creative Commons Lizenz öffentlich gemacht wurden, verwendet. Diese wurden ohne manuelle Filter von Hobbyfotografen geschossen. 19.487 Bilder konnten mit Geschlecht und Alter gelabelt werden. Die Anzahl der Bilder über die verschiedenen Altersgruppen sowie das Geschlecht ist in Tabelle 3.3 zu sehen. Die Anzahl an Bilder der zwei Geschlechter ist ungefähr gleich verteilt. Gleichwohl ist zu beachten, dass wenige Bilder in den Altersklassen über 43 Jahren vorhanden sind. Durchschnittlich enthält jede Altersklasse 2.436 Bilder. Hingegen sind in den Klassen 48 bis 53 Jahren nur 825 Bilder und in der Altersklasse größer gleich 60 Jahren 869 Bilder enthalten. Dies liegt vermutlich daran, dass ältere Menschen seltener Smartphones nutzen und daher weniger Fotos aufnehmen. Die meisten Bilder sind für die Personengruppe zwischen 25 und 32 Jahren mit 4.897 Bilder verfügbar. Die ungleiche Verteilung der Bilder auf die Altersgruppen ist kritisch zu betrachten, da es durch die geringe Trainingsmenge schwieriger ist ältere Menschen richtig zu erkennen. Allerdings ist dies wichtig für Anwendungsbereiche in denen die Zielgruppe Best Ager relevant ist, wie es beispielsweise im Radtourismus der Fall ist.

Das bisher größte Problem bei der Altersbestimmung ist die Abwesenheit von Datensätzen [77]. Aus diesem Grund findet viel Forschung im Bereich Gesichtserkennung, aber wenig in der Alterserkennung statt. Für die Gesichtserkennung gibt es viele öffentliche Datensätze. Im Vergleich dazu gibt es für die Bestimmung des Alters und Geschlechts kaum Daten. Um ein großes, gelabeltes Bildset zur Geschlechts- und Altersschätzung aufzubauen sind entweder persönliche Informationen wie das Geburtsdatum und das Geschlecht, der im Bild dargestellten Personen nötig, welche oft privat sind oder es muss sehr zeitaufwendig manuell gelabelt werden. Aus diesem Grund gibt es wenige und nur relativ kleine Datensätze in diesem Bereich.

Kleine Trainingsdatensätze führen zur Überanpassung von (tiefen) neuronalen Netzes [78]. Dies ist ein größeres Problem bei der Alters- und Geschlechtserkennung als bei der reinen Gesichtserkennung, da nicht nur weniger Daten zum Training zu Verfügung sind, sondern es sich außerdem um ein Mehrklassenproblem handelt. Dadurch ist mehr Variabilität in den Beispielen nötig als bei binären Klassen. Das Problem der Überanpassung wird verstärkt durch die Verwendung von tiefen CNNs, da es hier eine Vielzahl an Modellparametern gibt [75]. Aus diesem Grund ist eine relativ simple Netzwerkarchitektur gewählt worden. Der grundlegende Aufbau und die Schichten eines CNNs wurden in Abschnitt 2.3 bereits ausführlich beschrieben.

Das verwendete vortrainierte Netz besteht aus nur drei Faltungsschichten und drei vollverknüpften Schichten mit einer kleinen Zahl an Neuronen [75]. Die überschaubare Netzarchitektur ist ausreichend, da lediglich acht Altersklassen und zwei Geschlechtertypen unterschieden werden. Die erste der drei Faltungsschichten besteht aus 96 Filtern der Größe $3 \times 7 \times 7$ Pixel und es wird die ReLU Aktivierungsfunktion verwendet. Gefolgt wird diese von einer Pooling Schicht, welche die Maximalwerte von 3×3 Regionen mit einer Schrittweite von zwei Pixeln ermittelt. Daraufhin findet eine Normalisierung der lokalen Reaktion statt [22]. Dies setzt eine Art laterale Hemmung, ein Verschaltungsprinzip der Nervenzellen, um, welches aus der Natur bekannt ist. Dabei hemmt eine aktive Nervenzelle die Aktivität der benachbarten Zellen. Dieses allgemeine neurophysiologische Prinzip wird auf künstliche neuronale Netz zu überführen, da es der Generalisierung hilft [22]. Vor allem bei der ReLU Aktivierungsfunktionen wird dies verwendet, da hier die Aktivierungen unbeschränkt groß werden können. Gleichmäßig große Aktivierungen in einer lokalen Nachbarschaft werden gedämpft, sodass wenn alle großen Werte verkleinert werden. Falls nur wenige Aktivierungswert in einer Nachbarschaft groß sind, findet eine Art Verkleinerung der anderen Neuronen statt. Dies steigert die Neuronen mit relativ größeren Aktivierungen, sodass sie aus der Nachbarschaft herausstechen.

Die zweite Faltungsschicht besteht aus 256 Filtern der Größe $96 \times 5 \times 5$ Pixel. Auf ihr folgt ebenfalls eine Poolingschicht und Normalisierung der lokalen Reaktion mit den gleichen Hyperparamtern. Die letzte Faltungsschicht verwendet 384 Filter der Größe $256 \times 3 \times 3$ Pixel, gefolgt von einer Poolingschicht und drei vollvernetzten Schichten. Auf die erste und zweite vollvernetzten Schicht wird Dropout Lernen angewendet (vergleiche Abschnitt 2.4). Die zweite vollvernetzte Schicht gibt die Klassen für das Alter und Geschlecht zurück. Ihre Ausgabe wird in die letzte vollverknüpfe Schicht, eine Soft-Max Schicht, gefüttert, welche die Wahrscheinlichkeit für jede Klasse zurückgibt. Die Klasse mit der höchsten Wahrscheinlichkeit für das Bild wird gewählt.

Zum Training wurde kein vortrainiertes Netz zur Initialisierung verwendet sondern alle Gewichte wurden zufällig initialisiert. Das verwendete Netz enthält zwei Dropout Schichten, in denen die Werte der Hälfte der Neuronen Null werden. Dies sorgt dafür, dass das Netz besser ausgeglichen ist. Zum Training des neuronalen Netzes wurde das Mini Batch Gradientenverfahren mit je 50 Fotos verwendet. Vergleiche Abschnitt 2.4, hier wird das Training ausführlich beschrieben. Nachdem das Training abgeschlossen ist werden die Gewichte genutzt um eine Vorhersage zu ermitteln.

Beim Einlesen der Bilder in das kNN werden alle drei Farbkanäle geladen. Zunächst werden die Bilder auf 256×256 Pixel skaliert und ein 227×227 Pixel Ausschnitt des Bildes ins Netz eingelesen. Die Verkleinerung der Pixelanzahl erfolgt aus Effizienzgründen, da jedes Bildpixel Eingabe für ein Neuron in der Eingabeschicht ist. Es wird auf verschiedene Art und Weise der 227×227 Pixel große Bildausschnitt ausgewählt. Einmal wird der Bildausschnitt zentriert um das Gesicht abgetrennt und zusätzlich vier mal ausgehend von allen Ecken des Bildes. Diese fünf Bildausschnitte werden gespielt, sodass 10 Ausschnitte eines Gesichtes verwendet werden. Der Datensatz wird so aufgebläht und dem Netz werden alle 10 Bildausschnitte gezeigt. Die schlussendliche Vorhersage wird aus der Durchschnittsvorhersage über alle Bildvariationen gebildet. Kleine Fehlausrichtungen im Datensatz können einen bemerkbaren Einfluss auf die Qualität der Vorsage haben. Die

Überabtastung der Bilder kompensiert diese Fehlausrichtung, indem verschiedene Versionen des gleichen Bildes ins Netz eingelesen werden [75].

Für die Implementierung wurde das öffentliche Caffe Framework [66] verwendet. Nach einer Trainingszeit von circa vier Stunden auf dem Adience Datensatz kann das Alter und Geschlecht eines Bild innerhalb von ungefähr 200ms vorhergesagt werden [75]. Dabei wurde eine durchschnittliche Genauigkeit ±Standardfehler von 86,8 ±1,4% bei der Erkennung des Geschlechts erreicht [75]. Das Alter wird etwas schwieriger erkannt. Hierbei wurde eine durchschnittliche Genauigkeit ±Standardfehler von 50,7 ±5,1% für die exakte Altersgruppe erlangt [75]. Werden nicht die genauen Altersgruppen benötigt, können auch Altersgruppen die nach oben oder unten um eine Klasse von der tatsächlichen Altersgruppe abweichen, als richtig erachtet werden. Dann wird eine durchschnittliche Genauigkeit ±Standardfehler von 84,7 ±2,2% erreicht [75]. Die meisten falschen Klassifizierungen im Adience Datensatz wurden durch Bewegungsunschärfe, Verdeckung (vorallem durch starkes Schminke) oder durch schlechte Auflösung verursacht [75]. Geschlechtermisklassifizierung fand vor allem bei kleinen Kindern und Babys statt, bei denen die offensichtlichen Geschlechterattribute noch nicht sichtbar sind [75]. Dies ist wenig kritisch zu sehen, da in vielen Anwendungen das Geschlecht von Babys und Kleinkindern nicht relevant ist. So ist es beispielsweise im Radtourismus bei einem Familienfoto nicht entscheidend, ob die Kinder weiblich oder männlich sind, nur die Tatsache, dass Kinder den Radurlaub begleiten ist entscheidend für die Zielgruppenzuordnung. Die Geschlechterkennung funktioniert im Allgemeinen auf den verwendeten Testdatensätzen sehr gut, die Alterserkennung ist deutlich schlechter, allerdings ist es ebenso für Menschen schwierig das Alter von Personen zu schätzen. Es gibt verschiedene Einflussfaktoren wie die ungleiche Alterung von Personen, Schönheitskorrekturen, Schminke und auch Sonneneinstrahlung sowie Rauchen, welche die Haut unterschiedlich alt aussehen lässt und die Altersbestimmung erschwert.

Vergleicht man die Erkennungsraten der Geschlechts- und Altersvorhersage mit dem aktuellen Stand der Literatur, muss angemerkt werden, dass noch höhere Vorhersagegenauigkeiten erreicht werden konnten. Das aktuell beste Ergebnis für die Geschlechts- und Alterserkennung wurde 2020 erreicht. [79] testete sein CNNs ebenfalls auf dem Advience Datensatz und erreiche für die Vorhersage des Geschlechts eine Genauigkeit von 96,2% (im Vergleich zu 86,8%) und bei der Altersklassifizierung eine Genauigkeit von 83% (verglichen mit 50,7%). Die höhere Vorhersagegenauigkeit des Netzes von [79] ist durch das mehrfache Training auf verschiedenen Datensätzen erklärbar. Das Netz wurde auf dem IMDb-WIKI [80] und MORPH-II Datensatz [81] vortrainiert und schließlich auf dem Advience Datensatz verfeinert. Das IMDb-WIKI Bilddatensatz enthält 460,723 Gesichtsfotos von 20,284 Berühmtheiten aus der Internet Movie Database und 62,328 von Wikipedia, also insgesamt 523,051 Fotos mit exaktem Alter und Geschlecht gelabelt [80]. Des Weiteren wurde das Netz mit 5.134 Polizeifotos aus der MORPH-II Datenbank trainiert, welches verschiedene Ethnien, allerdings weit mehr Männer als Frauen enthält [81]. Zu guter Letzt hat das CNN auf dem Advience Datensatz seinen Feinschliff bekommen. Dahingegen wurde das Netz von Levi und Hassner nur auf dem Advience Datensatz trainiert und hatte damit weitaus weniger Trainingsbilder zu Verfügung. Ein weiterer Unterschied der Netze ist, dass [79] vier Faltungsschichten im Vergleich zu [75] nur drei Faltungsschichten verwendet. Das Training der Netze ist ebenfalls nicht identisch. Für das Training der Altersvorhersage verwendet [79] Adam, ein Algorithmus zur gradientenbasierten Optimierung

stochastischer Zielfunktionen erster Ordnung, basierend auf adaptiven Schätzungen von Momenten niedrigerer Ordnung [82]. Die Methode berechnet individuelle adaptive Lernraten für verschiedene Parameter aus Schätzungen des ersten und zweiten Moments der Gradienten. Im Gegensatz dazu wird beim stochastische Gradientenverfahren von [75] verwendet, eine einzige Lernrate für alle Gewichtsaktualisierungen beibehalten. Da der selbe Datensatz zur Evaluation verwendet wurde sind beide Netze gut direkt vergleichbar, dennoch ist nicht genau zu sagen welchen Einfluss die Implementierungsunterschiede auf die Vorhersagegenauigkeit haben. Ob die Netzarchitektur mit einer weiteren Faltungsschicht, das neuere Trainingsverfahren oder das größere Trainingsset den größten Gewinn für die Vorhersagegenauigkeit habe, lässt sich an dieser Stelle nicht sagen. Leider ist für das Netz von [79] keine Implementierung oder Gewichte öffentlich verfügbar. Aus Zeitgründen wurde das vortrainierte Netz von [75] verwendet, da es viel Zeit und Rechenleistung erfordern würde das in [79] beschriebene CNN nachzubauen und zu trainieren.

3.2.2 Datensatz und Evaluierung der Gesichtserkennung

Die Evaluation der Personenklassen- und Zielgruppenerkennung des Radtourismus mit der zuvor beschrieben Gesichtserkennung wird auf dem Datensatz RTG durchgeführt. Dieses Bilddatensatz besteht wie bereits in Abschnitt 1.2 beschrieben ohne Duplikate aus circa 1.000 Bildern. Im Datensatz sind Fotos enthalten mit Radfahrern, mit Personen ohne Fahrrad und auch von Landschaftsaufnahmen ohne Menschen. Um zu testen ob Gesichter erkannt wurden, obwohl keine Person auf dem Bild zu sehen ist, enthält der Datensatz 24 Bilder, auf denen keine Menschen zu sehen sind. Zur Überprüfung der Zielgruppen können Bilder verwendet werden, welche von den Experten der RTG bereits in entsprechende Order gespeichert wurden, die auf die Zielgruppen verweisen. Es ist zu beachten, dass nicht alle Bilder im Ordner *Familie* auch formal der Zielgruppe Familie zugeordnet gehören. Teilweise sind auf diesen Bildern nur Mutter und Vater oder andere Teile der Familie abgebildet. Diese Bilder sind ggf. anderen Zielgruppen zuzuordnen. Deshalb wurden die Bilder der Zielgruppenfotoshootings manuell aussortiert. Zur Evaluation können ebenfalls nur die Bilder verwendet werden auf denen das Gesicht der Personen zu erkennen ist, da sonst die Identifizierung des Geschlechts und Alters beruhend auf den Gesichtsmerkmalen nicht möglich ist. So wurden für die Zielgruppe Best Ager aus verschiedenen Fotoshootings 35 Bilder zur Evaluation ausgesucht. Von der Zielgruppe junges Paar konnten 41 Bilder verwendet werden, aus zwei Familienfotoshootings nur 11 Bilder, acht Bilder von Frauengruppen und 23 Bilder von gemischtgeschlechtlichen Gruppen. Die Anzahl an Bildern pro Zielgruppe ist im Verhältnis zur Gesamtzahl an Bildern relativ gering, da nur wenige Bilder von den Experten zu ihren Zielgruppen zugeordnet wurden. Ein weiteres Problem ist, dass die Models teilweise von hinten oder seitlich fotografiert wurden und somit das Gesicht nicht ausreichend erkennbar ist. Außerdem sind auf einige Aufnahmen die Personen auf dem Bild zu weit weg, sodass das Gesicht nicht genau genug erkenntlich ist.

Der eben beschriebene Datensatz wurde mit der Implementierung aus Abschnitt 3.2 getestet, welche mit entsprechenden Regeln zur Einordnung der Personenklassen und Zielgruppen erweitert wurde. Die Ergebnisse der Zielgruppenerkennung sind in Tabelle 3.4

Tabelle 3.4: Insgesamt wurden 118 Fotos mit Personen evaluiert, auf diesen sind 322 Gesichter zu sehen. Die Spalten *Gesicht erkannt* und *Personenklasse erkannt* beziehen sich auf alle Personen auf den Bildern, die Spalte *Zielgruppe erkannt* bezieht sich auf die einzelnen Bilder. Für die Zielgruppen standen unterschiedlich viele Bilder zur Evaluation zu Verfügung. In der Zielgruppe junges Paar wurden beispielsweise 41 Bilder mit je 2 Personen, also 82 junge Leute, evaluiert.

Zielgruppe	Gesicht erkannt	Personenklasse erkannt	Zielgruppe erkannt
Junges Paar	19/82	12/82	1/41
Best Ager Paar	3/70	0/70	0/35
Familie	3/44	1/44	0/11
Frauengruppe	0/33	0/33	0/8
gemischtegeschlichtliche Gruppe	4/93	4/93	0/23
Summe	29/322	18/322	1/118

dargestellt.

