HAW HAMBURG

BACHELORARBEIT

Super-Resolution in der Emotionserkennung: Effekte auflösungsoptimierter Bilder auf die Klassifikation von Gesichtsausdrücken durch Deep Neural Networks

vorgelegt am 7. Mai 2024

Leander Wernst

Erstprüferin: Prof. Dr. Larissa Putzar
Zweitprüfer: Thorben Ortmann

HOCHSCHULE FÜR ANGEWANDTE WISSENSCHAFTEN HAMBURG
Department Medientechnik
Finkenau 35
22081 Hamburg

Zusammenfassung

Deep Neural Networks (DNNs) benötigen zur effektiven Aufgabenbewältigung der Emotionserkennung anhand von Gesichtsausdrücken (Facial Expression Recognition [FER]) eine große Menge hochqualitativer Bilddaten. Dabei geht diese Arbeit davon aus, dass eine hohe Datenqualität mit einem hohen Detailgrad durch hohe Auflösungen einhergeht. Bestehende oder in unterschiedlichen Anwendungsszenarien generierte Daten können diesen Anforderungen oft nicht gerecht werden. Super-Resolution (SR) ist die Rekonstruktion hochaufgelöster Bilder aus niedrig aufgelösten und stellt einen vielversprechenden Ansatz zur Leistungssteigerung von FER durch Bild- und Auflösungsoptimierung dar, der in dieser Arbeit untersucht wird. Das Ziel der Analyse ist zu klären, welchen Effekt die Bildauflösung auf FER mit DNNs hat und inwiefern SR die Klassifikationsleistung durch Verbesserung degradierter Bilder beeinflussen kann. Hierzu werden mit EfficientNet-B0 und den FER-Datensätzen AffectNet und RaFD zunächst Versuche in unterschiedlichen durch Downsampling generierten Auflösungen durchgeführt. Anschließend werden die Bilder durch bikubische Interpolation sowie die SR-Verfahren HAT und GFPGAN upgesampelt und erneut getestet. Die höchste Accuracy des Auflösungsvergleichs wird jeweils bei 224 × 224 Pixeln mit 60,34 % (AffectNet, native Auflösung) bzw. 93,4 % (RaFD, downgesampelt) erzielt. Eine verbesserte Leistung durch das Upsampling niedriger aufgelöster Bilder durch SR gegenüber bikubischer Interpolation wird abgesehen von einem speziellen Fall nicht festgestellt. Die Ergebnisse deuten darauf hin, dass in SR Potenzial zur Leistungssteigerung von FER steckt, aber eine bessere Anpassung der SR-Modelle an die FER-Aufgabe vonnöten ist, um stärkere Verbesserungen zu bewirken.

Abstract

Deep neural networks (DNNs) require a large amount of high-quality image data to effectively handle the task of facial expression recognition (FER). This work assumes that high data quality goes hand in hand with a high level of detail due to high resolutions. Existing data or data generated in various application scenarios often cannot meet these requirements. Super-Resolution (SR) is the reconstruction of high-resolution images from low-resolution images and represents a promising approach for increasing the performance of FER through image and resolution optimization, which is investigated in this thesis. The aim of the analysis is to study the effect of image resolution on FER with DNNs and to what extent SR can influence classification performance by improving degraded images. For this purpose, experiments with EfficientNet-B0 and the FER datasets AffectNet and RaFD are first performed at different resolutions generated by downsampling. The images are then upsampled using bicubic interpolation and the SR methods HAT and GFPGAN and tested again. The highest accuracy of the resolution comparison is achieved at 224 × 224 pixels with 60.34 % (AffectNet, native resolution) and 93.4 % (RaFD, downsampled). An improved performance by upsampling lower resolution images by SR compared to bicubic interpolation is not observed except for one specific case. The results indicate that SR has the potential to improve the performance of FER, but that SR models need to be adapted more closely to the FER task in order to achieve stronger improvements.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

AD	Action Descriptor
AI	Artificial Intelligence
AU	Action Unit
AUC	Area Under the Curve
CNN	Convolutional Neural Network
DNN	Deep Neural Network
FACS	Facial Action Coding System
FER	Facial Expression Recognition
FSR	Feature Super-Resolution
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
HQ	high-quality
HR	high-resolution
IQA	Image Quality Assessment
ISR	Image Super-Resolution
KI	Künstliche Intelligenz
KNN	Künstliches neuronales Netzwerk
LPIPS	Learned Perceptual Image Patch Similarity
LQ	low-quality
LR	low-resolution
MCC	Matthews Correlation Coefficient
ML	Machine Learning
MSE	Mean Squared Error
PSNR	Peak Signal-to-Noise Ratio
RNN	Recurrent Neural Network
RMSE	Root Mean Squared Error
SOTA	State of the Art
SSIM	Structure Similarity Index Measure
SR	Super-Resolution

1 Einleitung

Das Erkennen menschlicher Emotionen in Gesichtsausdrücken, das in der englischen Literatur unter den Überbegriff *Facial Expression Recognition* (FER) fällt und synonym verwendet wird [83], [84], [86], [108], ist ein faszinierendes und sich stetig weiterentwickelndes Forschungsgebiet [84], da es spannende Anwendungsfälle in unterschiedlichen Feldern mit sich bringt. Diese liegen bspw. im Gesundheitswesen (Beobachtung von Patienten, Identifizieren von Erkrankungen), dem Personalwesen (Unterstützung von Personalvermittlern), im Bildungssektor (adaptive Anpassung des Lernwegs), in der öffentlichen Sicherheit (präventive Überwachung, Videoanalyse) [44] oder der Unterhaltungsindustrie (*Affective Gaming*) [76] sowie in vielen mehr.
Durch die immer besser werdenden Verarbeitungsmöglichkeiten von Daten, bspw. mittels schnellerer Grafikprozessoren (engl. *Graphics Processing Units* [GPUs]) und dem Aufkommen neuer, gut durchdachter Netzwerkarchitekturen, konnten Deep-Learning-Algorithmen die Ergebnisse von bis dahin traditionellen Methoden deutlich übertreffen. So wird automatische Emotionserkennung mittlerweile hauptsächlich mit dem Einsatz von *Deep Neural Networks* (DNNs) realisiert [84]. Diese benötigen für das Training und eine möglichst gute Generalisierung große und/oder hochqualitative Mengen an (Bild-)Daten [24]. Wurden Datensätze für FER-Tasks anfangs hauptsächlich unter kontrollierten Laborbedingungen entwickelt, hat sich die Forschung zu s. g. *In-the-Wild-Daten* hinbewegt, die versuchen, den Anforderungen der realen Welt bspw. durch unterschiedliche Belichtungssituationen oder Kopfhaltungen gerecht zu werden. Sie werden meist mithilfe von Bildern aus Filmen, vor allem aber durch das Sammeln von Bildergebnissen aus Suchmaschinen im Internet gewonnen [84]. Dadurch wird es deutlich einfacher große Datenmengen zu generieren. Da die Bildauflösung aufgrund der unterschiedlichen Quellen aber stark variiert, wird diese durch *Up-* bzw. *Downsampling* vereinheitlicht. So liegen manche der bekannten In-the-Wild-Datensätze für FER bspw. in 48×48 Pixeln vor (FER2013-Datensatz [51]), andere in 100×100 (RAF-DB [83]) oder 224×224 Pixeln (AffectNet-Datensatz [108]).
Gering aufgelöste Bilder bieten durch ihre kleinere Datenmenge den Vorteil einer schnelleren Verarbeitung bzw. einer geringeren Inanspruchnahme von Ressourcen. Auch kann u. U. das Risiko von *Overfitting* durch Bilder mit weniger Pixeln verringert werden, da so je nach Netzwerk die Anzahl der Parameter weniger werden, die von einem DNN optimiert werden müssen [119]. Doch durch Downsampling bzw. eine geringe Auflösung können auch Informationen verloren gehen resp. von vornherein fehlen, die ggf. für die Emotionserken-

nung nützlich sein könnten. Durch die immer besser werdenden Verarbeitungsmöglichkeiten von Computern und ihren Prozessoren können auch Bilder mit einer höheren Pixelanzahl immer problemloser verarbeitet werden, sodass die Vorteile kleiner Bilder abnehmen. Da sich mehr Informationen oftmals positiv auf die Leistung von DNNs auswirken [24], liegt die Vermutung nahe, dass auch bestimmte höhere Bildauflösungen, die mehr Details enthalten können, einen positiven Effekt auf die Emotionserkennung haben. Es ist also zu überprüfen, ob dies tatsächlich der Fall ist und in welchem Bereich diese optimalen Auflösungen liegen.

Ein weiteres spannendes Feld, das sich über die Jahre u. a. mit dem Aufkommen von Deep-Learning-Methoden stark entwickelt hat und genau an der Herausforderung von Emotionserkennung anhand kleiner Bilder ansetzen kann, ist Super-Resolution (SR). Es handelt sich dabei um Verfahren, die ein hochaufgelöstes Bild von einem Bild niedriger Auflösung rekonstruieren. Entgegen der sonst üblichen Interpolation, die Unschärfe und unrealistische Ergebnisse bei der Vergrößerung einführt, versucht Super-Resolution diese Effekte insbesondere mithilfe von Deep Learning zu vermeiden und eventuell vorhandene Degradierungen wieder zu entfernen [125], [141], [142], [148].
Bezogen auf die Emotionserkennung stellt sich also zusätzlich die Frage, ob niedrig aufgelöste Daten von der Hochskalierung und Optimierung durch moderne Super-Resolution-Verfahren profitieren und damit für einen Leistungszuwachs in DNNs bei der Bewältigung von FER-Aufgaben (engl. *FER-Tasks*) sorgen können, zumal das Upsampling zwar emotionsgetreu passieren muss, der Erhalt von Identitäten aber vernachlässigbar wäre, sofern ein entsprechendes Verfahren diese verändern sollte.

Im medizinischen Bereich wurden die Einflüsse der Bildauflösung auf DNNs bereits anhand endoskopischer Bilder sowie Röntgenaufnahmen untersucht. Sie zeigen, dass besser aufgelöste Daten durchaus einen positiven Effekt haben können, insbesondere wenn die Klassifizierung der Befunde von Details abhängt [119], [131]. Im direkten Bezug zur Emotionserkennung gibt es mehrere Studien, die demonstrieren, dass SR-Methoden effektiv dabei helfen können, relevante Merkmale aus niedrig aufgelösten Bildern zu extrahieren, um so die Erkennungsleistung zu steigern [67], [91], [110], [123], [130], [137]. Die genannten Arbeiten werden in Abschnitt 2.2 zusammengefasst.

1.1 Fragestellungen & Erkenntnisinteresse

Die vorliegende Arbeit beschäftigt sich mit der Klärung der folgenden Fragen:

i) Haben höhere Bildauflösungen einen positiven Effekt auf die Erkennungsraten von Deep Neural Networks beim Bewältigen von FER-Tasks?

ii) Wo liegt der optimale Auflösungsbereich, der sich positiv auf die Leistung von Deep Neural Networks bei der Emotionserkennung auswirkt?

iii) Lassen sich potenziell positive Effekte höherer Bildauflösungen bei der Emotionserkennung mit Deep Neural Networks durch das Anwenden von Super-Resolution-Methoden auf Bilder mit niedriger Auflösung replizieren?

Die Erkenntnisse der Arbeit könnten nützlich sein, um den Wert und die Qualität von (älteren) Datensätzen mit niedrig aufgelösten Bildern zu steigern. Stellt sich heraus, dass FER-Tasks durch DNNs, die mit höher aufgelösten Trainingsdaten trainiert wurden, auch bessere Ergebnisse erzielen und Super-Resolution effektiv dazu beitragen kann, könnten diese Datensätze hochskaliert und somit die Trainingsmenge eines Netzwerks mit qualitativ hochwertigen Daten deutlich erhöht werden.
Der Super-Resolution-Ansatz könnte sich neben der Trainingsdatenperspektive außerdem auch in Szenarien als vorteilhaft erweisen, in denen es aufgrund von Einschränkungen nicht möglich ist, hochaufgelöste Bilder für die Erkennung zu generieren. So liefern z. B. Überwachungskameras und Webcams oft nur gering aufgelöste Bilder oder es sollen mehrere Gesichter innerhalb eines Bildes, also kleine Teilausschnitte, erkannt werden. Bei letzterem entstehen Abhängig von diesen Ausschnitten auch unterschiedlich hohe Detailgrade. Super-Resolution könnte daran ansetzen, die Auflösung nachträglich optimieren und damit ggf. auch die Erkennungsraten erhöhen.
Zuletzt lassen sich die Ergebnisse voraussichtlich auch auf andere Computer-Vision-Aufgaben übertragen, sofern diese von höher aufgelösten Bildern profitieren.

Vor diesem Hintergrund zielt diese Forschungsarbeit darauf ab, den Einfluss der Bildauflösung und des Einsatzes von Super-Resolution-Verfahren auf die Klassifikationsleistung von Deep Neural Networks, die zur Emotionserkennung genutzt werden, zu analysieren.

1.2 Vorgehensweise

Nachdem sich ein Überblick über aktuelle, relevante Studien verschafft wurde, werden zur Beantwortung der Fragen Versuche durchgeführt. Hierzu findet eine Auswahl geeigneter Datensätze sowie einer Netzwerk-Architektur für die Aufgabe der Facial Expression Recognition statt, deren Daten zu unterschiedlichen Auflösungen herunterskaliert werden, um

die Leistungsunterschiede eines damit trainierten Netzwerks zu untersuchen. Im Anschluss werden vortrainierte Super-Resolution-Modelle aufgrund von geeigneten quantitativen Metriken sowie ihrer subjektiv wahrgenommenen visuellen Ergebnisse bewertet und ausgewählt, um damit die durch das Herunterskalieren degradierten Datensätze bezogen auf Auflösung und Bilddetails zu verbessern. Auch mit diesen Daten wird die Klassifikationsleistung des Netzwerks zur Facial Expression Recognition gemessen, damit festgestellt werden kann, welchen Effekt diese auflösungsoptimierten Bilder darauf haben.

1.3 Aufbau

Diese Arbeit gliedert sich wie folgt: Kapitel 2 beinhaltet Erklärungen zu den grundlegenden Begriffen sowie Konzepten, die für das Verständnis vonnöten sind, grenzt diese voneinander ab und wirft einen Blick auf die aktuelle Forschungslandschaft zu den hier aufgeworfenen Kernfragen. In Kapitel 3 werden die gewählten Methoden erläutert, ebenso wie die Herangehensweise der zur Beantwortung der Fragen i), ii) und iii) durchgeführten Untersuchungen. Die daraus resultierenden Ergebnisse werden in Kapitel 4 dargestellt, in Kapitel 5 diskutiert sowie interpretiert und in Kapitel 6 abschließend resümiert.

2 Theoretischer Hintergrund

2.1 Begriffliche Grundlagen

Nachfolgend werden Begriffe erklärt und abgegrenzt, die für das Verständnis der vorliegenden Arbeit vonnöten sind. Dadurch soll ein grundlegender, aber vereinfachter Einblick in die vorliegenden Themen und Schlüsselkonzepte vermittelt werden.

2.1.1 Facial Expression Recognition (Emotionserkennung)

In der (englischsprachigen) Literatur wird das Erkennen von Gesichtsausdrücken häufig unter dem Begriff *Facial Expression Recognition* (FER) zusammengefasst [84], [86], [108], was zum Forschungsgebiet des *Affective Computing* gehört, das sich mit der maschinellen Analyse sowie Nutzung menschlicher Gefühlszustände befasst [86]. Da hierfür visuelle Eingabedaten repräsentiert durch Bilder verwendet werden, um daraus aussagefähige Informationen zu extrahieren, bewegt man sich mit der Aufgabenstellung gleichzeitig im Feld der *Computer Vision*.
Das Ziel von FER ist es, Gesichtszüge und ggf. deren Veränderung zu analysieren, um diesen Werte oder Kategorien zur Interpretation zuzuordnen [86]. In [104] nennen Martinez und Valstar noch genauer drei Problemdefinitionen, die sich hinter FER verbergen und zwischen denen unterschieden werden muss: dem Erkennen von prototypischen Gesichtsausdrücken, der Analyse von Gesichtsmuskelbewegungen und der s. g. dimensionalen Affekterkennung. Dabei wird differenziert, ob es sich um ein reines zu beobachtendes Signal, wie bspw. der physischen Muskelbewegung eines Lächelns, oder die Nachricht dieses Signals (z. B. Freude) handelt.
Bei FER-Systemen kann zwischen zwei Arten bzgl. des zu analysierenden Inputs unterschieden werden: Methoden, die auf statischen Bildern und somit rein räumlichen Informationen basieren sowie dynamischen Verfahrensweisen, in denen zusätzlich die zeitliche Beziehung zwischen Bildern einer Sequenz einbezogen wird. Um diese auf visuellen Informationen basierenden Verfahren zu unterstützen, können in multimodalen Systemen auch auditive und physiologische Daten eingesetzt werden [84].

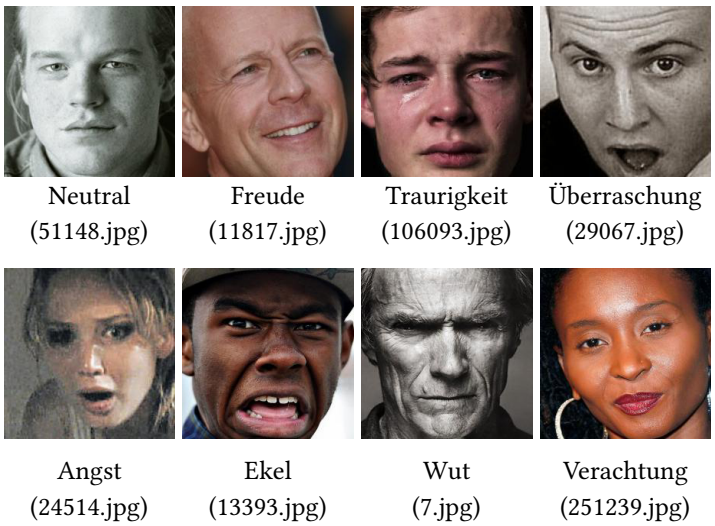
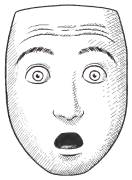


Abbildung 2.1: Beispiele der Basisemotionen (kategorisches Modell) inkl. „Neutral“ aus dem AffectNet-Datensatz [108].
(Bildquelle: [108])

Das Erkennen prototypischer Gesichtsausdrücke fällt unter den kategorischen Ansatz, bei dem einem Signal, dem Gesichtsausdruck, direkt eine Nachricht zugeordnet wird [104]. Es ist das, was in dieser Arbeit mit *Emotionserkennung* gemeint ist. In [40] beschrieben Ekman und Friesen sechs Basisemotionen mit der durch ihre Studie bestätigten Annahme, dass Menschen diese unabhängig von ihrer Kultur wahrnehmen könnten [84], [104]. Es handelte sich um *Freude, Wut, Traurigkeit, Ekel, Überraschung* und *Angst* (engl. *happiness, anger, sadness, disgust, surprise* und *fear*). Später wurde noch *Verachtung* (engl. *contempt*) als siebte Basisemotion dazu gezählt [84], [104], nachdem Ekman und Friesen diese in [42] bereits als weitere universell gültige Emotion entdeckt hatten und ein paar Jahre später die Ergebnisse noch einmal bestätigt wurden [105]. Neuere Studien, wie bspw. [65], argumentieren allerdings, dass das Modell dieser sechs bzw. sieben Basisemotionen nicht universell hinsichtlich der kulturellen Zugehörigkeit eines Menschen ist [84]. Weiter wird bspw. in [39] beschrieben, dass weitaus mehr als sechs Emotionen existieren und diese teilweise durch die Kombination von Basisemotionen entstehen, weshalb sie als *zusammengesetzte Emotionen* (engl. *compound emotions*) bezeichnet werden. Die vorliegende Arbeit beschränkt sich auf FER im Sinne einer Erkennung der Basisemotionen.

Zum objektiven Beschreiben von Gesichtsausdrücken (den Signalen) wird üblicherweise das *Facial Action Coding System* (FACS) [41] (siehe Tabelle 2.1) verwendet [104], das ebenfalls von Ekman und Friesen entwickelt und 2002 [43] erneuert wurde. Es ist das meistverbreitete System, verwendet objektive Parameter und erfordert keine Interpretation des beobachteten Signals. So können damit 32 kleine Gesichtsmuskelbewegungen, s. g. *Action Units* (AUs), sowie zusätzlich vierzehn *Action Descriptors* (ADs) (bspw. *beißen*) mit fünf unterschiedlichen Intensitätsleveln beschrieben werden. Zwar hat das Beschreiben von Gesichtsausdrücken mit

Tabelle 2.1: Facial Action Coding System (FACS) Action Units (AUs) für sechs Basisemotionen (Quelle: In Anlehnung an [73, S. 617, Tab. 1]. Bildquelle: [106, S. 83])

Basisemotion	**Gesichtsausdruck**	**Action Units**	**Action**
Freude		AU 1	Inner corner of eyebrow raised
		AU 6	Cheek raiser and lid compressed
		AU 12	Lip corner pulled
		AU 14	Dimpled
Überraschung		AU 1	Inner corner of eyebrow raised
		AU 2	Outer corner of eyebrow raised
		AU 5	Upper lid raised
		AU 15	Lip corner depressed
		AU 16	Lower lip depressed
		AU 20	Lip stretched
		AU 26	Jaw dropped
Angst		AU 1	Inner corner of eyebrow raised
		AU 2	Outer corner of eyebrow raised
		AU 4	Brow lowered
		AU 5	Upper lid raised
		AU 15	Lip corner depressed
		AU 20	Lip stretched
		AU 26	Jaw dropped
Ekel		AU 2	Outer corner of eyebrow raised
		AU 4	Brow lowered
		AU 9	Nose wrinkled
		AU 15	Lip corner depressed
		AU 17	Chin raised
Wut		AU 2	Outer corner of eyebrow raised
		AU 4	Brow lowered
		AU 7	Lid tightened
		AU 9	Nose wrinkled
		AU 10	Upper lip raised
		AU 20	Lip stretched
		AU 26	Jaw dropped
Traurigkeit		AU 1	Inner corner of eyebrow raised
		AU 4	Brow lowered
		AU 15	Lip corner depressed
		AU 23	Lip tightened

dem FACS eine hohe Beurteilerübereinstimmung, ist aber zunächst sehr zeitaufwändig und erfordert aufgrund der feinen Abstufungen eine intensive Schulung der Beurteilenden [104]. Beschäftigt sich der kategorische Ansatz nur mit wenigen und vor allem diskreten Emotionen, versucht der dimensionale die Komplexität menschlicher Ausdrücke durch kontinuierliche und multidimensionale Informationen zu repräsentieren [104]. Hierzu wird meist das *Circumplex Model of Affect* [118] verwendet, das den affektiven Zustand anhand der kontinuierlichen Variablen für die *Valenz* bzw. emotionale *Wertigkeit* (engl. *valence*), positiv bis negativ, und die Intensität der *Erregung* (engl. *arousal*), entspannt bis erregt, beschreibt (siehe Abb. 2.2). Dabei wird angenommen, dass jede Basisemotion mit Wertebereichen innerhalb dieses Modells korrespondiert. Im Allgemeinen werden diese Werte von einem Bild zum nächsten während einer Sequenz ermittelt [104].

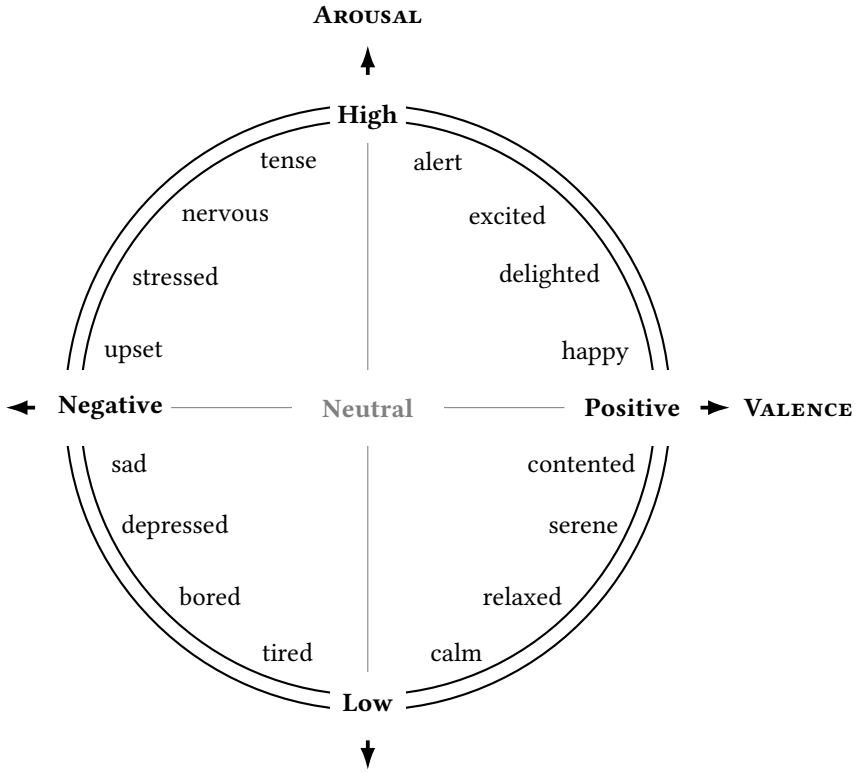


Abbildung 2.2: Grafische Repräsentation des Circumplex Model of Affect
(Quelle: In Anlehnung an [114, S. 716, Abb. 1] und [73, S. 618, Abb. 3])

Nicht zu verwechseln mit FER ist *Facial Recognition*, das sich auf das Identifizieren einer Person anhand ihrer Gesichtsmerkmale konzentriert [86]. Ebenfalls abzugrenzen ist die *Face Detection*, welche sich mit dem Erkennen von beliebigen Gesichtern in einem Bild befasst und oftmals der erste Schritt in automatisierten Systemen ist, die Gesichtsinformationen analysieren [86].

2.1.2 Machine Learning

Wie Görz *et al.* in [53] schreiben, ist das *Machine Learning* (ML, dt. *Maschinelles Lernen*) ein zentraler Bestandteil der *Künstlichen Intelligenz* (KI, engl. *Artificial Intelligence* [AI]). Sie beziehen sich zur Erklärung auf eine Definition von Mitchell in [107], wonach sich dieses Feld mit der Frage beschäftigt, wie sich Computerprogramme konstruieren lassen, die sich automatisch durch Erfahrung verbessern. Maschinelle Lernverfahren vereint, dass sie eine Menge von Trainingsdaten als Eingabe benötigen und das daraus gewonnene Wissen in einem s. g. *Modell* darstellen [53]. Sie werden also eher trainiert als explizit programmiert und finden statistische Strukturen in diesen Daten, um damit Regeln aufzustellen [24].

ML-Methoden lassen sich u. a. nach ihrer Lernart unterscheiden:
Beim *Supervised Learning* wird mit vorab gelabelten Daten trainiert, bei denen das Ergebnis bereits bekannt ist. Dabei wird versucht, die Beziehung zwischen Eingabedaten und Ergebnissen zu erlernen, um später Voraussagen über neue nicht im Training enthaltene Daten machen zu können. Die Vorhersagen sind entweder kategorische Klassen oder kontinuierliche Werte, weshalb man diese Lernform auch noch in zwei Unterkategorien, der Klassifikation und Regression, unterteilt [117]. Die Aufgabe (engl. *task*), Gesichtsausdrücken Kategorien von Emotionen zuzuordnen (Emotionserkennung), ist also ein Klassifikationsproblem.
Das *Unsupervised Learning* hingegen arbeitet mit ungelabelten bzw. unstrukturierten Daten. Ziel ist es, diese zu strukturieren und aussagekräftige Informationen zu extrahieren [117].
Der dritte Lerntyp nennt sich *Reinforcement Learning* und hat Ähnlichkeiten zum Supervised Learning. Im Unterschied dazu spielen aber nicht einzelne richtige oder falsche Ergebnisse eine Rolle, sondern das Resultat einer längeren Folge von Aktionen, wie bspw. bei einem Schachspiel [53], [117]. Beim Reinforcement Learning lernt ein Modell durch Interaktionen mit einer Umgebung, um eine Belohnungsfunktion (engl. *reward function*) zu maximieren und ein anvisiertes Ziel zu erreichen (wie das Gewinnen eines Spiels), ohne den Weg dorthin vorab zu kennen.

2.1.3 Datenrepräsentation & Datensätze

Eingabeinformationen werden von allen aktuellen ML-Systemen im Allgemeinen in Form von multidimensionalen *Arrays*[1] verarbeitet, die *Tensoren* genannt werden. Diese lassen sich wie Koordinaten im geometrischen Raum betrachten [24]. Chollet beschreibt Tensoren als Container für (meist) numerische Daten und als Verallgemeinerung von Matrizen mit einer beliebigen Anzahl von *Dimensionen* (oft *Achsen* genannt), welche den *Rang* (engl. *rank*) eines Tensors definieren. So sind Skalare Rang-0-Tensoren, Vektoren Rang-1-Tensoren,

[1] Arrays sind eine „Zusammenfassung von zusammengehörigen Daten des gleichen Typs“ [59, S. 331].