Der Evaluationsdatensatz enthält 322 Personen auf 118 Bilder. Für erkannte Gesichter kann das Geschlecht und Alter vorhergesagte werden und damit eine Zuordnung zu einer Personengruppe stattfinden. Anschließend kann mit allen Personengruppen auf einem Bilder eine Zielgruppenzuordnung erfolgen. Der Tabelle 3.4 ist bei der Summe erkannter Gesichter zu entnehmen, dass bereits die Erkennung des Gesichts bei lediglich 29 Personen von 322 möglich war. Wird das Gesicht nicht erkannt, so ist eine Zuordnung der Personenklasse und Zielgruppe unmöglich. Besonders auffällig ist, dass Gesichter jüngerer Personen am besten erkannt wurden und dagegen Best Ager sehr schlecht. Dies könnte daran liegen, dass im Trainingsdatensatz weitaus mehr Bilder von jungen Menschen enthalten sind, als von Best Agern. Personen auf Gruppenbildern wurden ebenfalls kaum erkannt (nur 4 aus 93). Hier könnte es problematisch sein, dass die Gesichter nicht nah genug zu erkennen waren, da die Aufnahmen von einer weiteren Entfernung aufgenommen wurden, um alle Gruppenmitglieder in einem Bild einzufangen. Zudem werden Gesichter die nahe nebeneinander sind als ein Gesicht erkannt. In der Zeile Junges Paar stellen wir fest, dass 12 der 19 erkannten Gesichter richtig als junge Leute vorhergesagt wurden. Dies ist ein relativ gutes Ergebnis. Nichtsdestotrotz wurden wenige Gesichter erkannt und somit war nur in wenigen Fällen eine Zuordnung zu Personengruppen überhaupt möglich. Die Erkennungsgenauigkeit aller Personenklasse liegt bei circa 6% (18/322). Die Zielgruppe wurde für ein einzigen Bild aus 118 Fotos richtig erkannt. Dies ist ein sehr unbefriedigendes Ergebnis. Die größten Probleme weshalb die Gesichter nicht erkannt werden ist vermutlich, dass die Aufnahmen der Gesichter nicht nah genug sind sowie Sonnenbrillen und Fahrradhelme zu viel vom Gesicht verdecken. Außerdem sollen die Bildsituationen möglichst natürlich wirken, weshalb die Models oft nicht direkt in die Kamera sehen, wodurch das Gesicht meist nur seitlich zu erkennen ist. Weiterhin könnte es durch die Translationsinvarianz schwierig sein wenn Gesichter zu nah nebeneinander sind. Dies ist gerade in Gruppenbildern problematisch. Obwohl bereits versucht wurde für eine Gesichtserkennung geeignete

Tabelle 3.5: Konfusionsmatrix der Gesichtserkennung

		Tatsächlich		
		Gesicht da	Gesicht nicht da	Summe
Vorhersage	Gesicht da	29	0	29
	Gesicht nicht da	293	24	317
	Summe	322	24	346

Bilder aus dem Datensatz der RTG herauszufiltern waren kaum nutzbare Bilder für eine Gesichtserkennung, die Nahaufnahmen der Gesichter benötigt, vorhanden.

Tabelle 3.5 zeigt die Konfusionsmatrix der Gesichtserkennung. Der Aufbau einer Konfusionsmatrix und die resultierende Kennzahlen wurde in Abschnitt 2.5 bereits erläutert. Der Tabelle ist zu entnehmen, dass sehr viele Gesichter nicht erkannt wurden. Dies konnte ebenfalls bereits mit Tabelle 3.4 festgestellt werden. Aus der Konfusionsmatrix können wir zusätzlich entnehmen, dass in keinem Fall ein Gesicht erkannt wurde wenn keines zu sehen war. Aus diesem Grund liefert der Anteil der richtig positiv klassifizierten Objekte an der Gesamtheit aller positiven Beispiele ein perfektes Ergebnis:

$$\text{Genauigkeit} = \frac{29}{29+0} = 1$$

Sehr klein ist dagegen der Anteil der korrekt positiv eingestuften Objekte, bezogen auf die Menge aller tatsächlich positiven Klassen. Dies liegt daran, dass 293 der 322 Gesicht nicht erkannt wurden.

$$\text{Trefferquote} = \frac{29}{29+293} = 0,09$$

Durch den geringen Wert der Genauigkeit liefert der F1-Kennwert der Gesichtserkennung ein schlechtes Ergebnis nahe an 0.

$$\text{F1-Kennwert} = \frac{2 \cdot 0,09 \cdot 1}{0,09+1} = 0,17$$

Anhand des F1-Wertes können wir aussagen, dass die Erkennung der Gesichter nicht praxistauglich für den Radtourismus ist. Dennoch wollen wir genauer betrachten, wie oft das Geschlecht bei den 29 erkannten Gesichtern richtig vorhergesagt wurde. Da in sechs Fällen das Gesicht von Kinder oder Jugendlicher erkannt wurde (bei denen kein Geschlecht unterschieden wird), werden bei der Geschlechtsklassifizierung insgesamt 23 Bilder betrachtet. Das Ergebnis der Konfusionsmatrix ist in Tabelle 3.6 abgebildet. Es wird kein Mann als Frau erkannt und lediglich zwei Frauen als Männer, ansonsten werden Frauen und Männer richtig klassifiziert. Die Genauigkeit hat somit den bestmöglichen Wert von 1 und die Trefferquote erreicht 0,78.

$$\text{Genauigkeit} = \frac{7}{7+0} = 1$$

$$\text{Trefferquote} = \frac{7}{7+2} = 0,78$$

Tabelle 3.6: Konfusionsmatrix des Geschlecht von der Implementierung zur Gesichtserkennung

		Tatsächlich		
		weiblich	männlich	Summe
Vorhersage	weiblich	7	0	7
	männlich	2	14	16
	Summe	9	14	23

Werden beide Kennzahlen zum F1-Kennwert verrechnet erhalten wir einen guten Wert von 0,88.

$$\text{F1-Kennwert} = \frac{2 \cdot 0,78 \cdot 1}{0,78 + 1} = 0,88$$

Damit ist die Vorhersage des Geschlechts insgesamt gut. Allerdings ist dieser Wert in Bezug zur vorherigen Gesichtserkennung zu setzen und bewerten. Es muss berücksichtigt werden, dass nur sehr wenige Gesichter erkannt wurden und deshalb die Testmenge für die Geschlechtserkennung klein ist.

Betrachten wir abschließend noch die Konfusionsmatrix der verschiedenen Personengruppen. Die Unterscheidung in die Personengruppen Mann und Frau wurde dabei nicht betrachtet, da sich diese mit den Personengruppen junge Frau, junger Mann und weiblicher, männlicher Best Ager überschneiden sowie die Geschlechtserkennung bereits untersucht wurde.

Tabelle 3.7: Konfusionsmatrix der Personengruppen von der Implementierung zur Gesichtserkennung

Vorhergesagte Personengruppe \ Tatsächliche Personengruppe	junge Frau	junger Mann	w. Best Ager	m. Best Ager	Jugendlicher	Kind	Summe
junge Frau	6	0	0	0	0	0	6
junger Mann	0	6	0	2	0	0	8
w. Best Ager	0	0	0	0	0	0	0
m Best Ager	0	0	0	0	0	0	0
Jugendlicher	0	4	0	0	0	0	4
Kind	0	2	0	0	0	0	2
Summe	6	12	0	2	0	0	20

Der Matrix in Tabelle 3.7 ist zu entnehmen, dass Männer oft zu jung eingeschätzt werden. Junge Männer wurden vier mal als Jugendliche und sogar zwei mal als Kinder eingestuft. Best Ager Männer wurden zwei mal als junge Männer erkannt. Somit wurden männliche Personen eher zu jung eingeschätzt. Best Ager Frauen, Kinder und Jugendliche wurden kein einziges mal richtig erkannt, da auf der Diagonalen der Konfusionsmatrix kein positiver Wert zu finden ist. Junge Frauen wurden hingegen immer richtig erkannt (falls das Gesicht erkannt wurde).

Um ein besseres Verständnis für den Datensatz zur Gesichtserkennung zu bekommen, wurden zwei Beispielbilder in Abbildung 3.2 und Abbildung 3.3 angefügt, auf denen die Gesichter nicht erkannt wurden. Es ist zu bemerken, dass die Erkennung des Geschlechts und Alters anhand der Gesichter in dieser Genauigkeit nicht für den Radtourismus geeignet ist. Für andere Branchen, in denen Nahaufnahmen von Gesichtern vorhanden sind, ist zu erwarten das diese Implementierung gute Ergebnisse liefert, da auf dem Adience Datensatz hohe Vorhersagegenauigkeiten erreicht wurden (vergleiche Abschnitt 3.2). Sobald ein Gesicht mit Alter und Geschlecht richtig erkannt wird, kann dies durch die zusätzlich festgelegten Regeln zu 100% der richtigen Personengruppen zugeordnet werden. Aus den korrekten Personenklassen kann dann mit 100%iger Sicherheit eine richtige Zuordnung zur Zielgruppe erfolgen. Allerdings sind diese Einordnungen zu Personenklassen und Zielgruppen an die Erkennung der Gesichter und richtige Klassifizierung des Geschlechts und Alters gebunden, welche in den wenigsten Bildern des Datensatzes richtig funktioniert hat.

Da diese Implementierung im Anwendungsfall nicht genügend ist, muss nach einer anderen Möglichkeit gesucht werden, um mit einer höheren Genauigkeit Personen in Bildern ihren Zielgruppen zuzuordnen. Eine Möglichkeit ist es lediglich Personen und keine Gesichter zu erkennen. Dies ist weitaus einfacher als Gesichter mit Geschlecht und Alter zu klassifizieren. Damit ist detaillierte Zuordnung zu Zielgruppen nicht möglich, da keine Aussage über die Personenklassen getroffen werden kann. Dennoch können mit der Anzahl der Personen auf einem Bild Rückschlüsse daraus gezogen werden, ob eine Reisegruppe auf dem Bild zu sehen ist. Eine genaue Klassifizierung ob eine Familie, Frauen- oder eine gemischtgeschlechtliche Gruppe auf den Fotos dargestellt ist kann damit nicht vorgenommen werden. Nichtsdestotrotz wird im nächsten Abschnitt eine Erkennung der Personen auf den Bildern des RTG vorgenommen um die Personenanzahl auf den Bildern zu erhalten.

3.3 Erkennung von Personen

Da die Erkennung der in Abschnitt 3.1 ermittelten exakten Zielgruppen basierend auf einer Alters- und Geschlechterkennung ungenügende Ergebnisse geliefert hat wird im Folgenden die Personenanzahl auf Bildern betrachtet. Dafür wird ein vortrainiertes CNN vorgestellt mit dem Personen erkannt werden können. Dieses wurde am Datensatz des RTG getestet und ausgewertet.

Abbildung 3.2: Beispielbild aus dem Datensatz der RTG von der Zielgruppe Best Ager Paar

Abbildung 3.3: Weiteres Beispielbild aus dem Datensatz der RTG von der Zielgruppe Best Ager Paar

3.3.1 Beschreibung des vortrainiertes Netzes zur Personenerkennung

Für die Erkennung von Personen wird ebenfalls, wie bei der Vorhersage des Alters und Geschlechts, auf ein vortrainiertes Netz zurückgegriffen. Es wird die MobileNet Netzwerkarchitektur [83] verwendet. Das Netz wurde MobileNet benannt, da es sehr ressourceneffizient arbeitet und deshalb auf Mobilgeräten wie Smartphones ausführbar ist. Die Besonderheit von MobileNets ist die Verwendung von schrittweise getrennter Faltung. Dabei wird die Faltung in zwei Schritte unterteilt um die Anzahl an Netzparametern zu reduzieren. Eine gewöhnliche Faltung wird aufgesplittet in eine sogenannte tiefe Faltung und eine punktweise Faltung. Die tiefe Faltung wendet einen Filter auf jeden Kanal an. Die punktweise Faltung wendet eine 1×1 Faltung an um die Ausgaben der tiefen Faltung zu kombinieren. Mit der standardmäßigen Faltung wird auf einmal gefiltert und die Eingaben kombiniert. Bei der schrittweise getrennter Faltung findet dies in zwei separaten Schichten statt. Durch diese Auftrennung werden die Netzgröße und der Rechenaufwand stark minimiert. Für eine schnelle und effiziente Objekterkennung wird das MobileNet mit einem SSD, vergleiche Abschnitt 2.6, kombiniert. Das Netz wurde zuerst auf COCO [50] trainiert und dann auf den VOC 2007 [54] abgestimmt.

Der COCO Datensatz enthält, zur Detektion und Segmentierung von Objekten des täglichen Lebens in ihrer natürlichen Umgebung, komplexe Bilder um Szenen zu verstehen [50]. Im Bilddatensatz sind 91 Objektklassen enthalten, die ein 4-jähriger leicht erkennen könnte. Es stehen 2,5 Millionen gelabelte Instanzen in 328.000 Bildern zur Verfügung. Im Vergleich mit ImageNet [55] hat COCO weniger Kategorien aber mehr Instanzen pro Kategorie. Ebenso hat COCO viel mehr Instanzen pro Kategorie als der VOC 2007 [54] und SUN [52]. COCO ist ein großer Datensatz mit Kontextinfos und nicht-ikonischen Ansichten von Objekten. Die Bilder stammen von `flickr.com` und wurden von Amateurfotografen aufgenommen und mit Metadaten und Schlagwörtern versehen. Es wurde nicht isoliert sondern paarweise nach Objektkategorien gesucht. Fragt man nur nach einer Kategorie wie z.B. Fahrrad dann bekommt man in der Regel ein Bild auf dem isoliert, zentriert, aus ikonischem Blickwinkel ein Fahrrad zu sehen ist. Wird nach Fahrrad + Mensch gesucht, werden komplexe Bilder auf denen oft noch weiter Objektkategorien zu sehen sind erhalten. Dadurch ist COCO nicht nur ein großer sondern auch variantenreicher Bilddatensatz.

Die VOC Datensätze bilden guten Vergleichsmaßstäbe für die Objekterkennung mit jährlich erweiterten Datensatz und Wettbewerben bis 2012. Sie beinhalten fünf verschiedene Aufgaben: Klassifizierung, Detektion, Segmentierung, Aktivitäten Klassifizierung, Personenlayout [84]. VOC 2007 besteht ebenfalls wie COCO aus Fotos der Webseite Flickr. Dabei beinhalten die Bilder verschiedene Blickwinkel, Posen und Lichtverhältnisse. Außerdem steht nicht das Objekt im Fokus des Bildes sondern Szenen. Dies wurde erreicht indem die Suchanfragen auf Flickr nicht nur nach bestimmtem Objektklasse wie z.B. Boot, sondern nach Synonymen und Szenen, in der die Klasse vorkommt wie Schiff, Fähre, Kanu, Bootfahren, Passagierschiff, Wasserfahrzeug, Regatta, Rennen, Marine, Wasser, Kanal, Fluss, Strömung, See, Yacht, segeln und rudern, gesucht wurden. Dadurch ist der Datensatz sehr vielseitig. Für die Objekterkennung besteht das VOC 2007 aus 9.963 Bildern von 20 Objektklassen mit 24.640 kommentierten Objekten [85].

Tabelle 3.8: Ergebnisse Personenerkennung

Zielgruppe	Erkennungsgenauigkeit
Junges Paar	58/66
Best Ager Paar	62/66
Gruppe	121/144
Summe	241/272

Nach dem Training auf den beschriebenen Datensätzen wurde eine mittlere Vorhersagegenauigkeit von 72,7% über alle 20 Objekte erreicht, die mit dem vortrainiertem Netz erkannt werden können. Es können Personen, Fahrräder, Motorräder, Züge, Flugzeuge, Vögel, Boote, Flaschen, Autos, Busse, Esstische, Stühle, Sofas, Katzen, Kühe, Hunde, Pferde, Schafe, Topfpflanzen und Fernseher erkannt werden. Die Größe der Eingabebilder wird auf 300×300 Pixel angepasst. Über eine Schleife im Code wird über alle 20 möglichen Klassen iteriert und geprüft ob das gesuchte Objekt im Bild zu sehen ist. Der Code kann angepasst werden, sodass nur eine bestimmte oder mehrere ausgewählte Objektklassen gesucht werden. Um schwache Erkennungsraten herauszufiltern, wurde ein minimaler Wert der Erkennungssicherheit von 20% festgelegt. Ab diesem Wert wird ein erkanntes Objekt ausgegeben. Wird der Schwellwert überschritten, wird das Klassenlabel sowie der Wert der Erkennungssicherheit zurückgegeben. Außerdem wird das Bildobjekt mit einer farbigen Box umrandet an der Position auf der es sich im Foto befindet. Eine detaillierte Codebeschreibung ist auf der Internetseite `https://www.pyimagesearch.com/2017/09/11/object-detection-with-deep-learning-and-opencv/` zu finden.

3.3.2 Datensatz und Evaluierung der Personenerkennung

Aus dem zuvor beschrieben Datensatz der RTG wurden 124 Bilder ausgewählt. Davon sind 100 Bilder mit Personen und 24 Bilder ohne Menschen auf den Bildern. Von den 100 Fotos mit Personen sind 34 Gruppenbilder (Frauengruppen, gemischtgeschlechtliche Gruppen oder Familien), 33 Bilder von jungen Paaren und 33 Bilder von Best Ager Paaren. Es ist zu beachten, dass allein mit der Personenerkennung keine Aussage darüber getroffen werden kann ob eine Person männlich oder weiblich, jung oder alt ist. Wir wollen im Folgenden unterscheiden ob auf Bildern ein Paar oder eine Gruppe zu sehen ist. Die getrennte Betrachtung von Best Agern und jungen Paaren bei der Personenerkennung ist dem geschuldet, dass es relativ starke Erkennungunterschiede junger und älterer Personen bei der Gesichtserkennung gab. Deshalb soll überprüft werden ob es derartige Ungleichheit ebenfalls mit diesem Netz gibt. Das CNN für die Personenerkennung ist nicht dazu in der Lage eine Zuordnung zur Klasse junges Paar oder Best Ager Paar vorzunehmen. Wird vom Netz die Personenzahl zwei zurückgegeben wird manuell überprüft ob es sich um ein junges oder Best Ager Paar handelt.

Der Tabelle 3.8 ist zu entnehmen das Best Ager nicht schwieriger, sondern ein bisschen besser erkannt werden als junge Leute. Dies deutet darauf hin, dass der verwendete Trainingsdatensatz über alle Altersgruppen gut ausgeglichen ist. Da die Erkennungsrate von

Tabelle 3.9: Konfusionsmatrix Personenerkennung

		Tatsächlich		
		Person da	keine Person da	Summe
Vorhersage	Person da	241	0	241
	keine Person da	31	24	55
	Summe	272	24	296

Gruppen ähnlich hoch ist, wie die von Paaren lässt sich ebenfalls schließen, dass Personen die auf Bildern in größerer Entfernung zur Kamera stehen, wie es bei Gruppenbildern üblich ist, trotzdem gut erkannt werden. Dies war mit der Implementierung der Gesichtserkennunng aus Abschnitt 3.2 nicht gegeben. Über alle Bilder im Evaluationsdatensatz wurde eine Genauigkeit der Personenerkennung von 89% erreicht. Dies ist ein sehr gutes Ergebnis. Nach [86] konnte 2019 die höchste Genauigkeit von 91% auf dem VOC 2012 [87] für die Erkennung von Personen erreicht werden. Dabei wurde ein R-FCN [44] auf dem COCO Datensatz vortraininert und dann auf VOC 2007 und 2012 verfeinert. Mit den erreichten 89% auf dem Praxisdatensatz ist dies sehr nahe an dem aktuellen Stand der Forschung.

Um die Ergebnisse der Personenerkennung noch weiter zu analysieren wurde eine Konfusionsmatrix erstellt. In der Tabelle 3.9 ist zu sehen, dass nie eine Person erkannt wurde wenn niemand auf dem Bild zu sehen war. Allerdings wurden 31 Personen nicht erkannt, obwohl sie auf Bildern dargestellt waren. Mit diesen Werten lassen sich die folgenden Kennzahlen ermitteln:

$$\text{Genauigkeit} = \frac{241}{241 + 0} = 1$$

$$\text{Trefferquote} = \frac{241}{241 + 31} = 0,89$$

$$\text{F1-Kennwert} = \frac{2 \cdot 0,89 \cdot 1}{0,89 + 1} = 0,94$$

Wir erreichen eine bestmögliche Genauigkeit von 1,0 und eine Trefferquote von 0,89, da einige Personen nicht erkannt wurden. Insgesamt ergibt sich ein F1-Kennwert von 0,94, der sehr nahe am Maximalwert 1 und damit ein sehr gutes Ergebnis ist.

Da Personen teilweise nicht erkannt wurden, wurde näher betrachtet auf welchen Bildern dies der Fall war. In einigen Fällen wurden Personen, die nahe auf den Bildern nebeneinander standen als eine Person erkannt. Ein anderes Problem ist, das verdeckte Personen nicht erkannt wurden. Alles in allem ist das Ergebnis der Personenerkennung erheblich vielversprechender als das Ergebnis der exakten Zielgruppenzuordnung durch die Gesichtserkennung.

Mit dieser Implementierung ist eine Einordnung möglich wie viele Personen auf einem Bild fotografiert sind. Damit kann ausgesagt werden, ob eine Personengruppe auf dem Bild zu sehen ist oder nicht. Sind drei oder mehr Personen auf einem Bild dargestellt so wird von einem Gruppenfoto gesprochen. Wenn weniger als drei Personen auf einem Foto sind

handelt es sich nicht um ein Gruppenbild. Es ist somit eine binäre Entscheidung möglich: Gruppe ja/nein. Eine Gruppe wurde mit dieser Implementierung mit einer Genauigkeit von 33/34 richtig erkannt. Das keine Gruppe auf den Bildern dargestellt ist wurde zu 100% richtig vorhergesagt. Genauere Aussagen um welche Art von Gruppe es sich handelt, sind damit nicht möglich. Es kann nicht klassifiziert werden, ob eine Familie, Frauengruppe oder gemischte Gruppe auf dem Bild dargestellt sind. Dennoch kann mit einer hohen Genauigkeit festgestellt werden, ob im Bild Personen sind und wie viele. Es könnte auch analysiert werden ob sich ein Paar auf einem Foto befindet oder nicht, indem überprüft wird ob genau zwei Personen im Foto abgebildet sind. Der Begriff eines Paares müsste dann allerdings weiter gefasst werden und nicht nur als Liebespaar verstanden werden. Es würde dann eine Mutter mit Kind zum Beispiel auch als Paar erkannt werden.

In einigen Anwendungsbereichen ist die Personenanzahl bei Fotos als Information ausreichend. Soll beispielsweise überwacht werden wie viele Personen sich in einem Ort befinden ist diese Implementierung sehr gut geeignet. Für den Anwendungsfall im Radtourismus ist die Erkennung von Personen eine gute Ausgangslage. Allerdings sind genauere Informationen über die Personen wichtig für die Zuordnung in die konkreten Zielgruppen. Deshalb wurde nach weiteren Möglichkeiten gesucht mehr Informationen über die Personen aus Bildern herauszulesen.

3.4 Erkennung des Geschlechts anhand des gesamten Körpers

Im Folgenden wird eine Implementierung betrachtet, mit der das Geschlecht anhand eines Fotos bestimmt werden kann, wenn der gesamte Körper der Person zu sehen ist. Dafür wird ein 3D-Abbild des Körpers erstellt. Zunächst wird das Verfahren beschrieben, wie ein solches Körpermodell erstellt und das Geschlecht vorhergesagt werden kann. Anschließend wird das Modell mit den Praxisdaten evaluiert.

3.4.1 Beschreibung der Methodik zur Erstellung eines geschlechtsspezifischen 3D-Körpermodells

Mit einem 3D-Modell des Körpers soll das Geschlecht von Personen vorhergesagt werden. Dafür tauchen wir in den Bereich der geometrischen Modellierung ein, in der Methoden und Algorithmen zur mathematischen Beschreibung von Formen behandelt werden. In Abbildung 3.4 sind die einzelnen Schritte dargestellt, die benötigt werden, um aus einem einzelnen Foto einer Person ein 3D-Körpermodell zu erzeugen. Zunächst kann mit OpenPose [88] aus einem Bild die Position der wichtigsten Körperpunkte ausgelesen werden. Diese werden dann zu einem Skelett verknüpft. Anschließend kann mit dem Körpermodell SMPL [89] ein 3D-Gitter der Körperoberfläche erzeugt werden. Um das Gesicht und die Hände detailliert nachzubilden, wird das SMPL Modell erweitert zu SMPL-X [90], mit dem eine realistische Abbildung von Gesicht und Händen möglich ist.

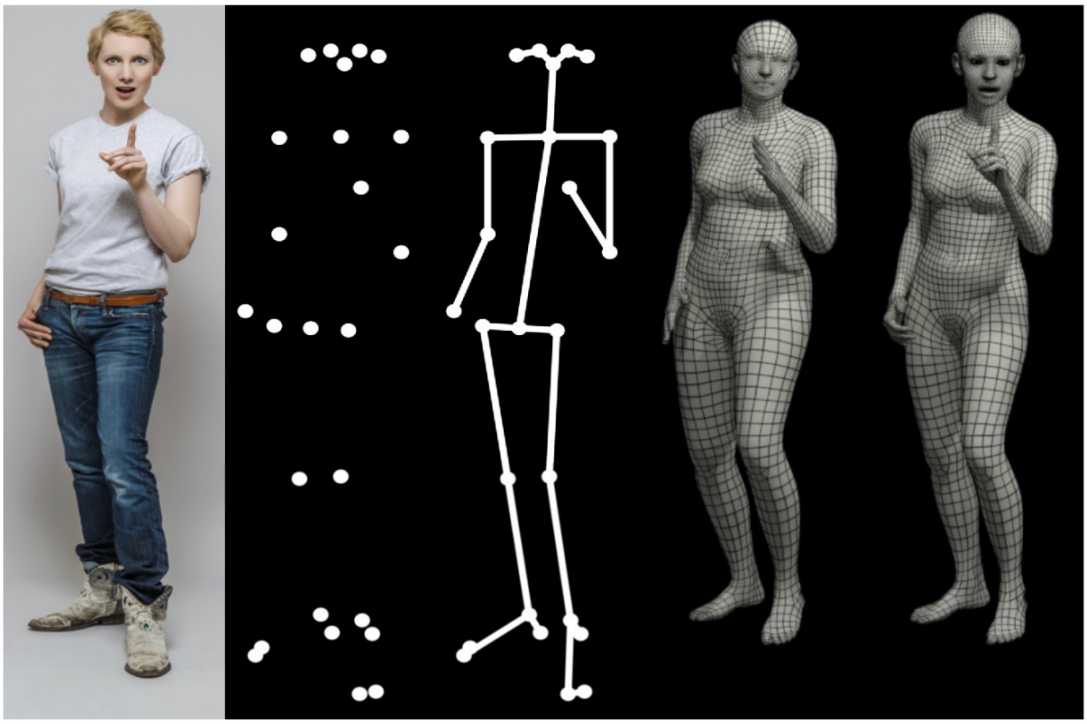


Abbildung 3.4: In der Abbildung ist von links nach rechts zu sehen: ein RGB Bild einer Frau, wichtigste Gelenke des Körpers als Punkte dargestellt (Körperpunkte ermittelt durch OpenPose), die Körperpunkte verbunden zu einem Skelett, das SMPL Körpermodell und das SMPL-X Modell der Frau mit detailliertem Gesicht und Händen. Die Abbildung wurde entnommen aus [90].

Mit einem 3D-Körpermodell von Personen können neben dem Geschlecht auch andere Merkmale wie soziale Hinweise, die kommuniziert werden, analysiert werden. Außerdem können Interaktionen mit der Umwelt für ein ganzheitliches Szenenverständnis dienen [90] und die Körperhaltung untersucht werden. Dafür ist es besonders wichtig, das Gesicht und die Hände detailliert mit zu modellieren, da der Gesichtsausdruck und die Handhaltung viele Rückschlüsse ermöglichen. Besonders für die Medizin ist die Körperhaltung und Körperpunkte interessant, da damit Krankheiten wie Skoliose, eine seitliche Verbiegung der Wirbelsäule, erkannt werden können [91]. Genauso könnten die motorischen Bewegungsfähigkeiten von Kindern oder Ähnliches analysiert werden. Forschung in der 3D-Modellierung entsteht auch im Bereich der Film- und Spieleentwicklung [92, 93, 94]. Hier werden verschiedene Wesen erschaffen und mit deformierbaren Gittern erzeugt.

Die bisherige Literatur in Bereich 3D-Körpermodellierung hat sich auf die Körperpose ohne Gesicht und Hände fokussiert [90]. Diese Methoden setzen voraus, dass sich die Hand entweder in einer Faust oder in einer offenen Pose befindet und das Gesicht einen neutralen Ausdruck hat. Daneben gibt es Modelle, die lediglich das Gesicht oder die Hände isoliert vom restlichen Körper betrachten [95, 96, 97, 98, 99]. Erst 2019 wurde in [90] ein realistisches Modell entwickelt, das die 3D-Oberfläche eines Körpers mit Händen und Gesicht aus einem einzigen Bild herstellen kann. Dies war zuvor nur mit mehreren Bildern aus verschiedenen Blickwinkeln möglich. Das Modell *Frank* [100] modelliert Gesicht,

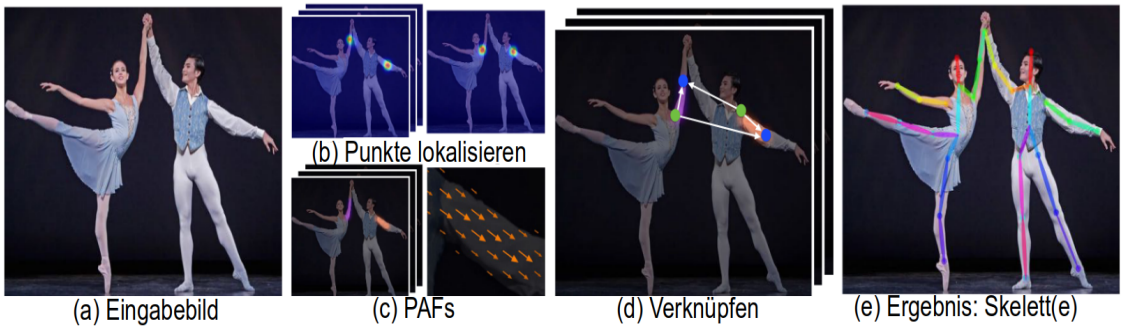


Abbildung 3.5: Darstellung des Ablaufes von OpenPose: (a) gesamtes Bild als Eingabe in ein CNN (b) Lokalisierung der Körperpunkte (c) PAFs um die Orientierung der Punkte zu erhalten (d) Verknüpfung der zusammengehörigen Körperpunkte (e) Skelett(e) der Ganzkörperpose aller Personen als Ergebnis.

Hände und Körper zusammen. Allerdings werden die individuellen Teile nur aneinander geheftet, was zu unrealistischen Ergebnissen führt.

Ein anderer Vorteil von [90] gegenüber der Literatur ist, dass die Methode auf Skinning basiert und deshalb kompatibel mit Grafiksoftware ist. Beim Skinning wird eine grafische Haut auf ein zugrundeliegendes Skelett gelegt, welche sich deformiert, wenn sich das Skelett verändert [101]. Skelettbasierte Skinning-Techniken können in zwei verschiedene Arten unterteilt werden: geometriebasierte Methoden und beispielbasierte Methoden [102]. Geometriebasierte Methoden [103] nutzen die Standardverformung des Skelettunterraums. Ein bekannter Algorithmus ist der Linear Blendskinning (LBS) [103], der aufgrund seiner hohen Recheneffizienz oft in Echtzeitanimationen verwendet wird. Allerdings gibt es bei LBS Probleme mit visuellen Verzerrungen. Wenn die Gelenke erheblich verdreht sind, erzeugt LBS das sogenannte Candy-Wrapper Artefakt [104]. Dabei wird ein signifikanter Verlust des Gitternetzvolumens verursacht. Durch z.B. Dual Quaternion Skinning (DQS) [105], kann die Geometrieverzerrung von LBS ausgeglichen werden. DQS führt allerdings zu unerwünschten Ausbeulungen an Gelenken, welche mit manuellem Aufwand behoben werden müssen [106]. Beispiel basierte Methoden sind die zweite Möglichkeit, um die Hauthülle zu deformieren. Dabei werden die Geometrieverzerrung durch Beispielposen [107] oder zusätzliche Gewichte verhindert [108]. Im Folgenden wird die beispiel-basierte Methode verwendet.

Um ein 3D-Körpermodell zu erhalten, müssen zunächst alle wichtigen Körperpunkte ermittelt werden. Mit OpenPose können in Echtzeit 2D-Posen von mehreren Personen ermittelt werden. Dabei können auch Körperpunkte im Gesicht, den Händen und Füßen gefunden werden. Es gibt zwei verschiedene Varianten um 2D-Posen von Personen zu erhalten: von oben nach unten (engl. Top-down) oder von unten nach oben (engl. Bottom-up). Bei einem Top-down Verfahren wird zuerst die Person erkannt und dann die zugehörigen Schlüsselpunkte der Person. Im Gegensatz dazu werden beim Bottom-up Verfahren zuerst die Körperpunkte gesucht und anschließend die Punkte zusammen verbunden, die zu einer Person gehören. OpenPose verwendet einen Bottom-up Algorithmus. In Abbildung 3.5 ist der schematische Ablauf dargestellt.

Das RGB Eingabebild wird zunächst in ein VGG19 [109] Netz, benannt nach der Forschungsgruppe Visual Geometry Group der Universität Oxford, eingelesen. Das VGG19 besteht aus 16 Faltungsschichten, 3 vollverknüpften Schichten und am Ende des Netzes einer Softmax Schicht. Es wird die ReLU Aktivierungsfunktion genutzt. Das besondere ist, dass kleine 3×3 Filter mit Schrittweite 1 verwendet werden. Mehrere Faltungsschichten werden von Maxpooling Schichten der Größe 2×2 mit Schrittweite 2 gefolgt. Die Idee von VGG Netzen ist es, tiefe Netze mit sehr kleinen 3×3 Filtern zu verwenden, um eine hohe Vorhersagegenauigkeit zu erreichen. Das VGG Netz sagt eine Menge von 2D-Karten voraus, an deren Orten Körperteile vermutet werden. Anschließend werden Part Affinity Fields (PAFs) [110] angewendet. Sie geben eine Menge von 2D-Vektorfeldern zurück, welche die Orientierung der Glieder im Bildkontext anzeigen. Als Glieder werden zusammengehörige Punktpaare bezeichnet. Anschließend werden die Orte an denen sich Glieder befinden und die Orientierungen verknüpft um ein 2D-Punktmodell aller Personen auf einem Bild zu erhalten. Das Körperskelett besteht am Ende aus insgesamt 135 verbundenen Körperpunkten.