Matrizen Rang-2-Tensoren und Tensoren mit n Achsen Rang-n-Tensoren. Die Dimensionalität kann sich entweder auf die Summe der Achsen oder auf die Menge der Werte einer diesen beziehen, weshalb man bei letzterem auch von der *Form* (engl. *shape*) spricht – einem Tupel aus Ganzzahlen, das angibt, wie viele Dimensionen sich auf einer Achse befinden [24]. Ein (digitales) Bild, das aus Pixeln und Farbwerten besteht, wird entsprechend in einem Rang-3-Tensor der Form

$$(\textit{Pixelhöhe}, \textit{Pixelbreite}, \textit{Farbkanalanzahl})$$

gespeichert.[2] Da die Verarbeitung von ML-Systemen durch Stapel (engl. *batches*) mehrerer Bilder erfolgt, werden diese Rang-3-Tensoren in Rang-4-Tensoren verpackt. Die Anzahl der Proben (engl. *samples*) bildet im Allgemeinen die erste Achse dieses Tensors, welcher dann der Form

$$(\textit{Probenanzahl}, \textit{Pixelhöhe}, \textit{Pixelbreite}, \textit{Farbkanalanzahl})$$

entspricht [24]. Eine schematische Darstellung eines solchen Tensors kann der Grafik 2.3 entnommen werden, die einen Stapel aus drei Bildern mit jeweils drei Farbkanälen zeigt.

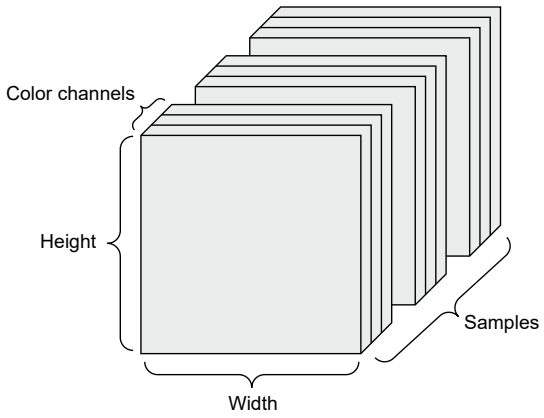


Abbildung 2.3: Schematische Darstellung von Bildern als Rang-4-Tensor
(Quelle: [24, S. 37, Abb. 2.4])

Um evaluieren zu können, wie gut ein ML-Algorithmus Aussagen über neue, unbekannte Daten machen kann (also solchen, die nicht Bestandteil des Trainings waren), wird nicht die Gesamtheit eines Datensatzes für den Trainingsprozess verwendet, sondern diese in zwei Teilmengen geteilt, die *Trainingsdatensatz* und *Testdatensatz* genannt werden. So kommen die Testdaten erst zur Beurteilung des fertig trainierten Modells zum Einsatz und können von diesem nicht vorab erlernt oder zur Anpassung verwendet werden [117].
Vom Trainingsdatensatz wird (üblicherweise) wiederum eine Teilmenge abgespalten, mit der das Modell nicht trainiert wird, um damit während eines Trainingsdurchlaufs, genannt *Epoche*,

[2] Chollet benennt hierzu die s. g. *channels-last convention*, was oftmals die bevorzugte ist, sowie die *channels-first convention*, bei der die Farbkanäle vor den Angaben der Höhe und Breite genannt werden [24].

Fehlerwerte (engl. *loss values*) und Metriken (eng. *metrics*) zu ermitteln, die eine Anpassung des Modells nach Bearbeitung eines Datenstapels ermöglichen [24]. Passt sich ein Modell zu stark an die Trainingsdaten an und verliert dadurch die Möglichkeit der Generalisierung gegenüber fremden, neuen Daten, wird von *Overfitting* (dt. *Überanpassung*) gesprochen [24]. Für die Evaluation werden bei Klassifikationsproblemen üblicherweise eine oder mehrere der Metriken

$$\text{Accuracy} = \frac{\text{Anzahl korrekt klassifizierter } x \in \mathbb{S}_{test}}{\text{Anzahl aller } x \in \mathbb{S}_{test}}, \tag{2.1}$$

$$\text{Precision}_i = \frac{\text{Anzahl korrekt als } i \text{ klassifizierter } x \in \mathbb{S}_{test}}{\text{Anzahl aller als } i \text{ klassifizierter } x \in \mathbb{S}_{test}}, \tag{2.2}$$

$$\text{Recall}_i = \frac{\text{Anzahl korrekt als } i \text{ klassifizierter } x \in \mathbb{S}_{test}}{\text{Anzahl aller } x \in \mathbb{S}_{test} \text{ mit Klasse } i}, \tag{2.3}$$

$$\text{F-Score}_i = 2 \cdot \frac{\text{Precision}_i \cdot \text{Recall}_i}{\text{Precision}_i + \text{Recall}_i} \tag{2.4}$$

für eine zufällige Testdatenmenge $\mathbb{S}_{test}$ mit den Elementen x für eine Klasse i ermittelt. Die *Accuracy* zeigt an, wie gut die vorhandenen Klassen durchschnittlich zugeordnet wurden. Diese Metrik eignet sich weniger gut bei einer sehr unausgeglichenen Klassenverteilung in der Testmenge, da bspw. bei 80 % zu einer Klasse i_a und 20 % zu einer Klasse i_b gehörenden Testdaten immer eine Accuracy von 80 % erreicht werden kann, wenn alle Daten i_a zugeordnet werden. Die drei anderen Metriken erlauben durch das Betrachten von einzelnen Klassen eine genauere Bewertung. Die *Precision* gibt an, wie hoch der Anteil der einer Klasse i zugeordneten Elemente ist, die auch tatsächlich zu dieser Klasse gehören. Der *Recall* bewertet, welcher Prozentsatz der tatsächlich vorhandenen Elemente der Klasse i in der Testmenge korrekt identifiziert wurde. Der *F-Score* ist das harmonische Mittel der Precision- und Recall-Metrik [53].

2.1.4 Deep Learning

Deep Learning ist ein spezifischer Unterbereich von Machine Learning, der (künstliche) neuronale Netzwerke (KNNs, engl. *Artificial Neural Networks*) zum Lösen von Aufgaben verwendet. Diese bestehen aus mehreren aufeinanderfolgenden Schichten (engl. *layers*), die dafür sorgen, dass Eingabedaten während des Lernprozesses zunehmend aussagekräftiger repräsentiert werden[3]. Je mehr davon aufeinanderfolgen, desto *tiefer* (engl. *deep*) ist ein

[3] Mit *Repräsentation* meint Chollet eine zusätzliche Art die Eingangsdaten darzustellen, sodass sie zur Lösung der zugrundeliegenden Aufgabe bestmöglich beitragen. Denn das Ziel ist es, Eingabedaten während des „Lernens“ so zu transformieren, dass sie näher an die erwartete Ausgabe führen. So können Aufgaben, die in der einen Repräsentation von Daten schwierig zu lösen sind, mit einer anderen leicht werden [24].

solches Netz(-werk). Deshalb spricht man bei Ansätzen, die sich auf ein oder zwei Schichten beschränken, auch gelegentlich von *shallow learning*. Im modernen Deep Learning hingegen sind oftmals dutzende oder hunderte Schichten beteiligt [24]. Solche neuronalen Netzwerke werden in der Literatur daher oftmals als *tiefe neuronale Netze* [53] oder englisch als *Deep Neural Networks* (DNNs) [1], [12] bezeichnet.

2.1.5 Deep Neural Networks

Wie in 2.1.4 beschrieben, ist ein Deep Neural Network ein künstliches neuronales Netz und besteht aus vielen Schichten, die zur Datenrepräsentation verwendet werden [24]. Ein solches versucht einen Wissensspeicher zu schaffen, der dem menschlichen Gehirn ähnelt [26]. Doch obwohl manche Konzepte von neuronalen Netzwerken durch das bisherige Verständnis der Funktionsweise unseres Gehirns inspiriert wurden, sind KNNs keine genaue Abbildung davon und funktionieren anders [24].
„Ein neuronales Netz besteht aus einer Menge von Neuronen, die durch gerichtete und gewichtete Verbindungen miteinander verknüpft sind." [27, S. 196] Weiter definieren Cleve und Lämmel „ein [künstliches] Neuron [als] [...] Verarbeitungseinheit, die die über die gewichteten Verbindungen eingehenden Werte geeignet zusammenfasst [...] und daraus mittels einer Aktivierungsfunktion unter Beachtung eines Schwellenwertes einen Aktivierungszustand ermittelt. Aus dieser Aktivierung bestimmt eine Ausgabefunktion die Ausgabe eines Neurons." [27, S. 192]. Die s. g. Eingabeschicht (engl. *input layer*) in einem KNN ist die ers-

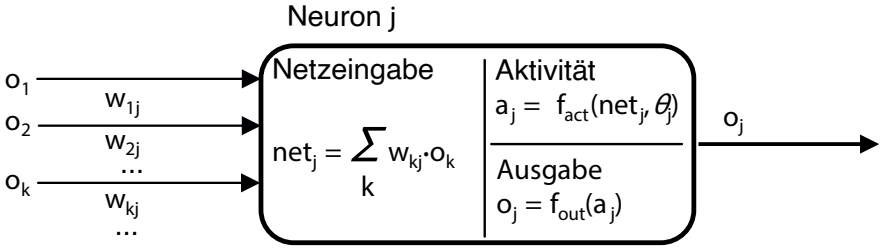


Abbildung 2.4: Schematische Darstellung eines künstlichen Neurons. Die Eingabeinformationen o_k führen mit gewichteten Verbindungen w_{kj} zum Neuron j, um die Netzeingabe durch net_j zu ermitteln. Die Aktivität a_j wird mittels Aktivierungsfunktion f_{act} bestimmt, die einen Schwellwert θ_j berücksichtigt. Mit der Ausgabefunktion f_{out} wird schließlich der Ausgabewert o_j berechnet [27].
(Quelle: [27, S. 192, Bild 5.2])

te Schicht und setzt sich aus Eingabe-Neuronen zusammen, die so bezeichnet werden, da es zu ihnen keine gerichteten Verbindungen gibt [27]. Die letzte Schicht eines neuronalen Netzes nennt man Ausgabeschicht (engl. *output layer*) und besteht aus Ausgabe-Neuronen, die mit ihren Ausgabewerten das Ergebnis einer Verarbeitung abbildet [27]. Schichten, die

zwischen diesen beiden liegen, heißen verdeckte Schichten (engl. *hidden layer*) [27]. Durch das Verknüpfen dieser Schichten bzw. Neuronen entsteht das neuronale Netz [27]. Die Menge sowie Art und Weise wie sie miteinander verknüpft sind, wird in der Literatur als *Architektur* bezeichnet [1], [12], [27].
Um die Netzeingabe net_n eines Neurons n zu ermitteln, wird eine s. g. *Propagierungsfunktion* f_{prop_n} verwendet [26] wofür i. d. R. die Summe aus mit den Gewichten w_{in} multiplizierten Ausgaben der Vorgängerneuronen (bzw. den Eingabeinformationen bei der ersten Schicht) o_i gebildet wird [26], [27]:

$$f_{prop_n} = net_n = \sum_i w_{in} \cdot o_i$$

Als Aktivierungsfunktion, die den Aktivierungszustand bestimmt, werden in Deep Neural Networks häufig nichtlineare Funktionen aus dem folgenden Grund verwendet [24]: Neuronale Netze bestehen aus Tensor-Operationen, die einfache geometrische Transformationen der Eingangsdaten im multidimensionalen Raum sind [24]. Dazu gehören bspw. die Translation, die Rotation oder die Skalierung, wobei das Rotieren und Skalieren jeweils lineare Transformationen sind [24]. Die Kombination aus einer solchen linearen Transformation und einer Translation wird *affine Abbildung* (auch *affine Transformation*) genannt [24], [149]. Werden mehrere dieser affinen Transformationen hintereinander durchgeführt, wie es bei vielen Schichtarten in DNNs der Fall ist, ergibt sich wiederum eine solche [24]. Ohne eine nichtlineare Aktivierungsfunktion würden sich diese verketteten Operationen also auch durch eine einzige darstellen lassen und die Verkettung vieler Schichten (wodurch die Tiefe eines neuronalen Netzes entsteht) hätte keinen Vorteil [24]. Durch sie werden nichtlineare Transformationen und damit die Modellierung sehr komplexer Muster überhaupt erst möglich, was für einen erweiterten Hypothesenraum sorgt [24].
Die Gewichte eines Modells sind der Ort, an dem das erlangte Wissen gespeichert wird. Sie repräsentieren also die Informationen, die durch das Verarbeiten der Trainingsdaten erlernt werden und werden initial mit zufälligen Werten belegt [24]. Das Ermitteln des Abstands (auch *Fehlerwert* oder *Distanz*), im Englischen als *prediction error* oder *loss value* bezeichnet, zwischen dem bekannten, erwarteten Ergebnis und dem während des Trainings bestimmten Resultat dient als Feedback-Signal, um den verwendeten *Algorithmus* anpassen zu können. Der Abstand wird dabei mit einer Fehlerfunktion berechnet. Die graduelle Anpassung der Gewichte ist, was als Lernen bezeichnet wird [24]. Im Allgemeinen wird dafür eine Technik benutzt, die sich *Gradientenabstieg* nennt und sich die Eigenschaften der in den Modellen verwendeten Funktionen zu eigen macht, dass diese alle *stetig*[4] und *glatt*[5] sind. Solche Funktionen lassen sich in jedem Punkt differenzieren, eine Verkettung dieser also ebenfalls [24]. Das ermöglicht den s. g. *Gradienten* zu berechnen, der beschreibt, wie der *Loss* von den Koeffizienten (Gewichten) eines Modells beeinflusst wird. Ist dieser bekannt, können die Gewichte in eine Richtung angepasst werden, die den Fehler zwischen Vorhersage und

[4] „Eine Funktion ist stetig, wenn es in ihrem Funktionsgraphen keine ‚Sprünge' gibt [...]." [149, S. 521]
[5] „Eine [...] Funktion, die unendlich oft differenzierbar ist, nennt man *glatt* [...]." [149, S. 631]

tatsächlichem Wert der Trainingsdaten verkleinert. Der Gradient ist also die Ableitung einer Tensor-Operation bzw. -Funktion. Werden die Gewichte leicht in die gegenüberliegende Richtung des Losses angepasst, wird dieser mit jeder Anpassung etwas kleiner. Das Anpassen erfolgt mit der Lernrate (engl. *learning rate*), einem Skalar als Parameter, der mit dem Gradienten multipliziert wird und so gewählt werden muss, dass weder zu große Sprünge stattfinden, noch lokale Minima nicht mehr überwunden werden. S. g. *Optimizer* können dabei auch bisher erfolgte Aktualisierungen der Gewichte einbeziehen und beim Überwinden eines lokalen Minimums helfen [24]. Mithilfe der *Backpropagation* lässt sich ein Gradient auch von beliebig komplex verketteten Tensor-Operationen, wie sie in einem neuronalen Netz vorkommen, berechnen, was im Grunde nur die Anwendung der Kettenregel

$$(f \circ g)'(x) = f'(g(x)) \cdot g'(x)$$

ist. Damit wird rückwärtsgehend von der letzten Schicht aus berechnet, wie stark ein Neuron zum Entstehen des Loss Values beigetragen hat, um die Gewichte entsprechend anpassen zu können [24].

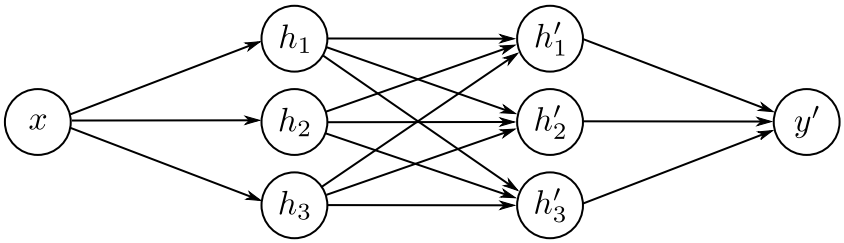


Abbildung 2.5: Schematische Darstellung eines neuronalen Netzwerks mit einem Eingabeneuron x, zwei verdeckten Schichten mit den Neuronen h_1, h_2, h_3 und h'_1, h'_2, h'_3 sowie einem Ausgabeneuron y'.
(Quelle: In Anlehnung an [115, S. 45, Abb. 4.4])

Es gibt verschiedene Arten von DNNs, darunter insbesondere *Convolutional Neural Networks* (CNNs), die mit ihrem Aufkommen sehr gute Ergebnisse im Feld der Computer Vision, das zuvor lange ohne neuronale Netze bearbeitet wurde, vor allem bei der Klassifikation von Bildern erzielten und seitdem universell in solchen Anwendungen eingesetzt werden [24]. Hauptbestandteile eines solchen Netzwerks sind Faltungsschichten (engl. *convolution layers*) und Pooling-Operationen [24], [53]. Im Gegensatz zu voll verknüpften Schichten (engl. *dense layers*/*fully connected layers*), die globale Muster im gesamten Merkmalsraum erlernen, lernen Faltungsschichten im Falle von Bildern lokale Muster anhand von kleineren Fenstern, die verschiebungsinvariant sind und damit überall im Bild wiedererkannt werden können [24], [53]. Außerdem können CNNs räumliche Hierarchien erlernen, wodurch komplexere Konzepte zusammengesetzt werden können: Eine erste Schicht erkennt bspw. Kanten, eine zweite bestimmte Muster, die aus diesen Kanten zusammengesetzt werden [24]. CNNs arbeiten mit

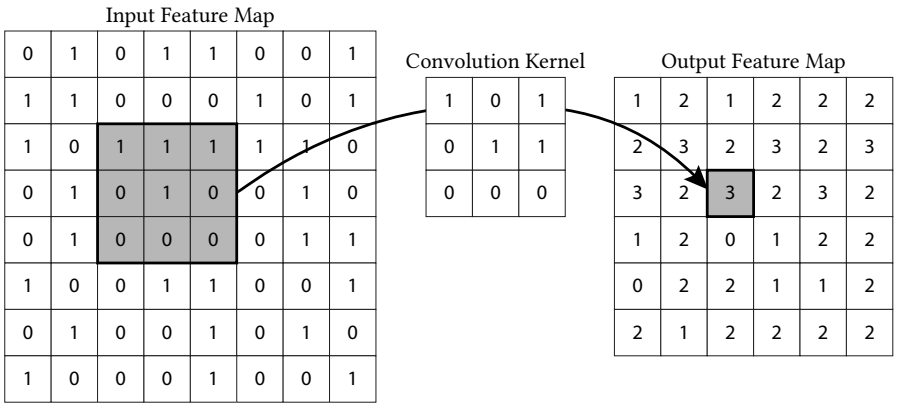


Abbildung 2.6: Beispiel eines Faltungsschritts im Zweidimensionalen. Der grau gefärbte Wert in der Output Feature Map ergibt sich durch die Berechnung aus Formel (2.5), in der die Werte des grau gefärbten Fensters innerhalb der Input Feature Map mit den Werten der Faltungsmatrix (Convolution Kernel) miteinander multipliziert und aufsummiert werden: $1 \cdot 1 + 1 \cdot 0 + 1 \cdot 1 + 0 \cdot 0 + 1 \cdot 1 + 0 \cdot 1 + 0 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 = 3$
(Quelle: In Anlehnung an [27, S. 254, Bild 6.43])

Rang-3-Tensoren (Höhe, Breite, Tiefe), die Merkmalskarten bzw. engl. *Feature Maps* genannt werden [24], [53]. Durch das Anwenden der Faltungsoperation auf kleinere Ausschnitte entsteht eine *Output Feature Map*, die ein Rang-3-Tensor bleibt, aber deren (beliebige) Tiefe anders als noch beim Input nicht mehr für die Farbkanäle sondern für s. g. *Filter* steht und spezifische Merkmale des Input-Tensors repräsentiert. Die kleineren Ausschnitte werden erzeugt, indem typischerweise ein 3×3- oder 5×5-Fenster über die *Input Feature Map* gelegt und Schritt für Schritt verschoben wird [24]. Jeder dieser dreidimensionalen Ausschnitte wird durch ein Tensorprodukt mit einer erlernten Faltungsmatrix bestehend aus Gewichten (Anfangs zufällig gewählte Werte [27]) in eindimensionale Vektoren transformiert, die zuletzt wieder zu einem dreidimensionalen Tensor zusammengesetzt werden [24]. Durch das Anwenden der Fenster auf die Input Feature Map wird die Output Feature Map entsprechend kleiner, wenn kein *Padding* angewendet wird, das für zusätzliche Zeilen und Spalten sorgt und es somit ermöglicht, das Zentrum des Convolution-Fensters um jeden Input-Wert zu legen [24]. Im Zweidimensionalen wird der Wert der Output Feature Map O an der Position i, j mit

$$O(i,j) = \sum_{m=0}^{dim_F - 1} \sum_{n=0}^{dim_F - 1} I(i+m, j+n) \cdot F(m,n) \tag{2.5}$$

berechnet, wobei dim_F die Dimension der Faltungsmatrix darstellt, I die Input Feature Map und $F(m,n)$ den Wert der Faltungsmatrix an der Position m, n [27]. Ein vereinfachtes Beispiel dazu lässt sich Abbildung 2.6 entnehmen.

Die Pooling-Operation ist die zweite zentrale Operation, um lokale Merkmale zusammenzufassen. Dabei werden Werte statistisch vereint, wodurch eine neue Feature Map entsteht [53].

Aus den Input Feature Maps werden Fenster extrahiert und bspw. deren Mittelwert (Average-Pooling) oder Maximum (Max-Pooling) berechnet [24]. Auch andere Operationen, wie z. B. der Einsatz von größeren Schrittweiten (engl. *strides*) während den Faltungen, sind möglich [24], [53]. Das Ziel davon ist eine Verringerung der zu verarbeitenden Feature-Map-Koeffizienten, um Overfitting zu vermeiden, sowie das Schaffen von räumlichen Filterhierarchien, indem aufeinanderfolgende Faltungsschichten immer größere Fenster (durch das Zusammenfassen benachbarter) betrachten [24].
CNNs enden üblicherweise mit einem *Flatten Layer* oder einem *Global-Pooling Layer*, die die räumlichen Feature Maps in Vektoren umwandeln. Anschließend folgt ein Dense Layer, der mit diesen eindimensionalen Vektoren arbeitet. Bei der Klassifikation hat dieser Layer üblicherweise genauso viele Neuronen wie es Klassen gibt, um entsprechende Werte zur Zuordnung auszugeben [24].

Während CNNs insbesondere auf die Verarbeitung von zweidimensionalen Bildern ausgelegt sind [53], gibt es noch weitere Arten von Deep Neural Networks. Darunter u. a. *Rekurrente Neuronale Netze* (engl. *Recurrent Neural Networks* [RNNs]), welche sich gut für die Analyse von Sequenzen eignen [24], [53], *Transformer* [136], die mit einem Aufmerksamkeitsmechanismus arbeiten und RNNs bereits in der Sequenzverarbeitung abgelöst haben, da sie meist bessere Ergebnisse erzielen können [53] oder *Generative Adversarial Networks* (GANs) [52], die sich dafür eignen neue Daten zu synthetisieren und für die Generierung von Bildern von Goodfellow *et al.* 2014 vorgestellt wurden [117]. Bei dieser Bildsynthetisierung erzielen sie recht realistische Ergebnisse [24]. GANs gehören zu den generativen Modellen und bestehen aus zwei Teilen: Einem *Generator*, der versucht auf Grundlage der Trainingsdatenverteilung neue Daten zu generieren, und einem *Diskriminator*, der versucht zu ermitteln, wie hoch die Wahrscheinlichkeit ist, dass eine Probe aus den Trainingsdaten stammt und nicht generiert wurde. Während des Trainingsprozesses versucht der Generator immer bessere Daten zu generieren, die den Diskriminator täuschen [52]. Dabei sieht der Generator nie direkt die Trainingsdaten, sondern erhält alle Informationen vom diskriminativen Modell [24].
Transformer betrachten eine Menge aus Vektoren und verwenden einen Aufmerksamkeitsmechanismus, der auch über weite Distanzen Abhängigkeiten unter diesen Eingaben ermittelt. Im Kontext von *Natural Language Processing* können so Zusammenhänge zwischen einzelnen Wörtern „verstanden“ werden [24], [117]. Populäre Sprachmodelle wie BERT [30] und GPT [116] sind aus der Transformer-Architektur hervorgegangen [117]. Wurden Transformer vor allem für die Verarbeitung von Textdaten entwickelt, werden sie mittlerweile auch für mit Bildern kontextualisierte Aufgaben eingesetzt und können hierbei in einigen Fällen die Leistung von CNNs übertreffen. Ein s. g. *Vision Transformer* [38] teilt ein Bild in kleinere Ausschnitte bzw. Fenster (nach Dosovitskiy *et al.* 16×16 Pixel [38]) auf, die in eine niedriger dimensionale Repräsentation umgewandelt und dann vom Netzwerk verarbeitet werden, um so globale Zusammenhänge zwischen diesen Ausschnitten zu erfassen. *Multi-Scale Vision Transformers* wie der *Swin Transformer* [95] nutzen dafür unterschiedliche Auflösungen,

die schrittweise verringert werden. Während CNNs mit Faltungen Merkmale extrahieren und hierarchische Filter generieren, stellen Vision Transformers Abhängigkeiten zwischen Bildausschnitten her, was pro Schicht eine vergleichsweise geringe rechnerische Komplexität darstellt [115].
Oftmals werden unterschiedliche Modelle mit verschiedenen Architekturen und Ansätzen miteinander kombiniert, um die bestmöglichen Ergebnisse zu erzielen, was *Model Ensembling* genannt wird [24].

2.1.6 Bildinterpolation

Die Interpolation von Bildern ist der Prozess, bekannte Werte zu verwenden, um neue Werte an unbekannten Punkten zu berechnen [50]. Dies ist nötig, da diskrete Pixelpositionen eines Bildes oftmals nicht auf diskrete Rasterpositionen eines anderen Bildes abgebildet werden können [13]. Beim Vergrößern im zweidimensionalen Raum kann man sich, wie Gonzalez und Woods erklären, eine der einfachsten Methoden wie folgt vorstellen: Ein Raster, das bspw. mit 750 × 750 Pixeln dem vergrößerten Bild entsprechen soll, wird so verkleinert, dass es auf das Originalbild passt, welches nur 500 × 500 Pixel hat. Nun wird für ein Pixel des überlagernden Rasters das nächstgelegene Pixel im Originalbild gesucht und dessen Wert zugewiesen. Dies erfolgt für alle Pixel des neuen Rasters. Im Anschluss wird das Raster wieder auf seine ursprüngliche Größe skaliert, womit das vergrößerte Bild entsteht [50]. Da immer der Wert des nächstgelegenen Nachbarpixels übernommen wird, nennt man diese Methode *Nearest Neighbor Interpolation* [13], [50].

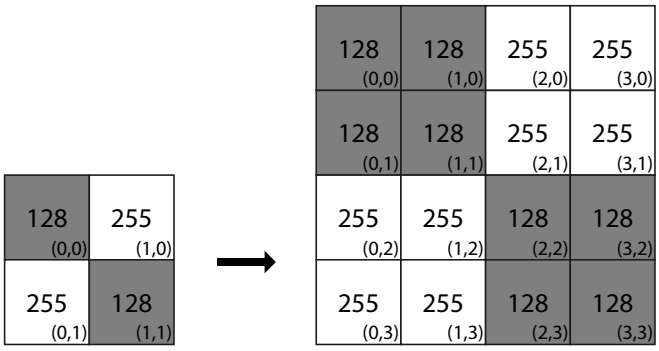


Abbildung 2.7: Vereinfachtes Beispiel der Nearest-Neighbor-Interpolation mit 8-Bit-Grauwerten (Weiß = 255, Grau = 128) und Pixelkoordinaten anhand eines 2 × 2-Bildes, das zu einer 4 × 4-Abbildung interpoliert wird.
(Quelle: eigene Grafik)

Möchte man also im vergrößerten Bild den Wert v eines neuen Punktes (x', y') ermitteln, muss zunächst sein nächster Nachbar N_n bestimmt werden. N_n kann dabei mithilfe der neuen

Koordinaten (x', y'), Breite W und Höhe H des Originalbildes sowie Breite W' und Höhe H' des neuen Formats mit

$$N_n = \begin{pmatrix} x_n \\ y_n \end{pmatrix} = \begin{pmatrix} \lfloor x' \times W/W' \rfloor \\ \lfloor y' \times H/H' \rfloor \end{pmatrix}$$

berechnet werden. Der Wert des Punktes im Originalbild wird dann als Wert für den Punkt im vergrößerten Bild übernommen:

$$v(x', y') = v(N_n)$$

Eine vereinfachte Darstellung kann Abbildung 2.7 entnommen werden.
Da mit dieser Methode nur bestehende Pixel an neuen Koordinaten wiederholt werden, entstehen blockartige Artefakte bei Vergrößerungen, die meist unerwünscht sind, weshalb sie nur selten in der Praxis zum Einsatz kommt [13], [50].