Mithilfe des 2D-Punktmodells kann ein 3D-Körpermodell entwickelt werden. Um das Ziel zu erreichen, aus einem einzigen Bild ein realistischen 3D-Abbild der Person zu kreieren, ist ein realistisches Modell des Körpers, das die Komplexität von Händen, Gesicht und Körperposen darstellen kann, nötig. Die Basis bildet deshalb das realistische Körpermodel Skinned Multi-Person Linear Model (SMPL) [89], ein punktebasiertes Modell, das eine Vielzahl von Körperformen natürlicher, menschlicher Posen genau darstellt. Dieses wurde in [90] weiterentwickelt zu SMPL-X, welches detaillierte Modelle für Gesicht und Hände enthält. Bevor die Erweiterung SMPL-X erklärt wird, wird zunächst SMPL erläutert. Die Parameter von SMPL werden aus Daten gelernt, einschließlich der Ruheposenvorlage, der Körperpunktgewichte, der Varianzen menschlicher Formen und der posenabhängigen Formen. In Abbildung 3.6 sind die einzelnen Teilformen dargestellt, die aus Beispielen gelernt werden. Die verschiedenen Formen werden addiert und mit einer Ruheposenvorlage kombiniert. Skinning kann dann die Körperknotenpunkte in die ausgeführte Pose transformieren.

Das SMPL Modell wird erweitert durch die Kombination mit dem FLAME Gesichtsmodell [111] und dem MANO Handmodell [112]. Das kombinierte Modell wird dann mit 5586 3D-Scans trainiert. Indem das Modell von Daten gelernt wird, können natürliche Korrelationen zwischen den Formen des Körpers, Gesicht und Händen, ohne Verzerrungen des Modells, abgebildet werden.

FLAME (Faces Learned with an Articulated Model and Expressions) [111] modelliert im Gegensatz zu anderen Gesichtsmodellen den gesamten Kopf und die Halsregion (und erlaubt dem Kopf relativ zum Hals zu rotieren). Dies ist nötig, um den Kopf mit dem Körper zu verknüpfen. Keine der bisherigen Methoden der Literatur modelliert Korrelationen zwischen Gesichts- und Körperformen [90]. Das FLAME Modell wurde mit ca. 4800 Köpfen mit neutralem Gesichtsausdruck aus dem CAESAR-Datensatz [113] trainiert. Um Posen und Ausdrücke zu lernen, wurden 4D-Gesichtssequenzen aus dem Dynamic 3D FACS (D3DFACS) Datensatz [114] und zusätzliche 4D-Sequenzen verwendet.

Ebenso wird das parametrische Handmodell MANO (Hand Model with Articulated and Non-rigid Deformations) [112], welches einen großen Posen und Formraum besitzt, ver-

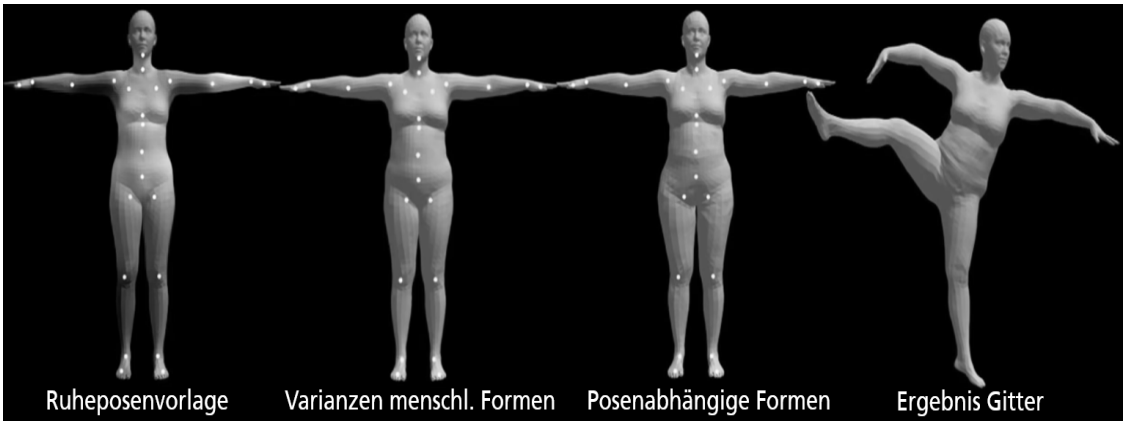


Abbildung 3.6: Darstellung der Abschnitte des SMPL Modells. Die weißen Punkte stellen die wichtigen Körperpunkte da. Sie wurden zuvor durch OpenPose ermittelt. Links ist die Ruheposenvorlage abgebildet. Zu dieser werden die Varianzen menschlicher Formen, wie die Größe verschiedener Körperteile, hinzugefügt. Da sich die menschliche Körperform unter verschiedenen Posen ändert durch z.B. das Anspannen von Muskeln, werden posenähhängige Formänderungen ebenfalls berücksichtigt. Die Ruheposenvorlage kombiniert mit der Form- und Posenvarianz ergeben dann das Körpermodell, welches durch Skinning transformiert werden kann.

wendet. Wie SMPL faktorisiert das Modell geometrische Änderungen in solche, die der Identität des Subjekts inhärent sind und in solche, die durch die Pose verursacht werden. Im Posenraum wird LBS verwendet. Die Handformen und -posen werden automatisch von Beispielen gelernt. Es wurden 1000 3D-Handscans von 31 Personen in 51 Posen zum Training des Handmodells verwendet. Das SMPL Modell, erweitert mit dem FLAME Gesichtsmodell und dem MANO Handmodell ergibt das neue SMPL-X Modell aus [90]. SMPL-X besteht aus insgesamt 119 Modellparametern: 75 Parameter für globale Körperrotationen und Körper, Augen, Kiefer Knoten, 24 Parameter für Handposen, 10 Parameter für Körperformen, 10 Parameter für Gesichtsausdrücke.

Als SMPLify [115] wird SMPL-X bezeichnet angewendet auf ein einziges Bild. Das Modell SMPLify wurde zu SMPLify-X mit einigen Verbesserungen erweitert. Es werden unmögliche Posen bestraft, da es oft Kollisionen/ Durchdringungen von verschiedenen Körperteilen gibt, die physisch nicht möglich sind. Männner und Frauen haben unterschiedliche Proportionen und Körperformen. Aus diesem Grund sollte auch bei der Modellierung eines 3D-Körpers das Geschlecht mit einbezogen werden. Bisher gibt es außer SMPLify-X noch kein Modell, das dies anwendet [90]. Die Merkmale der Geschlechter werden mit einem Residual Neuronal Network (ResNet) [116] vorhergesagt. Um zu verstehen, warum ein ResNet verwendet wird, sollte man sich zunächst klar machen, dass theoretisch ein tieferes neuronales Netz detailliertere Merkmale erkennen kann und somit ein besseres Verständnis der Daten und eine höhere Vorhersagegenauigkeit haben sollte. Allerdings haben Experimente gezeigt, dass flachere Netze teilweise bessere Ergebnisse liefern als tiefere [116]. Das liegt daran, dass im Fall, in dem eine Identitätskarte optimal wäre, es schwierig ist, diese auf mehrere Schichten zu passen. Einfacher ist es, wenn Verbindungen

ausgelassen und Informationen einer Schicht in eine tiefe Schicht direkt gefüttert werden können. Dazu können eine oder mehrere Schichten übersprungen werden. Vor der ReLU Aktivierungsfunktion werden dann die Informationen einer vorherigen Schicht zu einer Späteren hinzugefügt. Dabei entstehen keine neuen Parameter oder Rechenkomplexität. Das Netz kann immer noch mit dem stochastischen Gradientenverfahren und Fehlerrückführungsalgorithmus (vergleiche Abschnitt 2.4) trainiert werden.

Es wird eine ResNet18, d.h. ein ResNet mit 18 Schichten verwendet. Die Aktivierungsfunktion sind Leaky ReLU (LReLU) [117]. Im Gegensatz zu ReLU (vergleiche Abschnitt 2.3) werden Funktionseingaben kleiner als Null nicht auf Null gesetzt, sondern steigen leicht an. Die Funktion dafür lautet: $f(x) = 0.01 \cdot x$, falls $x < 0$. Dies führt dazu, dass sich negative Werte leichter erholen können und nicht Null bleiben. Dadurch wird verhindert, dass Neuronen nicht "sterben" . Aus diesem Grund muss weniger auf die Initialisierung des neuronalen Netzen und auf die Normalisierung der Daten geachtet werden.

An das ResNet18 werden zwei vollverknüpfte Schichten zur Klassifikation des Geschlechts angehängt. Die Größe der Bilder wird auf 224 × 224 Pixel verändert um kompatibel mit der ResNet18 Architektur zu sein. Zur Optimierung wurde Adam [82] verwendet. Trainiert wurde mit verschiedenen Datensätzen. Es wurden 50216 Trainingsbeispiele und 16170 Testbeispiele aus fünf verschiedenen Datensätzen verwendet. Dazu gehören unter anderem COCO [50], MPII Human Pose [118], welcher 800 verschiedene Aktivitäten und unterschiedliche Blickwinkel enthält sowie ein Datensatz mit 2.000 Ganzkörperbildern aufgenommen beim Durchführen verschiedener Sportarten [119]. Für die Implementierung wurden die offiziellen Teilungen in Trainings- und Testsätzen der verschiedenen Datensätze verwendet.

Die Implementierung von [90] nutzt einen Schwellwert von 0,9, ab dem die vorhergesagte Geschlechtsklasse angenommen wird. Unterhalb des Schwellwertes wird eine Person als neutrales Geschlecht klassifiziert. Mit diesen Einstellungen wurde mit einer Wahrscheinlichkeit von nur 7,5% das Geschlecht falsch vorhergesagt. Die Rate für richtige Vorhersagen lag aber dennoch bei nur 62,38%, da öfters das Geschlecht als neutral klassifiziert wurde. In der verwendeten Implementierung dieser Arbeit wurde der Schwellwert auf 0 gesetzt, sodass dieser immer überschritten wird. Dadurch erfolgt immer eine Zuordnung des Geschlechts als männlich oder weiblich. Das neutrale Geschlecht wird aus diesem Grund nie vorhergesagt. Dies wurde so gewählt, da in der originalen Implementierung sonst in vielen Fällen das neutrale Geschlecht vorhergesagt wird. Damit ist allerdings keine Zuordnung zu Zielgruppen möglich, die im Praxiskontext erwünscht ist. Deshalb wurde in der verwendeten Implementierung das neutrale Geschlecht umgangen.

Zur Erstellung des 3D-Körpermodells wurde die Implementierung[5] von Nima Ghorbani verwendet. Diese nutzt TensorFlow [65] (vergleiche Abschnitt 2.7) und das trainierte Modell und die Gewichte werden mit PyTorch[6] verwaltet. Als Eingabe benötigt diese Implementierung die Körperpunkte der abgebildeten Personen, welche durch OpenPose ermittelt werden können. Dazu wurde mit einer Implementierung[7] die Körperpunkte des

[5] `https://github.com/nghorbani/homogenus`

[6] PyTorch ist eine auf maschinelles Lernen ausgerichtete öffentliche Programmbibliothek für Python, vergleiche `https://pytorch.org/`

[7] `https://github.com/CMU-Perceptual-Computing-Lab/openpose`

Tabelle 3.10: Konfusionsmatrix Geschlechterkennung anhand des ganzen Körpers

		Tatsächlich		
		weiblich	männlich	Summe
Vorhersage	weiblich	60	7	67
	männlich	62	78	140
	Summe	122	85	207

Gesichts, der Hände und des Körpers ermittelt und als json File zurückgegeben. Dieses json File sowie das zugehörige Bild sind als Eingabe der Implementierung zur Erstellung des 3D-Körpermodells erforderlich. Dann kann das Geschlecht der Personen und ein 3D-Modell ihrer Körper zurückgegeben werden.

3.4.2 Datensatz und Evaluierung der Geschlechterkennung anhand des ganzen Körpers

Im Folgenden werden die Ergebnisse der Evaluation des Modells ohne neutrales Geschlecht und mit dem Datensatz der RTG beschrieben. Es wurden 100 Bilder mit 261 Personen, davon 153 Frauen und 108 Männer verwendet. Die Personen auf den Fotos fahren Rad, sitzen, stehen oder gehen um verschiedene Posen in den Evaluationsdatensatz vorzufinden, da es nicht trivial ist verschiedene Posen zu modellieren. 54 Personen wurden auf den Bildern nicht erkannt, dementsprechend konnte kein Geschlecht vorhergesagt werden. Dies entspricht ca. 26% aller Personen im Evaluationsdatensatz. Personen die nicht erkannt wurden, waren oft verdeckt durch andere Personen oder Fahrräder oder waren zu dicht nebeneinander, sodass sie als eine Person erkannt wurden. Teilweise war auch der Körper nicht vollständig auf dem Foto zu sehen (z.B. nur bis zur Hüfte) oder ein Teil des Körpers auf dem Bild abgeschnitten. In Tabelle 3.10 ist das Ergebnis der Evaluation der erkannten Personen in einer Konfusionsmatrix dargestellt.

Wir können feststellen, dass 62 mal weibliche Personen als männlich erkannt wurden. In nur 7 Fällen wurden allerdings Männer als Frauen erkannt. Dementsprechend wurden sehr oft männliche Personen vorhergesagt (140 mal) und Frauen nur ungefähr halb so oft (67 mal). Deshalb ist die Genauigkeit mit 0,90 sehr hoch, da Männer selten falsch erkannt wurden, allerdings liegt die Trefferquote nur bei 0,49, da weibliche Personen häufig als männlich vorhergesagt wurden.

$$\text{Genauigkeit} = \frac{60}{60+7} = 0,90$$

$$\text{Trefferquote} = \frac{60}{60+62} = 0,49$$

$$\text{F1-Kennwert} = \frac{2 \cdot 0,49 \cdot 0,90}{0,49+0,90} = 0,63$$

Abbildung 3.7: Beispielbilder der Ergebnisse der Geschlechtererkennung mit SMPLify-X anhand des gesamten Körpers. Die Personen auf den oberen, grün hinterlegten Bilder wurden alle richtig zugeordnet. Die Frauen auf den unteren, rot hinterlegten Bildern wurden als Männer erkannt.

Verrechnet man Genauigkeit von Trefferquote zum F1-Kennwert ergibt dies ein Ergebnis von 0,63, welches sehr nah an der Richtigkeit liegt. Insgesamt wurden bei 138 von 207 Personen das Geschlecht richtig vorhergesagt.

$$\text{Richtigkeit} = \frac{60 + 78}{207} = 0,67$$

Da es ungleich viele Frauen und Männer auf den Bildern gibt, wird zusätzlich der MCC berechnet, da bei nicht ausgeglichenen Daten der F1-Kennwert und die Genauigkeit zu optimistisch sein könnten.

$$\text{MCC} = \frac{60 \cdot 78 - 7 \cdot 62}{\sqrt{(60 + 7) \cdot (60 + 62) \cdot (78 + 7) \cdot (78 + 62)}} = 0,43$$

Der MCC ist allerdings schwer direkt mit F1-Kennwert und Richtigkeit vergleichbar, da hier Wertebereich zwischen 0 und 1, hingegen beim MCC Wertebereich zwischen -1 und 1.

In Abbildung 3.7 sind verschiedene Beispielbilder des verwendeten RTG Datensatzes zur Geschlechtserkennung dargestellt. Grün hinterlegt wurde zunächst eine Auswahl verschiedener richtig klassifizierter Personen. Das Geschlecht von Personen unterschiedlichen Alters und in verschiedenen Körperposen wurde richtig erkannt. Im rot hinterlegten Abschnitt sind Beispielbilder abgebildet, auf denen Frauen als Männer erkannt wurden. Es ist dabei kein Altersunterschied festzustellen, ob bei jungen oder älteren Personen schlechter das Geschlecht erkannt wurde. Es ist zu beachten, dass es wenige Bilder mit Kindern im Datensatz gibt und daher nicht nachvollzogen werden konnte, ob das Geschlecht bei

Kindern meist richtig erkannt wird. Im Kontext ist das Geschlecht von Kindern ohnehin nicht relevant für die Zuordnung der Zielgruppe. Es trat kaum das Problem auf, dass Personen nicht erkannt wurden. Es ergab keinen Unterschied, ob das Geschlecht von Personen im Stehen, im Sitzen oder beim Rad fahren erkannt wurde.

Verglichen mit der Geschlechtserkennung anhand des Gesichts (in Abschnitt 3.2), ist diese Methode eine interessante Alternative für den Fall, dass der ganze Körper einer Person auf einem Bild zu sehen ist. Für Nahaufnahmen von Gesichtern ist eine Gesichtserkennung zu bevorzugen, da hier eine genauere Vorhersagegüte in der Literatur erreicht wurde. In unserem Praxisbeispiel in dem Personen meist vollständig zu sehen sind und das Gesicht weiter entfernt, durch Sonnenbrillen und Helme verdeckt ist, ist die Verwendung eines Körpermodells eine sinnvolle Alternative.

3.5 Erkennung von Fahrrädern

Neben der Erkennung von Personen auf Bildern ist es für die Redakteure im Radtourismus ebenfalls wichtig, nach Vorhandensein von Rädern zu filtern. Aus diesem Grund ist eine Forschungsfrage dieser Arbeit ob bestehende Implementierungen zur Fahrraderkennung praxistauglich sind. Deshalb wird in dieser Sektion eine Fahrraderkennung evaluiert, um in Kapitel 4 die Forschungsfrage beantworten zu können.

3.5.1 Beschreibung des vortrainierten Netzes der Fahrraderkennung

Um Fahrräder auf Bildern zu erkennen kann das gleiche vortrainierte Netz wie in Abschnitt 3.3.1 zur Erkennung von Personen verwendet werden. Mit diesem Netz ist es möglich 20 verschiedene Objekte zu erkennen. Dazu gehören Personen, Fahrräder, Motorräder uvm.

3.5.2 Datensatz und Evaluierung der Fahrraderkennung

Der Datensatz zur Erkennung von Fahrrädern wurde ebenfalls aus dem Datensatz der RTG entnommen. Es wurden insgesamt 200 Bilder ausgewählt, davon sind 100 Bilder mit mindestens einem Rad pro Bild und 100 Bilder ohne Fahrrad. Die Evaluationsdaten beinhalten 100 Bilder ohne Fahrräder, um zu überprüfen, ob Räder erkannt werden, obwohl keine auf dem Bild zu sehen sind. Dieser Test lieferte ein sehr gutes Ergebnis, da auf keinem der 100 Bilder fälschlicherweise ein Rad erkannt wurde. Auch auf den restlichen Bildern wurde nie ein Rad an einer Stelle erkannt, an der keines vorhanden war. Dies ist in Tabelle 3.11 am falsch positiven Wert 0 abzulesen. Aus der Konfusionsmatrix ergibt sich eine Genauigkeit von 1.

$$\text{Genauigkeit} = \frac{87}{87+0} = 1$$

Tabelle 3.11: Konfusionsmatrix Raderkennung

		Tatsächlich		
		Rad da	kein Rad da	Summe
Vorhersage	Rad da	87	0	87
	keine Rad da	140	100	240
	Summe	227	100	327

Tabelle 3.12: Ergebnisse Fahrraderkennung

Untersuchungsobjekt	Wahrscheinlichkeit
Rad erkannt	0,38
Rad nicht erkannt	0,51
Motorrad erkannt	0,11
Rad oder Motorrad erkannt	0,49
Mind. 1 Rad erkannt	0,76

Da der falsch negativ Wert mit 140 nicht erkannten Rädern sehr hoch ist, erhalten wir eine Trefferquote von nur 0,38.

$$\text{Trefferquote} = \frac{87}{87 + 140} = 0,38$$

Der F1-Kennwert ist, aufgrund des niedrigen Anteils der korrekt erkannten Räder, bezogen auf die Menge aller tatsächlichen Räder, mit 0,55 eher gering.

$$\text{F1-Kennwert} = \frac{2 \cdot 0,38 \cdot 1}{0,38 + 1} = 0,55$$

In Tabelle 3.12 ist aufgeführt, mit welchen Wahrscheinlichkeiten Fahrräder identifiziert wurden. Dabei wurde ebenfalls untersucht, ob ein Fahrrad mit einem Motorrad verwechselt wurde. Mit einer Wahrscheinlichkeit von 38% sind die Räder auf den Beispielbildern vom RTG erkannt worden. Knapp die Hälfte der Räder wurde nicht erkannt und 11% wurden mit Motorrädern verwechselt. Geht man davon aus, dass im Datensatz keine Bilder von Motorrädern enthalten sind und deshalb alle erkannten Motorräder Fahrräder sein müssen, dann werden Räder mit einer Wahrscheinlichkeit von fast 50% erkannt. Dieses Ergebnis scheint nicht vielversprechend, allerdings werden auf 76% aller Bilder mindestens ein Rad identifiziert. Dieses Ergebnis ist hinreichend für den konkreten Anwendungsfall, da die exakte Zahl an Fahrrädern im Foto nicht entscheidend ist. Vielmehr ist es für den Redakteur im Radtourismus interessant zu wissen, ob auf dem Foto Personen mit Fahrrad zu sehen sind und nicht wie viele. In anderen Anwendungsfällen wie etwa beim autonomen Fahren ist eine solche Betrachtung nicht zielführend. Im Straßenverkehr ist es extrem wichtig mit einer sehr hohen Wahrscheinlichkeit alle Objekte zu erfassen.

Vergleicht man die Wahrscheinlichkeit von 38%, dass ein Rad erkannt wird, mit dem aktuellen Stand der Technik, der eine Genauigkeit von fast 90% auf dem VOC 2012 Datensatz

Abbildung 3.8: Beispielbild aus dem Datensatz der RTG auf dem kein Rad erkannt wurde

erreicht [86], ist unser Ergebnis weit davon entfernt. Die Modelle wurden sehr ähnlich trainiert. Beide wurden auf dem COCO [50] und VOC 2007 [85] Datensatz traininert und das Netz aus [86] zusätzlich noch auf PASCAL VOC 2012 [87]. Allerdings wird in der Implementierung aus [86] die R-FCN [44] Objekteerkennungsmethode und ein ResNet [116] (vergleiche Abschnitt 3.4) genutzt, im Vergleich dazu wird in der verwendeten Implementierung ein SSD [46] mit einer MobileNets Netzwerkarchitektur [83] abgewendet. Bei einer R-FCN Methode handelt es sich um eine zweistuftige Objekterkennung, hingegen bei der SSD Methode um eine einstufige Objekterkennung. Die Unterschiede einer der zwei verschiedenen Objekterkennungsvarianten wurden bereits in Abschnitt 2.6 erläutert. Das ResNet-101 hat 101 Schichten [116] und ist damit sehr tief im Vergleich zum MobileNet mit 28 Schichten [83]. Die unterschiedlichen Objekterkennungsmethoden und Netzwerkarchitekturen könnten einen großen Einfluss auf die Performance haben, aber genauso ist es denkbar, dass der verwendete Datensatz zur Evaluation komplexer ist als das Testset von VOC 2012. Deshalb ist es schwierig, eine Aussage darüber zu treffen, welche Objekterkennungsmethode und welches CNN für den Anwendungsfall besser geeignet sind.

Probleme bei der Erkennung von Rädern im Datensatz der RTG sind vor allem, dass einige Räder auf den Bilder zu weit entfernt und damit nur klein abgebildet sind. Außerdem sind die Räder oft verdeckt oder nicht vollständig auf dem Bild erkennbar. Dadurch wurden die Räder vielmals nicht erkannt. Außerdem wurden gelegentlich Räder mit Motorrädern verwechselt. Dies war bei Elektrofahrrädern oder Rädern mit Gepäck festzustellen. Auf dem Beispielbild in Abbildung 3.8 sind zwei Räder zu sehen, die nicht erkannt wurden. In diesem Fall ist zu vermuten, dass die Räder zu klein auf dem Foto und nicht vollständig zu erkennen sind und deshalb nicht richtig klassifiziert wurden.

Im Projektkontext ist die Erkennungsgenauigkeit soweit hilfreich, dass die Mitarbeiter der RTG nur noch knapp die Hälfte der Bilder manuell überprüfen müssten, auf denen kein Rad erkannt wurde, ob wirklich kein Fahrrad auf den Bildern zu sehen ist. Das spart Zeit.

Dennoch könnte, verglichen mit dem aktuellen Stand der Forschung, die Erkennunngsrate höher sein. In Anwendungen wie beim autonomen Fahren o.Ä., die für Personen sicherheitsrelevant sind, ist diese Genauigkeit nicht ausreichend.

4 Beantwortung der Forschungsfragen

Zu Beginn der Arbeit wurden in Abschnitt 1.3 vier Forschungsfragen formuliert. Zunächst sollten die Zielgruppen des Radtourismus entwickelt werden. Drauf aufbauend war herauszufinden, wie eine Erkennung dieser auf Bildern möglich ist. Die dritte Forschungsfrage bezieht sich auf die Praxistauglichkeit bestehender Implementierungen zur Fahrraderkennung, bevor die Letzte die Frage stellt, ob vorhandene Datensätze ausreichen um eine Generalisierung der Objekterkennung auf Praxisdaten zu ermöglichen. Diese Forschungsfragen werden mithilfe der Ergebnisse aus Kapitel 3 beantwortet.

4.1 Welche Zielgruppen gibt es im Radtourismus?

Die gezielte Kommunikation mit Zielgruppen ist sehr wichtig. Um mögliche Kunden anzulocken, soll speziell auf ihre Bedürfnisse eingegangen werden. Dafür ist es wichtig, den Markt und seine Zielgruppen zu kennen. Aus diesem Grund ist es sinnvoll, die Zielgruppen des Radtourismus zu ermitteln.

In Abschnitt 3.1 ist die Ausarbeitung der Zielgruppen ausführlich beschrieben. Die Zielgruppen wurden durch Interviews mit der RTG ermittelt. Die Destinationsmanagementorganisation hat ihren Fokus auf dem Radtourismus. Die Mitarbeiter sind deshalb Fachexperten in diesem Bereich. Zunächst wurden in den Interviews verschiedene Personenklassen unterschieden: Frau, Mann, junge Frau, junger Mann, weiblicher und männlicher Best Ager sowie Kind und Jugendlicher. Mit diesen können die Zielgruppen des Radtourismus genau definiert werden. Als relevante Zielgruppen wurden ermittelt: Junges Paar, Best Ager Paar, Familie, Frauengruppe und gemischtgeschlechtliche Gruppe (vergleiche Tabelle 3.2). Für eine Zuordnung von Personen zu ihren Zielgruppen ist Wissen über das Geschlecht und Alter nötig. Bei Kindern ist das Geschlecht nicht relevant. Vor allem durch den stark wachsenden Markt an Elektrofahrrädern sind Best Ager die wichtigste Zielgruppe im Radtourismus [72]. Aber auch junge Paare werden im Radtourismus immer aktiver. Besonders interessant ist ebenso die Unterscheidung, ob es sich um eine Gruppenreise handelt.

In der Literatur werden die Zielgruppen des Radtourismus kaum thematisiert, obwohl es Statistiken zum Radfahrverhalten von Männern und Frauen sowie auch Altersverteilungen gibt [73, 72]. In [74] werden die Zielgruppen nicht nach Personenmerkmalen unterschieden sondern nach Art des Rades: Tourenrad, Mountainbike oder Rennrad. Aufgrund dieser Forschungslücke ist es ein wichtiges Ergebnis, dass die Zielgruppen des Radtourismus anhand ihrer Geschlechts- und Altersmerkmale unterschieden werden. Zumal die relevanten Zielgruppen sehr unterschiedlich sind und deshalb eine individuelle Kommunikation erfordern.

4.2 Wie können Zielgruppen auf Bildern erkannt werden?

Zu wissen, welche Zielgruppen im Radtourismus interessant sind, ist der Grundstein. Mit diesen Erkenntnissen sollen die Zielgruppen auf Fotos erkannt werden, um sie geeignet anzusprechen. Es ist sinnvoll, junge Paare mit Bildern von jungen Paaren anzuwerben und auch Best Ager mit Bildern ihrer Altersgruppe. Dazu ist es nötig die Zielgruppen auf den Bildern des Datensatzes der RTG zu erkennen. Wie eine Vorhersage der Zielgruppen möglich ist, wird in diesem Abschnitt zusammengefasst.

Für die ermittelten Zielgruppen sind Informationen über das Geschlecht und Alter von Personen nötig. Aus diesem Grund wurde nach Möglichkeiten gesucht, diese Merkmale auf Bildern zu erkennen. Zunächst wurde in Abschnitt 3.2 mit einem Algorithmus zur Gesichtserkennung begonnen, mit dem das Geschlecht und Alter vorhergesagt werden kann. Das Problem war hierbei, dass die Gesichter der Personen in nur wenigen Fällen erkannt wurden. Aus diesem Grund konnten kaum das Geschlecht und Alter vorhergesagt werden. Lediglich 29 von 322 Gesichtern wurden identifiziert. Besonders wenige Gesichter von Best Agern wurden vom neuronalen Netz wahrgenommen. Das könnte der Problematik geschuldet sein, dass es deutlich weniger Fotos von älteren Personen im Trainingsdatensatz gibt. Ebenso wurden kaum Personen auf Gruppenbildern erkannt. Dies könnte daran liegen, dass Gruppenaufnahmen aus weiterer Entfernung aufgenommen werden und somit die Gesichter nicht nah genug zu erkennen sind. Die Gesichter des RTG Datensatzes sind oft nicht ausreichend erkennbar, da seitlich oder von hinten fotografiert wurde. Helme und Sonnenbrillen verdecken zusätzlich das Gesicht.

Das Geschlecht der erkannten Gesichter wurde mit relativ hoher Trefferquote vorhergesagt. Allerdings muss der F1-Kennwert von 0,88 vorsichtig betrachtet werden, da nur bei 23 Gesichtern das Geschlecht vorhergesagt werden konnte. Durch die geringe Anzahl an erkannten Gesichtern konnte ebenfalls nicht umfänglich die Alterserkennung evaluiert werden. Trotzdem war die Tendenz festzustellen, dass vor allem Männer eher zu jung eingeschätzt werden. Die Zielgruppe konnte nur bei einem einzigen Foto richtig vorhergesagt werden. Aus diesem Grund ist eine Gesichtserkennung auf dem Datensatz des Radtourismus nicht zielführend.

Als Alternative zur Gesichtserkennung wurde anschließend die Personenanzahl auf Bildern in Abschnitt 3.3.1 evaluiert. Damit ist keine detaillierte Zielgruppenzuordnung möglich, allerdings können wertvolle Informationen darüber erhalten werden, wie viele Personen sich auf einem Bild befinden. Dadurch ist es möglich vorherzusagen, ob ein Gruppenfoto vorliegt. Personen konnten auf dem Evaluationsdatensatz mit einer Wahrscheinlichkeit von 89% richtig vorhergesagt werden. Dies ist ein sehr gutes Ergebnis verglichen mit dem Stand der Technik von 91%. Junge und ältere Menschen konnten gleich gut erkannt werden. Das neuronale Netz hatte wenig Probleme, Personen auf Gruppenbildern zu erkennen. Mit einer Wahrscheinlichkeit von 97% wurde eine Gruppe auf Bildern erkannt, wenn mindestens drei Personen auf dem Foto zu sehen sind. Lediglich, wenn Personen sehr nah nebeneinander stehen, ist es möglich, dass zwei Personen als eine einzige Person erkannt werden. Die Erkennung von Personen ist eine gute Ausgangslage. Genauere

Informationen über das Geschlecht oder Alter der Personen sind jedoch wichtig für die Zuordnung in die konkreten Zielgruppen. Deshalb wurde nach weiteren Möglichkeiten gesucht mehr Informationen über die Personen aus Bildern herauszulesen.

Mittels eines 3D-Körpermodells kann das Geschlecht von Personen in Abschnitt 3.4.1 vorhergesagt werden. Dafür ist ebenfalls lediglich ein einzelnes Bild der Personen nötig. Da im Datensatz der RTG ohnehin der ganze Körper fast aller Personen abgebildet ist, war es kein Problem für die Evaluation geeignete Bilder zu finden. Es wurden Fotos verwendet von Personen, die radfahren, sitzen, stehen oder gehen, um verschiedene Posen in den Bildern vorzufinden. Die Personen wurden meist erkannt, allerdings wurden Frauen häufiger als Männer falsch vorhergesagt. Bei 138 von 207 erkannten Personen wurde das Geschlecht richtig erkannt. Das ergibt eine Richtigkeit von 0,67. Es war dabei kein Altersunterschied festzustellen, ob bei jungen oder älteren Personen schlechter das Geschlecht erkannt wurde. Es wurde kein Unterschied erkannt, dass das Geschlecht von Personen im Stehen besser erkannt wurde als das Geschlecht von Personen im Sitzen oder beim Rad fahren. Dies ist für den Datensatz im Aktivtourismus sehr entscheidend.

Liegt ein Datensatz vor, in dem vorwiegend Nahaufnahmen von Gesichtern enthalten sind, ist eine Gesichtserkennung zu bevorzugen. Damit konnte eine genauere Vorhersagegüte in der Forschungsliteratur erreicht werden. Bei unserem Praxisbeispiel, in dem Personen meist vollständig zu sehen sind, das Gesicht weiter entfernt sowie durch Sonnenbrillen und Helme verdeckt ist, ist die Verwendung eines Körpermodells eine sinnvolle Alternative. Die Vorhersagegüte mit 0,67 ist sicherlich noch ausbaubar durch weitere Forschung im Bereich der Körpermodellierung. Insgesamt kann empfohlen werden, weiter an der Geschlechtserkennung anhand des gesamten Körpers zu forschen, um eine höhrere Vorhersagegüte zu erreichen. Eine Alterserkennung anhand des Körpers ist nach bekanntem Stand der Technik noch nicht erforscht. Falls diese Forschungslücke noch geschlossen werden kann ist eine detaillierte Zuordnung von Bildern gemäß ihrer in Abschnitt 3.1 ermittelten Zielgruppe möglich. Bis dahin können bereits interessante Rückschlüsse aus der Personenzahl und der Geschlechtserkennung geschlossen werden.

4.3 Sind bestehende Implementierung zur Fahrraderkennung praxistauglich?

Wir haben das Glück, dass es für viele verschiedene Problemstellungen frei zugängliche Implementierungen zur Nutzung für die Forschung gibt. So auch für die Raderkennung auf Bildern. Aber sind diese auch abseits der Forschung in der Praxis einsetzbar? Diese Frage soll im Folgenden mit den Ergebnissen aus Abschnitt 3.5 beantwortet werden.