Die *bilineare Interpolation* bezieht die Werte der nächstgelegenen diagonalen Nachbarpixel im Originalbild in die Berechnung der neuen, interpolierten Werte ein [13], [50]. Für jeden Pixel mit den Koordinaten (x', y') des neuen Rasters $W' \times H'$ müssen zunächst die Koordinaten (x, y) im alten Raster $W \times H$ durch Skalierung bestimmt werden. Das heißt für

$$\forall (x', y') \in \{0, 1, \ldots, W' - 1\} \times \{0, 1, \ldots, H' - 1\}, \text{ berechne:}$$

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{x'}{W'-1} \cdot (W - 1) \\ \frac{y'}{H'-1} \cdot (H - 1) \end{pmatrix}$$

Anschließend wird für jeden skalierten Punkt P der Wert I der benachbarten Eckpixel durch das Bestimmen der nächstgelegenen ganzzahligen Koordinaten u und v ermittelt, wobei hierfür $u = \lfloor x \rfloor$ und $v = \lfloor y \rfloor$ gelten[6] [13]:

$$A = I(u, v) \qquad B = I(u + 1, v)$$
$$C = I(u, v + 1) \qquad D = I(u + 1, v + 1)$$

Die Werte A, B und C, D werden schließlich horizontal zu den Zwischenwerten E, F mit den Formeln

$$E = A + d_h \cdot (B - A)$$
$$F = C + d_h \cdot (D - C)$$

und dem horizontalen Abstand $d_h = (x - u)$ interpoliert, um mit diesen und dem vertikalen Abstand $d_v = (y - v)$ dann den endgültigen Interpolationswert $I(x, y)$ wie folgt zu berechnen:

$$I(x, y) = E + d_v \cdot (F - E)$$

[6] Werden die Nachbarn eines Randpixels berechnet, können einige davon außerhalb des Bildbereichs liegen, womit unterschiedlich umgegangen werden kann. So können sie je nach Verfahrensweise ignoriert oder mit spezifischen Werten belegt werden [50].

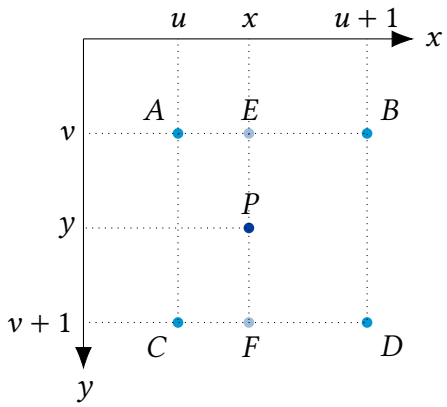


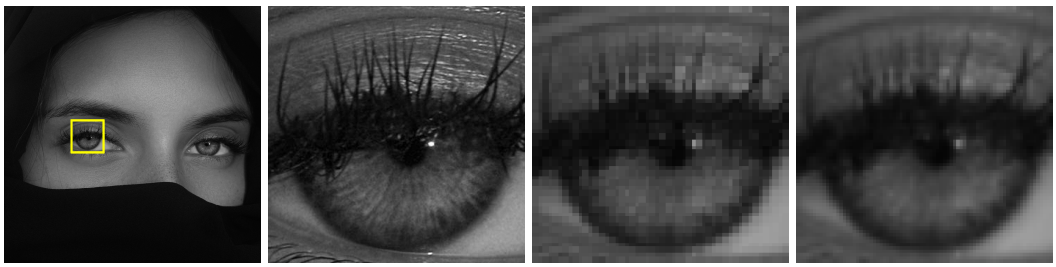
Abbildung 2.8: Veranschaulichung der beteiligten Punkte bei der bilinearen Interpolation (Quelle: In Anlehnung an [8])

Etwas komplexer ist die *bikubische Interpolation*, welche statt vier die Werte von sechzehn direkt benachbarten Pixeln zur Berechnung eines Interpolationswertes I einbezieht [50]. Der Wert zu einem Punkt (x, y) wird mit der Formel

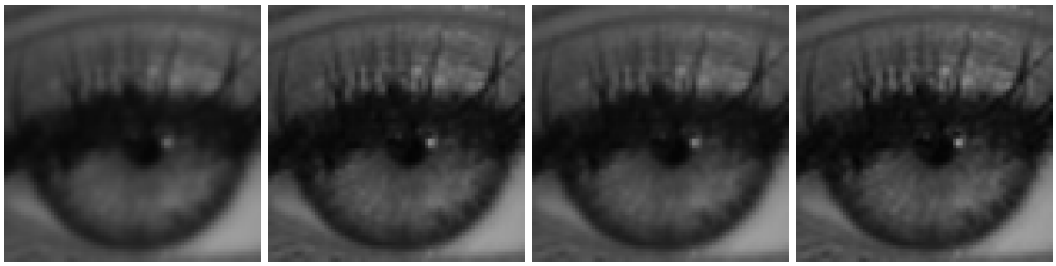
$$I(x, y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j$$

berechnet, wobei die Koeffizienten a_{ij} aus einem System von sechzehn Gleichungen abgeleitet werden, das auf den Werten der umliegenden Pixel basiert. Im Allgemeinen kann die bikubische Interpolation feine Details wie Kanten besser erhalten als die bilineare [13], [50] bei vergleichbarem Rechenaufwand, weshalb es in vielen Bildverarbeitungsprogrammen die Standardmethode geworden ist [13].
Es lassen sich auch mehr Nachbarn zur Interpolation verwenden und es gibt noch komplexere Techniken, die zur Verbesserung der Ergebnisse führen können. So lässt sich bspw. das bikubische Verfahren durch die Verwendung von *Splines*[7] erweitern [50] und bestimmte Nachteile des Verfahrens vermieden werden [66].
Weitere Methoden sind u. a. *Catmull-Rom* und *Mitchell-Netravali*, die eine schärfere Rekonstruktion als die bikubische Interpolation leisten können, allerdings bei fast gleichem Rechenaufwand. *Lanczos* ist ebenfalls eine beliebte Methode, die allerdings nicht wie die zuvor genannten auf polynomischen Funktionen basiert, sondern trigonometrische Funktionen nutzt, die mehr Rechenaufwand erfordern, die Bildergebnisse aber nicht viel besser sind. Für hochqualitative Bilder sollten entsprechend Catmull-Rom oder Mitchell-Netravali in Betracht gezogen werden [13].

[7] So werden Kurven genannt, die eine Funktion stückchenweise durch mehrere Polynome approximieren [149].



(a) Original (b) Original Ausschnitt (c) Nearest Neighbor (d) Bilinear



(e) Bikubisch (f) Catmull-Rom (g) Mitchell-Netravali (h) Lanczos

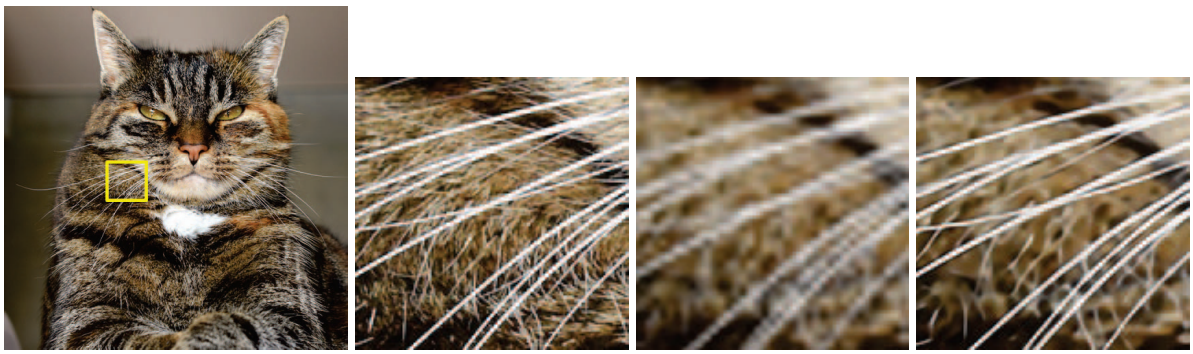
Abbildung 2.9: Vergleich von Bildinterpolationsmethoden. Der Bildausschnitt (b) des hochauflösenden Originalbilds (a) wurde bikubisch auf 50 × 50 Pixel herunterskaliert. Im Anschluss wurde dieser mithilfe der entsprechenden Interpolationsmethoden (c)-(h) auf 100 × 100 hochskaliert.
(Bildquelle: Jonaorle, 2020 [69])

2.1.7 Super-Resolution

(Image) Super-Resolution (SR) (auch *Image Upsampling/Upscaling* genannt) bezeichnet das Problem, ein hochaufgelöstes (engl. *high-resolution* [HR]) Bild von einem (oder mehreren) mit niedriger Auflösung (engl. *low-resolution* [LR]) zu rekonstruieren. Im Gegensatz zur klassischen (räumlichen) *Bildinterpolation* (siehe 2.1.6) durch bspw. bilineare oder bikubische Methoden, die im Vergleich zum natürlich hochauflösenden Bild nur sehr unscharfe und unrealistische Ergebnisse liefert, versucht das Feld der Super-Resolution diese Einschränkungen bei der Vergrößerung zu überwinden. Insbesondere SR-Algorithmen, die auf CNNs (siehe 2.1.5) basieren, haben auch hierbei wieder gezeigt, dass sie konventionelle Methoden ohne Deep Learning übertreffen und bemerkenswerte Erfolge bei der fotorealistischen Rekonstruktion verzeichnen können [125]. Einige der aktuellen SR-Methoden beruhen auf s. g. *Transformern* [136] statt auf Architekturen mit *Convolutions* (dt. *Faltungen*) [18], [20], [80], [89], [94] und zeigen teilweise sogar noch bessere Leistung [167]. Abbildung 2.10 zeigt einen Vergleich zwischen Interpolation und Super-Resolution.
Der Prozess die Bildqualität zu verbessern während versucht wird, die Bildcharakteristik zu bewahren, nennt sich *Image Enhancement* bzw. *Image Restoration*, ist oftmals Bestandteil von Super-Resolution (resp. umgekehrt [141]) und wird dann teils auch als *Super-Resolution Zooming* bezeichnet. Er beinhaltet aber nicht zwangsläufig nur die Verbesserung von gering

aufgelösten Bildern, sondern versucht alle möglichen Bildverschlechterungen wie Unschärfe, Rauschen oder Blockartefakte zu entfernen. *Restoration* bezieht sich in der Literatur meist auf den mathematischen Prozess eine Bilddegradierung rückgängig zu machen, *Enhancement* auf die Verbesserung der allgemeinen Bildqualität ohne eine mathematische Umkehrung des Degradierungsprozesses [155]. Ist dem Verfahren zur Auflösungs- und Bildverbesserung die Ursache einer Degradierung unbekannt, wird es auch *blinde Super-Resolution* (engl. *Blind Super-Resolution*) oder *Real-World-Super-Resolution* genannt [141], [142], [148]. Weitere Begrifflichkeiten finden sich in domänenspezifischeren Methoden, wie der *(Blind) Face Restoration*, die entsprechend explizit auf die Rekonstruktion hochqualitativer Bilder von Gesichtern abzielt [141]. Diese SR-Aufgabe wird in der Literatur auch mit dem Begriff *Face Hallucination* beschrieben [4], [92]. Er wurde 2000 von Baker und Kanade eingeführt mit Bezug auf die zusätzlichen Pixel, die beim Erhöhen der Auflösung generiert werden müssen [4]. In der vorliegenden Arbeit wird vorwiegend der Oberbegriff *Super-Resolution* verwendet. Es gibt auch Ansätze von Super-Resolution, die sich nicht auf den Pixelraum beziehen, hier aber der Vollständigkeit halber erwähnt werden. So schlagen Tan *et al.* in der Arbeit „Feature Super-Resolution: Make Machine See More Clearly“ [130] eine neue Methode für das Problem der Identifizierung von sehr kleinen Objekten bzw. solchen in niedrig aufgelösten Bildern vor, die sich nicht auf das zuvor erläuterte visuelle Vergrößern und Verbessern von Bildern konzentriert. Diese wird *Feature Super-Resolution* (FSR) von ihnen genannt und zielt darauf ab, die Trennschärfe zwischen Bildern auf Merkmalsebene zu erhöhen, um höhere Erkennungsraten durch Computer zu erreichen, indem schwache Merkmale (engl. *features*) des gesamten Merkmalsraums extrahiert und in hoch diskriminative Merkmale umgewandelt werden. Super-Resolution bzw. *Image Super-Resolution* (ISR), wie Tan *et al.* es zur Abgrenzung nennen und auf das die vorliegende Arbeit sich konzentriert, versucht also die visuelle Qualität durch Erhöhung der gesamten Pixel-Auflösung zu steigern, Feature Super-Resolution hingegen die Qualität der für die Erkennung durch ML-Modelle entscheidenden Eigenschaften.

Die meisten Ansätze sind auf das Supervised Learning (siehe 2.1.2) fokussiert [151] und versuchen eine Zuordnung zwischen niedrig aufgelösten und hochaufgelösten Bildern herzustellen [81], [89]. So werden die Netzwerke i. d. R. mit Datensätzen trainiert, die zu hochauflösenden Referenzbildern, welche zur Überprüfung der SR-Leistung dienen, jeweils ein oder mehrere korrespondierende niedrig auflösende Bilder haben [133]. Die Degradierung der HR-Bilder zum Erstellen dieser LR-Bilder kann verschieden vorgenommen werden und bspw. das Hinzufügen von Rauschen, Unschärfe oder anderen Artefakten beinhalten, was dazu führt, dass die hochfrequenten Bildinformationen verloren gehen, die mittels SR schließlich wiederhergestellt werden sollen. Dabei ist die meist verbreitete Methode das bikubische *Downsampling* (dt. *Heruntertaktung*; Reduzierung der Auflösung) ist [167]. Die Verkleinerungsfaktoren liegen meist bei ×2, ×3, ×4, teilweise auch bei ×8, was die üblichen in der Literatur und daher auch in den entsprechenden Wettbewerben sind [2], [133], [161]. Bei größeren Faktoren wird auch von *Extreme Super-Resolution* gesprochen [16], [159]. Das erste Modell, dass auf

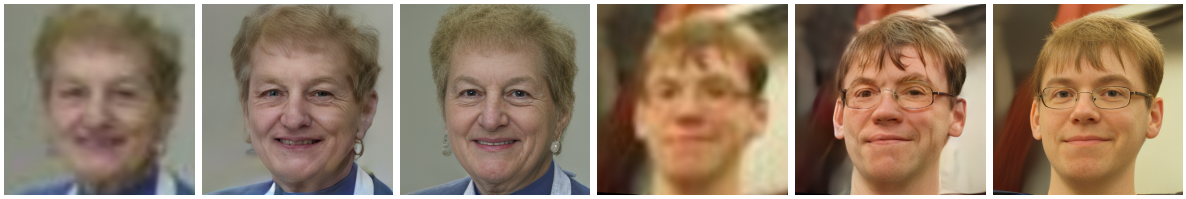


(a) hochaufgelöstes Originalbild | (b) vergrößerter Original-Ausschnitt | (c) bikubische Interpolation | (d) Ergebnis Super-Resolution

Abbildung 2.10: Beispiel zum Vergleich konventioneller Interpolation mit Super-Resolution. (a) ist ein hochaufgelöstes Originalbild, wovon in (b) ein vergrößerter Ausschnitt gezeigt wird, um den Detailgrad des Originals zu verdeutlichen. (c) zeigt die Rekonstruktion des Ausschnitts, wenn das Ausgangsmaterial in deutlich niedrigerer Auflösung vorliegt und bikubische Interpolation angewandt wird, (d) die Ergebnisse durch CNN-basierte Super-Resolution.
(Quelle: In Anlehnung an [90, S. 1137, Abb. 6])

CNNs basierte, war das *Super-Resolution Convolutional Neural Network* (SRCNN) [35], das 2014 von Dong *et al.* vorgestellt wurde, woraufhin immer mehr CNN-basierte Methoden entwickelt wurden [81], [82], die konventionelle Ansätze mit bspw. *Sparse Coding* übertrafen [167]. Daraufhin entstanden immer tiefer werdende Netzwerke [167], in die schließlich auch *Residual Connections* [58] Einzug fanden, um dem *Problem des verschwindenden Gradienten* (engl. *vanishing gradient problem*) entgegenzuwirken [80], [167]. Es beschreibt das durch die Verkettung vieler Netzwerkebenen entstehende Informationsrauschen, das die Fehlerinformation für die Backpropagation so stark überdecken kann, dass das Modell nichts mehr lernt. Eine *Residual Connection* sorgt dafür, dass Informationen rauschfrei an destruktiven Funktionen vorbei gereicht werden können, was tiefere Netzwerke ermöglicht [24], [80].
Auch GANs (siehe 2.1.5), die 2014 von Goodfellow *et al.* vorgestellt wurden [52], sind erstmals 2017 von Ledig *et al.* im SRGAN [79] für Super-Resolution eingesetzt worden, welches das erste Netzwerk war, das Bilder mit einem Downsample-Faktor von ×4 rekonstruieren konnte [82]. Sie werden außerdem sehr häufig aufgrund ihrer visuell ansprechenden Ergebnisse für Face Restoration eingesetzt [156]. Vor allem bei der Rekonstruktion von HR-Bildern aus sehr niedrig aufgelösten haben aktuelle Methoden mit GANs, wie GFPGAN [141] oder CodeFormer [170], gezeigt, dass sie bemerkenswerte Ergebnisse erzielen können (vgl. Abb. 2.11). Diese beiden Ansätze sind nicht wie andere Verfahren auf traditionelle A-priori-Informationen (z. B. geometrische Orientierungspunkte, Texturen, Farben) der Eingabebilder angewiesen, die bei einer starken Reduzierung von Auflösung und Qualität verloren gehen, sondern verwenden eigene durch ihr Training erlernte und generierte [141], [170].

In heutigen SR-Herangehensweisen ist das Verwenden und Anpassen einer Transformer-Architektur, die auf Rekursion verzichtet und stattdessen einen Aufmerksamkeitsmechanismus nutzt, um globale Abhängigkeiten zwischen dem Input und dem Output ausfindig zu machen [136], die gängigste Methode [167].



(a) Eingabebild (b) GFPGAN (c) CodeFormer (d) Eingabebild (e) GFPGAN (f) CodeFormer

Abbildung 2.11: Beispiele für Face Reconstruction mit aktuellen Methoden, die auf GANs basieren. (a) und (b) sind jeweils das Eingabebild mit geringer Qualität, (c) und (d) zeigen die Rekonstruktion mit GFPGAN, (e) und (f) die Ergebnisse mit CodeFormer. (Quelle: In Anlehnung an [170, S. 7, Abb. 4])

Zur Bewertung der unterschiedlichen SR-Verfahren bzw. allgemeinen Beurteilung der Qualität eines Bildes (auf englisch bekannt als *Image Quality Assessment* [IQA]) wurden viele unterschiedliche Metriken entwickelt und verglichen [55], [100], [113], [124], [161]. IQA beinhaltet einerseits qualitative (subjektive) Bewertungsverfahren, die ein Bild auf Grundlage der menschlichen Wahrnehmung bewerten, sowie quantitative (objektive) Verfahren. Diese lassen sich außerdem darin unterscheiden, ob ein Referenzbild (eine s. g. *full-reference*), nur extrahierte Merkmale davon (*reduced-reference*) oder keines von beidem (*no-reference*), also nur das generierte Bild selbst, zur Beurteilung herangezogen wird [55], [80], [147]. Da subjektive Ansätze Menschen benötigen, sind diese teuer und zeitintensiv, weshalb objektive Methoden, wie der *Root Mean Squared Error* (RMSE), das *Peak Signal-to-Noise Ratio* (PSNR) oder *Structure Similarity Index Measure* (SSIM) [147], welche sich automatisch und genau berechnen lassen, in Forschung und Literatur meist vorgezogen werden [55], [80], [164]. Auch in aktuellen SR-Wettbewerben, wie der NTIRE Challenge [2], oder Studien zu neu entwickelten SR-Modellen kommen vor allem PSNR und SSIM zum Einsatz [2], [18], [20], [87]–[89], [94], [133], [138], [144], [167], [171]. Sie vergleichen jeweils das *super-resolved* Bild mit dem hochaufgelösten Original (Referenzbild) [167], entsprechend muss letzteres bei der Evaluation verfügbar sein (full-reference).

Das Peak Signal-to-Noise Ratio (PSNR) ist eine skalierte Variante des *Mean Squared Error* (MSE) (dt. *mittlerer quadratischer Fehler*), der als

$$\mathrm{MSE}(U, V) = \frac{1}{WH} \sum_{x,y} (U_{x,y} - V_{x,y})^2$$

angegeben ist, wenn $U, V \in \mathbb{R}^{W \times H}$ diskrete Bilder sind [11]. Das PSNR wird in Dezibel angegeben und wurde ursprünglich für das Messen von Rauschen bzw. Kompressionsartefakten

entwickelt, nicht um allgemein eine Ähnlichkeit zwischen zwei unterschiedlichen Bildern festzustellen. Bei diskreten Bildern $U, V \in \mathbb{R}^{W \times H}$, bei denen $U_{x,y}, V_{x,y} \in [0, 255]$ gilt[8], ist

$$\mathrm{PSNR}(U, V) = 10 \log_{10} \left(\frac{255^2}{\mathrm{MSE}(U, V)} \right) \mathrm{dB}.$$

Zu beachten ist hierbei, dass ein höherer PSNR-Wert entgegen dem MSE auch eine höhere Bildqualität bedeutet, wobei der Unterschied zwischen zwei Bildern ab 40 dB nicht mehr wahrzunehmen ist [11]. Kleine Pixelverschiebungen wirken sich bereits sehr negativ auf die PSNR (und den MSE) aus, auch wenn die zu vergleichenden Inhalte abgesehen von der Verschiebung identisch sind [2].

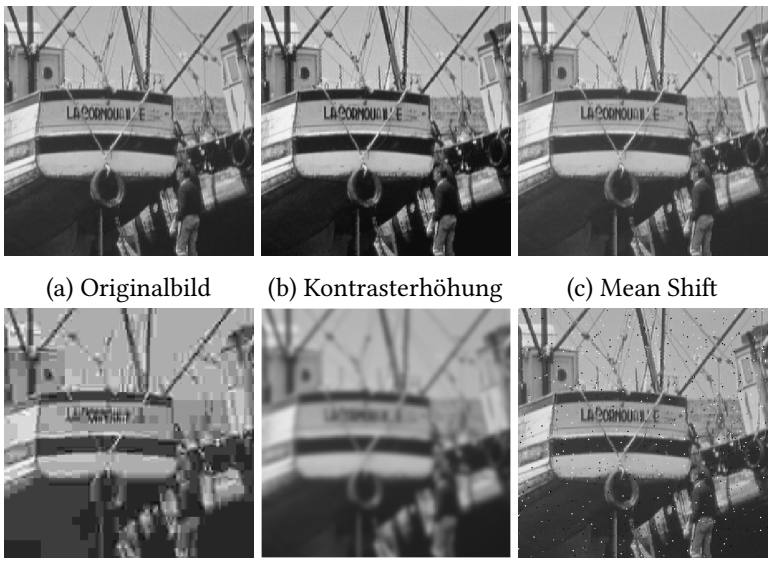
Das Structure Similarity Index Measure (SSIM) soll sich entgegen dem MSE oder PSNR, die zwar einfach zu berechnen sind und eine klare physikalische Bedeutung haben, stärker mit der tatsächlich wahrgenommenen (subjektiven) visuellen Bildqualität decken. So wurde es basierend auf der Annahme entwickelt, dass das menschliche visuelle System sehr effizient darin ist, Strukturen zu erkennen [147]. Wang *et al.* zeigen in [147] an einem Beispiel, das in Abbildung 2.12 betrachtet werden kann, dass Vergleiche, die wie der MSE auf einer Fehlersensitivität beruhen, nicht erklären können, warum manche der gezeigten Bilder eine hohe wahrgenommene Qualität aufweisen, obwohl der messbare Fehler zwischen allen nahezu gleich ist. Es lässt sich allerdings mit dem Ansatz des SSIM erklären, da dieser die Struktur eines Bildes berücksichtigt und bspw. im kontrasterhöhten Bild fast alle strukturellen Informationen erhalten bleiben, es somit auch eine höhere wahrnehmbare Qualität hat. Der Gedanke dahinter ist, dass die Struktur eines Objekts in der Szene unabhängig von Luminanz (Helligkeit) und Kontrast ist, weshalb das System des SSIM ein Bild nach diesen drei Kriterien (Luminanz, Kontrast und Struktur) bewertet. Dabei wird diese Bewertung am besten anhand lokaler Bildausschnitte vorgenommen, nicht global für das gesamte Bild. Die mathematische Herleitung für

$$\mathrm{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

wird detailliert in [147] beschrieben, wobei μ_x,μ_y die Mittelwerte der Pixelintensitäten zweier Bildausschnitte x und y (die Luminanz) repräsentieren, σ_x^2 und σ_y^2 die Varianzen der Pixelintensitäten (den Kontrast) und σ_{xy} die Kovarianz zwischen zwei Bildfenstern (die strukturelle Ähnlichkeit). C_1 und C_2 sind Konstanten, die eingeführt werden, um die Berechnung stabiler zu machen und eine Division durch Null zu vermeiden, vor allem wenn die Mittelwerte und Varianzen sehr klein sind. Da in der Praxis meist ein Wert für die Qualität des gesamten Bildes benötigt wird, verwenden Wang *et al.* [147] einen Mittelwert

$$\mathrm{MSSIM}(X, Y) = \frac{1}{M} \sum_{j=1}^{M} \mathrm{SSIM}(x_j, y_j),$$

[8] Bei einem 8-Bit-Graustufenfarbraum ergeben sich die Werte von 0 bis $2^8 - 1 = 255$.



(a) Originalbild (b) Kontrasterhöhung (c) Mean Shift

(d) JPEG-Kompression (e) Unschärfe (f) Rauschen

Abbildung 2.12: Vergleich verschieden verzerrter Bilder mit gleichem MSE = 210. (a) Original, (b) $MSSIM = 0.9168$, (c) $MSSIM = 0.9900$, (d) $MSSIM = 0.6949$, (e) $MSSIM = 0.7052$, (f) $MSSIM = 0.7748$
(Quelle: [147, S. 603, Abb. 2])

wo X und Y die zu vergleichenden Bilder sind, x_j und y_j der Inhalt des j-en lokalen Fensters und M die Anzahl dieser Fenster. Der höchste SSIM-Wert liegt bei 1, wobei dieses Maximum nur erreicht wird, wenn $X = Y$ [147].
Zusammengefasst kann das PSNR den absoluten Fehler zwischen zwei Bildern gut ermitteln, das SSIM ist ein wahrnehmungsbasiertes Modell, das eine Bildverschlechterung als Veränderung in den strukturellen Informationen betrachtet [2]. Bei zweidimensionalen Bildern steht das SSIM im Gegensatz zu anderen Algorithmen, wie MSE und PSNR, in hoher Korrelation mit den subjektiven Bewertungsverfahren [55].

Doch auch wenn das SSIM (sowie Variationen davon) bereits näher an das Urteil der menschlichen Wahrnehmung kommt, bleibt die Entwicklung einer wahrnehmungsbezogenen Metrik, die misst wie ähnlich sich zwei Bilder sind und mit der menschlichen Beurteilung von Ähnlichkeit übereinstimmt, herausfordernd [164]. Denn diese Beurteilung, so schreiben Zhang *et al.* in *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric* [164], hängt von komplexeren Bildstrukturen ab, sind gleichzeitig kontextabhängig und lassen sich nicht ohne weiteres durch eine Distanz darstellen. Durch die Abhängigkeit von einem Kontext ergeben sich außerdem unterschiedliche Auffassungen von Ähnlichkeit: Ist bspw. ein roter Kreis ähnlicher zu einem roten Quadrat oder zu einem blauen Kreis? Eine Funktion direkt an die menschliche Beurteilung anzupassen ist daher schwierig. Zhang *et al.* entwickeln in ihrer Arbeit 2018 eine neue Metrik, die sie *Learned Perceptual Image Patch Similarity* (LPIPS) nennen

und die auf einer Entdeckung der Computer-Vision-Community basiert: So sind die internen Aktivierungen von CNNs, die für hochrangige Bildklassifizierungsaufgaben trainiert wurden, oft auch nützlich als Repräsentationsraum für eine viel größere Vielfalt von Aufgaben. Zhang *et al.* finden heraus, dass diese Aktivierungen sogar ohne weitere Kalibrierung und über verschiedene Netzwerkarchitekturen mit der menschlichen Wahrnehmung korrespondieren, deutlich besser als bspw. SSIM. Die Ergebnisse der Arbeit deuten also darauf hin, dass CNNs, die für komplexe visuelle Vorhersageaufgaben trainiert werden, eine der menschlichen Wahrnehmung ähnliche Weltrepräsentation erlernen. Ein qualitativer Vergleich von SSIM und LPIPS kann Abb. 2.13 entnommen werden.