Für die Implementierung der Fahrraderkennung wurde eine MobileNets [83] Netzwerkarchitektur verwendet, die 20 verschiedene Objekte erkennen kann. Dazu gehören Personen, Fahrräder, Motorräder uvm. Aufgrund der Ressourceneffizienz von MobileNets sind auch mobile Anwendungen möglich. Das verwendete Netz wurde vortrainiert auf dem COCO [50] und auf dem VOC 2007 [54]. Beides sind große, bekannte Datensätze, die aus Fotos der Internetseite `flickr.com` von Hobbyfotografen bestehen. Die Bilder enthalten meist

Szenen im Gegensatz zu ikonischen Bildern. Allerdings ist zu bedenken, dass das neuronale Netz auf verschiedene Objektklassen und nicht ausschließlich auf die Erkennung vom Fahrrädern trainiert wurde. Der VOC 2007 Datensatz enthält rund 500 Fotos von 700 Fahrrädern, von Motorrädern etwa ähnlich viele, allerdings ca 10.000 Personen in ungefähr 4.000 Bildern. Dadurch ist es nicht verwunderlich, dass die Personenerkennung, welche mit vielen Personenbildern trainiert wurde gut generalisiert und eine Erkennungswahrscheinlichkeit von fast 90% erreicht. Während die Fahrraderkennung, traininert auf erheblich weniger Bildern von Rädern, nur 38% richtige Vorhersagen, gemessen auf dem Datensatz der RTG, schafft. Knapp die Hälfte der Räder wurde nicht erkannt und 11% wurden mit Motorrädern verwechselt.

Geht man davon aus, dass in diesem speziellen Datensatz keine Bilder von Motorrädern enthalten sind und deshalb alle erkannten Motorräder Fahrräder sein müssen, dann werden Räder mit einer Wahrscheinlichkeit von fast 50% erkannt. Es ist im konkreten Praxiskontext nicht entscheidend, alle Räder auf einem Foto zu erkennen. Mindestens ein Rad wurde auf 76% aller Bilder mit Rädern erkannt. Dieses Ergebnis ist hinreichend für den konkreten Anwendungsfall, da die exakte Zahl an Fahrrädern im Foto nicht entscheidend ist. Für den Redakteur im Radtourismus ist es interessant zu wissen, ob auf dem Foto Personen mit Fahrrad zu sehen sind und nicht wie viele Räder.

Prinzipiell sind die existierenden Implementierungen in der Praxis mit Vorsicht zu nutzen. Die Erkennungsgüte sollte immer auf diversen Beispielen aus dem konkreten Praxisdatensatz evaluiert werden. Je nachdem wie sicherheitskritisch die Verwendung der Objekterkennung ist, ist im Einzelfall zu entscheiden, ob die Implementierung ausreichend ist. Im Kontext der RTG kann mit der vorliegenden Implementierung gearbeitet werden. Hier befinden wir uns in keinem sicherheitskritischen Bereich, sondern reduzieren die manuelle Zuordnungsarbeit der Mitarbeiter. Durch die Implementierung kann die Arbeitszeit, in der Mitarbeiter monoton Bilder ihren Klassen zuordnen, um 76% reduziert werden. Dabei legen wir für diesen speziellen Fall fest, dass alle Motorräder Fahrräder sein müssen. Dies ist ist nur in diesem konkreten Kontext sinnvoll. Durch die Zeiteinsparung bei der Klassenzuordnung können sich die Redakteure der RTG auf komplexere Aufgaben konzentrieren. In anderen Praxisanwendungsfällen können Motorräder nicht Fahrrädern gleichgesetzt werden zudem ist die genaue Anzahl und Position von Rädern ist wichtig. In diesen Fällen ist die Implementierung mit nur 38% richtig erkannten Rädern nicht ausreichend. Soll beispielsweise das Nutzungsverhalten überprüft werden, indem gezählt wird, wie viele Fahrräder in einem Bereich unterwegs sind, würde die Implementierung das Ergebnis extrem verfälschen. Denkt man an sicherheitskritische Anwendungen, wie das autonome Fahren, wäre eine Implementierung mit einer solchen Erkennungsgüte auf keinen Fall einsetzbar. Es ist zu vermuten, dass mit einem vielfältigerem Datensatz beim Training des neuronalen Netzes eine besseres Ergebnis der Fahrraderkennung möglich ist. Sind im Evaluationsdatensatz Bilder von Rädern mit Gepäcktaschen enthalten, aber im Trainingssatz nicht, ist es nicht verwunderlich, wenn solche Räder als Motorräder erkannt werden.

4.4 Sind vorhandene Datensätze ausreichend um eine Generalisierung der Objekterkennung auf Praxisdaten zu ermöglichen?

Allgemein schaffen es kNN aufgrund ihrer Datensatzvoreingenommenheit oft schlecht zu generalisieren. Dies wird in [51] mit einer Kreuzvalidierung auf verschiedenen Datensätzen gezeigt. Alle sechs Datensätze liefern mit Abstand bessere Ergebnisse, wenn der Testdatensatz aus dem gleichen Datensatz entnommen wurde wie der Trainingsdatensatz, verglichen mit dem Testen auf einem anderen Datensatz. Vergleiche dazu Abschnitt 2.6. Im gleichen Abschnitt werden außerdem die vier verschiedenen Arten von Problemen, die Datensätze haben, beschrieben: Auswahlvoreingenommenheit, Erfassungsvoreingenommenheit, Benennungsvoreingenommenheit und Negativdatensatz-Voreingenommenheit.

Um zu betrachten, wie gut vortrainierte Implementierungen der Objekterkennung auf Praxisdaten vorhersagen, wurden verschiedene getestet und die Trainingsdatensätze genau betrachtet. Ziel ist es herauszufinden, ob die Vorhersagegüte auf einem Praxisdatensatz abnimmt, im Vergleich zur Güte getestet auf Forschungsdatensätzen. Außerdem sollen mögliche Probleme der Datensätze aufgedeckt werden, welche eine Generalisierung verhindern. Es soll bewertet werden, ob vorhandene Datensätze die Realität gut genug abbilden, damit eine Generalisierung auf Realweltdaten möglich ist. Dabei wird angemerkt, dass auch Praxisdatensätze nicht vollständig die reale Welt abbilden. Diese sind ebenfalls aus einem bestimmten Verwendungszweck gesammelt worden und deshalb eingeschränkt.

Das erste vortrainierte Netz das evaluiert wurde, war ein CNN zur Gesichtserkennung. Damit wurden kaum Gesichter auf dem Datensatz der RTG erkannt. Vergleiche dazu Abschnitt 3.2. Dies ist besonders auf die Verschiedenheit des Trainingsdatensatzes zum Praxisdatensatz, auf dem evaluiert wurde, zurückzuführen. Der Trainingsdatensatz des neuronalen Netzes enthält vor allem Nahaufnahmen von einzelnen, jüngeren Personen, die direkt in die Kamera schauen. Hingegen enthält der Datensatz der RTG Bilder von Best Agern, vorwiegend Ganzkörperbilder, die Gesichter werden durch Sonnenbrillen und Helme verdeckt sowie einige Gruppenbilder. Besonders wenige Gesichter von Best Agern wurden vom kNN wahrgenommen. Das könnte der Problematik geschuldet sein, dass es deutlich weniger Fotos von älteren Personen im Trainingsdatensatz gibt. Insgesamt konnte die Gesichtserkennung auf dem Datensatz der RTG sehr schlecht generalisieren. Dies ist allerdings durch die starken Unterschiede der Datensätze zu begründen. Im Bereich der Alters- und Geschlechtserkennung besteht allgemein das Problem der Verfügbarkeit von großen, geeigneten Datensätzen. Die Schwierigkeit ist hierbei, dass persönliche Informationen über das Geschlecht und Alter der Personen nötig sind. Diese Informationen sind in der Regel nicht öffentlich. Es muss angemerkt werden, dass es außer dem Datensatz noch weitere Aspekte gibt, welche Einfluss auf die Fähigkeit eines kNN zur Generalisierung haben. Diese werden bei der Forschungsfrage nicht berücksichtigt. Trotzdem soll angefügt werden, dass beim Training des Netzes Dropout Lernen [26] verwendet wurde, was eine Überanpassung des Netzes an den Trainingsdatensatz verhindern soll.

Für die Personenerkennung und Fahrraderkennung wurde das gleiche Netz verwendet. Das

MobileNet [83] wurde mit den Datensätzen COCO [50] und VOC 2007 [54] trainiert. Die Personenerkennung erreichte eine sehr hohe Erkennungsgüte von 89%. Die Generalisierung auf dem Datensatz der RTG ist demnach ohne Probleme erfolgt. Die Fahrraderkennung funktionierte allerdings nicht einmal halb so gut. Räder wurden nur mit einer Wahrscheinlichkeit von 38% richtig erkannt. Leider kann diese Wahrscheinlichkeit nicht direkt mit der Wahrscheinlichkeit auf dem Testset der Implementierung verglichen werden, da diese nicht konkret für Fahrräder veröffentlicht wurde. Es ist eine mittlere Vorhersagegenauigkeit von 72,7% über alle 20 Objekte angegeben worden. Geht man davon aus, dass die Fahrraderkennung auf dem VOC 2007 [54] ungefähr bei 73% liegt, ist dies erheblich besser als die Vorhersagegüte auf dem Datensatz der RTG. Vergleicht man die Wahrscheinlichkeit von 38%, dass ein Rad erkannt wird, mit dem aktuellen Stand der Technik von fast 90% auf dem VOC 2012 Datensatz [86], ist unser Ergebnis sehr weit davon entfernt.

Räder aus dem Datensatz der RTG wurden vor allem nicht erkannt, wenn sie auf den Bildern zu weit entfernt und damit nur klein abgebildet sind. Außerdem sind die Räder im Datensatz teilweise verdeckt oder nicht vollständig auf dem Bild zu erkennen. Dies ist in den Trainingsdatensätzen kaum der Fall gewesen, wodurch die Räder vielmals in der Evaluation nicht erkannt wurden. Außerdem wurden gelegentlich Räder mit Motorrädern verwechselt. Dies war vor allem bei Elektrofahrrädern oder Rädern mit Gepäck der Fall. Die Verwechslung von Fahrrädern mit Motorrädern sollte vermieden werden, indem mit mehr Beispielen von Motorrädern trainiert wird. Damit ist eine stärkere Abgrenzung beider Klassen möglich. Um solche Vertauschungen der Objektklassen Fahrrad und Motorrad zu vermeiden, müsste das kNN mit mehr Bildern trainiert werden, welche die Realität umfänglicher abbilden. Entstehen Fotos lediglich in einem Stadtkern, fehlen gegebenenfalls Bilder von verschiedenen Elektrofahrradmodellen, Mountainbikes oder Rennrädern, die vorwiegend im ländlichen Bereich gefahren werden. Außerdem mangelt es an Bilder von Personen mit seitlichen Gepäcktaschen, die für längere Touren verwendet werden. Das Netz wurde nicht mit einem speziellen Datensatz zur Raderkennung trainiert, sondern mit Beispielen für viele verschiedene Objekte. Dadurch das die Raderkennung kein Fokus des Trainingsdatensatzes war, ist dieser nicht groß und divers genug. Zusammengefasst fehlen besonders Bilder mit Elektrofahrrädern, Räder mit seitlichen Gepäcktaschen, verdeckte Räder oder welche, die im Bild abgeschnitten sind.

Damit eine gute Generalisierung möglich ist, muss die Umwelt ausreichend gut abgebildet werden. Dafür ist ein großer, diverser Datensatz von Rädern nötig. Der Tsinghua-Daimler Cyclist Benchmark ist ein großer Datensatz, in dem 22.161 Radfahrer auf 30.000 Bildern gelabelt wurden [120]. Dabei ist zu beachten, dass nicht das Objekt Fahrrad im Datensatz gelabelt wurde sondern Fahrradfahrer, d.h. Personen auf einem Rad. Es ist somit beim Training mit einem solchen Netz nicht möglich abgestellte Fahrräder ohne Personen in unmittelbarer Nähe zu erkennen. Zudem wurden lediglich Objekte gelabelt, die weniger als 10% verdeckt sind. Dies ist eine starke Einschränkung der Realwelt. Die Bilder stammen von einem, sich durch die Stadt Beijing bewegenden, Fahrzeug. Dadurch wurde ein großer Datensatz mit vielen Bilder von Radfahrern kreiert, allerdings beschränkt auf den Stadtverkehr. In Städten werden in der Regel andere Fahrräder gefahren als für Radtouren verwendet werden. Es ist zu vermuten, dass der Datensatz nur eine spezielle Umgebung im Stadtstraßenverkehr abbildet und nicht die gesamte Realität. Es sollen nicht nur Stadträder, sondern auch Mountainbikes, Rennräder und Elektrofahrräder er-

kannt werden, die vor allem im ländlichen Bereich gefahren werden. Außerdem sollte ein Datensatz nicht nur Fotos, auf denen aktives Radfahren zu sehen ist, enthalten sein, sondern auch Situationen, in denen Räder geschoben werden oder Personen vor Fahrrädern stehen oder sitzen. Diese verschiedenen Situationen sowie die unterschiedlichen Arten von Rädern sind in keinem bekannten Datensatz ausreichend abgebildet. Deswegen wird die Empfehlung gegeben, einen neuen Datensatz speziell für die Raderkennung aufzubauen und dafür verschiedene Umgebungen und Radmodelle zu verwenden. Es ist zu überlegen ob Fotofallen an Radwegen aufgestellt, Bilder von Hobbyfotografen verwendet oder spezielle Fotoshooting, aufgenommen werden können um die Datensatzvoreingenommenheit zu minimieren. Damit könnte ein kNN spezielle für die Raderkennung trainiert werden, welches die Realwelt besser abbilden sollte. Infolgedessen wird eine bessere Generalisierung auf unbekannten Daten erwartet.

Außerdem wurde die Geschlechterkennung anhand des gesamten Körpers ebenfalls auf einem Teil des Praxisdatensatzes der RTG evaluiert. Frauen wurden dabei öfter falsch als Männer vorhergesagt. Mit einer Wahrscheinlichkeit von 67% wurde das Geschlecht richtig vorhergesagt. Dies ist sogar leicht höher als die auf dem Testdatensatz angegebene Wahrscheinlichkeit von 62% das richtige Geschlecht zu erkennen [90]. Allerdings ist zu beachten, dass bei dieser Auswertung ein neutrales Geschlecht verwendet wurde, falls das neuronale Netz unsicher bei der Wahl des Geschlechts ist. In der auf dem Datensatz des RTG verwendeten Implementierung wurde auf das neutrale Geschlecht verzichtet. Vergleicht man die Wahrscheinlichkeiten, mit denen das Geschlecht richtig erkannt wurde, können wir feststellen, dass sehr gut generalisiert wurde, da die Wahrscheinlichkeit auf dem Praxisdatensatz nicht abgenommen, sondern sogar leicht zugenommen hat. Erfreulich ist es keine Unterschiede bei der Geschlechtserkennung zwischen jungen oder älteren Personen gab. Es wurde kein Unterschied festgestellt, dass das Geschlecht von Personen im Stehen besser erkannt wurde als von Personen im Sitzen oder beim Rad fahren. Daraus lässt sich schließen, dass der Datensatz im Hinsicht auf Personenalter ausgeglichen ist und verschiedene Aktivitäten enthält. Dies führte zu einer guten Generalisierung.

Um die Ergebnisse der verschiedenen Implementierungen zusammenzufassen, lässt sich sagen, dass es teilweise große Probleme mit der Generalisierung gibt. Die Personenerkennung und Geschlechtserkennung anhand des gesamten Körpers konnten gut auf neuen Daten vorhersagen. Problemtisch war die Gesichts- und Fahrraderkennung. Die Schwierigkeiten der Gesichts- und Fahrraderkennung waren durch die eingeschränkte Abbildung der Realwelt in den Trainingsdatensätzen zurückzuführen.

Bezüglich der in [51] dargestellten vier Datensatzprobleme, welche in 2.6 vorgestellt wurden, ist festzustellen, dass die Auswahlvoreingenommenheit eines der größten Probleme der Datensätze ist. Dies liegt daran, dass die Fotos meist eine bestimmte Art wie Straßenszenen oder Nahaufnahmen darstellen. Außerdem problematisch ist die Erfassungsvoreingenommenheit. Diese ist allerdings auf verschiedenen Bilddatensätzen ähnlich, da Fotografen meist ihre Bilder auf eine ähnliche Art und Weise aufnehmen, sodass das Foto ästhetisch wirkt. Unterschiede gibt es oft nur dann, wenn Bilder nicht von Menschen sondern oder automatisch mit fester und sich bewegender Kamera fotografiert wurden. Die Benennungsvoreingenommenheit kann nicht beurteilt werden, da keine Informationen vorliegen wie genau semantische Kategorien definiert wurden. Der Negativdatensatz, welcher

als Rest der Welt angesehen wird, war in den getesteten Implementierungen genügend, da nie ein Objekt erkannt wurde, wenn es nicht zu sehen waren.