Die Autoren entwickeln für ihre Studie einen neuen Datensatz für *Full Reference Image Quali-*

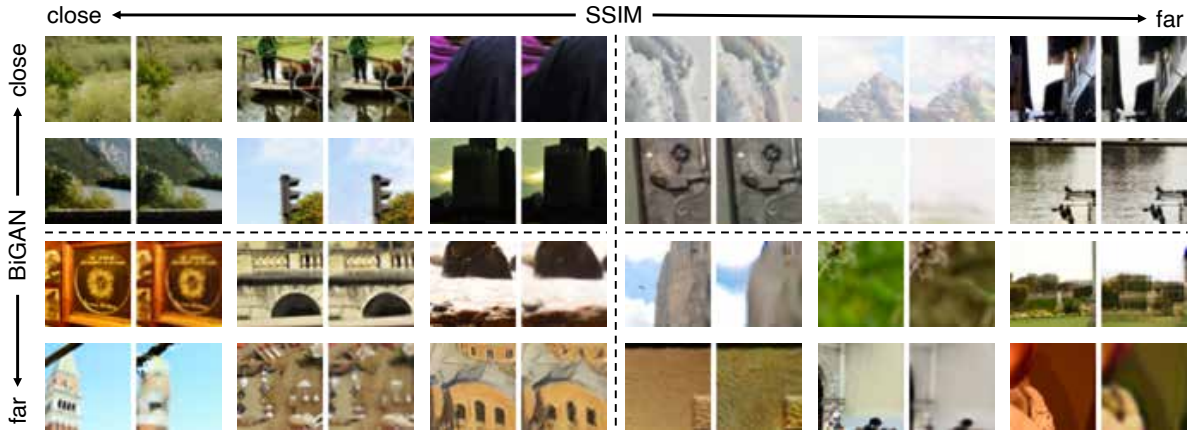


Abbildung 2.13: Qualitativer Vergleich von SSIM und LPIPS anhand von CNN-basierten Verzerrungen. Die LPIPS-Werte wurden mit einem BiGAN [34] ermittelt. Die Grafik zeigt Bildpaare, deren Ähnlichkeit von den beiden Metriken bewertet wird (*close* = höhere Ähnlichkeit, *far* = niedrigere Ähnlichkeit).
(Quelle: [164, S. 6, Abb. 4b])

ty Assessment (FR-IQA), der 484 Tsd. menschliche Beurteilungen von Bildern mit einer Vielfalt an Verzerrung (Filter, Rauschmuster, Kontrast- und Farbveränderungen, algorithmisch generierte Bilder etc.) beinhaltet. Damit trainieren sie bestehende Netzwerke, die einfache lineare Schichtaktivierungen erlernen. Sie zeigen außerdem, dass auch self-supervised Netzwerke ganz ohne menschlich annotierte Trainingsdaten mit der Wahrnehmung von Menschen korrespondieren, dann allerdings mit einer deutlich geringeren Leistung, die aber trotzdem noch den sonst üblichen Metriken, wie SSIM, überlegen ist.

Um nun eine Distanz d zwischen einem Referenzbild x und einer verzerrten Variante x_0 zu berechnen, werden beide in ein wie zuvor erläutertes Netzwerk F eingespeist und verarbeitet. Aus Netzwerk F werden L Schichten extrahiert und deren Aktivierungen in einem *Feature Stack* gesammelt. Die Kanaldimensionen dieser Aktivierungen werden schließlich normalisiert und mit $\hat{F}(x)^l, \hat{F}(x_0)^l \in \mathbb{R}^{H_l \times W_l \times C_l}$ beschrieben, wobei H_l, W_l, C_l jeweils für die Höhe, Breite und Anzahl der Kanäle der l-ten Schicht stehen. Für jede Schicht werden die Aktivierungen

in den Kanälen mit einem Vektor $w^l \in \mathbb{R}^{C_l}$ skaliert, um dann die Euklidische Distanz ℓ_2 zu berechnen. Zum Schluss werden diese Distanzen über die räumlichen Dimensionen gemittelt und über die Kanaldimensionen summiert, was sich durch jede Schicht als

$$d(x, x_0) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \left\| w_l^T \odot \left(\hat{F}(x)^l_{hw} - \hat{F}(x_0)^l_{hw} \right) \right\|_2^2$$

darstellt, wobei $\odot$ die Multiplikation zwischen den Kanälen repräsentiert [164]. Ein geringerer LPIPS-Distanzwert bedeutet entsprechend eine höhere Ähnlichkeit.
Seit der Vorstellung von LPIPS wurde die Metrik oft zum Bewerten der wahrgenommenen Qualität von Bildern verwendet [48] und weiterentwickelt [48], [71]. Auch kann sie als Fehlerfunktion in neuronalen Netzen eingesetzt werden, um bspw. visuell ansprechende Bilder zu generieren [48] oder zum Bestimmen der Diversität [15]. Dennoch findet sich die Metrik in Wettbewerben wie der NTIRE-Challenge nur selten [159], was einerseits historische Gründe (bessere Vergleichbarkeit mit älteren SR-Methoden, da LPIPS erst 2018 vorgestellt wurde) haben könnte, andererseits daran liegen mag, dass PSNR und SSIM eindeutig sowie nachvollziehbar berechenbar sind. In manchen Anwendungen könnten Vergleiche auf Pixelebene wichtiger sein, in anderen der Vergleich in der Wahrnehmung.

Tabelle 2.2: Übersicht geläufiger Super-Resolution-Datensätze.
(Quelle: In Anlehnung an [148, S. 3367, Tab. 1] und [81, S. 4, Tab. 1])

Datensatz	Einsatz	Anzahl	∅ Auflösung	Format	Motive
General-100 [37]	Train	100	435 × 381	BMP	Tiere, Gegenstände, Essen, Menschen, Pflanzen, Texturen etc.
T91 [150]	Train	91	264 × 204	PNG	Autos, Blumen, Früchte, Gesichter etc.
WED [101]	Train	4744	k. A.	PNG	Menschen, Tiere, Pflanzen, Landschaft, Stadt, Stillleben, Verkehr
Flickr2K [132]	Train	2650	2040	PNG	Menschen, Tiere, Umgebung, Stadt, Pflanzen von Flickr
DIV2K [2]	Train/Val	1000	1972 × 1437	JPG	Umwelt, Flora, Fauna, Gegenstände, Menschen etc.
BSDS300 [102]	Train/Val	300	432 × 370	JPG	Tiere, Gebäude, Essen, Landschaft, Menschen, Pflanzen etc.
BSDS500 [3]	Train/Val	500	432 × 370	JPG	Tiere, Gebäude, Essen, Landschaft, Menschen, Pflanzen etc.
RealSR [14]	Train/Val	100	k. A.	JPG	Bilder der realen Welt
OutdoorScene [143]	Train/Val	10624	653 × 440	PNG	Tiere, Gebäude, Berge, Gras, Pflanzen, Himme, Wasser
City100 [17]	Train/Test	100	k. A.	RAW	Stadt
Flickr1024 [146]	Train/Test	100	k. A.	RAW	Stereo-Bilder für Stereo-SR
SR-RAW [169]	Train/Test	7×500	k. A.	JPG/RAW	RAW-Sensor-Daten für Computational Zoom
PIPAL [68]	Test	200	k. A.	PNG	Texturen, Gebäude, Bäume, Gras, Fell, Gesichter, Text etc.
Set5 [6]	Test	5	313 × 336	PNG	Baby, Vogel, Schmetterling, Kopf, Frau
Set14 [158]	Test	14	492 × 446	JPG	Menschen, Tiere, Insekten, Blumen, Gemüse, Comic, Rutschen etc.
BSD100 [102]	Test	100	k. A.	JPG	Tiere, Gebäude, Essen, Landschaft, Menschen, Pflanzen etc.
Urban100 [63]	Test	100	984 × 797	PNG	Architektur, Stadt, Strukturen etc.
Manga109 [46]	Test	109	826 × 1169	PNG	Manga-Bilder
L20 [134]	Test	20	3834 × 2870	PNG	Tiere, Gebäude, Landschaft, Menschen, Pflanzen etc.
PIRM [7]	Test	200	617 × 482	PNG	Umgebung, Flora, Natur, Gegenstände, Menschen etc.

Es gibt eine Reihe an Datensätzen, die für das Training von Super-Resolution-Netzwerken zum Einsatz kommen. Manche davon enthalten bereits Paare aus LR- und HR-Bildern, während bei anderen die LR-Bilder durch Downsampling erst erstellt werden müssen. In Tabelle 2.2

sind SR-Datensätze aufgelistet, die oft verwendet werden [148]. Außerdem kommt es vor, dass neben diesen auch für andere Computer-Vision-Aufgaben eingesetzte Datensätze zum Zeck von SR eingesetzt werden [81], [148].

2.2 Relevante Studien

2.2.1 Bildauflösung in Deep Neural Networks

In „Impact of Image Resolution on Deep Learning Performance in Endoscopy Image Classification: An Experimental Study Using a Large Dataset of Endoscopic Images“ [131] befassen sich Thambawita *et al.* mit dem Einfluss der Bildauflösung auf die Leistung von DNNs bzw. CNNs im Hinblick auf die Klassifikation von endoskopischen Bildern. Sie beziehen sich einerseits auf vorhergegangene Studien wie [119], die eine bessere Modellleistung mit niedrig aufgelösten Daten bzw. nur eine teilweise Steigerung durch höher aufgelöste festgestellt haben und erklären dies mit einer geringeren Anzahl benötigter Parameter, was das Risiko von Overfitting vermeiden kann. Andererseits vermuten sie, dass durch die starke Auflösungsreduzierung (oftmals zum Einsparen von Zeit und Hardware-Ressourcen) auch entscheidende Details verloren gehen, vor allem, wenn die wichtigen Informationen in kleinen Details liegen, was letztlich die Netzwerkleistung schmälern kann und bestätigen dies durch ihre Ergebnisse.
Ausgehend von der Aussage, dass die Auflösung für das Training von CNNs üblicherweise zwischen 64×64 und 256×256 Pixeln liegt, untersuchen sie Bilder in 32×32, 64×64, 128×128, 256×256 und 512×512 Pixeln, wobei die Obergrenze durch den von ihnen verwendeten Hyper-Kvasir-Datensatz [9] festgelegt ist, in dem die höchste gemeinsame Auflösung ebendiese ist. Für die Klassifikation von 23 Klassen nutzen Thambawita *et al.* zwei unterschiedliche Architekturen, ein DenseNet-161 [62] sowie ein ResNet-152 [58], die mit durch ImageNet [28] vortrainierten Gewichten initialisiert werden. Die Wahl dieser grundlegenden Architekturen wird damit begründet, dass das Ziel ihrer Studie ist, den Einfluss der Auflösung auf die Leistung zu beurteilen, anstatt die Methoden mit den höchsten Leistungen aufzuzeigen. Das Training wird mit einer initialen Lernrate von 1e−3 gestartet, die um den Faktor 10 verringert wird, sobald die Modelle nach 25 aufeinanderfolgenden Epochen keine Verbesserung zeigen, und beendet mit einem *Early-Stopping*-Mechanismus, der bei einer Lernrate von 1e−5 greift. Für die Evaluierung werden einerseits dieselben Auflösungen für Trainingsdaten und Testdaten verwendet, als auch unterschiedlich aufgelöste Testdaten bei einheitlicher Auflösung der Trainingsdaten genutzt.
Die Ergebnisse von Thambawita *et al.* zeigen insgesamt, dass eine höhere Auflösung auch zu einer erhöhten Leistung in beiden Modellen bei fast allen Metriken (Precision, Sensitivity, F1-Score, *Matthews Correlation Coefficient* [MCC]) führt. Eine geringe Auflösung führt signifikant zu einer geringeren Leistung, sogar wenn die Verringerung nur relativ klein ist.

Sie stellen fest, dass je größer der Unterschied zwischen der Auflösung des Inputs und den Testdaten ist, desto niedriger fällt die Leistung aus. Bei diesen „mixed resolution cases“ [131, S. 7] wird außerdem beobachtet, dass das Upscaling von niedrig aufgelösten Bildern eine höhere Leistungseinbuße verursacht, als das Verringern der Auflösung von höher aufgelösten. Thambawita *et al.* nennen als Einschränkung ihrer Arbeit, dass für das Training aller Modelle eine feste *Batch-Size* (dt. *Stapelgröße*, siehe auch 2.1.3) von 8 gewählt werden musste, da die von ihnen genutzte Hardware die Verarbeitung von HR-Bildern mit einem größeren Wert nicht zuließ. Als Ausblick mutmaßen sie, dass Super-Resolution die negativen Auswirkungen geringer Bildauflösung auf aktuelle Klassifikationsprobleme reduzieren könne, eine genauere Untersuchung vor allem im Kontext endoskopischer Bilder aber noch ausstehe.

In der Studie „The Effect of Image Resolution on Deep Learning in Radiography“ [119], die ebenfalls in [131] genannt wird, untersuchen Sabottke und Spieler den Einfluss von unterschiedlich aufgelösten Röntgenbildern der Brust in CNNs mithilfe des NIH-ChestX-ray14-Datensatzes [140], dessen Bilder in 1024×1024 Pixel vorliegen. Sie nennen vorausgegangene Studien, die mit unterschiedlichen Auflösungen dieses Datensatzes gearbeitet haben, wobei die Arbeit mit der niedrigsten Auflösung von 224×224 Pixeln im Vergleich eine erhöhte Leistung zeige. Sabottke und Spieler argumentieren (wie später auch Thambawita *et al.*), dass dies paradox erscheinen mag, aber an der geringeren Anzahl zu optimierender Parameter und dem vermeiden von Overfitting liegen kann. Außerdem kann eine durch Hardware-Limitierungen erzwungene geringe Batch-Size aufgrund hoch aufgelöster Bilder dazu führen, dass der Gradient (siehe 2.1.5) nicht so gut berechnet werden kann, wie mit einer höheren Batch-Size, was letztlich zu einer schlechteren Leistung führt. Um die mit bilinearer Interpolation generierten Auflösungen von 32×32, 64×64, 128×128, 224×224, 256×256, 320×320, 448×448, 512×512 und 600×600 gegeneinander zu testen, verwenden sie eine ResNet34-[58] sowie eine DenseNet121-Architektur [62], deren Gewichte mit durch ImageNet [28] vortrainierten Werten initialisiert, aber sämtliche Schichten neu trainiert werden. 20% der für das Training vorgesehenen Daten werden für den Validierungsdatensatz abgespalten und die Modelle bei einer Batch-Size von 8 (aufgrund von Hardware-Beschränkungen) und einer Lernrate von $5e{-}4$ trainiert.
Die Ergebnisse zeigen, dass nur zwei von acht Klassen von einer höheren Auflösungen profitieren (Steigerung der *Area Under the Curve* [AUC]), diese aber zu Befunden gehören, die im Vergleich anhand sehr kleiner Merkmale gestellt werden müssen und damit auch gewisse Bilddetails erfordern. Sabottke und Spieler kommen außerdem zu dem Schluss, dass eine höhere Leistung durch höhere Bildauflösung durchaus gegeben sein könnte, wenn bessere Hardware auch eine größere Batch-Size ermöglichen.

Diese beiden Studien befassen sich zwar nicht mit der speziellen Aufgabe der Facial Expression Recognition, geben aber Hinweise darauf, dass die Auflösung nicht nur eine Rolle bei der Klassifikation von Bildern spielt, sondern eine höhere durchaus positive Auswirkungen haben kann. Das gilt insbesondere, wenn Details für die Klassifizierung eine Rolle spielen,

wie es auch bei Gesichtsausdrücken der Fall sein kann. Beide Arbeiten betonen außerdem die Wichtigkeit einer ausreichenden Bildmenge pro Trainingsschritt, um den Gradienten bestmöglich berechnen zu können, was sich auf die Leistung der Modelle auswirkt.

2.2.2 Super-Resolution in Facial-Expression-Recognition-Tasks

Vo *et al.* befassen sich in ihrer Arbeit [137] mit den unterschiedlichen Auflösungen von In-the-Wild-Datensätzen und der Beobachtung, dass gerade CNNs, die oftmals für Computer-Vision- bzw. FER-Tasks eingesetzt werden, empfindlich auf die Größe der Eingabedaten reagieren. Um ihr Netzwerk robust gegenüber unterschiedlich großen Daten zu machen, wie sie auch in der Realität vorkommen, trainieren sie es mit verschiedenen Auflösungen und setzen dabei, um den Detailgrad niedrig aufgelöster Bilder zu erhöhen und eine Alternative zur in der Forschung üblichen Interpolation zum Erreichen einer einheitlichen Auflösung zu entwickeln, auf einen Block mit Super-Resolution. Sie erarbeiten hierfür eine spezielle Architektur, die sie *Pyramid with Super-Resolution* (PSR) nennen und setzen für den SR-Task auf die EDSR-Architektur von Lim *et al.* [90]. So soll die Fähigkeit des CNNs verbessert werden, relevante Merkmale aus niedrig aufgelösten Bildern zu extrahieren. Nach Vo *et al.* ist die sonst übliche Vorgehensweise, SR bereits in der Datenvorverarbeitung anzuwenden.
Auch, wenn die Autoren das Training mit mehreren Auflösungen vornehmen, zeigen die Ergebnisse, dass Super-Resolution insbesondere die Leistung des Netzwerks bei der Verarbeitung von Bildern mit niedriger Auflösung positiv beeinflussen kann. Da die Ergebnisse sich aber auf die Gesamtleistung des Netzwerks für den FER-Task beziehen, ist nicht genau nachvollziehbar, wie groß der Einfluss von Super-Resolution tatsächlich auf die Leistung ist.

Shao und Cheng [123] entwickeln ein *Edge-Aware Feedback Convolutional Neural Network* (E-FCNN) zum Erkennen von Gesichtsausdrücken in sehr kleinen Bildern ausgehend von der Beobachtung, dass viele FER-Datensätze aus Daten bestehen, die nicht kleiner als 48×48 Pixel aufgelöst sind, Gesichter, die von Überwachungskameras aufgenommen werden, aber oft eine kleinere Auflösung als 40 Pixel je Seite haben. Sie klassifizieren sieben Emotionen in den Datensätzen CK+ [97], FER2013 [51], BU-3DFE [154] und RAF-DB [83], deren Bilder sie auf 16×16 Pixel mittels bikubischer Interpolation downsampeln. Das vorgeschlagene Netzwerk kombiniert SR-Techniken mit einem Block zur Kantenschärfung, da Shao und Cheng argumentieren, dass Kanteninformation in Bildern von Gesichtern die effektivsten Informationen für FER darstellen. Im Kantenverbesserungsblock setzen sie auf SRCNN [36] als SR-Methode, um Texturdetails zu verstärken.
Um ihre Ergebnisse zu evaluieren, vergleichen sie ihr Netzwerk mit anderen zum Zeitpunkt der Studie aktuellen Algorithmen, die ebenfalls mit Bildern in einer Auflösung von 16×16 Pixeln trainiert werden, und kommen zum Schluss, dass ihre Methode die anderen im Bezug auf die *Accuracy* (dt. *Genauigkeit*) übertrifft.

In „Effective image super resolution via hierarchical convolutional neural network“ [91] präsentieren Liu und Ait-Boudaoud ein *Hierarchical Convolutional Neural Network* (HCNN), das sich Kanteninformationen zunutze macht, um die Leistung von Super-Resolution zu verbessern. Die Idee basiert auf ihrer Beobachtung, dass die wahrnehmungsstärksten Merkmale eines Bildes durch Kanten repräsentiert werden und es einfacher ist, diese in degradierten Bildern zu extrahieren als Texturinformationen. Sie vermuten, dass die Erkennung von Kanten auch die Rekonstruktion von HR-Bildern verbessert. Liu und Ait-Boudaoud erklären, dass es trotz sich weiterentwickelnder Bildsensor-Technologie in einigen Szenarios, wie Videoüberwachung oder Human-Computer-Interaction, weiterhin schwierig bleibt, hochauflösende Bilder von Gesichtern zu generieren. Sie nehmen an, dass eine niedrige Auflösung auch die Leistung von bestehenden FER-Algorithmen schmälert und sehen so ein Potenzial in Super-Resolution, die Leistung in vielen Computer-Vision-Tasks zu verbessern. Um dies zu ergründen, untersuchen sie die Leistung ihres SR-Netzwerks am Beispiel eines FER-Tasks. Sie verwenden dafür die Bilder des FER2013-Datensatzes [51] und nutzen für den Vergleich AlexNet [77] als Architektur. Die Ergebnisse zeigen, dass sich die Leistung des Netzwerks deutlich erhöht, wenn die Bilder des Datensatzes vorher mit dem HCNN um den Faktor 2 vergrößert[9] und verbessert werden. Liu und Ait-Boudaoud benennen AlexNet als ein Beispiel zum Bewältigen eines FER-Tasks, gehen aber davon aus, dass sich diese Leistungssteigerung auch auf andere neuronale Netze übertragen lässt.

Jin *et al.* vergleichen in ihrer Arbeit [67] drei unterschiedliche FER-Algorithmen, die auf Convolutional Neural Networks basieren, darunter ein einfaches CNN, MobileNet [60] und RepVGG [33]. Sie verwenden ebenfalls den FER2013-Datensatz. Da sie für das MobileNet eine Eingabegröße von 224×224 Pixeln wählen, die Bilder des Datensatzes aber nur in 48×48 Pixeln vorliegen, was nach Jin *et al.* zu einer Bildvergrößerung und damit einer Reduktion der Qualität und letztlich auch der Leistung führen würde, verwenden sie Super-Resolution mit einem SRGAN [79] zur Vergrößerung und Optimierung. Da die Bilder gemäß ihrer Beobachtung außerdem stark im Helligkeitsgrad variieren, passen sie dies im Anschluss mit einer Graustufennormalisierung an.
Die Ergebnisse zeigen, dass das MobileNet bereits die höchste Accuracy der drei Modelle vorweist, diese aber jeweils leicht durch das Anwenden von SR mit dem SRGAN, der Graustufennormalisierung sowie der Kombination aus beidem verbessert werden kann.

Tan *et al.* entwickeln ein neues Super-Resolution-Verfahren, das sie als *Feature Super Resolution* (FSR) bezeichnen [130] und die diskriminativen Merkmale für ein Modell verbessert statt visuelle Details im reinen Pixelraum. In ihren Experimenten, die sich auf das Wiedererkennen von Bildern (engl. *Image Retrieval*) konzentrieren, vergleichen sie dies mit zwei herkömmlichen Super-Resolution-Methoden (SRCNN [36] und VDSR [72]) und zeigen, dass FSR in diesem Falle besser funktioniert. Gleichzeitig argumentieren sie, dass Super-Resolution im

[9] Die Bilder des FER2013-Datensatzes liegen in Graustufen und einer Auflösung von 48×48 Pixeln vor [51].

Pixelraum (*Image Super-Resolution* [ISR]) nur gute Leistungen bei bereits vergleichsweise wenig in der Auflösung reduzierten Bildern erbringen kann (bis zu einem Downsampling-Faktor von ×4).
In [110] verwenden die Autoren Nan *et al.* ein sehr ähnliches Verfahren, genannt FSR-FER, speziell für den FER-Task. Ihre Ergebnisse bestätigen, dass dieses eine bessere Leistung als ISR erzielt, wenn der Downsample-Faktor mehr als 4 beträgt[10]. Allerdings erklären sie auch, dass ISR dann besser funktioniert, wenn die Bildauflösung nicht allzu stark reduziert ist.

Diese Studien zeigen, dass SR-Techniken einen positiven Einfluss auf die Erkennungsleistung von FER-Systemen haben können, insbesondere bei der Verarbeitung von Bildern mit niedriger Auflösung, wie sie auch in realen Anwendungsszenarien häufig vorkommen. Die genannten Arbeiten betonen die Wichtigkeit der Bildauflösung sowie der Qualität von Bildvorverarbeitungstechniken und liefern Einblicke, wie Super-Resolution gezielt eingesetzt werden kann, um die Leistungsfähigkeit von Deep Neural Network im Bereich der Emotionserkennung zu verbessern.

[10] Ein Downsample-Faktor von 4 ergibt bei Nan *et al.* 25×25 Pixel ausgehend von einer Eingabeauflösung von 100×100 Pixeln [110].

3 Methodik

Um die Effekte der Bildauflösung zu untersuchen und Antworten auf Fragestellung i) sowie ii) zu erhalten, werden zunächst geeignete Datensätze ausgewählt. Ausgehend von der höchstmöglich verfügbaren Auflösung dieser Daten, werden diese auf die zu testenden niedrigeren Auflösungen herunterskaliert und damit die Bildinformation reduziert. Anschließend wird ein Netzwerk mit den sich in der Auflösung unterscheidenden Datensätzen trainiert und die Leistung der trainierten Modelle miteinander verglichen.
Zur Beantwortung von Fragestellung iii), ob sich positive Effekte höherer Bildauflösung (sofern messbar) auch durch Anwendung von Super-Resolution-Verfahren auf LR-Bilder erzielen lassen, werden die auflösungsreduzierten Bilder mit SR vergrößert. Das Verfahren hierfür wird in Unterabschnitt 3.2.4 ausgewählt. Anschließend werden damit die Netzwerke aus den ersten Versuchen erneut trainiert und getestet, um die Ergebnisse mit den nativ höher aufgelösten Bildern zu vergleichen.
Auch wenn sich bereits während den Versuchen zu i) und ii) herausstellen sollte, dass eine höhere Auflösung nicht zu einer verbesserten Leistung von Deep Neural Network führt, werden LR-Bilder mit SR hochgerechnet, um die Effekte zu untersuchen, die solche Verfahren ggf. haben können.

Die Versuche werden mit folgender Hardware durchgeführt:

- **CPU:** AMD Ryzen 5 5600X 6-Core CPU 3.70 GHz
- **GPU:** NVIDIA GeForce RTX 4070 (12 GB)
- **RAM:** 32 GB DDR4-3200

Die Implementierung, das Training und die Auswertung der Netzwerke bzw. Modelle wird mit der Python-basierten Machine-Learning-Plattform *TensorFlow* [103] Version 2.15.0 und der Multi-Framework API *Keras* [22] Version 2.15.0 umgesetzt.

3.1 FER-Datensätze

Für den Versuch wird aus Facial-Expression-Datensätzen ausgewählt, die entsprechend dem kategorischen Modell (siehe Unterabschnitt 2.1.1) mit den Basisemotionen gelabelte Bilder

enthalten. Dieses Modell ist einfach, da es auf einem einzigen statischen Bild je Emotion basiert, und eignet sich dadurch gut zur Untersuchung der Fragestellungen. Eine Übersicht geläufiger Datensätze für Facial Expression Recognition kann Tabelle 3.1 entnommen werden. Es werden insgesamt zwei unterschiedliche Datensätze aus den nachfolgenden Gründen eingesetzt. Als erstes kommt ein In-the-Wild-Datensatz zum Einsatz, da ein solcher, wie einleitend kurz erwähnt, versucht, die Herausforderungen der Realität u. a. durch unterschiedliche Belichtungssituationen, Kopfhaltungen, Verdeckung oder geringer Ausdrucksintensität nachzubilden und entsprechend den aktuellen Forschungsfokus darstellt [84]. Da die Ursprungsdaten eines In-the-Wild-Datensatzes aufgrund der verschiedenen Bildquellen i. d. R. allerdings sehr unterschiedlich in ihrer Auflösung sind und nur durch Down- und Upsampling vereinheitlicht werden, variiert auch der Detailgrad dieser Bilder sehr stark. So könnte es sein, dass diese Vereinheitlichung bereits für eine Verzerrung der Ergebnisse sorgt, vor allem im Bezug auf Frage i). Um letzteres möglichst auszuschließen und einen Vergleich zu haben, wird noch ein zweiter Datensatz verwendet, der unter kontrollierten Bedingungen (Labor) entwickelt wurde und eine konstante Auflösung der Ursprungsbilder vorweist. Dieser dient nur zum Vergleich der Netzwerkleistung hinsichtlich Fragestellung i) und ii), bzgl. des positiven Einflusses einer höheren Auflösung bzw. des optimalen Auflösungsbereichs und wird nicht mit einem Super-Resolution-Verfahren bearbeitet.

Die Auswahl beläuft sich auf die nachfolgend genannten zwei Datensätze, wobei einerseits die Größe der Datensätze (Menge der Proben) als auch die Qualität und Übereinstimmung der Klassen eine Rolle spielen sowie die grundsätzliche Beschaffbarkeit[1] der Datensätze.

3.1.1 AffectNet

Der *AffectNet*-Datensatz (kurz für „**Affect** from the Inter**Net**“ [108, S. 19]) ist nach Angaben der Autoren Mollahosseini *et al.* die größte Datenbank kategorischer (sowie dimensionaler) Emotionsmodelle von *Affect in the Wild*. Sie wurde mit den Suchmaschinen Google, Bing und Yahoo zusammengestellt, wofür 1250 emotionsbezogene Stichwörter in sechs unterschiedlichen Sprachen verwendet wurden. Die gesamte Datenbank der Arbeit in 2019 besteht aus über 1 000 000 Bildern, wovon 450 000 durch zwölf Expert*innen sowohl für das diskrete kategorische als auch das dimensionale Modell manuell gelabelt wurden, wobei laut Mollahosseini *et al.* aus Zeit- und Kostengründen jedes Bild nur von einer Person beschriftet wurde. Der Rest, also die nicht manuell beschrifteten Bilder, wurde automatisiert gelabelt. Um die Übereinstimmung der labelnden Personen zu messen, die bei 60,7 % liegt, wurden 36 000 Bilder von zwei Personen unabhängig beschriftet [108].

Aufgrund der Größe der AffectNet-Datenbank (122 GB) wurde zusätzlich eine kleinere Version

[1] Aufgrund des Alters der Datensätze und den zugehörigen Forschungsarbeiten sind manche Websites, Kontaktformulare oder Emailadressen zum Beantragen eines Zugriffs nicht mehr funktionsfähig oder erreichbar.

Tabelle 3.1: Übersicht geläufiger Facial-Expression-Datensätze.
(Quelle: In Anlehnung an [84, S. 1197, Tab. 1]) und [120, S. 9807, Tab. 1]

Datensatz	Jahr	Inhalt	Motive	Beding.	Auflösung	Ausdrücke
JAFFE [99]	1998	213 Bilder	10	Labor P	562 × 762	6 Basisemotionen + Neutral
MMI [111], [135]	2005, 2010	740 Bilder, 2900 Videos	25	Labor P	640 × 480	7 Basisemotionen + Neutral
BU-3DFE [154]	2006	2500 3D-Bilder	100	Labor P	1040 × 1329	6 Basisemotionen + Neutral
BU-4DFE [153]	2008	606 3D-Sequenzen	101	Labor P	1040 × 1329	6 Basisemotionen + Neutral
Multi-PIE [54]	2008	755 370 Bilder	337	Labor P	k. A.	Schrei, Überraschung, Ekel, Blinzeln, Lächeln, Neutral
Oulu-CASIA [127]	2008	2880 Bildsequenzen	80	Labor P	320 × 240	6 Basisemotionen
KDEF [49]	2008	4900 Bilder	70	Labor P	562 × 762	6 Basisemotionen + Neutral
CK+ [97]	2010	593 Bildsequenzen	123	Labor P & S	640 × 490	7 Basisemotionen + Neutral
TFD [126]	2010	112 234 Bilder, davon 4178 gelabelt [120]	k. A.	Labor P	48 × 48	6 Basisemotionen + Neutral
RaFD [78]	2010	8040 Bilder	49	Labor P	1024 × 681	7 Basisemotionen + Neutral
DEAP [75]	2012	40 Videos	22 (32)	Labor S	640 × 490	16 Emotionen (zzgl. physiologische Daten)
FER2013 [51]	2013	35 887 Bilder	k. A.	Internet P & S	48 × 48	6 Basisemotionen + Neutral
SFEW 2.0 [32]	2015	1766 Bilder	k. A.	Film P & S	720 × 576	6 Basisemotionen + Neutral
EmotioNet [5]	2016	1 000 000 Bilder	k. A.	Internet P & S	k. A.	23 Basis- & Compound-Emotionen
FER-Wild [109]	2016	19 562 Bilder	k. A.	Internet P & S	k. A.	6 Basisemotionen + Neutral
AFEW 7.0 [31]	2017	1809 Videos	k. A.	Film P & S	k. A.	6 Basisemotionen + Neutral
ExpW [168]	2017	91 793 Bilder	k. A.	Internet P & S	k. A.	6 Basisemotionen + Neutral
RAF-DB [83], [85]	2019, 2017	29 672 Bilder	k. A.	Internet P & S	k. A.	6 Basisemotionen + Neutral, 12 Compound-Emotionen
4DFAB [21]	2018	1,8 Mio. 3D-Gesichter	180	Labor P & S	k. A.	6 Basisemotionen + Neutral
AffectNet [108]	2019	450 000 Bilder, davon 291 651 verfügbar	k. A.	Internet P & S	224 × 224	7 Basisemotionen + Neutral

Beding. = Erhebungsbedingungen; P = posiert; S = spontan

Neutral (17197.jpg) | Freude (6135.jpg) | Traurigkeit (18255.jpg) | Überrascht (3190.jpg) | Angst (1054.jpg) | Ekel (23588.jpg) | Wut (15881.jpg) | Verachtung (49619.jpg)

Abbildung 3.1: Beispiele aus dem AffectNet-Datensatz.
(Bildquelle: [108])

veröffentlicht, die nur die manuell gelabelten Bilder mit acht kategorischen Klassen (Neutral, Freude, Traurigkeit, Überraschung, Angst, Ekel, Wut, Verachtung), Werten für Valenz & Wertigkeit sowie 68 Orientierungspunkten im Gesicht (engl. *facial landmarks*) beinhaltet und bereits in einen Trainings- sowie Validierungsdatensatz aufgeteilt wurde [56]. Abbildung 3.1 zeigt je ein Beispiel für jede Klasse des Datensatzes. Wie Hasani *et al.* erklären, wird bislang ein Testdatensatz zurückgehalten, um in der Zukunft einen Wettbewerb ausrichten und so dessen Ergebnisse evaluieren zu können. In der für diese Arbeit genutzten und vorliegenden, kleinen Variante sind 291 651 Bilder enthalten. Die offiziell als Validierungsdatensatz veröffentlichten Daten dienen in dieser Arbeit, wie auch in vielen anderen [47], [122], [137], [139], [157], als Testdatensatz, sind damit nicht Bestandteil des Modelltrainings und werden nur zur Evaluierung der fertig trainierten Modelle verwendet. Dieser Testdatensatz ist mit 500 Bildern je Klasse[2] ausbalanciert, im Gegensatz zu den Trainingsdaten (siehe hierzu die Klassenverteilung in Tab. 3.2).

Tabelle 3.2: Klassenverteilung im AffectNet-Datensatz (Trainingsset)

Klasse	**Definition**	**Anzahl**	**Anteil, %**
0	Neutral	74 874	26,0
1	Freude	134 415	46,7
2	Traurigkeit	25 459	8,9
3	Überraschung	14 090	4,9
4	Angst	6378	2,2
5	Ekel	3803	1,3
6	Wut	24 882	8,7
7	Verachtung	3750	1,3
Gesamt		287 651	100

Die Bilder liegen im RGB-Farbraum[3] in einer Auflösung von 224 × 224 Pixeln im JPG-

[2]Eine Ausnahme bildet die Klasse zu „Verachtung“, die 499 Bilder enthält.
[3]Der RGB-Farbraum ist der meist genutzte Farbraum für digitale Bilder. Er kodiert eine Farbe als additive Kombination der drei Primärfarben Rot (R), Grün (G) und Blau (B) [86].

Format vor, ausgehend von den Ursprungsbildern mit einer durchschnittlichen Auflösung von 425 × 425 Pixeln und einer Standardabweichung von 349 × 349 Pixeln. Vo *et al.* zeigen in ihrer Arbeit [137] ein Diagramm, dass die Auflösungsverteilung aller Bilder des Datensatzes darstellt und in Abbildung 3.2 betrachtet werden kann. Dabei ist allerdings unklar, auf welchen Teil des Datensatzes (gesamte Datenbank, kleine Version oder nur Test- und Validierungsmenge) die Grafik sich bezieht und woher die Informationen kommen, da sie nicht von Mollahosseini *et al.* in [108] angegeben werden.

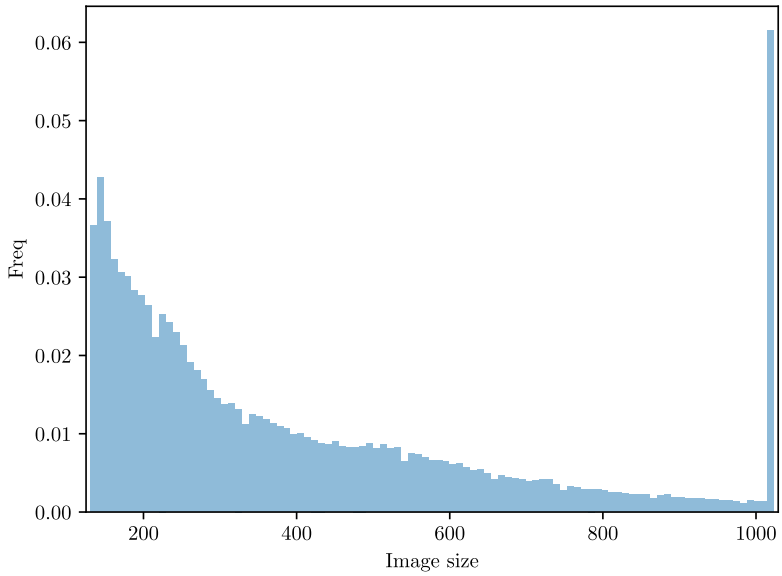


Abbildung 3.2: Auflösungsverteilung der Bilder von AffectNet.
(Quelle: [137, S. 131989, Abb. 1b])

Durch seine große Datenmenge bei vergleichsweise hoher Auflösung von In-the-Wild-Bildern eignet sich der AffectNet-Datensatz gut, um damit ein DNN für den geplanten Versuch zu trainieren.

3.1.2 RaFD

Als geeigneter Labordatensatz wird *RaFD* (kurz für „**Ra**dboud **F**aces **D**atabase“) ausgewählt, der von Langner *et al.* entwickelt wurde und nicht nur wie AffectNet die sieben Basisemotionen sowie Neutral enthält, sondern pro Ausdruck auch drei unterschiedliche Blickrichtungen (links, frontal, rechts) und fünf Aufnahmewinkel (0° [Seitenprofil rechts], 45°, 90° [frontal],

135°, 180° [Seitenprofil links]) [78]. Das ergibt bei 67 unterschiedlichen Fotomodellen[4] insgesamt 8040 RGB-Bilder (120 pro Modell, bzw. 735 pro Klasse; im JPG-Format) im Datensatz. Die Verteilung der Daten nach angegebenen Diversitätskategorien, kann Tabelle 3.3 entnommen werden. Dieser ist also bzgl. seiner Klassenverteilung ausbalanciert. Die Daten haben außerdem aufgrund einer einheitlichen Belichtungssituation und Kleidung sowie einer konstanten Auflösung von 1024 × 681 Pixel eine hohe Datenqualität. Da die Label bereits mit der Aufnahme der Daten feststanden, wurden diese im Nachgang noch einmal von 276 Studierenden validiert, sodass jedes Bild mind. zwanzig Mal bewertet wurde. Bzgl. der Gesichtsausdrücke ergab es über alle Klassen eine durchschnittliche Übereinstimmung von 82 % [78]. Beispiele des Datensatzes können Abb. 3.3 entnommen werden.
Die Anzahl der Daten ist zwar deutlich kleiner als bei AffectNet, was letztlich auch die Ergebnisse des trainierten Netzwerks beeinträchtigt, dennoch lässt sich mit den Ergebnissen der relative Leistungsunterschied zwischen den Auflösungstests vergleichen.

Neutral (A1.1.1) | Freude (A1.1.2) | Traurigkeit (A1.1.3) | Überrascht (A1.1.4) | Angst (A1.1.5) | Ekel (A1.1.6) | Wut (A1.1.7) | Verachtung (A1.1.8)

Abbildung 3.3: Beispiele aus dem RaFD-Datensatz.
(Bildquelle: [78], siehe genaue Dateibezeichnung in Unterabschnitt A1.1)

Tabelle 3.3: Verteilung der RaFD-Fotomodelle nach Diversitätskategorien. Pro Fotomodell enthält der Datensatz 120 Bilder.

Kategorie	**Geschlecht**	**Anzahl**	**Anteil Kategorie, %**	**Anteil gesamt, %**
Caucasian	female	19	48,7	58,2
	male	20	51,3	
Kid	female	6	60,0	14,9
	male	4	40,0	
Moroccan	male	18	100,0	26,9
Gesamt		67		100,0

[4] Langner *et al.* geben in [78] 49 Modelle („[...] 39 Caucasian Dutch adults [...], and 10 Caucasian Dutch children [...]“ [78, S 1378.]) an, im vorliegenden Datensatz sind zusätzlich 18 Fotomodelle enthalten, denen die Kategorie „Moroccan“ zugeordnet ist.

3.1.3 Datensplit

Wie unter 3.1.1 beschrieben, wird der offizielle Validierungsdatensatz von AffectNet als Testdatensatz verwendet. Außerdem wird vom Trainingsdatensatz eine Teilmenge zur Validierung abgespaltet, die wie die Testdaten eine ausbalancierte Klassenverteilung aufweist. Dadurch wird gewährleistet, dass das Modell auf ähnliche Daten hin optimiert wird, wie die, mit denen es später auch getestet wird. Die Abspaltung der Teilmenge von den Trainingsdaten wird auf Grundlage der Klasse mit der geringsten Datenmenge vorgenommen, was Klasse 7 (Verachtung) entspricht (siehe 3.2). Hiervon werden 10 % berechnet, was bei 3750 Bildern 375 sind, und diese Anzahl Bilder schließlich von jeder Klasse zufällig abgespaltet. Der Trainingsdatensatz besteht so aus 284 651 Daten, bei einer weiterhin unbalancierten Klassenverteilung, der nun neue Validierungsdatensatz aus $375 \times 8 = 3000$ Bildern.

RaFD enthält keinen vordefinierten Datensplit, weshalb Trainings-, Validierungs- und Testdatensatz erst erstellt werden müssen. Da der Datensatz mehrere Bilder derselben Person enthält, wird die Aufteilung nicht allein von der absoluten Zahl der Bilder im Datensatz, sondern von der Anzahl der Fotomodelle abhängig gemacht. Dabei wird darauf geachtet, dass jeweils alle Bilder mit allen dargestellten Emotionen einer Person im selben Teildatensatz enthalten sind. So wird vermieden, dass das Netzwerk im Trainingsprozess nur bestimmte Emotionen eines Fotomodells bzw. eine verzerrte Verteilung der Klassen zu einer Person erlernt. Die Klassenverteilung insgesamt ist ausbalanciert, weshalb auch die Teilmengen ausbalanciert werden. Auch die Diversität (Kategorie und Geschlecht) des Gesamtdatensatzes wird möglichst beibehalten. Für den Trainingsdatensatz werden 43 der Fotomodelle verwendet, für Validierungs- und Testdatensatz jeweils 12. Das entspricht grob einer Verteilung von ~64 % fürs Training, ~18 % zur Validierung und ~18 % zum Testen. Tabelle 3.4 zeigt weitere Details zur Verteilung der Daten.

Tabelle 3.4: Aufteilung der RaFD-Fotomodelle nach Diversitätskategorien und Datensatz-Splits.

Kategorie	**Geschlecht**	**Anzahl**			**Anteil Kategorie, %**			**Anteil gesamt, %**		
		Train	**Val**	**Test**	**Train**	**Val**	**Test**	**Train**	**Val**	**Test**
Caucasian	female	13	3	3	52,0	42,9	42,9	58,1	58,3	58,3
	male	12	4	4	48,0	57,1	57,1			
Kid	female	4	1	1	66,7	50,0	50,0	14,0	16,7	16,7
	male	2	1	1	33,3	50,0	50,0			
Moroccan	male	12	3	3	100,0	100,0	100,0	27,9	25,0	25,0
Gesamt		43	12	12				100,0	100,0	100,0

3.2 Bildvorverarbeitung

3.2.1 Auflösungen

Wie in Unterabschnitt 3.1.1 beschrieben, liegen die Bilder des AffectNet-Datensatzes in 224×224 Pixeln vor, was entsprechend die höchstmögliche verfügbare Auflösung dieser Daten darstellt. Ausgehend davon werden die niedrigste Auflösung von 48×48 sowie einer mittleren von 112×112 Pixeln generiert. 48 Pixel stellen dabei die untere Grenze dar, da einerseits die kleinste Auflösung der üblichen FER-Datensätze (siehe bspw. FER2013 [51] oder TFD [126]) bei dieser Untergrenze liegt und andererseits ein Großteil der unter Unterabschnitt 3.2.4 erläuterten SR-Verfahren darauf ausgelegt ist, einen maximalen Vergrößerungsfaktor von $\times 4$ zu leisten. Das entspricht $48 \times 4 = 192$ Pixel. Für das Erreichen der Obergrenze von 224 Pixeln muss also durch Interpolation weiter upgesampelt werden.
Die Daten des RaFD-Datensatzes besitzen in ihrer unveränderten Form eine höhere Auflösung, womit auch das obere Limit theoretisch höher als bei den AffectNet-Bildern liegt. Wie Abbildung 3.3 entnommen werden kann, haben die Bilder von RaFD aber einerseits ein anderes Seitenverhältnis sowie andererseits einen ganz anderen Motivausschnitt und viel „motivfreie Fläche“ um die relevanten Gesichtsmerkmale herum. Damit die relative Leistung zwischen den mit unterschiedlichen Datensätzen trainierten Modellen vergleichbar bleibt, werden die Bilder des RaFD-Datensatzes quadratisch zugeschnitten und dieselben drei Auflösungen erstellt. Zusätzlich werden die RaFD-Daten auch auf 280×280 Pixel gebracht, um bzgl. Fragestellung i) noch eine Auflösung über der bei AffectNet maximal verfügbaren zu testen. Durch den Zuschnitt der Gesichter (engl. *Face Cropping*) reduziert sich die Auflösung im Vergleich zu den Ausgangsbildern von 1024×681 Pixeln sehr stark (in Unterabschnitt 3.2.2 näher beschrieben), weshalb der Test eines viel größeren Formats nicht möglich ist.

Auf Basis der downgesampelten Bilder in 48×48 Pixeln, werden dann wiederum mit den nachfolgend definierten Verfahren die jeweils höheren Auflösungen (112×112 und 224×224 Pixel, sofern möglich auch 448×448 Pixel) erstellt und zusätzlich ausgehend von den in 224×224 Pixel vorliegenden Originalbildern noch Bilder in 448×448 Pixel generiert, um zu beobachten, ob eine noch höher generierte Auflösung ausgehend von den meisten verfügbaren Bildinformationen für einen Leistungszuwachs sorgen kann.
Die genauen Kombinationen von verwendeten Auflösungen nach Verfahren für Modell-Training und -Tests können Tabelle 3.9 entnommen werden.

Die Auflösungen sind so gewählt, dass diese durch acht teilbar sind, was mit der Verarbeitung durch das Netzwerk (siehe Abschnitt 3.4) zu tun hat.

3.2.2 Zuschnitt RaFD

Zum Beschneiden der RaFD-Bilder wird der *Face Detector* von Googles *MediaPipe* [98], [163] in der Version 0.10.11 verwendet. Dieser ermöglicht die automatische Gesichtserkennung in Bildern durch Nutzung eines vortrainierten *BlazeFace-Modells (short-range)*, das eine *Single-Shot-Detector*-Technik [93] verwendet, einem sehr effizienten und schnellen CNN für Objekterkennung [96], [163]. Die durch den Face Detector erkannten Bildbereiche haben eine durchschnittliche Auflösung von 310, 97 × 310, 97 Pixeln mit einer Standardabweichung von 29,71 Pixeln in einem Bereich von 205 × 205 bis 403 × 403 Pixeln. Zum Sicherstellen eines quadratischen Zuschnitts wird jeweils die längste Seite des Erkennungsbereichs als Zuschnittswert für Breite und Höhe genommen. Anschließend werden die quadratischen Formate durch bikubische Interpolation auf eine einheitliche Größe skaliert, sodass sie nach der Verarbeitung eine Auflösung von 224×224 (bzw. für die Zusatzauflösung 280×280) Pixeln haben. Beispiele der bereits in Abbildung 3.3 verwendeten Bilder können als beschnittene Variante in Abbildung 3.4 betrachtet werden (vgl. Abb. 3.1 der AffectNet-Bilder).

Neutral (A1.1.1)	Freude (A1.1.2)	Traurigkeit (A1.1.3)	Überrascht (A1.1.4)	Angst (A1.1.5)	Ekel (A1.1.6)	Wut (A1.1.7)	Verachtung (A1.1.8)

Abbildung 3.4: Zugeschnittene Beispiele aus dem RaFD-Datensatz.
(Bildquelle: [78], siehe genaue Dateibezeichnung in Unterabschnitt A1.1)

Die Zusatzauflösung von 280 × 280 Pixeln wird deshalb der durchschnittlichen Auflösung von 310 × 310 Pixeln der Zuschnitte vorgezogen, da sonst 3547 Bilder (44,12 %) hochskaliert werden müssten, während nur 4493 Bilder (55,88 %) nativ mind. 310 × 310 Pixel beinhalten. Fast die Hälfte müsste somit leicht interpoliert werden, was die Ergebnisse stärker beeinträchtigen könnte, als bei der Wahl von 280 × 280 Pixeln. Hier müssen nur 1223 Bilder (15,21 %) hochskaliert werden, weil bereits 6817 Bilder (84,79 %) mindestens diese Auflösung haben.

3.2.3 Down- & Upsampling

Müssen Bilder interpoliert werden, findet das Down- sowie Upsampling dieser Bilder grundsätzlich mit bikubischer Interpolation statt (siehe Unterabschnitt 2.1.6). Dafür wird zum einen die *Pillow*-Bibliothek [25] (`Resampling.BICUBIC`) in Version 10.2.0 verwendet. Pillow erweitert den Python-Interpreter um Bildverarbeitungsfunktionen [25]. So werden sowohl die Ausgangsbilder als auch die durch die SR-Netzwerke vergrößerten Bilder (soweit nötig) neu berechnet, um die zuvor unter Unterabschnitt 3.2.1 aufgeführten Auflösungen zu erhalten.

Zum anderen werden die durch den Zuschnitt der RaFD-Daten (vgl. Unterabschnitt 3.2.2) entstehenden unterschiedlich großen Formate im Zuge der Verarbeitung durch OpenCV [10] (`INTER_CUBIC`) in Version 4.9.0, einer Open-Source-Bibliothek für Computer Vision und Machine Learning, auf die Ausgangsgröße von 224×224 Pixeln gebracht.
Das Verfahren zum Upsampling mit Super-Resolution wird im nachfolgenden Unterabschnitt ausgewählt und erläutert.

3.2.4 Super-Resolution-Verfahren

Um ein geeignetes Super-Resolution-Verfahren zu finden, werden verschiedene miteinander verglichen. Die Vorauswahl der nachfolgend vorgestellten Methoden basiert auf Aktualität, Relevanz und Leistung sowie der freien Verfügbarkeit des zur Verwendung benötigten Programmcodes. Es handelt sich dabei sowohl um allgemeine Super-Resolution-Verfahren als auch solche, die speziell auf Face Restoration ausgelegt sind.

- *SwinIR* [89] wurde 2021 von Liang *et al.* vorgestellt und basiert auf dem Swin Transformer [95]. Das Netzwerk besteht aus drei Teilen: einem Modul zur Extraktion von oberflächlichen Features, einem zur Extraktion von tiefen und einem Rekonstruktionsmodul. Das erste nutzt eine Convolution-Schicht und übermittelt die Features direkt an das Rekonstruktionsmodul, um niedrig frequente Bildinformationen zu erhalten. Das zweite Modul besteht hauptsächlich aus s. g. *Residual Swin Transformer Blocks* und einer Convolution-Schicht. Im Rekonstruktionsmodul werden dann die extrahierten Features für die Bildrekonstruktion zusammengeführt. Vor SwinIR wurden nach Angaben von Liang *et al.* nur wenige Versuche zur Nutzung von Transformern für Super-Resolution gemacht. Das Netzwerk wurde mit DIV2K und Flickr2K[5] (vgl. Tab. 2.2) trainiert [89]. Das Modell übertrifft zahlreiche der bis 2021 bestehenden SR-Methoden [81]. Und auch viele der in aktuellen Wettbewerben vorgestellten Methoden basieren auf dieser Architektur [167]. Für das Training eines zusätzlichen Modells zur Anwendung von *Real-World-Super-Resolution*, also Super-Resolution von Bildern mit unbekannter Degradierung, setzt SwinIR auf das Degradierungsmodell von *BSRGAN* [160], das low-quality (LQ)-Bilder synthetisiert [89], [160].
- *Real-ESRGAN* [142] ist eine Erweiterung von ESRGAN [144] mit der Absicht bessere Ergebnisse in realen Szenarien zu erzielen, in denen die Degradierung unbekannt ist. Laut Wang *et al.*, die Real-ESRGAN 2021 vorgestellt haben, basiert der Großteil der bis dahin entwickelten SR-Methoden auf der Annahme, dass die LR-Bilder bikubisch downgesampelt wurden. In echten Szenarien ist aber nicht bekannt, welche Degradierung stattgefunden hat. Der Algorithmus muss also „blind“ arbeiten. Um das

[5] Die Kombination aus DIV2K und Flickr2K wird auch als *DF2K* bezeichnet [18], [171].

Potenzial von ESRGAN weiter zu verbessern, wird ein hochwertiges Degradationssmodell eingeführt, das verschiedene Echtweltdegradierungen (z. B. Gaußsches Rauschen, Poisson-Rauschen, JPEG-Kompression, Ringing- und Overshoot-Artefakte) für die Daten von DIV2K, Flickr2K und OutdoorScene simuliert, und das Modell so mit rein synthetischen Daten trainiert. Dadurch kann es viel besser echte Bilder restaurieren als Algorithmen, die auf bikubisches Downsampling optimiert sind, was sich auch in seiner überlegenen Leistung zu früheren Arbeiten zeigt [142].

- *GFPGAN* [141], entwickelt und vorgestellt von Wang *et al.* in 2021, nutzt ***G**enerative **F**acial **P**riors* zur Face Restoration aus für Gesichtsgenerierung vortrainierten GANs, statt wie sonst übliche Methoden auf geometrisches A-priori-Wissen oder Referenzbilder zurückzugreifen. Geometrische Priors sind zwar entscheidend zur Wiederherstellung von genauen Gesichtsformen und -details, werden aber i. d. R. anhand von den Eingabebildern geschätzt und verschlechtern sich damit nicht nur unweigerlich bei Eingaben von sehr niedriger Qualität, sondern beinhalten auch nur wenige texturbezogene Informationen zu Details wie bspw. Pupillen. Ansätze mit Referenzpriors hingegen bieten meist nur eine stark limitierte Diversität und Reichhaltigkeit von Gesichtsdetails. Durch die Verwendung der in großen vortrainierten GANs enthaltenen Informationen über Gesichter kann GFPGAN diese Limitierungen überwinden und damit eine hohe Bildqualität bei gleichzeitig hoher Originaltreue (engl. *fidelity*) erzielen. GFPGAN wurde mit FFHQ [70] trainiert, einem Datensatz, der 70 000 high-quality (HQ)-Bilder von Gesichtern in einer Auflösung von 1024 × 1024 enthält [70], die von der Fotoplattform Flickr[6] stammen. Das Netzwerk nutzt zusätzlich ein Modul zum Entfernen von Bilddegradierungen [141]. Da sich das eigentliche Netzwerk auf die Gesichtsrekonstruktion fokussiert, wird im Programmcode zusätzlich auf RealESRGAN zum optionalen Upsampeln des Hintergrunds zurückgegriffen (siehe auch die Quelle unter A1.2.3).
- *HAT* [18] basiert auf demselben Architekturdesign wie SwinIR hinsichtlich der drei Module sowie der Tiefe und Breite des Netzwerks. Chen *et al.*, die den Ansatz 2022 vorgestellt haben, stellen sich in ihrer Arbeit die Frage, warum der Einsatz des Swin Transformers bessere Ergebnisse als CNN-basierte Methoden erzielt und untersuchen, welche Eingabepixel am meisten Einfluss auf ausgewählte Regionen eines SR-Ergebnisses haben. Die intuitive Annahme dabei ist, dass eine größere Menge verwendeter Eingabeinformation auch eine bessere SR-Leistung erzielt. Sie stellen fest, dass dies zwar bei der Verwendung von CNNs (hier EDSR [90] und RCAN [166]) der Fall ist, SwinIR als Beispiel für eine Transformer-basierte Methode allerdings keinen größeren Bereich der Verwendung von Bildinformationen aufzeigt. Das deutet darauf hin, dass SwinIR eine deutlich stärkere Zuordnungsfähigkeit besitzt und daher weniger Informationen für einer besseren Leistung benötigt. Außerdem vermuten

[6] siehe https://flickr.com/

Chen *et al.*, dass SwinIR aufgrund der begrenzten Verwendung von Eingangspixeln möglicherweise falsche Texturen herstellt und stellen mit HAT ein Netzwerk vor, dass dies verbessert, indem deutlich mehr Pixel für die Bildrekonstruktion aktiviert werden. Sie verwenden hierfür eine Kombination aus *kanalbezogenen Attention-Mechanismen* sowie *Self-Attention-Mechanismen* und integrieren einen *Cross-Attention-Block*, der die Interaktion benachbarter Fenstermerkmale verstärkt und dazu beiträgt Blockartefakte, die bei SwinIR durch den *Shifted-Window-Mechanismus* hervorgerufen werden, zu verringern. HAT wurde wie SwinIR mit den DIV2K- und Flickr2K-Daten trainiert [18] und bietet oft auch die Grundlage für in aktuellen Wettbewerben entwickelte SR-Modelle [167]. Um mit Echtweltdegradierungen umgehen zu können, ist mittlerweile auch ein RealESRGAN-basiertes Modell verfügbar [19] (siehe auch A1.2.4).