Zusammenfassend ist festzuhalten, dass die Diversität und Nähe an der Realwelt eines Datensatzes einen großen Einfluss auf die Generalisierbarkeit hat. Die Datensätze zur Gesichts- und Fahrraderkennung waren nicht vielfältig genug, da nur kleine Teile der Realwelt abgebildet wurden. Beide Datensätze werden daher als ungenügend für Praxisanwendungen eingestuft. Hingegen ließen sich die Datensätze der Personenerkennung und Geschlechtserkennung gut generalisieren. Das zur Geschlechtserkennung trainierte CNN konnten auf dem Datensatz der RTG mit sogar höherer Testrichtigkeit das Geschlecht vorhersagen, als auf dem gleichem Datensatz mit dem trainiert wurde. Die Personenerkennung erreicht fast den aktuellen Stand der Technik. Mit 89% liegt er nur 2% unterhalb der 2019 erreichten 91% auf dem VOC 2012 [86].

5 Schlussbetrachtungen

Ziel dieser Masterarbeit war es, die Zielgruppen des Radtourismus zu identifizieren. Diese wurden durch Interviews mit der RTG herausgestellt. Das Ergebnis sind die fünf relevanten Zielgruppen: Junges Paar, Best Ager Paar, Familie, Frauengruppe und gemischtgeschlechtliche Gruppe. Im Bereich der Zielgruppen im Radtourismus war wenig Forschung vorhanden. Eine Einteilung der Zielgruppen nach den Personenmerkmalen Geschlecht und Alter war bisher nicht bekannt und ist ein wichtiger Schritt für effektives Marketing im Radtourismus.

Zur Erkennung der Zielgruppe auf Bildern wurde auf eine bestehende Implementierung aufgebaut, um mit einer Gesichtserkennung das Alter und Geschlecht von Personen vorherzusagen. Die Implementierung wurde mit Regeln erweitert, die zunächst einem Gesicht eine der Personengruppen Frau, Mann, junge Frau, junger Mann, weiblicher Best Ager, männlicher Best Ager, Kind oder Jugendlicher zuordnen. Dies wird für alle Gesichter auf dem Bild durchgeführt und die Anzahl je Personengruppe gezählt. Mit den entwickelten Charakteristika der Zielgruppen als Regeln formuliert, kann damit ein Bild einer Zielgruppe zugewiesen werden. Diese Gesichtserkennung hat ungenügend auf dem Datensatz der RTG vorhergesagt, da dieser zu unterschiedlich vom Datensatz ist auf dem das kNN trainiert wurde.

Anschließend wurde eine Personenerkennung analysiert, um damit die Anzahl an Personen in einem Bild zu ermitteln. Mit einer solchen Implementierung ist keine genaue Zielgruppenzuordnung möglich, da keine Informationen über Geschlecht und Alter vorliegen. Es können allerdings Aussagen darüber getroffen werden, ob eine Personengruppe auf einem Foto dargestellt ist. Zu einer Gruppe gehören nach Meinung der Radtourismusexperten mindestens drei Personen. Die Personenerkennung hat exzellente Ergebnisse erreicht und konnte mit dem aktuellen Stand der Technik mithalten. Gruppenbilder wurden mit einer Wahrscheinlichkeit von 97% richtig erkannt.

Da meist Ganzkörperaufnahmen im Datensatz der RTG sind, wurde eine Geschlechtserkennung anhand des gesamten Körpers durchgeführt. Dafür wurden zunächst mit OpenPose die wichtigsten Körperpunkte ermittelt und darauf aufbauend ein realistisches 3D-Körpermodell erstellt, welches das Gesicht und die Hände detailliert modelliert. Mit diesem Verfahren konnte bei 138 von 207 Personen das Geschlecht richtig vorhergesagt werden. Frauen wurde dabei häufig falsch als Männer vorhergesagt.

Ein weiteres Ziel ist die Beurteilung der Praxistauglichkeit bestehender Implementierungen zur Fahrraderkennung. Mit der verwendeten Implementierung wurden 38% der Räder richtig erkennt. In einigen Fällen wurden Räder mit Motorrädern verwechselt oder nicht erkannt. Aus diesem Grund ist die Raderkennung im konkreten Kontext anwendbar, allerdings für andere Praxisbereiche aufgrund der niedrigen Trefferquote weniger geeignet.

Die Datensatzvoreingenommenheit ist ein großes Problem, welches die Generalisierung auf unbekannten Datensätzen erschwert. Diese Thematik ist in der Literatur bekannt. Beispielsweise wurde in [51] gezeigt, dass Training und Testen auf verschiedenen Datensätzen zu niedrigerer Vorhersagegüte führt. Aus diesem Grund wurde die Forschungsfrage beantwortet, ob vorhandene Datensätze ausreichend sind, um eine Generalisierung der Objekterkennung auf Praxisdaten zu ermöglichen. Dabei konnte festgestellt werden, dass es an Datensätzen zur Bestimmung des Alters und Geschlechtes mangelt. Der Datensatz zur Fahrraderkennung ist ebenfalls nicht divers genug. Hingegen konnte die Geschlechtserkennung anhand des gesamten Körpers gute Ergebnisse auf dem Datensatz der RTG liefern, im Vergleich zum Testen auf dem selben Datensatz mit dem trainiert wurde. Daraus konnte gefolgert werden, dass mit dem Trainingsdatensatz eine gute Generalisierung möglich ist und somit die Realität im Trainingsdatensatz nah genug abgebildet wird. Die Personenerkennung lieferte ebenfalls hervorragende Ergebnisse, welche auf einen realitätsnahen Datensatz schließen lassen.

Um eine höhere Erkennungsgenauigkeit von Rädern zu erhalten, wird empfohlen, einen größeren Datensatz zur Fahrraderkennung aufzubauen. Dadurch sollen Elektroräder, Räder mit Gepäck und verdeckte Räder besser erkannt werden und weniger Verwechslungen mit Motorrädern passieren. Weiterer Forschungsbedarf besteht bei der Geschlechtserkennung anhand des gesamten Körpers. Hier konnte nicht geklärt werden, warum Frauen des öfteren falsch als Männer vorhergesagt wurden. Außerdem sind dem Autor der Masterarbeit lediglich Implementierung bekannt, die das Alter von Personen anhand des Gesichtes bestimmen. Dies ist im Anwendungskontext mit vorwiegend Ganzkörperbildern nicht zielführend gewesen. Jedoch ist das Alter von Personen nötig, um die Zielgruppe richtig zuordnen zu können. Eine offene Forschungsfrage ist demnach, ob das Alter von Personen ebenfalls anhand des gesamten Körpers erkannt werden kann. Alternativ kann untersucht werden, ob die Gesichtserkennung trainiert auf einem Datensatz mit Ganzkörperaufnahmen bessere Ergebnisse liefert. Ebenfalls könnte das Problem, dass Personen nicht direkt in die Kamera sehen, durch eine Frontalisierung des Gesichtes gelöst werden [121]. Gegenbebenfalls finden andere Forschende weitere Möglichkeiten das Geschlecht und Alter von Personen präzise zu bestimmen und können damit die Ergebnisse der Zielgruppenzuordnung nutzen. Mit der Arbeit soll Wissenschaftlern das Problem der Datensatzvoreigenommenheit als eine Ursache für schlechte Generalisierung auf neuen Datensätzen vor Augen geführt und ins Gedächtnis gerufen werden. Dadurch, dass die Probleme der Raderkennung in einem Praxisdatensatz aufgezeigt wurden, konnten klare Verbesserungspotentiale neuer Datensätze herausgestellt werden.

6 Anhang

Python-Code zur Geschlechts- und Alterserkennung mit der Erweiterung zur Personenklassen- und Zielgruppenerkennung.

```
#Geschlechts und Alterserkennung von Mahesh Sawant
#erweitert mit Personenklassen und Zielgruppen

import cv2
import math
import argparse

def highlightFace(net, frame, conf_threshold=0.7):
    frameOpencvDnn=frame.copy()
    frameHeight=frameOpencvDnn.shape[0]
    frameWidth=frameOpencvDnn.shape[1]
    blob=cv2.dnn.blobFromImage(frameOpencvDnn, 1.0, (300, 300), [104, 117, 123], True, False)

    net.setInput(blob)
    detections=net.forward()
    faceBoxes=[]
    for i in range(detections.shape[2]):
        confidence=detections[0,0,i,2]
        if confidence>conf_threshold:
            x1=int(detections[0,0,i,3]*frameWidth)
            y1=int(detections[0,0,i,4]*frameHeight)
            x2=int(detections[0,0,i,5]*frameWidth)
            y2=int(detections[0,0,i,6]*frameHeight)
            faceBoxes.append([x1,y1,x2,y2])
            cv2.rectangle(frameOpencvDnn, (x1,y1), (x2,y2), (0,255,0), int(round(frameHeight/150)), 8)
    return frameOpencvDnn,faceBoxes


parser=argparse.ArgumentParser()
parser.add_argument('--image')

args=parser.parse_args()

faceProto="opencv_face_detector.pbtxt"
faceModel="opencv_face_detector_uint8.pb"
ageProto="age_deploy.prototxt"
ageModel="age_net.caffemodel"
genderProto="gender_deploy.prototxt"
genderModel="gender_net.caffemodel"

MODEL_MEAN_VALUES=(78.4263377603, 87.7689143744, 114.895847746)
```

```
ageList=['(0-2)', '(4-6)', '(8-12)', '(15-20)', '(25-32)', '(38-43)', '(48-53)', '(60-100)']
genderList=['Male','Female']

faceNet=cv2.dnn.readNet(faceModel,faceProto)
ageNet=cv2.dnn.readNet(ageModel,ageProto)
genderNet=cv2.dnn.readNet(genderModel,genderProto)

video=cv2.VideoCapture(args.image if args.image else 0)
padding=20

hasFrame,frame=video.read()

resultImg,faceBoxes=highlightFace(faceNet,frame)

# Initialisierung der Anzahl der Personenklassen
countwoman=0
countman=0
countchild=0
countBestAger=0
countteenager=0
countwomanyoung=0
countmanyoung=0
countBestAgerfemale=0
countBestAgermale=0

if not faceBoxes:
    print("No face detected")

for faceBox in faceBoxes:
    face=frame[max(0,faceBox[1]-padding):
               min(faceBox[3]+padding,frame.shape[0]-1),max(0,faceBox[0]-padding)
               :min(faceBox[2]+padding, frame.shape[1]-1)]

    blob=cv2.dnn.blobFromImage(face, 1.0, (227,227), MODEL_MEAN_VALUES, swapRB=False)
    genderNet.setInput(blob)
    genderPreds=genderNet.forward()
    gender=genderList[genderPreds[0].argmax()]
    print('Gender:', gender)

    ageNet.setInput(blob)
  agePreds=ageNet.forward()
    age=ageList[agePreds[0].argmax()]
    print('Age:', age[1:-1],'years')

# Ordnet dem erkannten Gesicht mithilfe des
# Geschlechts und Alters die Personenklasse zu
# und die Anzahl der Personen in dieser Klasse wird gezaehlt
    if gender == 'Female' and (age == '(25-32)' or age == '(38-43)' or age == '(48-53)' and age == '(60-100)'):
        countwoman=countwoman+1

    if gender == 'Male' and (age == '(25-32)' or age == '(38-43)' or age ==
```

```
        '(48-53)' and age == '(60-100)'):
1092            countman=countman+1

1094        if age == '(0-2)' or age == '(4-6)' or age == '(8-12)':
                print('Kind')
1096            countchild=countchild+1

1098        if age == '(60-100)' or age == '(48-53)':
                countBestAger=countBestAger+1
1100
            if gender =='Female' and (age == '(60-100)' or age == '(48-53)'):
1102            print('weiblicher Best Ager')
                countBestAgerfemale=countBestAgerfemale+1
1104
            if gender =='Male' and (age == '(60-100)' or age == '(48-53)'):
1106            print('maennlicher Best Ager')
                countBestAgermale=countBestAgermale+1
1108
            if age == '(15-20)':
1110            print('Jugendlicher')
                countteenager=countteenager+1
1112
            if gender == 'Female' and age == '(25-32)':
1114            print('Junge Frau')
                countwomanyoung=countwomanyoung+1
1116
            if gender == 'Male' and age == '(25-32)':
1118            print('Junger Mann')
                countmanyoung=countmanyoung+1
1120
            label="{},{}".format(gender,age)
1122        cv2.putText(resultImg, label, (faceBox[0], faceBox[1]-10), cv2.
        FONT_HERSHEY_SIMPLEX, 0.8, (0,255,255), 2, cv2.LINE_AA)
            cv2.imshow("Detecting age and gender", resultImg)
1124        cv2.waitKey(1)

1126    # Ausgabe der Personenklassen und der Zielgruppe
        if  countwoman >= 1:
1128        print(countwoman, 'Frau/en')

1130    if countman >= 1:
            print(countman, 'Mann/Maenner')
1132
        if countchild >= 1:
1134        print(countchild, 'Kind/er')

1136    if countteenager >= 1:
            print(countteenager,'Jugendliche/r')
1138
        if countBestAger >= 1:
1140        print(countBestAger, 'BestAger')

1142    if countBestAgerfemale >= 1:
            print(countBestAgerfemale, 'weibliche(r) BestAger')
1144
```

```
if countBestAgermale >= 1:
    print(countBestAgermale, 'maennliche(r) BestAger')

if (countwomanyoung >= 1 or countmanyoung >=1 or countBestAger >= 1) and (countchild >= 1 or countteenager >= 1):
    print('Familie')

if countwomanyoung == 1 and countmanyoung == 1 and countchild == 0 and countBestAger == 0 and countteenager == 0:
    print('junges Paar')

if countwoman >= 3 and countman == 0 and countBestAger == 0 and countchild == 0 and countteenager == 0:
    print('Frauengruppe')

if countman >= 3 and countwoman == 0 and countBestAger == 0 and countchild == 0 and countteenager == 0:
    print('Maennergruppe')

if countBestAgerfemale == 1 and countBestAgermale ==1 and countman == 0 and countwoman == 0 and countchild == 0 and countteenager == 0:
    print('Best Ager Paar')

if countmanyoung+countwomanyoung+countBestAger >= 3 and (countmanyoung >= 1 or countBestAgermale >= 1) and (countwomanyoung >=1 or countBestAgerfemale >= 1) and countchild == 0 and countteenager == 0:
    print('Gemischte Gruppe')
```

codeGesichtserkennung.py

Literatur

[1] Zhengxia Zou u. a. „Object detection in 20 years: A survey“. In: *arXiv preprint arXiv:1905.05055* (2019).

[2] Paul Viola und Michael Jones. „Rapid object detection using a boosted cascade of simple features“. In: *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*. Bd. 1. IEEE. 2001, S. I–I.

[3] David J Robertson u. a. „Face recognition by metropolitan police super-recognisers“. In: *PloS one* 11.2 (2016), e0150036.

[4] Markus Vincze, Michael Zillich und Johann Prankl. „Roboter lernen mit Gegenständen umzugehen: neue Entwicklungen und Chancen“. In: *e & i Elektrotechnik und Informationstechnik* 134.6 (2017), S. 304–311.

[5] A. Chayeb u. a. „HOG based multi-object detection for urban navigation“. In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. 2014, S. 2962–2967.

[6] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020.

[7] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[8] Oludare Isaac Abiodun u. a. „State-of-the-art in artificial neural network applications: A survey“. In: *Heliyon* 4.11 (2018), e00938.

[9] Raúl Rojas. *Neural networks: a systematic introduction*. Springer Science & Business Media, 2013.

[10] Rudolf Kruse u. a. *Computational intelligence*. Springer, 2011.

[11] Gopinath Rebala, Ajay Ravi und Sanjay Churiwala. *An introduction to machine learning*. Springer, 2019.

[12] Ian Goodfellow, Yoshua Bengio und Aaron Courville. *Deep learning*. MIT press, 2016.

[13] Frieder Stolzenburg, Olivia Michael und Oliver Obst. „The Power of Linear Recurrent Neural Networks-Predictive Neural Networks“. In: *arXiv preprint arXiv:1802.03308* (2018).

[14] David H Hubel und Torsten N Wiesel. „Receptive fields, binocular interaction and functional architecture in the cat's visual cortex“. In: *The Journal of physiology* 160.1 (1962), S. 106.

[15] Kunihiko Fukushima und Sei Miyake. „Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition“. In: *Competition and cooperation in neural nets*. Springer, 1982, S. 267–285.

[16] David Marr und Ellen Hildreth. „Theory of edge detection“. In: *Proceedings of the Royal Society of London. Series B. Biological Sciences* 207.1167 (1980), S. 187–217.

[17] Franz Pauer und Florian Stampfer. „Was ist ein Skalarprodukt und wozu wird es verwendet?“ In: *Schriftenreihe zur Didaktik der Österreichischen Mathematischen Gesellschaft* 49 (2016), S. 100–109.

[18] Wei Wang u. a. „Development of convolutional neural network and its application in image classification: a survey“. In: *Optical Engineering* 58.4 (2019), S. 040901.

[19] Yann LeCun, Koray Kavukcuoglu und Clément Farabet. „Convolutional networks and applications in vision“. In: *Proceedings of 2010 IEEE international symposium on circuits and systems*. IEEE. 2010, S. 253–256.

[20] O Rebecca Vincent, Olusegun Folorunso u. a. „A descriptive algorithm for sobel image edge detection“. In: *Proceedings of Informing Science & IT Education Conference (InSITE)*. Bd. 40. Informing Science Institute California. 2009, S. 97–107.

[21] Yann LeCun u. a. „Backpropagation applied to handwritten zip code recognition“. In: *Neural computation* 1.4 (1989), S. 541–551.