- *CodeFormer* [170] ist ein Transformer-basierter Ansatz für Blind Face Restoration, vorgestellt von Zhou *et al.* in 2022. Die Arbeit setzt auf die innovative Methode, Face Restoration als Vorhersage von Codes in einem kleinen, endlichen Proxy-Raum bzw. einer Zwischenrepräsentationsebene zu betrachten. Dieser Raum wird bei CodeFormer als vortrainiertes, diskretes Codebuch dargestellt, wobei jeder Code für eine spezifische Kombination von Merkmalen oder Texturen steht, die in hochqualitativen Bildern von Gesichtern vorkommen. Das hat den Vorteil, dass das Netzwerk nicht so stark vom Input abhängt, wie bspw. GFPGAN. Durch die sorgfältig ausgewählte Sammlung von Bildmerkmalen wird das Risiko von Artefakten oder unnatürlichen Texturen, wie sie bei GANs auftreten können, verringert. Durch die begrenzte Kardinalität des Proxy-Raums werden außerdem Unsicherheiten in der Abbildung der degradierten und niedrigqualitativen (engl. *low-quality* [LQ]) Eingaben auf hochqualitative (engl. *high-quality* [HQ]) Ausgaben deutlich reduziert, womit das System robuster ggü. unterschiedlichsten Degradierungen wird. CodeFormer nimmt also LQ-Eingaben und sagt Code-Sequenzen voraus, die als diskrete Repräsentation der Gesichtsbilder im Codebuch-Raum betrachtet wird. Als zusätzliche Besonderheit wird ein kontrollierbares Modul mit anpassbarem Koeffizienten eingeführt, das den Informationsfluss vom LQ-Encoder zum Decoder steuern kann. So wird mit einem höheren Koeffizienten und damit einem höheren Zufluss an LQ-Merkmalen des Inputs zwar eine höhere Originaltreue erreicht, aber ggf. eine geringere Qualität. Umgekehrt lässt sich für eine höhere Qualität sorgen, wenn ein niedrigerer Koeffizient gewählt wird, da weniger der degradierten Merkmale einfließen, was aber für weniger Originaltreue sorgt. CodeFormer wurde wie GFPGAN mit FFHQ [70] trainiert [170] und greift im Programmcode ebenfalls auf RealESRGAN zurück, um optional eine Möglichkeit zu bieten, den Hintergrund eines Bildes zu verbessern.
- *DAT* [20], vorgestellt von Chen *et al.* in 2023, setzt wie SwinIR und HAT ebenfalls auf eine Transformer-Architektur mit drei Modulen zur Extraktion von oberflächlichen Merkmalen, tiefen Merkmalen und Bildrekonstruktion. Zusätzlich führt das Netzwerk ***D**ual **A**ggregation **T**ransformer Blocks* ein, die das zentrale Element des Verfahrens

bilden. Diese verwenden abwechselnd Mechanismen der *Spatial Window Self-Attention* und der *Channel-wise Self-Attention*, wodurch lokale als auch globale Abhängigkeiten der Eingabe erfasst und zusammengeführt werden können, was eine umfassendere Bildanalyse und damit eine bessere Rekonstruktion ermöglicht [171].

- *SRFormer* [171] basiert auch auf einer Transformer-Architektur und wurde 2023 von Zhou *et al.* auf Grundlage der Beobachtung vorheriger Arbeiten entwickelt, dass größer gewählte Bildfenster zur Verarbeitung und Erfassung von Abhängigkeiten durch die Attention-Mechanismen auch signifikant zu einer höheren Modellleistung führen, damit aber auch für einen höheren Rechenaufwand verantwortlich sind. Im Kern der Arbeit steht ein neuer *Permuted-Self-Attention*-Mechanismus, der es durch das Transferieren räumlicher Informationen in die Kanaldimension erstmals erlaubt, einen 24×24 großen Fenster-Aufmerksamkeitsmechanismus bei einer akzeptablen Rechenzeit zu implementieren. Der Ansatz zeigt Leistungsverbesserungen bei weniger Parametern und geringerem Rechenaufwand im Vergleich zu früheren Modellen [171].

SwinIR, HAT, DAT und SRFormer sind SR-Verfahren, deren Standardmodelle am besten mit Degradierung durch bikubisches Downsampling umgehen können, wodurch sie sich untereinander bereits anhand der in den jeweiligen Publikationen angegebenen Metriken einfacher vergleichen lassen. Tabelle 3.5 zeigt hierzu die in den Arbeiten aufgeführten Testergebnisse auf den üblichen dafür verwendeten Testdatensätzen Set5, Set14, BSD100, Urban100 und Manga109 (vgl. Tab. 2.2). GPFGAN und CodeFormer sind wie beschrieben auf Face Restoration auslegt, weshalb ein Vergleich mit diesen Testsätzen nicht geeignet ist. Sie wurden in [170] mit *CelebA-Test* [96] getestet, einer Teilmenge mit 3000 Bildern des CelebA-Datensatzes [96], der insgesamt 202 599 Bilder von 10 177 unterschiedlichen prominenten Gesichtern beinhaltet. Die Ergebnisse dazu können in Tabelle 3.6 betrachtet werden. Real-ESRGAN lässt sich nur schwer mit den anderen Methoden vergleichen, da Wang *et al.* in [142] nur einen qualitativen Vergleich vornehmen und durch den Fokus der Arbeit auf alle möglichen unbekannten Degradierungen ein Vergleich mit klassischen Methoden anhand der üblichen Testdatensätze, die auf bikubisches Downsampling abzielen, ohnehin ungeeignet wäre.

(a) Original (b) Interpolation (c) Classic-SR (d) Real-World-SR

Abbildung 3.5: Beispiel zum Qualitätsunterschied zwischen Classic-SR und Real-World-SR anhand von SwinIR. Modell A1.3.12 (Classic) und A1.3.11 (Real) aus Tab. A1.3. (Bildquelle: AffectNet [108] 4063.jpg)

Tabelle 3.5: Testergebnisse der vorausgewählten, klassischen SR-Methoden auf dafür üblichen Testdatensätzen (vgl. Tab. 2.2) bei einem Vergrößerungsfaktor von ×4. Die besten Werte sind jeweils unterstrichen.
(Quelle: In Anlehnung an [89, S. 6, Tab. 2], [18, S. 7, Tab. 6], [20, S. 7, Tab. 2] und [171, S. 7, Tab. 4])

	Method	Training Dataset	Set5 [6]		Set14 [158]		BSD100 [102]		Urban100 [63]		Manga109 [46]	
			PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
SR ×4	**SwinIR** [89]	DIV2K+Flickr2K	32.92	0.9044	29.09	0.7950	27.92	0.7489	27.45	0.8254	32.03	0.9260
	HAT [18]	DIV2K+Flickr2K	33.04	0.9056	29.23	0.7973	28.00	0.7517	27.97	0.8368	32.48	0.9292
	DAT [20]	DIV2K+Flickr2K	33.08	0.9055	29.23	0.7973	28.00	0.7515	27.87	0.8343	32.51	0.9291
	SRFormer [171]	DIV2K+Flickr2K	32.93	0.9041	29.08	0.7953	27.94	0.7502	27.68	0.8311	32.21	0.9271

Tabelle 3.6: Testergebnisse GFPGAN & CodeFormer auf dem CelebA-Test-Datensatz. Die besten Werte sind jeweils unterstrichen.
(Quelle: In Anlehnung an [170, S. 7, Tab. 1])

Method	PSNR↑	SSIM↑	LPIPS↓
GFPGAN [141]	20.37	0.545	0.391
CodeFormer [170]	22.18	0.610	0.299

Um einen Eindruck davon zu gewinnen, wie gut die vorausgewählten SR-Methoden im Anwendungsfall dieser Arbeit funktionieren und ein geeignetes Netzwerk bzw. Modell zur Verwendung in den beschriebenen Versuche auszuwählen, werden die genannten Verfahren noch einmal spezifisch gegeneinander getestet. Die Grundlage dafür bietet der aus 3999 Bildern bestehende Validierungsdatensatz von AffectNet, der in dieser Arbeit als Testdatensatz eingesetzt wird und dessen Klassenverteilung ausbalanciert ist. Die Originaldateien (gemeint sind die unveränderten Bilder des Datensatzes mit jeweils 224 × 224 Pixeln) werden bikubisch zunächst auf 48 × 48 Pixel verkleinert und dann mit den SR-Methoden als auch mit bikubischer Interpolation wieder auf die ursprüngliche Größe gebracht. Die genaue Referenz zum Programmcode sowie den verwendeten Modellen kann im Anhang den Tabellen A1.2 und A1.3 entnommen werden. Abgesehen von CodeFormer verwenden alle SR-Methoden feste Skalierungsfaktoren. Das bedeutet in den meisten Fällen, dass die Netzwerke Upscaling mit Faktoren von ×2, ×3 und ×4 ermöglichen und entsprechende bereits trainierte Modelle zur Verfügung gestellt werden. Wie zuvor beschrieben, sind außerdem in einigen Fällen Modelle für Real-World-Super-Resolution verfügbar, die teilweise (z. B. HAT oder das für DAT gewählte Modell) nur für einen Vergrößerungsfaktor ×4 ausgelegt sind, weshalb dieser Faktor mit den zugehörigen Modellen für den Vergleich und die späteren Versuche verwendet wird. Da AffectNet aus In-the-Wild-Daten besteht, die unterschiedlichen Degradierungen unterliegen, werden diese Modelle bevorzugt. Abbildung 3.5 zeigt anhand von SwinIR den

(a) Originalbild (b) 2× SwinIR ×2 (c) 1× SwinIR ×4

(d) Originalbild (e) 2× SwinIR ×2 (f) 1× SwinIR ×4

Abbildung 3.6: Beispiel verstärkter Artefakte durch (mehrfach) angewandte Super-Resolution mit SwinIR-Modellen [89] ausgehend von 48 × 48 Pixeln des downsgesampelten Originalbilds.
Siehe Tab. A1.3 für verwendete Modelle: Modell A1.3.9 (×2), Modell A1.3.10 (×4)
(Bildquelle: AffectNet [108] 2839.jpg [a-c], 2840.jpg [d-f]

Qualitätsunterschied zwischen dem klassischen und dem Real-World-Modell. Zum Erreichen der Zielgröße von 224 × 224 werden die restlichen Pixel bikubisch interpoliert. Statt der Interpolation nach einer Vervierfachung der Pixel wäre auch denkbar, das Upsampeln durch Super-Resolution mit kleineren Faktoren in mehreren Stufen vorzunehmen (bspw. ×3, dann ×2, denn $48 \times 3 \times 2 = 288$ Pixel), um dann nur noch auf die entsprechende Zielauflösung downsampeln zu müssen. Allerdings durchläuft das Ausgangsbild dann mehrmals die SR-Algorithmen (und u. U. aufgrund von unterschiedlichen Faktoren auch mit unterschiedlich trainierten Modellen), was unerwünschte Artefakte verstärken, zusätzliche erzeugen oder für die spätere Klassifikation wichtige Merkmale verfälschen könnte. Ein Beispiel hierzu zeigt Abbildung 3.6. CodeFormer generiert Bilder mit mindestens 512 × 512 Pixeln, weshalb diese mit bikubischer Interpolation wieder verkleinert werden. Da außerdem wie zuvor beschrieben ein Koeffizient (ein Wert aus dem Intervall [0,1]) zum Steuern der Balance zwischen Qualität und Originaltreue angegeben werden muss, wird ein Wert von 0,5 gewählt, um ein ausgewogene Verhältnis zu erreichen.

Für alle vergrößerten Bilder werden die Metriken PSNR, SSIM und LPIPS mit dem Originalbild als Referenz berechnet[7]. Die Durchschnittswerte jeder Methode lassen sich Tabelle 3.7

[7]Die Metriken werden mit *TorchMetrics* [29] in Version 1.3.2 berechnet. Für LPIPS wird das bereitgestellte AlexNet [77] als Netzwerk verwendet, da es nach Zhang *et al.*, das beste und schnellste Netzwerk zur Verwendung von LPIPS als vorwärts gerichtete Metrik ist [162], [164].

entnehmen. In Tabelle 3.8 sind außerdem Beispiele für einen qualitativen Vergleich abgebildet.

Tabelle 3.7: Durchschnittliche Vergleichswerte der Metriken PSNR↑, SSIM↑ und LPIPS↓ zu den genannten SR-Methoden. Zur Berechnung der Werte wurden alle 3999 Bilder des AffectNet-Datensatzes (Validierungsmenge, die in dieser Arbeit als Testmenge verwendet wird; siehe Unterabschnitt 3.1.1) mit den acht Verfahren verarbeitet. Die farbig hinterlegten Zellen markieren jeweils Höchst- und Tiefstwerte:
bester Wert, zweitbester Wert, schlechtester Wert, zweitschlechtester Wert

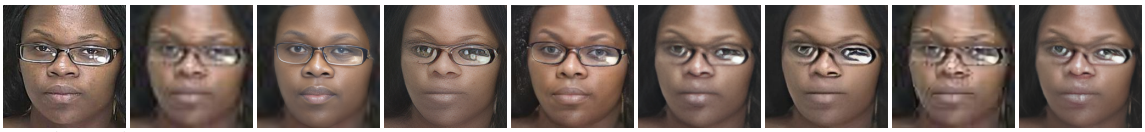
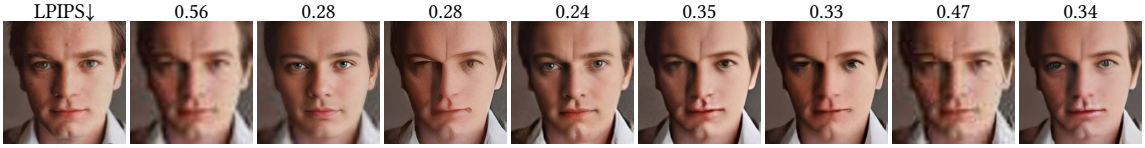
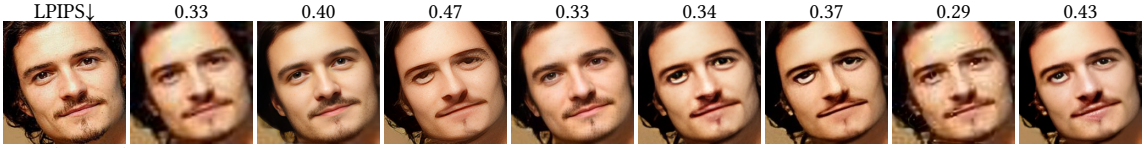
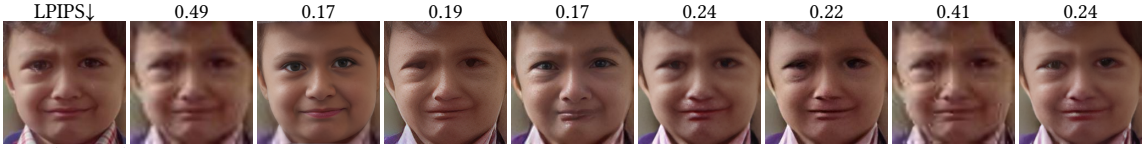
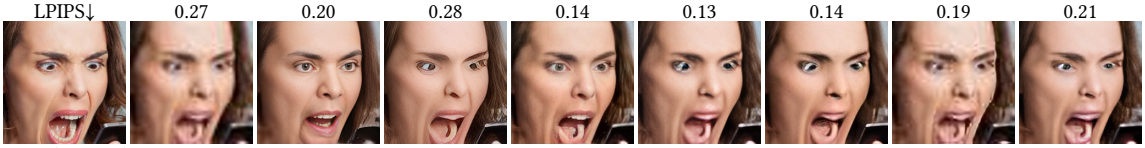
Methode	PSNR↑	SSIM↑	LPIPS↓
Bicubic	27.178	0.784	0.362
GFPGAN	25.665	0.758	0.164
DAT	25.126	0.704	0.245
CodeFormer	25.038	0.737	0.211
HAT	26.575	0.789	0.162
RealESRGAN	25.520	0.767	0.173
SRFormer	27.260	0.772	0.298
SwinIR	26.427	0.782	0.158

Tabelle 3.7 zeigt, dass HAT den Bestwert bei SSIM sowie den zweitbesten bei LPIPS aufweist. SRFormer hat den besten PSNR-Wert. Es fällt auf, dass die rein bikubische Interpolation die zweitbesten Werte bei PSNR und SSIM erreicht. Das mag einerseits daran liegen, dass klassische SR-Methoden mit deutlich höheren Auflösungen trainiert werden (vgl. DIV2K & Flickr2K in Tab. 2.2) und ihre Stärke deshalb nicht besser ausspielen können. Andererseits basieren die beiden Metriken auf pixelweisen vergleichen und da die Interpolation eine reine Berechnung aus den nächstgelegenen Pixeln des verkleinerten Originals ist, muss die grundsätzliche Ähnlichkeit, die diese Werte angeben, relativ hoch sein. Für die Methoden, die GANs oder Codebücher verwenden, sind diese außerdem nicht unbedingt aussagekräftig, da sie neue Daten synthetisieren bzw. aus einem bestehenden Codebuch generieren und damit keine pixelgenaue Ähnlichkeit mehr gewährleisten können. Interessant ist also vor allem der LPIPS-Wert, da er eine Aussage über die wahrgenommene Ähnlichkeit zwischen Originalbild und dem SR-vergrößerten trifft. Hier hat SwinIR den Bestwert und auch HAT, dicht gefolgt von GFPGAN, sind davon nicht weit entfernt. Die bikubische Interpolation zeigt hierbei den schlechtesten Wert. Da für den Versuch nicht die pixelgenaue Ähnlichkeit der Bilder eine Rolle spielt, sondern vor allem eine hohe Bildqualität durch viele Details bei Erhalt einer hohen Emotionstreue, bietet der LPIPS-Wert zwar schon einen recht guten Anhaltspunkt, gleichzeitig müssen aber auch konkrete Beispiele betrachtet werden.
Wie sich Tabelle 3.8 entnehmen lässt, bieten aus rein subjektiver Sicht CodeFormer und GFPGAN die höchste Bildqualität, was den wahrgenommenen Detailgrad der Gesichtsmerkmale betrifft. SRFormer hat eine starke Ähnlichkeit zur bikubischen Interpolation, was sich

auch in den Durchschnittswerten in Tabelle 3.7 widerspiegelt. Dabei muss erwähnt werden, dass die anderen Methoden mittlerweile für Real-World-Super-Resolution vortrainierte Modelle mitliefern, also nicht nur auf die klassische, bikubische Degradierung optimiert sind. Auch für SRFormer gibt es theoretisch ein solches, das in der aktuellen Version (vgl. A1.2.6) aber nicht ohne weiteres mit dem zur Verfügung gestellten Netzwerk geladen werden kann. DAT und RealESRGAN sowie HAT und SwinIR sind sich jeweils ebenfalls recht ähnlich in der subjektiven Wahrnehmung. DAT scheint Details etwas mehr zu verwaschen und zu glätten, während RealESRGAN Konturen stärker betont und dunkle Bereiche verstärkt. Sie produzieren beide außerdem teilweise leichte Artefakte. Die Ergebnisse von HAT und SwinIR sind bei Betrachtung auch nicht weit entfernt von denen der beiden gerade genannten, scheinen aber insgesamt etwas bessere Bilder zu produzieren und Details besser zu erhalten bzw. zu rekonstruieren, wobei die Bildqualität der Beispiele von SwinIR etwas höher wirkt, was sich auch mit dem leicht besseren LPIPS-Wert deckt.
Da für diese Arbeit eine hohe Bildqualität mit einem hohen Detailgrad gewünscht ist, kommen zunächst CodeFormer und GFPGAN in Betracht. Werden die produzierten Bilder dieser Methoden mit dem Originalbild verglichen, fällt zwar auf, dass CodeFormer tatsächlich eine etwas höhere Qualität leistet, die Identitäten der Personen aber stärker verändert[8] werden. Dies ist im Kontext von FER durchaus vernachlässigbar, die Beispiele zu den ausdrucksstärkeren Emotionen wie *Sadness*, *Anger*, *Fear*, *Surprise* und *Disgust* zeigen aber gleichzeitig, dass wichtige Merkmale teilweise komplett verändert bzw. „neutralisiert“ werden. So wirken sämtliche Emotionen der von CodeFormer produzierten Ergebnisse deutlich positiver und wären damit subjektiv auch eher den Klassen *Neutral* oder *Happiness* zuzuordnen. Aus diesem Grund wird für die geplanten Experimente GFPGAN verwendet, dessen bessere Ähnlichkeit zum Originalbild sich auch in den besseren LPIPS-Werten widerspiegelt. GFPGAN lässt außerdem einen beliebigen ganzzahligen Skalierungsfaktor zur, weshalb die auf 48×48 Pixel verkleinerten Bilder um das zehnfache auf 480×480 vergrößert werden, um davon mittels bikubischer Interpolation die benötigten kleineren Formate zu generieren und die Möglichkeit zu haben, auch eine größere Auflösung als die mit AffectNet in 224×224 Pixeln vorliegende zu testen. Außerdem wird ausgehend von 224×224 auch die zweifache Auflösung von 448×448 generiert. Das verwendete Modell ist das *GFPGANv1.3* (siehe A1.3.4), das natürlichere und bessere Ergebnisse für LQ-Eingaben erzielt [145].
Um auch ein konventionelles SR-Verfahren für Real-World-Super-Resolution ohne Face Restoration vergleichen zu können, das u. U. weniger Gefahr birgt, Identitäten und Emotionen zu verfälschen, wird außerdem HAT eingesetzt, das nicht nur den Bestwert bei SSIM, sondern auch einen sehr guten Wert bei LPIPS liefert. Hierfür wird das *Real_HAT_GAN_sharper*-Modell (siehe A1.3.6) eingesetzt, das derzeit aber nur eine Vergrößerung mit dem Faktor ×4

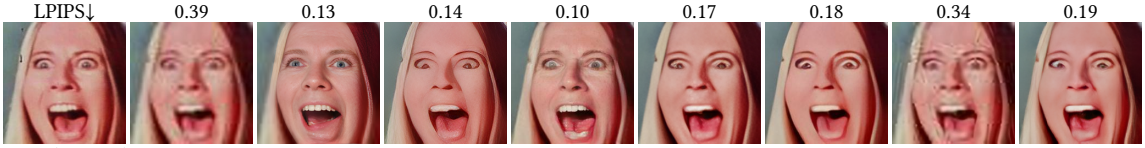
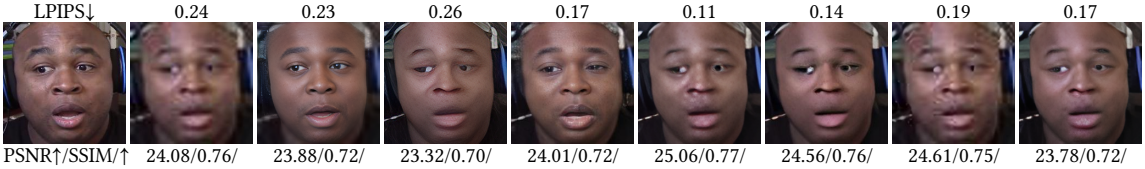
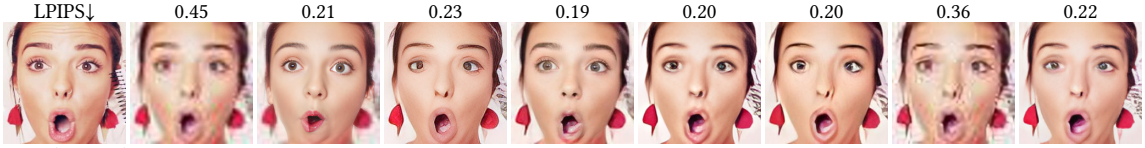
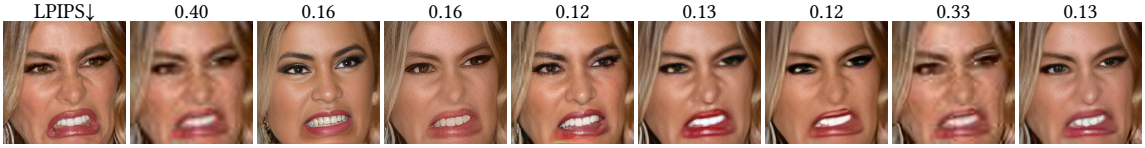
[8] Der gewählte Koeffizient von 0,5 spielt hier keine Rolle, da auch der höchste Wert 1 für eine hohe Originaltreue in Tests keine sichtbaren Unterschiede erkennen lässt.

Tabelle 3.8: Beispiele von mit den vorausgewählten SR-Methoden vergrößerten AffectNet-Bildern (je zwei Bilder der acht Klassen) zum qualitativen sowie quantitativen Vergleich unter Angabe der Metriken PSNR↑, SSIM↑ und LPIPS↓.
(Bildquellen: AffectNet (Trainingsdatensatz) [108])

	Original	Bicubic	CodeFormer [170]	DAT [20]	GFPGAN [141]	HAT [18]	RealESRGAN [142]	SRFormer [171]	SwinIR [89]
Neutral 1930.jpg									
PSNR↑/SSIM↑ LPIPS↓		23.32/0.63/ 0.56	21.97/0.60/ 0.28	22.12/0.58/ 0.28	22.34/0.60/ 0.24	23.34/0.64/ 0.35	22.36/0.63/ 0.33	23.46/0.63/ 0.47	23.01/0.63/ 0.34
Neutral 1932.jpg									
PSNR↑/SSIM↑ LPIPS↓		28.78/0.84/ 0.34	26.87/0.82/ 0.13	27.07/0.78/ 0.20	27.14/0.82/ 0.11	27.52/0.85/ 0.12	27.27/0.84/ 0.12	28.89/0.83/ 0.29	27.72/0.84/ 0.13
Happiness 2710.jpg									
PSNR↑/SSIM↑ LPIPS↓		27.42/0.78/ 0.33	25.03/0.73/ 0.40	24.63/0.67/ 0.47	25.19/0.74/ 0.33	25.07/0.75/ 0.34	23.82/0.73/ 0.37	27.67/0.78/ 0.29	23.51/0.71/ 0.43
Happiness 10826.jpg									
PSNR↑/SSIM↑ LPIPS↓		24.39/0.67/ 0.49	24.01/0.67/ 0.17	23.49/0.63/ 0.19	23.90/0.68/ 0.17	24.08/0.69/ 0.24	23.28/0.66/ 0.22	24.31/0.66/ 0.41	23.25/0.64/ 0.24
Sadness 1052.jpg									
PSNR↑/SSIM↑ LPIPS↓		31.42/0.89/ 0.22	27.90/0.85/ 0.16	27.50/0.80/ 0.28	27.28/0.84/ 0.16	29.25/0.89/ 0.10	28.10/0.86/ 0.13	31.11/0.88/ 0.17	28.53/0.87/ 0.14
Sadness 14923.jpg									
PSNR↑/SSIM↑ LPIPS↓		26.41/0.67/ 0.46	24.26/0.58/ 0.24	24.40/0.55/ 0.27	24.74/0.61/ 0.20	25.71/0.66/ 0.25	24.19/0.63/ 0.27	26.20/0.66/ 0.39	24.79/0.62/ 0.24
Anger 2288.jpg									
PSNR↑/SSIM↑ LPIPS↓		27.30/0.83/ 0.27	25.28/0.78/ 0.20	25.18/0.75/ 0.28	26.10/0.80/ 0.14	26.27/0.83/ 0.13	25.17/0.81/ 0.14	27.19/0.81/ 0.19	24.71/0.78/ 0.21
Anger 15165.jpg									
PSNR↑/SSIM↑ LPIPS↓		25.37/0.76/ 0.42	22.25/0.69/ 0.18	23.67/0.71/ 0.18	23.92/0.73/ 0.15	24.99/0.77/ 0.17	24.28/0.76/ 0.17	25.54/0.75/ 0.34	24.08/0.75/ 0.20

Fortsetzung auf nächster Seite

Tabelle 3.8 – Fortsetzung von vorheriger Seite

	Original	Bicubic	CodeFormer [170]	DAT [20]	GFPGAN [141]	HAT [18]	RealESRGAN [142]	SRFormer [171]	SwinIR [89]
Fear 1105.jpg									
PSNR↑/SSIM/↑ LPIPS↓		27.28/0.75/ 0.39	25.68/0.73/ 0.13	25.53/0.70/ 0.14	26.47/0.72/ 0.10	26.95/0.76/ 0.17	25.30/0.74/ 0.18	27.31/0.73/ 0.34	25.44/0.73/ 0.19
Fear 45009.jpg									
PSNR↑/SSIM/↑ LPIPS↓		28.57/0.83/ 0.24	23.97/0.75/ 0.23	25.99/0.78/ 0.26	26.55/0.80/ 0.17	27.43/0.85/ 0.11	27.00/0.83/ 0.14	28.78/0.81/ 0.19	25.29/0.82/ 0.17
Surprise 2268.jpg									
PSNR↑/SSIM/↑ LPIPS↓		24.08/0.76/ 0.45	23.88/0.72/ 0.21	23.32/0.70/ 0.23	24.01/0.72/ 0.19	25.06/0.77/ 0.20	24.56/0.76/ 0.20	24.61/0.75/ 0.36	23.78/0.72/ 0.22
Surprise 18448.jpg									
PSNR↑/SSIM/↑ LPIPS↓		23.02/0.71/ 0.51	22.06/0.72/ 0.22	22.21/0.70/ 0.20	22.77/0.75/ 0.19	21.60/0.75/ 0.18	22.08/0.75/ 0.19	23.22/0.71/ 0.40	22.19/0.75/ 0.19
Contempt 31772.jpg									
PSNR↑/SSIM/↑ LPIPS↓		28.16/0.83/ 0.34	25.50/0.76/ 0.19	26.65/0.76/ 0.21	26.98/0.81/ 0.13	27.77/0.83/ 0.12	26.38/0.81/ 0.13	28.48/0.82/ 0.26	27.55/0.79/ 0.16
Contempt 34431.jpg									
PSNR↑/SSIM/↑ LPIPS↓		26.85/0.76/ 0.43	25.24/0.69/ 0.16	25.65/0.68/ 0.23	26.56/0.76/ 0.12	27.28/0.78/ 0.16	26.36/0.77/ 0.15	26.54/0.75/ 0.36	26.10/0.73/ 0.16
Disgust 37780.jpg									
PSNR↑/SSIM/↑ LPIPS↓		26.45/0.76/ 0.40	23.72/0.72/ 0.16	25.42/0.69/ 0.16	26.31/0.77/ 0.12	26.40/0.78/ 0.13	25.70/0.76/ 0.12	26.49/0.74/ 0.33	25.08/0.71/ 0.13
Disgust 969.jpg									
PSNR↑/SSIM/↑ LPIPS↓		27.06/0.76/ 0.33	22.77/0.61/ 0.24	24.71/0.66/ 0.21	24.92/0.70/ 0.16	25.18/0.74/ 0.15	24.99/0.73/ 0.17	26.94/0.75/ 0.26	23.70/0.65/ 0.23

(In der digitalen Version vergrößern für mehr Details)

zulässt. Ein Upsampling von 48×48 auf 448×448 Pixel ist demnach nicht möglich[9]. Zusätzlich zu den Bildern mit 48 × 48 Pixeln, die mit HAT auf 112 × 112 sowie 224 × 224 Pixel upgesampelt werden, wird auch eine Vergrößerung der AffectNet-Originalbilder von 224 × 224 vorgenommen, wonach aufgrund der einzig verfügbaren Option der Vervierfachung einer

Auflösung bikubisch auf 448 × 448 downgesampelt wird.

3.3 Versuchsübersicht

Tabelle 3.9 gibt einen Überblick über die geplanten Versuche anhand der nach Verfahren aufgelisteten Auflösungen, mit denen je ein Modell trainiert und mit dem zugehörigen Testdatensatz getestet wird.
Neben den ausgewählten Super-Resolution-Methoden wird auch die bikubische Interpolation zum Upsampeln getestet und verglichen. Außerdem ist in Tabelle 3.9 je Datensatz (AffectNet und RaFD) ein zusätzlicher Versuch bzgl. dem s. g. *Zero-Padding* (Erweitern der Zeilen und Spalten durch Nullwerte [117]; siehe Abb. 3.8 für ein Beispiel) aufgeführt, der zwar nicht in direkter Verbindung zu den Kernfragen dieser Arbeit steht, aber dennoch aufschlussreiche Ergebnisse hinsichtlich der Bildverarbeitung durch CNNs geben kann. Damit soll festgestellt werden können, wie sich das Zero-Padding eines Bildes (also eine andere Art der Dimensionsskalierung), das eine geringere Auflösung als die von einem Netzwerk erwartete Eingabegröße, auf die Erkennungsraten im Vergleich zur Interpolation und dem Verwenden der nativen Auflösung auswirkt. Der Hintergrund wird im nachfolgenden Abschnitt 3.4 erläutert, der die Netzwerk-Architektur sowie die Eingabe der Bilder ins Netzwerk beschreibt.

Tabelle 3.9: Übersicht der zu trainierenden Auflösungen nach Verfahren und Datensatz.

AffectNet

Verfahren	Auflösung[1]	Ursprung[2]
Original/ Downsample Bicubic	48 × 48	224 × 224
	112 × 112	224 × 224
	224 × 224	224 × 224
Zero Padding	224 × 224	48 × 48
Upsample Bicubic	112 × 112	48 × 48
	224 × 224	48 × 48
	448 × 448	48 × 48
	448 × 448	224 × 224
SR GFPGAN	112 × 112	48 × 48
	224 × 224	48 × 48
	448 × 448	48 × 48
	448 × 448	224 × 224
SR HAT	112 × 112	48 × 48
	224 × 224	48 × 48
	448 × 448	224 × 224

RaFD

Verfahren	Auflösung[1]	Ursprung[2]
Original/ Downsample Bicubic	48 × 48	310 × 310*
	112 × 112	310 × 310*
	224 × 224	310 × 310*
	280 × 280	310 × 310*
Zero Padding	224 × 224	48 × 48

[1] Trainings- und Testauflösung
[2] Ausgangsauflösung, wovon Trainings- und Testauflösung erzeugt wurden
* Durchschnittliche Auflösung der Ausgangsdaten, siehe Unterabschnitt 3.2.2

[9] Das mehrmalige Durchlaufen eines Netzwerks wird aufgrund der in Unterabschnitt 3.2.4 (siehe auch Abb. 3.6) erläuterten Gründe vermieden.

3.4 Netzwerk-Architektur

Für das zu trainierende Deep Neural Network wird eine *EfficientNet*-Architektur [129], bzw. konkret ein EfficientNet-B0 verwendet, das auch in [122] sowie [152] für FER zum Einsatz kommt und mit 57,55 % [122] bzw. 59,0 % Accuracy [152] (jeweils Klassifizierung von acht Klassen) gute Ergebnisse auf dem AffectNet-Datensatz zeigt, die nicht weit entfernt von den Leistungen des Stands der Technik (engl. *State of the Art* [SOTA]) liegen. SOTA-Modelle erreichen bei diesem Datensatz eine Accuracy von bis zu 64,25 % bei acht Klassen [112], [165].
EfficientNet ist eine Netzwerkfamilie, die 2019 von Tan und Le [129] entwickelt wurde, ausgehend von zwei Beobachtungen in CNNs: Das Hochskalieren einer der Netzwerkdimensionen Breite, Tiefe oder Auflösung verbessert die Genauigkeit, wobei der Zugewinn bei größeren Netzwerken stetig abnimmt. Außerdem ist es für eine bessere Genauigkeit und Effizienz von hoher Bedeutung, alle Dimensionen balanciert zu skalieren, wohingegen es sonst eher üblich ist, nur eine Dimension wie bspw. die Tiefe eines Netzwerks zu verändern (mehr Schichten hinzuzufügen). Ändert sich also eine Dimension, sollten im besten Fall die anderen Dimensionen ebenfalls skaliert werden. Die Autoren lösen dies mit einer Menge an festen Koeffizienten zur Skalierung und bezeichnen dies als *Compund Scaling Method*. Sie zeigen anhand bestehender Standard-Netzwerkarchitekturen, dass ihr Ansatz funktioniert. Da die Skalierung der Dimensionen allein allerdings keine Veränderung an den Operationen einer Verarbeitungsschicht selbst vornimmt, ist ein gut strukturiertes Netzwerks als Grundlage ebenso entscheidend für die Effektivität der Skalierung. Tan und Le entwickeln daher die EfficientNet-Familie, die ähnlich wie *Mnas-Net* [128] aufgebaut ist. EfficientNet-B0 ist das Basisnetzwerk dieser Familie, EfficientNet-B1 bis B7 weitere Varianten mit anderen Koeffizienten zur höheren Skalierung. Die Arbeit zeigt, dass mit der neuen Methode und Netzwerk-Familie SOTA-Leistungen nicht nur auf dem ImageNet-Datensatz [28] zur Objekterkennung übertroffen, sondern auch SOTA-Leistungen bei Transfer-Aufgaben erreicht werden bei einer deutlich geringeren Parameterzahl und damit weniger benötigten Rechenleistung [129]. Die hohe Leistung bei verhältnismäßig geringem Rechenaufwand ist ein entscheidender Vorteil ggü. anderen Netzwerken, weshalb diese Arbeit, ebenso wie Savchenko *et al.* [122] und Yen und Li [152], auf diese Architektur setzt.
Die EfficientNet-Netzwerke mit höherer Versionsnummer sind durch die höhere Skalierung bzw. höhere Anzahl an Parametern i. d. R. auch leistungsstärker [122], [129]. Da die vorliegende Arbeit aber nicht versucht die höchstmögliche Leistung zu erzielen oder sogar SOTA-Modelle zu übertreffen, wird das kleinste Modell B0 verwendet.

Es wird die EffiecientNet-B0-Implementierung aus Keras verwendet, wobei die letzten drei Schichten (Average-Pooling-, Dropout- und Prediction-Layer) verworfen werden, sodass die letzte Schicht des aus Keras geladenen Netzwerks der Activation-Layer des siebten und letzten Blocks ist. Daran wird mit einem Global-Average-Pooling-, einem Batch-Normalization- sowie einem Dropout-Layer (mit einer Dropout-Rate von 20 %) angeschlossen, woraufhin der

Tabelle 3.11: EfficientNet-B0 Basisarchitektur. Hauptbaustein des Netzwerks ist der *Mobile Inverted Bottleneck MBConv* [121], [128] mit zusätzlicher *Squeeze-and-Excitation-Optimierung* [61].
(Quelle: [129, S. 5, Tab. 1])

Stage i	**Operator** $\hat{F}_i$	**Resolution** $\hat{H}_i \times \hat{W}_i$	**#Channels** $\hat{C}_i$	**#Layers** $\hat{L}_i$
1	Conv3x3	224 x 224	32	1
2	MBConv1, k3x3	112 x 112	16	1
3	MBConv6, k3x3	112 x 112	24	2
4	MBConv6, k5x5	56 x 56	40	2
5	MBConv6, k3x3	28 x 28	80	3
6	MBConv6, k5x5	14 x 14	112	3
7	MBConv6, k5x5	14 x 14	192	4
8	MBConv6, k3x3	7 x 7	320	1
9	Conv1x1 & Pooling & FC	7 x 7	1280	1

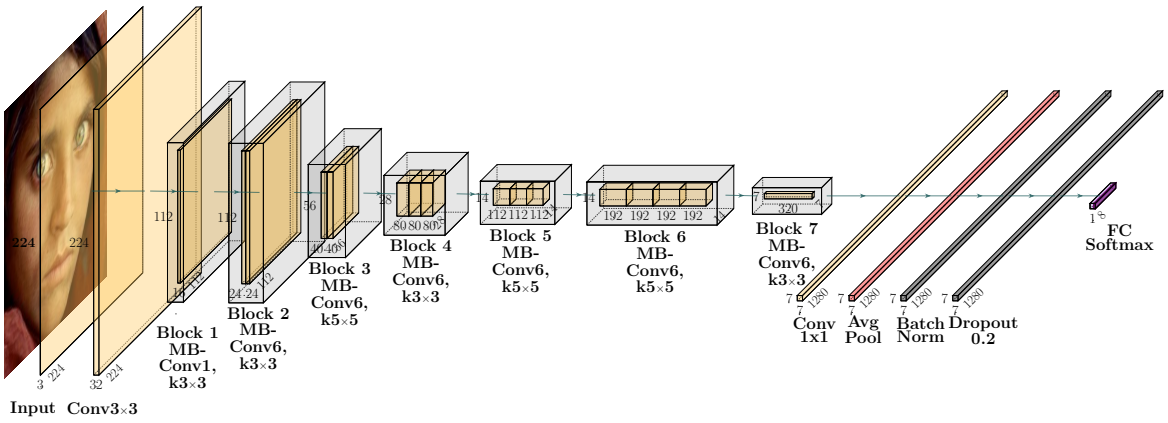


Abbildung 3.7: Schematische Darstellung des angepassten EfficientNet-B0
(Bildquelle: AffectNet [108] 5366.jpg; Plot erstellt mit *PlotNeuralNet* [64])

finale Dense-Layer mit acht Neuronen und Softmax-Aktivierung[10] zur Vorhersage der Wahrscheinlichkeiten der Klassenzugehörigkeit folgt. Der Batch-Normalization-Layer sorgt für ein schnelleres und stabileres Training, indem die Eingaben normalisiert und Veränderungen in ihrer Verteilung während des Trainings vermieden werden [117]. Der Dropout-Layer hilft bei der Generalisierung, indem der definierte Anteil der Ausgabewerte auf 0 gesetzt und damit „fallengelassen“ wird, was Overfitting entgegenwirkt [24]. Das Netzwerk ist schematisch in

[10]Die Softmax-Aktivierung sorgt für die Ausgabe einer Wahrscheinlichkeitsverteilung der acht Klassen, die zusammen 1 bzw. 100 % ergeben [24].

Abbildung 3.7 dargestellt.

Als Eingabegröße sollte eine Auflösung gewählt werden, die teilbar durch acht ist, da diese sonst in manchen Schichten dafür sorgt, dass Zero-Padding an den Rändern eingesetzt werden muss, was Rechenressourcen verschwendet [45]. Die unter Unterabschnitt 3.2.1 definierten Auflösungen stimmen mit dieser Vorgabe überein. Eine zentrale Frage mit unterschiedlichen Herangehensweisen ist, ob die Bilder in den Dimensionen eingegeben und verarbeitet werden, die von den Auflösungen vorgegeben sind, oder die Bilddaten vor bzw. nach Eingabe (bspw. durch eine zusätzlich implementierte Schicht) in ihren Dimensionen verändert werden.
Die eine Möglichkeit ist, ein Farbbild mit 128×128 Pixeln auch als Tensor mit den Dimensionen $(128, 128, 3)$ in das Netzwerk einzugeben und keine weitere Skalierung vorzunehmen. Wie in Unterabschnitt 2.1.5 erklärt, besteht ein zentraler Teil in CNNs aus dem Zusammenfassen von lokalen Merkmalen, bspw. durch Pooling-Operationen oder der Verwendung von größeren Strides als 1 in den Faltungen, um durch aufeinanderfolgende Faltungsschichten Filterhierarchien zu schaffen. Bei niedrigen Auflösungen, die tiefe Netzwerke mit vielen Reduzierungsoperationen durchlaufen, kann es passieren, dass die Output Feature Maps so klein werden, dass diese entweder gar nicht mehr verarbeitet werden können oder durch Zero-Padding an den Rändern aufgefüllt werden müssen, wie es in EfficientNet-B0 umgesetzt wird. Denn ein 3×3-Convolution-Fenster kann nicht mehr über eine 1×1-Feature-Map wandern. Ein tieferes Netzwerk bringt dann also nicht unbedingt mehr Vorteile, da keine neuen Filterhierarchien mehr geschaffen werden, weil immer derselbe Bereich betrachtet wird. Bei einer zu starken Reduktion der Feature-Map-Größe ist also keine effektive Extraktion von höherstufigen Merkmalen mehr möglich.
Eine andere Möglichkeit ist, die Dimensionen eines Bildes (vorab) so zu erweitern, bspw. durch (bikubische) Interpolation, dass das Bild beim Durchlaufen der Schichten nicht währenddessen gepaddet werden muss. Ein Nachteil davon ist, dass das Bild durch die Interpolation verändert wird und somit eine leichte Verzerrung der vorhandenen Bildinformationen stattfindet, sofern ein Kriterium ist, die Daten möglichst unberührt zu belassen. Alternativ lässt sich Zero-Padding bereits zu Anfang anwenden, sodass nicht nur die nötigsten Bereiche gepadded werden, wenn einem Convolution-Fenster Pixel zur Verarbeitung fehlen, sondern vor der eigentlichen Netzwerk-Verarbeitung das gesamte Bild mit ausreichend Nullwerten (schwarzem Rand) erweitert wird (siehe Beispiel in Abb. 3.8). Hierdurch bleiben die ursprünglichen Bildinformationen unberührt. Hashemi beschreibt in [57], dass trotz Zero-Padding die Möglichkeit der Feature-Extraktion am Rand eines Bildes gleich ist zu den zentrierten Teilen, das Padding durch Nullwerte also keine negativen Auswirkung darauf hat. Das liegt daran, dass die Gewichte aller Faltungsfenster einer Faltung gleich bleiben. Weiter erklärt Hashemi, dass die Anpassung der Gewichte durch Zero-Padding nicht gestört wird, weil Nullwerte nicht zu einer Aktivierung führen, die berücksichtigt werden muss, und liefert für seine Aussagen einen theoretischen Beweis. Gleichzeitig würde sich dies ggü. Interpolation aber positiv auf die Verarbeitungszeit bei gleichbleibender Accuracy auswirken [57].

Da die bikubische Interpolation als Upsampling-Verfahren bereits Bestandteil der Versuche ist und die direkte Auswirkung der Auflösung auf DNNs (ohne nachträgliche Veränderung der Dimensionen) bei FER-Tasks untersucht werden soll, wird für jeden der Versuche die zu testende Auflösung des jeweiligen Verfahrens auch als Eingabegröße definiert. Eine Ausnahme bilden die in Tab. 3.9 zusätzlich aufgeführten Tests der Bilder, die mithilfe von Zero-Padding von 48×48 auf 224×224 Pixel gebracht werden, um den Einfluss dieses Verfahrens anhand eines Beispiels zu untersuchen. Hier sorgt ein zusätzlicher Zero-Padding-2D-Layer nach der Eingabeschicht für das Padding während der Verarbeitung durch das Netzwerk.

Abbildung 3.8: Beispiel eines Bildes mit Zero-Padding von 48×48 auf 224×224 Pixel (Bildquelle: AffectNet [108] 452.jpg)

3.5 Training & Vorgehen

Sofern nicht explizit erwähnt, findet das Training aller Modelle unter denselben Bedingungen (Trainingsparametern) statt, die nachfolgend erläutert werden.

3.5.1 Metriken & Evaluierung

Zur Evaluierung der trainierten Modelle wird die Accuracy (siehe 2.1.3) mit den Testdatensätzen gemessen, also wie gut die vorhandenen Klassen im Durchschnitt zugeordnet werden. Das Training wird entsprechend auf Basis der Accuracy, die auf den Validierungsdatensätzen erreicht wird, gesteuert und ist maßgeblich für die Trainingsdauer bzw. Anzahl der trainierten Epochen. Das Modell der Epoche, die die höchste Validierungs-Accuracy erreicht, wird dann als endgültiges Modell zum Testen des Testdatensatzes verwendet, womit die finale Netzwerkleistung angegeben wird. Es wird mit den jeweiligen Testdatensätzen getestet, die unter denselben Bedingungen (Verfahren und Ausgangsauflösung) erstellt wurden, wie die für das Training verwendeten Daten.
Die Accuracy ist die am weitesten verbreitete Metrik für kategorische Facial Expression Recognition [84], [122], [137], [152] und eignet sich daher auch am besten für einen allgemeinen Vergleich mit vergangenen sowie zukünftigen Arbeiten, obwohl in Unterabschnitt

2.1.3 erläutert wird, dass sie weniger Aussagekraft bei unbalancierter Klassenverteilung hat. Da die ungleiche Verteilung der Klassen der AffectNet-Daten aber im Training durch eine Gewichtung ausgeglichen wird (siehe nachfolgender Unterabschnitt) und der Testdatensatz dieselbe Anzahl Bilder je Klasse enthält, kann die Accuracy dennoch als aussagekräftige Metrik verwendet werden, um die Leistung des Modells zu bewerten.

3.5.2 Trainingsparameter & Besonderheiten

Die Trainings- bzw. *Hyperparameter*[11] wurden durch mehrere Versuche festgelegt. Da diese Arbeit nicht versucht, SOTA-Leistung zu erreichen oder sogar zu übertreffen, wurden diese zwar manuell so optimiert, dass eine ähnliche Leistung zu den genannten Arbeiten [122], [129] erreicht wird, aber kein spezielles automatisiertes Verfahren zum Hyperparameter-Tuning angewendet.

Das EfficientNet-B0 wird mit Gewichten aus einem Vortraining mit ImageNet initialisiert, die ebenfalls über Keras geladen werden. Es zeigt sich in ersten Vorversuchen, dass die reine Initialisierung mit diesen Gewichten und dem *Unfreezen*[12] aller Schichten (mit Ausnahme der Batch-Normalization-Layer, die nicht unfreezed werden, um nicht das gesamte Vortraining zu zerstören [23]) ab der ersten Epoche die besten Ergebnisse im Hinblick auf die Accuracy bringt. Das partielle Einfrieren oder Freigeben von Layern führt in diesem Fall zu einer schlechteren Leistung.
Die Eingabegröße des Netzwerks wird wie in Abschnitt 3.4 beschrieben jeweils entsprechend der zu trainierenden Auflösung gewählt, d. h. eine nachträgliche Größenveränderung findet nicht statt[13].
Als Fehlerfunktion wird die *Categorical Crossentropy* verwendet, welche die Distanz zwischen der Wahrscheinlichkeit der vorhergesagten Klasse zur tatsächlichen Klasse misst [24].
Für den Optimizer wird *Adam* [74] eingesetzt, ein Verfahren zur Optimierung des stochastischen Gradientenabstiegs, das nur wenig Speicheranforderungen sowie eine hohe Recheneffizienz hat und sich bei der Verarbeitung von großen Datenmengen und/oder Modell-Parametern eignet [74]. Auch die Arbeiten von Savchenko *et al.* [122] sowie Yen und Li [152] nutzen Adam als Grundlage.
Eine Lernrate von 1e−4 wird für die ersten drei Epochen gewählt. Anschließend wird mit einer Lernrate von 1e−5 für maximal zehn Epochen weiter trainiert. Hierfür wird ein *Early-Stopping-Mechanismus* mit einer *Patience* von fünf Epochen verwendet, womit der Trainingsprozess nach fünf Epochen angehalten wird, sofern sich die Validierungs-Accuracy über diese Zeit nicht verbessert.

[11] Damit werden die Parameter auf Architektur-Ebene in Abgrenzung zu den trainierten Modell-Parametern bezeichnet [24].
[12] Die bestehenden Gewichte werden also nicht eingefroren und sind somit durch weiteres Training veränderbar.
[13] Hiervon ausgenommen ist der Versuch mit Zero-Padding, siehe auch Abschnitte 3.3 und 3.4

Die Batch-Size, also die Anzahl der Bilder in einem Verarbeitungsstapel nach dem eine Anpassung der Gewichte stattfindet, wird auf 32 gesetzt, wobei diese in Versuchen mit einer Auflösung von 448×448 Pixeln aufgrund von Hardware-Limitierungen auf 16 heruntergesetzt werden muss.

Da der AffectNet-Datensatz bzgl. der Klassenverteilung sehr unbalanciert ist, werden fürs Training inverse normalisierte Gewichte zu jeder der Klassen eingeführt, sodass solche mit weniger Bildern stärker gewichtet und die mit mehr Bildern weniger stark gewichtet werden. Dafür wird zunächst die Klassenverteilung V_i für Klasse i mit

$$V_i = \frac{n_i}{N}$$

berechnet, wobei n_i die Anzahl der Bilder einer Klasse und N die Gesamtzahl der Bilder des Datensatzes darstellt. Anschließend werden die inversen Gewichte w_i mit

$$w_i = \frac{1}{V_i} = \frac{N}{n_i}$$

berechnet und schließlich durch

$$w_{i,\text{norm}} = \frac{w_i}{\sum_{j=1}^{K} w_j}$$

so normalisiert, dass die Werte im Bereich $[0, 1]$ liegen. K steht dabei für die Anzahl der Klassen des Datensatzes.

Data Augmentation, was als Begriff verschiedene Techniken beschreibt, um mit einer limitierten Datenmenge umzugehen und (im Falle der Bildverarbeitung) bspw. durch Veränderung von Kontrast, Orientierung, Helligkeit, Skalierung oder Sättigung „neue“ Daten synthetisiert, womit oftmals Overfitting entgegengewirkt werden kann [117], führt in Vorversuchen mit AffectNet zu keinem Leistungszuwachs, sondern zu einer Verschlechterung, weshalb davon abgesehen wird. Auch der RaFD-Datensatz profitiert gemessen an der Modellleistung nicht von Data Augmentation, weshalb diese in keinem der Versuche angewendet wird.

4 Ergebnisse

Nach der Durchführung der unter 3.3 zusammengefassten Versuche lassen sich die Ergebnisse wie folgt zusammenfassen.

4.1 Vergleich Bildauflösung

Zur Klärung der Fragen i) und ii) bzgl. eines positiven Effekts einer höheren Bildauflösung auf die Erkennungsraten und eines optimalen Auflösungsbereichs für Facial Expression Recognition geben die Ergebnisse aus Tabelle 4.1 Aufschluss.

Tabelle 4.1: Ergebnisse (Accuracy) des Auflösungsvergleichs ohne Super-Resolution-/Upsampling-Verfahren durch Training von EfficientNet-B0 anhand der (bikubisch downgesampelten) Originalbilder von AffectNet und RaFD. Trainings- und Testdaten wurden jeweils unter denselben Bedingungen (Ausgangsauflösung & Verfahren) erzeugt.

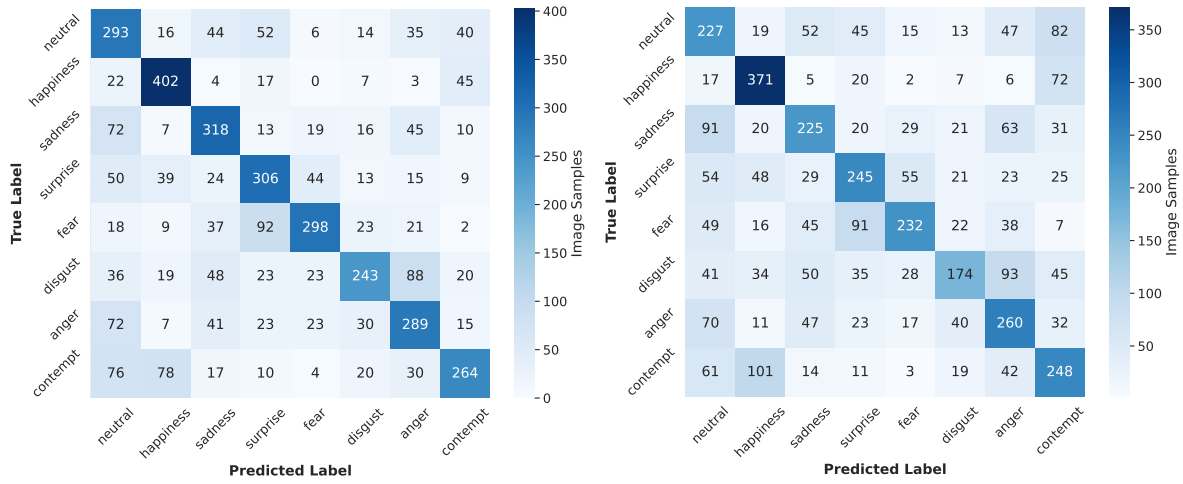
	Auflösung[1]	**Ursprung**[2]	**Verfahren**	**Accuracy, %**	**Veränderung, %**
AffectNet	224 × 224	224 × 224	Original	60,34	Basis
	112 × 112	224 × 224	Downsample Bicubic	57,31	-5,02
	48 × 48	224 × 224	Downsample Bicubic	49,56	-17,87
RaFD	224 × 224	310 × 310*	Downsample Bicubic	93,40	Basis
	280 × 280	310 × 310*	Downsample Bicubic	91,39	-2,20
	112 × 112	310 × 310*	Downsample Bicubic	90,14	-3,49
	48 × 48	310 × 310*	Downsample Bicubic	76,88	-17,69

[1] Trainings- und Testauflösung
[2] Ausgangsauflösung, wovon neue Trainings- und Testauflösung erzeugt wurde
* Durchschnittliche Auflösung der Ausgangsdaten, siehe Unterabschnitt 3.2.2

Bezogen auf das Training mit AffectNet liegt die höchste Accuracy bei 60,34 % und wird wie erwartet mit der höchsten Auflösung von 224 × 224 Pixeln erreicht. Wird das Modell mit einer Auflösung von 112 × 112 Pixeln trainiert und getestet, sinkt die Accuracy auf 57,31 %, was einer relativen Veränderung von -5,02 % entspricht. Das Training mit der niedrigsten

Auflösung von 48 × 48 Pixeln führt auch zur geringsten Accuracy mit 49,56 % und damit einer relativen Veränderung zur höchsten Accuracy von -17,87 %.
Die mit den RaFD-Daten trainierten Modelle zeigen sehr ähnliche Ergebnisse hinsichtlich der relativen Veränderung. Die höchste Accuracy wird ebenfalls mit einer Auflösung von 224 × 224 Pixeln erzielt und liegt hier bei 93,4 %. Allerdings erreichen die höher aufgelösten Bilder von 280 × 280 Pixeln mit 91,39 % Accuracy 2,2 % weniger, was sich nicht komplett mit der Hypothese deckt, dass mehr Bildinformationen auch für höhere Erkennungsraten sorgen. Wird mit den RaFD-Bildern in 112 × 112 Pixeln trainiert, sinkt die Accuracy um 3,49 % ggü. 224 × 224 Pixeln auf 90,14 %. Diese relative Veränderung ist 1,53 Prozentpunkte weniger stark, als bei dem Training mit AffectNet. Das Training mit den RaFD-Daten in 48 × 48 Pixeln zeigt auch hier mit -17,69 % den größten Abfall auf eine Accuracy von 76,88 % ggü. dem Training mit 224 × 224 Pixeln aufgelösten Daten. Der Unterschied in der relativen Veränderung zwischen RaFD und AffectNet liegt hier bei 0,18 Prozentpunkten.