[22] Alex Krizhevsky, Ilya Sutskever und Geoffrey E Hinton. „Imagenet classification with deep convolutional neural networks“. In: *Advances in neural information processing systems*. 2012, S. 1097–1105.

[23] Kazuyuki Hara, Daisuke Saito und Hayaru Shouno. „Analysis of function of rectified linear unit used in deep learning“. In: *2015 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2015, S. 1–8.

[24] Sebastian Ruder. „An overview of gradient descent optimization algorithms“. In: *arXiv preprint arXiv:1609.04747* (2016).

[25] Léon Bottou. „Stochastic gradient descent tricks“. In: *Neural networks: Tricks of the trade*. Springer, 2012, S. 421–436.

[26] Geoffrey E Hinton u. a. „Improving neural networks by preventing co-adaptation of feature detectors“. In: *arXiv preprint arXiv:1207.0580* (2012).

[27] Nitish Srivastava u. a. „Dropout: a simple way to prevent neural networks from overfitting“. In: *The journal of machine learning research* 15.1 (2014), S. 1929–1958.

[28] Pierre Baldi und Peter Sadowski. „The dropout learning algorithm“. In: *Artificial intelligence* 210 (2014), S. 78–122.

[29] Tom Fawcett. „An introduction to ROC analysis“. In: *Pattern recognition letters* 27.8 (2006), S. 861–874.

[30] David Martin Powers. „Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation“. In: (2011).

[31] Neha Sharma, Vibhor Jain und Anju Mishra. „An analysis of convolutional neural networks for image classification“. In: *Procedia computer science* 132 (2018), S. 377–384.

[32] Davide Chicco und Giuseppe Jurman. „The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation“. In: *BMC genomics* 21.1 (2020), S. 6.

[33] Davide Chicco. „Ten quick tips for machine learning in computational biology“. In: *BioData mining* 10.1 (2017), S. 35.

[34] Robert M Haralick und Linda G Shapiro. „Image segmentation techniques“. In: *Computer vision, graphics, and image processing* 29.1 (1985), S. 100–132.

[35] Navneet Dalal und Bill Triggs. „Histograms of oriented gradients for human detection“. In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Bd. 1. IEEE. 2005, S. 886–893.

[36] Edgar Osuna, Robert Freund und Federico Girosit. „Training support vector machines: an application to face detection“. In: *Proceedings of IEEE computer society conference on computer vision and pattern recognition*. IEEE. 1997, S. 130–136.

[37] Paul Viola, Michael Jones u. a. „Robust real-time object detection“. In: *International journal of computer vision* 4.34-47 (2001), S. 4.

[38] Ross Girshick u. a. „Rich feature hierarchies for accurate object detection and semantic segmentation“. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, S. 580–587.

[39] Jasper RR Uijlings u. a. „Selective search for object recognition“. In: *International journal of computer vision* 104.2 (2013), S. 154–171.

[40] Johan AK Suykens und Joos Vandewalle. „Least squares support vector machine classifiers“. In: *Neural processing letters* 9.3 (1999), S. 293–300.

[41] Bernhard Scholkopf und Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. Adaptive Computation und Machine Learning series, 2018.

[42] Ross Girshick. „Fast r-cnn“. In: *Proceedings of the IEEE international conference on computer vision*. 2015, S. 1440–1448.

[43] Shaoqing Ren u. a. „Faster r-cnn: Towards real-time object detection with region proposal networks“. In: *Advances in neural information processing systems*. 2015, S. 91–99.

[44] Jifeng Dai u. a. „R-fcn: Object detection via region-based fully convolutional networks“. In: *Advances in neural information processing systems*. 2016, S. 379–387.

[45] Jonathan Long, Evan Shelhamer und Trevor Darrell. „Fully convolutional networks for semantic segmentation“. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, S. 3431–3440.

[46] Wei Liu u. a. „Ssd: Single shot multibox detector“. In: *European conference on computer vision*. Springer. 2016, S. 21–37.

[47] Joseph Redmon u. a. „You only look once: Unified, real-time object detection“. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, S. 779–788.

[48] Alexander Neubeck und Luc Van Gool. „Efficient non-maximum suppression“. In: *18th International Conference on Pattern Recognition (ICPR'06)*. Bd. 3. IEEE. 2006, S. 850–855.

[49] Volker Blanz, Michael J Tarr und Heinrich H Bülthoff. „What object attributes determine canonical views?“ In: *Perception* 28.5 (1999), S. 575–599.

[50] Tsung-Yi Lin u. a. „Microsoft coco: Common objects in context“. In: *European conference on computer vision*. Springer. 2014, S. 740–755.

[51] Antonio Torralba und Alexei A Efros. „Unbiased look at dataset bias“. In: *CVPR 2011*. IEEE. 2011, S. 1521–1528.

[52] Jianxiong Xiao u. a. „Sun database: Large-scale scene recognition from abbey to zoo“. In: *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE. 2010, S. 3485–3492.

[53] Bryan C Russell u. a. „LabelMe: a database and web-based tool for image annotation“. In: *International journal of computer vision* 77.1-3 (2008), S. 157–173.

[54] Mark Everingham u. a. „The pascal visual object classes (voc) challenge“. In: *International journal of computer vision* 88.2 (2010), S. 303–338.

[55] Jia Deng u. a. „Imagenet: A large-scale hierarchical image database“. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, S. 248–255.

[56] Li Fei-Fei, Rob Fergus und Pietro Perona. „Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories“. In: *2004 conference on computer vision and pattern recognition workshop*. IEEE. 2004, S. 178–178.

[57] John Winn, Antonio Criminisi und Thomas Minka. „Object categorization by learned universal visual dictionary“. In: *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. Bd. 2. IEEE. 2005, S. 1800–1807.

[58] Tatiana Tommasi u. a. „A deeper look at dataset bias“. In: *Domain adaptation in computer vision applications*. Springer, 2017, S. 37–55.

[59] Aditya Khosla u. a. „Undoing the damage of dataset bias“. In: *European Conference on Computer Vision*. Springer. 2012, S. 158–171.

[60] Jeff Donahue u. a. „Decaf: A deep convolutional activation feature for generic visual recognition“. In: *International conference on machine learning*. 2014, S. 647–655.

[61] Tatiana Tommasi und Tinne Tuytelaars. „A testbed for cross-dataset analysis“. In: *European Conference on Computer Vision*. Springer. 2014, S. 18–31.

[62] Abhinav Nagpal und Goldie Gabrani. „Python for data analytics, scientific and technical applications“. In: *2019 Amity international conference on artificial intelligence (AICAI)*. IEEE. 2019, S. 140–145.

[63] Gary Bradski und Adrian Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. Ö'Reilly Media, Inc.", 2008.

[64] Ivan Culjak u. a. „A brief introduction to OpenCV“. In: *2012 proceedings of the 35th international convention MIPRO*. IEEE. 2012, S. 1725–1730.

[65] Martın Abadi u. a. „Tensorflow: Large-scale machine learning on heterogeneous distributed systems“. In: *arXiv preprint arXiv:1603.04467* (2016).

[66] Yangqing Jia u. a. „Caffe: Convolutional architecture for fast feature embedding“. In: *Proceedings of the 22nd ACM international conference on Multimedia*. 2014, S. 675–678.

[67] Myroslav Komar u. a. „Deep neural network for image recognition based on the caffe framework“. In: *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*. IEEE. 2018, S. 102–106.

[68] Emine Cengil, Ahmet Çınar und Erdal Özbay. „Image classification with caffe deep learning framework“. In: *2017 International Conference on Computer Science and Engineering (UBMK)*. IEEE. 2017, S. 440–444.

[69] John Nickolls und William J Dally. „The GPU computing era“. In: *IEEE micro* 30.2 (2010), S. 56–69.

[70] John D Owens u. a. „GPU computing“. In: *Proceedings of the IEEE* 96.5 (2008), S. 879–899.

[71] Danilo De Donno u. a. „Introduction to GPU computing and CUDA programming: A case study on FDTD [EM programmer's notebook]“. In: *IEEE Antennas and Propagation Magazine* 52.3 (2010), S. 116–122.

[72] Allgemeiner Deutscher Fahrrad-Club-ADFC u. a. „ADFC-Radreiseanalyse. 2020.“ In: (2020).

[73] Allgemeiner Deutscher Fahrrad-Club-ADFC und Bremen Bundesverband. „ADFC-Radreiseanalyse 2016. 17. bundesweite Erhebung zum fahrradtouristischen Markt. Internationale Tourismus-Börse Berlin, ITB.“ In: (2016).

[74] Bonn Deutscher Tourismusverband. „Grundlagenuntersuchung Fahrradtourismus in Deutschland. Langfassung.“ In: (2009).

[75] Gil Levi und Tal Hassner. „Age and gender classification using convolutional neural networks“. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2015, S. 34–42.

[76] Sebastian Lapuschkin u. a. „Understanding and comparing deep neural networks for age and gender classification“. In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2017, S. 1629–1638.

[77] Eran Eidinger, Roee Enbar und Tal Hassner. „Age and gender estimation of unfiltered faces“. In: *IEEE Transactions on Information Forensics and Security* 9.12 (2014), S. 2170–2179.

[78] Kitsuchart Pasupa und Wisuwat Sunhem. „A comparison between shallow and deep architecture classifiers on small dataset“. In: *2016 8th International Conference on Information Technology and Electrical Engineering (ICITEE)*. IEEE. 2016, S. 1–6.

[79] Olatunbosun Agbo-Ajala und Serestina Viriri. „Deeply Learned Classifiers for Age and Gender Predictions of Unfiltered Faces“. In: *The Scientific World Journal* 2020 (2020).

[80] Rasmus Rothe, Radu Timofte und Luc Van Gool. „Deep expectation of real and apparent age from a single image without facial landmarks“. In: *International Journal of Computer Vision* 126.2-4 (2018), S. 144–157.

[81] G Bingham u. a. „MORPH-II: Inconsistencies and Cleaning“. In: *University of North Carolina Wilmington NSF REU* (2017).

[82] Diederik P Kingma und Jimmy Ba. „Adam: A method for stochastic optimization“. In: *arXiv preprint arXiv:1412.6980* (2014).

[83] Andrew G Howard u. a. „Mobilenets: Efficient convolutional neural networks for mobile vision applications“. In: *arXiv preprint arXiv:1704.04861* (2017).

[84] Mark Everingham u. a. „The pascal visual object classes challenge: A retrospective“. In: *International journal of computer vision* 111.1 (2015), S. 98–136.

[85] Mark Everingham und John Winn. „The pascal visual object classes challenge 2007 (voc2007) development kit“. In: *University of Leeds, Tech. Rep* (2007).

[86] Zhong-Qiu Zhao u. a. „Object detection with deep learning: A review“. In: *IEEE transactions on neural networks and learning systems* 30.11 (2019), S. 3212–3232.

[87] Mark Everingham und John Winn. „The pascal visual object classes challenge 2012 (voc2012) development kit“. In: *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep* 8 (2011).

[88] Zhe Cao u. a. „OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields“. In: *arXiv preprint arXiv:1812.08008* (2018).

[89] Matthew Loper u. a. „SMPL: A skinned multi-person linear model“. In: *ACM transactions on graphics (TOG)* 34.6 (2015), S. 1–16.

[90] Georgios Pavlakos u. a. „Expressive body capture: 3d hands, face, and body from a single image“. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, S. 10975–10985.

[91] Nicola D'Apuzzo und Harvey Mitchell. „Medical applications“. In: *Advances in photogrammetry, remote sensing and spatial Information sciences: 2008 ISPRS congress book*. Taylor & Francis Group: London, UK. 2008, S. 425–438.

[92] Shian-Ru Ke u. a. „Real-time 3D human pose estimation from monocular view with applications to event detection and video gaming“. In: *2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance*. IEEE. 2010, S. 489–496.

[93] Angelos Barmpoutis. „Tensor body: Real-time reconstruction of the human body and avatar synthesis from RGB-D“. In: *IEEE transactions on cybernetics* 43.5 (2013), S. 1347–1356.

[94] Karteek Alahari u. a. „Pose estimation and segmentation of people in 3D movies“. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, S. 2112–2119.

[95] Sameh Khamis u. a. „Learning an efficient model of hand shape variation from depth images“. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, S. 2540–2548.

[96] Markus Oberweger, Paul Wohlhart und Vincent Lepetit. „Training a feedback loop for hand pose estimation“. In: *Proceedings of the IEEE international conference on computer vision*. 2015, S. 3316–3324.

[97] Anastasia Tkach, Mark Pauly und Andrea Tagliasacchi. „Sphere-meshes for real-time hand modeling and tracking“. In: *ACM Transactions on Graphics (ToG)* 35.6 (2016), S. 1–11.

[98] Tianye Li u. a. „Learning a model of facial shape and expression from 4D scans.“ In: *ACM Trans. Graph.* 36.6 (2017), S. 194–1.

[99] Alan Brunton u. a. „Review of statistical shape spaces for 3D data with comparative analysis for human faces“. In: *Computer Vision and Image Understanding* 128 (2014), S. 1–17.

[100] Hanbyul Joo, Tomas Simon und Yaser Sheikh. „Total capture: A 3d deformation model for tracking faces, hands, and bodies“. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, S. 8320–8329.

[101] Martin Komaritzan und Mario Botsch. „Projective skinning“. In: *Proceedings of the ACM on Computer Graphics and Interactive Techniques* 1.1 (2018), S. 1–19.

[102] Junjun Pan u. a. „Automatic skinning and weight retargeting of articulated characters using extended position-based dynamics“. In: *The Visual Computer* 34.10 (2018), S. 1285–1297.

[103] Nadia Magnenat-Thalmann, Richard Laperrire und Daniel Thalmann. „Joint-dependent local deformations for hand animation and object grasping“. In: *In Proceedings on Graphics interface'88*. Citeseer. 1988.

[104] Hao Yin und Ramakarishnan Mukundan. „Improved Vertex Skinning Algorithm Based On Dual Quaternions“. In: *2019 IEEE 21st International Workshop on Multimedia Signal Processing (MMSP)*. IEEE. 2019, S. 1–6.

[105] Steven Collins u. a. „Skinning with Dual Quaternions“. In: *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*. ACM. 2007.

[106] YoungBeom Kim und JungHyun Han. „Bulging-free dual quaternion skinning“. In: *Computer Animation and Virtual Worlds* 25.3-4 (2014), S. 321–329.

[107] Doug L James und Christopher D Twigg. „Skinning mesh animations“. In: *ACM Transactions on Graphics (TOG)* 24.3 (2005), S. 399–407.

[108] Alex Mohr und Michael Gleicher. „Building efficient, accurate character skins from examples“. In: *ACM Transactions on Graphics (TOG)* 22.3 (2003), S. 562–568.

[109] Karen Simonyan und Andrew Zisserman. „Very deep convolutional networks for large-scale image recognition“. In: *arXiv preprint arXiv:1409.1556* (2014).

[110] Zhe Cao u. a. „Realtime multi-person 2d pose estimation using part affinity fields“. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, S. 7291–7299.

[111] Tianye Li u. a. „Learning a model of facial shape and expression from 4D scans“. In: *ACM Transactions on Graphics, (Proc. SIGGRAPH Asia)* 36.6 (2017), 194:1–194:17. URL: https://doi.org/10.1145/3130800.3130813.

[112] Javier Romero, Dimitrios Tzionas und Michael J Black. „Embodied hands: Modeling and capturing hands and bodies together“. In: *ACM Transactions on Graphics (ToG)* 36.6 (2017), S. 245.

[113] Kathleen M Robinette u. a. *Civilian american and european surface anthropometry resource (caesar), final report. volume 1. summary*. Techn. Ber. SYTRONICS INC DAYTON OH, 2002.

[114] Darren Cosker, Eva Krumhuber und Adrian Hilton. „A FACS valid 3D dynamic action unit database with applications to 3D dynamic morphable facial modeling“. In: *2011 international conference on computer vision*. IEEE. 2011, S. 2296–2303.

[115] Federica Bogo u. a. „Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image“. In: *European Conference on Computer Vision*. Springer. 2016, S. 561–578.

[116] Kaiming He u. a. „Deep residual learning for image recognition“. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, S. 770–778.

[117] Andrew L Maas, Awni Y Hannun und Andrew Y Ng. „Rectifier nonlinearities improve neural network acoustic models“. In: *Proc. icml*. Bd. 30. 1. 2013, S. 3.

[118] Mykhaylo Andriluka u. a. „2d human pose estimation: New benchmark and state of the art analysis“. In: *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition*. 2014, S. 3686–3693.

[119] Sam Johnson und Mark Everingham. „Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation.“ In: *bmvc*. Bd. 2. 4. Citeseer. 2010, S. 5.

[120] Xiaofei Li u. a. „A new benchmark for vision-based cyclist detection“. In: *2016 IEEE Intelligent Vehicles Symposium (IV)*. 2016, S. 1028–1033. DOI: 10.1109/IVS.2016.7535515.

[121] Tal Hassner u. a. „Effective face frontalization in unconstrained images“. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, S. 4295–4304.

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Arbeit bisher bei keiner anderen Prüfungsbehörde eingereicht, sie selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt sowie Zitate kenntlich gemacht habe.

Bayreuth, 21. Februar 2022

Die Objekterkennung als wichtiger Teil der Bildanalyse hat schon seit einigen Jahren eine große Bedeutung und bildet einen komplexen Forschungszweig, in dem viele interessante Anwendungen möglich sind.