Hinsichtlich Frage i), scheinen höhere Bildauflösungen bei der Emotionserkennung durchaus einen positiven Effekt auf die Erkennungsraten von DNNs zu haben. Der optimale Auflösungsbereich, wonach in ii) gefragt wird, lässt sich anhand der höchsten Accuracys auf beiden Datensätzen auf einen Bereich um 224 × 224 Pixel eingrenzen.



(a) 224 × 224 (Original)

(b) 48 × 48 (Downsample Bicubic)

Abbildung 4.1: Confusion Matrizen AffectNet – Vergleich vorhergesagter und tatsächlicher Klassen der Testdaten zu den entsprechenden Auflösungen aus Tab. 4.1

Abbildung 4.1 zeigt in (a) die *Confusion Matrix* der Vorhersagen für die Testdaten in 224 × 224 Pixeln (Originalauflösung) sowie in (b) die Confusion Matrix für die Vorhersagen der Daten in 48 × 48 Pixeln (Downsample Bicubic) von AffectNet. Die Hauptdiagonale zeigt jeweils die Anzahl der korrekt klassifizierten Emotionen, Werte außerhalb davon sind Fehlklassifikationen. Die Summe der Werte einer Zeile ergibt die Anzahl der Bilder pro Klasse, die summierten Werte einer Spalte geben an, wie oft das Modell eine Klasse vorhergesagt hat.

Tabelle 4.2: AffectNet – Metriken nach Klassen für Vergleich von 224 × 224 & 48 × 48

	Precision, %	Recall, %	F-Score, %
neutral	45,85	58,60	51,45
happiness	69,67	80,40	74,65
sadness	59,66	63,60	61,57
surprise	57,09	61,20	59,07
fear	71,46	59,60	64,99
disgust	66,39	48,63	56,12
anger	54,94	57,80	56,34
contempt	65,19	52,91	58,41

(a) 224 × 224 (Original)

	Precision, %	Recall, %	F-Score, %
neutral	37,21	45,40	40,90
happiness	59,84	74,20	66,25
sadness	48,18	45,00	46,54
surprise	50,00	49,00	49,49
fear	60,89	46,40	52,67
disgust	54,89	34,80	42,59
anger	45,45	52,00	48,51
contempt	45,76	49,70	47,65

(b) 48 × 48 (Downsample Bicubic)

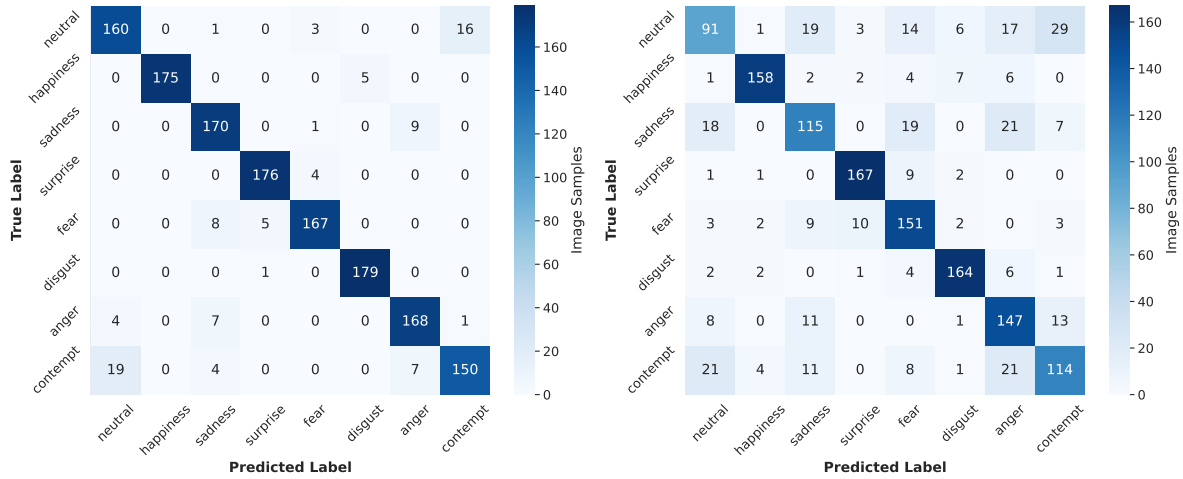
	neutral	happiness	sadness	surprise	fear	disgust	anger	contempt
Precision, %	-18,84	-14,11	-19,24	-12,42	-14,79	-17,32	-17,27	-29,81
Recall, %	-22,53	-7,71	-29,25	-19,93	-22,15	-28,40	-10,03	-6,07
F-Score, %	-20,51	-11,25	-24,41	-16,22	-18,96	-24,11	-13,90	-18,42

(c) Relative Veränderung der Werte von 224 × 224 (a) zu 48 × 48 (b)

Die Klasse „happiness“ wurde von beiden Modellen am öftesten korrekt vorhergesagt, „disgust“ hingegen am wenigsten oft. Tabelle 4.2 gibt noch einen detaillierteren Einblick zur Vorhersage der einzelnen Klassen durch die Modelle mit der Angabe von Precision, Recall und F-Score (siehe hierzu auch Unterabschnitt 2.1.3) sowie der relativen Veränderung der Werte zwischen den beiden darin verglichenen Modellen. Hieran lässt sich feststellen, dass das Modell mit den in 48 × 48 Pixeln aufgelösten Daten vor allem „sadness“ und „disgust“ mit einer Abnahme des Recalls von 29,25 % und 28,4 % weniger gut erkennt (mehr falsch negative Ergebnisse), bei der Klasse „contempt“ mit einer Abnahme der Precision um 29,81 % deutlich mehr Fehlklassifikationen stattfinden (mehr falsch positive Ergebnisse). Insgesamt nimmt die Leistung bei den niedriger aufgelösten Daten hinsichtlich Precision und Recall deutlich ab, wobei die Veränderungen hinsichtlich F-Score (harmonisches Mittel zwischen den beiden Metriken) bei „happiness“ und „anger“ nicht ganz so stark ausfallen, wie bei den anderen Klassen.

Die Detailergebnisse zum Vergleich der entsprechenden Modelle mit RaFD sind in Abbildung 4.2 und Tabelle 4.4 zu sehen. Wie bereits die Accuracys in Tabelle 4.1 zeigen, zeigt das Modell mit den in 224 × 224 Pixeln trainierten Daten eine sehr hohe Klassifizierungsleistung und hat so z. B. bei der Klasse „happiness“ keine Fehlklassifizierungen, was sich an der Confusion Matrix oder der Precision von 100 % ablesen lässt. Am schlechtesten erkennt es die Klasse „contempt“ mit einem Recall von 83,33 %. Das Modell mit den in 48 × 48 Pixeln trainierten Bildern nimmt insgesamt zwar in der Leistung ab, bleibt aber in einigen Klassen etwas stabiler, wie sich Tabelle 4.1 (c) bei der relativen Veränderung entnehmen lässt. So finden bei den

Klassen „surprise“, „happiness“ und „disgust“ mit -5,38 %, -7,9 % und -8,12 % im F-Score die wenigsten Veränderungen statt.



(a) 224 × 224 (Original)

(b) 48 × 48 (Downsample Bicubic)

Abbildung 4.2: Confusion Matrizen RaFD – Vergleich vorhergesagter und tatsächlicher Klassen der Testdaten zu den entsprechenden Auflösungen aus Tab. 4.1

Tabelle 4.4: RaFD – Metriken nach Klassen für Vergleich von 224 × 224 & 48 × 48

	Precision, %	Recall, %	F-Score, %
neutral	87,43	88,89	88,15
happiness	100,00	97,22	98,59
sadness	89,47	94,44	91,89
surprise	96,70	97,78	97,24
fear	95,43	92,78	94,08
disgust	97,28	99,44	98,35
anger	91,30	93,33	92,31
contempt	89,82	83,33	86,46

(a) 224 × 224 (Downsample Bicubic)

	Precision, %	Recall, %	F-Score, %
neutral	62,76	50,56	56,00
happiness	94,05	87,78	90,80
sadness	68,86	63,89	66,28
surprise	91,26	92,78	92,01
fear	72,25	83,89	77,63
disgust	89,62	91,11	90,36
anger	67,43	81,67	73,87
contempt	68,26	63,33	65,71

(b) 48 × 48 (Downsample Bicubic)

	neutral	happiness	sadness	surprise	fear	disgust	anger	contempt
Precision, %	-28,22	-5,95	-23,04	-5,63	-24,29	-7,87	-26,14	-24,00
Recall, %	-43,12	-9,71	-32,35	-5,11	-9,58	-8,38	-12,49	-24,00
F-Score, %	-36,47	-7,90	-27,87	-5,38	-17,49	-8,12	-19,98	-24,00

(c) Relative Veränderung der Werte von 224 × 224 (a) zu 48 × 48 (b)

4.2 Upsampling-Ergebnisse

Tabelle 4.6 liefert Ergebnisse zu Fragestellung iii), ob Super-Resolution-Methoden positive Effekte höherer Bildauflösung replizieren kann. Visuelle Beispiele des Upsamplings durch die verschiedenen Methoden können in Abbildung 4.4 betrachtet werden.

Tabelle 4.6: Vergleichsergebnisse (Accuracy) der durch Super-Resolution/Upsampling erzeugten Auflösungen und anschließendem Training von EfficientNet-B0 mit AffectNet. Trainings- und Testdaten wurden jeweils unter denselben Bedingungen (Ausgangsauflösung & Verfahren) erzeugt.

	Auflösung[1]	**Ursprung**[2]	**Verfahren**	**Accuracy, %**	**Veränderung, %**
(a)	224 × 224	224 × 224	Original	60,34	Basis
	224 × 224	48 × 48	Upsample Bicubic	56,09	-7,04
	224 × 224	48 × 48	Upsample HAT	55,29	-8,37
	224 × 224	48 × 48	Upsample GFPGAN	54,51	-9,66
(b)	112 × 112	224 × 224	Downsample Bicubic	57,31	Basis
	112 × 112	48 × 48	Upsample HAT	54,36	-5,15
	112 × 112	48 × 48	Upsample Bicubic	53,54	-6,58
	112 × 112	48 × 48	Upsample GFPGAN	52,81	-7,85
(c)	224 × 224	224 × 224	Original	60,34	Basis
	448 × 448	48 × 48	Upsample Bicubic	55,56	-7,92
	448 × 448	48 × 48	Upsample GFPGAN	55,01	-8,83
(d)	224 × 224	224 × 224	Original	60,34	Basis
	448 × 448	224 × 224	Upsample Bicubic	59,64	-1,16
	448 × 448	224 × 224	Upsample GFPGAN	58,49	-3,07
	448 × 448	224 × 224	Upsample HAT	35,16	-41,73

[1] Trainings- und Testauflösung
[2] Ausgangsauflösung, wovon neue Trainings- und Testauflösung erzeugt wurde

Die mit den in 224 × 224 Pixeln vorliegenden AffectNet-Originaldaten erreichte Accuracy von 60,34 % ist auch im Vergleich mit sämtlichen durch die Upsampling-Verfahren (Super-Resolution HAT & GFPGAN, bikubische Interpolation) generierten Auflösungen der höchste Wert und wird von keinem Verfahren sowie keiner höheren Auflösung übertroffen.
Tab. 4.6 (a) zeigt die Ergebnisse des Upsamplings von 48 × 48 Pixeln auf die Originalgröße von 224 × 224 Pixeln. Im Vergleich zwischen den Upsampling-Verfahren erreicht die bikubische Interpolation mit 56,09 % Accuracy einen höheren Wert als die beiden SR-Verfahren HAT und GFPGAN mit 55,29 % bzw. 54,51 %. Damit liegt die Interpolation 7,04 % unter dem Bestwert mit den Originalbildern, HAT 8,37 % und GFPGAN 9,66 % darunter.
In Tab. 4.6 (b) sind die Ergebnisse des Vergleichs der unterschiedlich erzeugten Bilder in

112 × 112 Pixeln aufgeführt. Auch hier erreichen die bikubisch downgesampelten Originalbilder die höchste Accuracy mit 57,31 %. Beim Upsampling von 48 × 48 auf 112 × 112 Pixel ist das beste Verfahren die Super-Resolution mit HAT, das 54,36 % Accuracy erreicht und damit 5,15 % unter der besten Accuracy dieses Auflösungsbereichs liegt. Danach folgt die bikubische Interpolation mit 53,54 % Accuracy, was einer Veränderung von -6,58 % zum downgesampelten Original entspricht. GFPGAN liefert mit 52,81 % Accuracy den schlechtesten Wert und liegt 7,85 % unter dem Bestwert des Auflösungsbereichs von 112 × 112 Pixeln.
Die Ergebnisse des Upsamplings von 48 × 48 auf 448 × 448 Pixel werden in Tab. 4.6 (c) dargestellt. Da die höchste verfügbare Auflösung der Originalbilder bei 224 × 224 Pixeln liegt, dient der Test mit diesen Daten als Basis für die relative Veränderung. Die bikubische Interpolation auf 448 × 448 erreicht eine Accuracy von 55,56 % was einer Veränderung von -7,92 % zum Bestwert durch Training sowie Test mit Bildern in 224 × 224 Pixeln entspricht. Die durch GFPGAN generierten Bilder erreichen hingegen eine Accuracy von 55,01 % und liegen 8,83 % unter der höchsten Accuracy.
Beim stichprobenartigen Betrachten von durch GFPGAN upgesampelten Bildern mit einer Ausgangsauflösung von 48 × 48 Pixeln fallen vereinzelt stärkere visuelle Merkmalsverfremdungen auf, von denen einige in Abbildung 4.3 dargestellt werden.

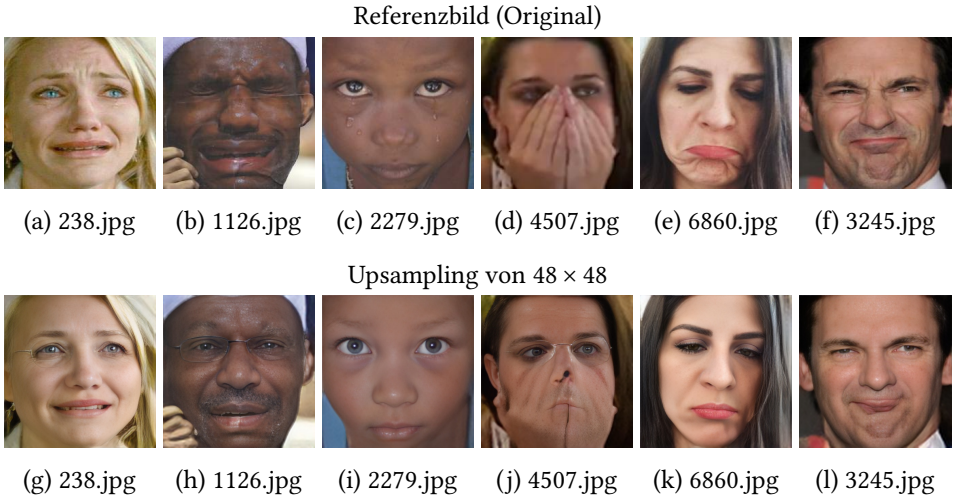


Abbildung 4.3: Beispiele starker visueller Merkmalsverfremdung durch GFPGAN beim Upsampling von Bildern in 48 × 48 Pixeln. (Bildquelle: AffectNet [108])

Tab. 4.6 (d) führt die Ergebnisse des Upsamplings der Bilder von 224 × 224 zu 448 × 448 Pixeln auf, wobei auch hier die mit den Originalbildern in 224 × 224 Pixeln höchste erzielte Accuracy von 60,34 % als Basis zum Bewerten der relativen Veränderung dient. Das Upsampling der höchstmöglichen verfügbaren Auflösung von 224 × 224 Pixeln zeigt keine Verbesserung in der Accuracy, weder mit den eingesetzten SR-Methoden, noch mit der bikubischen Interpolation. Dabei erreicht das bikubische Upsampling mit 59,64 % die höchste Accuracy und liegt damit 1,16 % unter dem Basiswert von 60,34 %. Die mit GFPGAN generierten Daten kommen auf

eine Accuracy von 58,49 % (-3,07 % relative Veränderung), was damit das erste Mal in den Versuchen vor den mit HAT generierten Daten liegt. Diese erreichen eine Accuracy von nur 35,16 % und liefern damit das schlechteste Ergebnis aller Versuche bei einer relativen Veränderung von -41,73 % zum Bestwert mit 224 × 224 Pixeln. Letzteres entspricht nicht der Erwartung, da das visuelle Ergebnis des Upsamplings von 224 × 224 auf 448 × 448 Pixel aus subjektiver Sicht deutlich besser wirkt, als bspw. das des Upsamplings von 48 × 48 auf 224 × 224 Pixel (vgl. Abb. 4.4 [c] und [h]). Im Bereich von 112 × 112 Pixeln (b) kann HAT als Upsample-Verfahren die bikubische Interpolation mit der Accuracy um 0,82 Prozentpunkte übertreffen, in allen anderen Versuchen liegt das Interpolationsverfahren stets vor den SR-Methoden.
Die Accuracys, die mit den von 48 × 48 Pixeln upgesampelten Bilddaten erreicht werden (a-c), deuten ebenfalls darauf hin, dass sich eine höhere Auflösung positiv auf die Erkennungsraten bzgl. Frage i) auswirkt.

Bei Betrachtung der Metriken nach einzelnen Klassen von den mit 224 × 224 Pixeln trainierten Modellen in Tabelle 4.7 lässt sich erkennen, dass die SR-Methoden HAT und GFPGAN nicht bei allen vorhergesagten Klassen schlechtere Werte als die bikubische Interpolation vorweisen. So erkennen beide Verfahren z. B. etwas mehr Bilder der Klasse „happiness“ (Recall) bei weniger Fehlklassifikationen (Precision). Über alle Verfahren bleibt diese Klasse aber am stabilsten mit einer maximalen Leistungsabnahme im F-Score um 4,57 % gegenüber dem Training mit den Originaldaten. Die Vorhersagen zur Klasse „contempt“ nehmen in allen Verfahren deutlich ab, wobei GFPGAN die größte Abnahme von 16,2 % im F-Score zeigt, verglichen mit -13,54 % bei HAT und -12,74 % bei der Interpolation.
Die größten Diskrepanzen (gemessen am F-Score) zwischen der bikubischen Interpolation und den SR-Verfahren bestehen in den Klassen „anger“ (GFPGAN mit 2,98, HAT mit 2,68 Prozentpunkten weniger als Interpolation) und „sadness“ (GFPGAN mit 2,53 und HAT mit 2,47 Prozentpunkten weniger als Interpolation). Wie bereits die Accuracys in Tabelle 4.6 zeigen, sind die Leistungswerte der Modelle mit upgesampelten Trainingsdaten nicht sehr weit voneinander entfernt.
Zusammenfassend zeigt sich, dass die Veränderung der Leistung durch Anwenden unterschiedlicher Upsampling-Verfahren inkl. Super-Resolution je nach Emotion variiert, also keine gleichmäßige Ab- oder Zunahme in den Klassen stattfindet.

Tabelle 4.7: AffectNet Upsampling – Metriken nach Klassen für mit 224 × 224 trainierte Modelle zu Tab. 4.6 (a)

	Precision, %	Recall, %	F-Score, %
neutral	45,85	58,60	51,45
happiness	69,67	80,40	74,65
sadness	59,66	63,60	61,57
surprise	57,09	61,20	59,07
fear	71,46	59,60	64,99
disgust	66,39	48,63	56,12
anger	54,94	57,80	56,34
contempt	65,19	52,91	58,41

(a) 224 × 224 (Original)

	Precision, %	Recall, %	F-Score, %
neutral	44,79	49,00	46,80
happiness	66,44	76,80	71,24
sadness	53,61	62,40	57,67
surprise	53,61	56,40	54,97
fear	64,86	55,00	59,52
disgust	65,07	43,60	52,22
anger	50,35	58,00	53,90
contempt	54,99	47,49	50,97

(b) 224 × 224 (Upsample Bicubic)

	Precision, %	Recall, %	F-Score, %
neutral	41,98	51,80	46,37
happiness	66,72	78,20	72,01
sadness	52,33	58,40	55,20
surprise	52,79	56,80	54,72
fear	63,95	56,40	59,94
disgust	64,16	42,60	51,20
anger	50,10	52,40	51,22
contempt	56,44	45,69	50,50

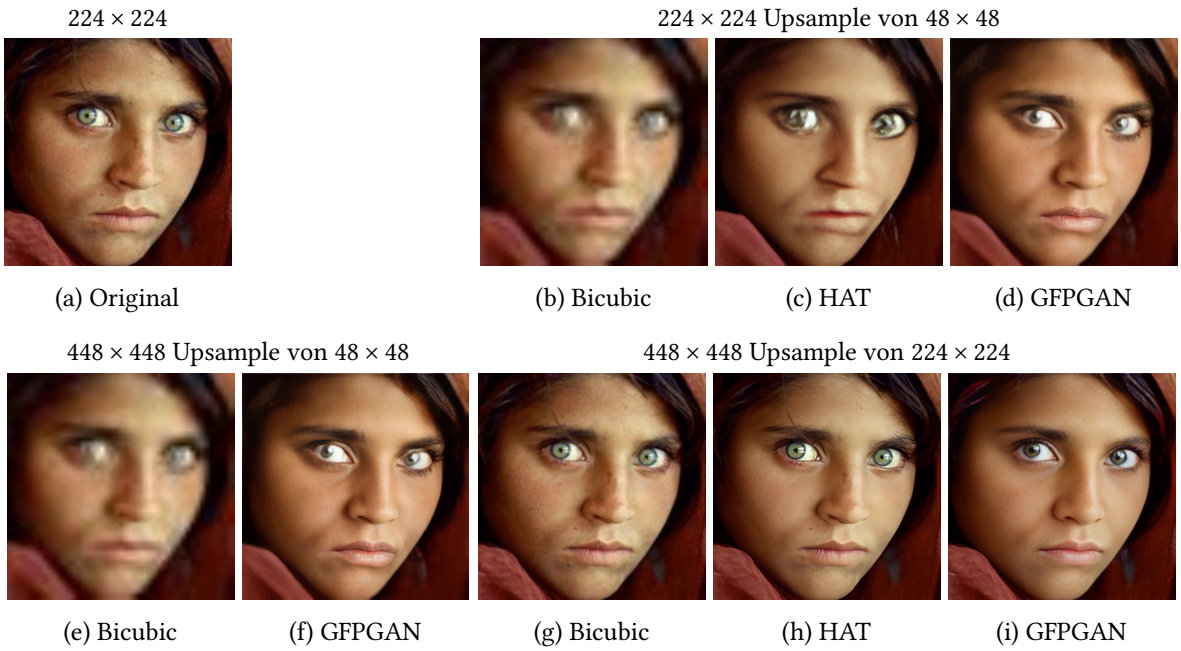
(c) 224 × 224 (Upsample HAT)

	Precision, %	Recall, %	F-Score, %
neutral	41,67	50,00	45,45
happiness	66,84	78,20	72,07
sadness	53,05	57,40	55,14
surprise	50,18	54,40	52,21
fear	62,72	56,20	59,28
disgust	61,71	43,20	50,82
anger	49,53	52,40	50,92
contempt	54,70	44,29	48,95

(d) 224 × 224 (Upsample GFPGAN)

		neutral	happiness	sadness	surprise	fear	disgust	anger	contempt
Bicubic Up.	**Precision, %**	-2,31	-4,64	-10,14	-6,10	-9,24	-1,99	-8,35	-15,65
	Recall, %	-16,38	-4,48	-1,89	-7,84	-7,72	-10,29	0,35	-10,24
	F-Score, %	-9,04	-4,57	-6,33	-6,94	-8,42	-6,95	-4,33	-12,74
HAT	**Precision, %**	-8,44	-4,23	-12,29	-7,53	-10,51	-3,36	-8,81	-13,42
	Recall, %	-11,60	-2,74	-8,18	-7,19	-5,37	-12,35	-9,34	-13,65
	F-Score, %	-9,87	-3,54	-10,35	-7,36	-7,77	-8,77	-9,09	-13,54
GFPGAN	**Precision, %**	-9,12	-4,06	-11,08	-12,10	-12,23	-7,05	-9,85	-16,09
	Recall, %	-14,68	-2,74	-9,75	-11,11	-5,70	-11,11	-9,34	-16,29
	F-Score, %	-11,66	-3,46	-10,44	-11,61	-8,79	-9,44	-9,62	-16,20

(e) Relative Veränderung der Werte jeweils von (a) zu (b), (c), (d)



(a) Original (b) Bicubic (c) HAT (d) GFPGAN

(e) Bicubic (f) GFPGAN (g) Bicubic (h) HAT (i) GFPGAN

Abbildung 4.4: Visuelle Beispiele der Upsampling-Ergebnisse. Alle Formate werden hier zur besseren Betrachtung in derselben Größe dargestellt.
(Bildquelle: AffectNet [108] 5366.jpg)

Um einen besseren Einblick zu bekommen, wie sich die Leistung des mit einer Accuracy von 35,16 % wider Erwarten am schlechtesten abschneidenden Modells in den einzelnen Klassen widerspiegelt, sind Beispiele sowie die Ergebnisse des mit HAT von 224 × 224 auf 448 × 448 Pixel upgesampelten AffectNet-Datensatzes in Abbildung 4.6 und Tabelle 4.9 dargestellt.

Neutral (1025.jpg) | Freude (660.jpg) | Traurigkeit (1412.jpg) | Überrascht (3645.jpg) | Angst (1037.jpg) | Ekel (3555.jpg) | Wut (5465.jpg) | Verachtung (3987.jpg)

Abbildung 4.5: Visuelle Beispiele des Upsamplings der AffectNet-Daten mit HAT von 224 × 224 auf 448 × 448 Pixel. (Bildquelle: AffectNet [108])

Wie die Werte zeigen, können die Klassen „neutral“, „happiness“, „surprise“ und „fear“ nicht mehr korrekt klassifiziert werden. In den Klassen „sadness“, „disgust“, „anger“ und „contempt“ werden zwar einige der Bilder noch korrekt identifiziert, es finden zusätzlich aber sehr viele Fehlklassifikationen statt, was sich in der niedrigen Precision zu diesen Klassen zeigt. Dieses Modell liefert Ergebnisse, die nicht den Erwartungen entsprechen, da das Super-Resolution-Netzwerk HAT sogar mehr Bildinformationen als beim Upsampling von 48 × 48

Pixeln zu Verfügung hatte. Auch aus subjektiver Sicht lassen die visuellen Ergebnisse dieses Upsamplings (siehe Beispiele in Abb. 4.5 pro Klasse) die schlechte Leistung dieses Modells nicht vermuten.

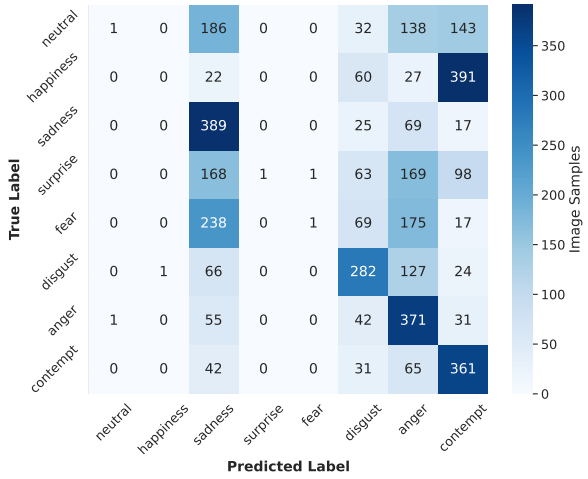


Abbildung 4.6: Confusion Matrix – 224 × 224 auf 448 × 448 (HAT)

	Precision, %	Recall, %	F-Score, %
neutral	50,00	0,20	0,40
happiness	0,00	0,00	0,00
sadness	33,36	77,80	46,70
surprise	100,00	0,20	0,40
fear	50,00	0,20	0,40
disgust	46,69	56,40	51,09
anger	32,52	74,20	45,22
contempt	33,36	72,34	45,67

Tabelle 4.9: Metriken nach Klassen für 224 × 224 auf 448 × 448 (HAT)

4.3 Ergänzende Ergebnisse

In den Abschnitten 3.3 und 3.4 wurde zusätzlich zu den Hauptfragestellungen i)-iii) dieser Arbeit die Frage aufgeworfen, wie sich initiales Zero-Padding (vor der Verarbeitung eines Bildes durch das Netzwerk) im Vergleich zur Verwendung der nativen Auflösung bzw. dem Interpolieren auf eine höhere auswirkt. Hierzu liefern die Ergebnisse in Tabelle 4.10 Aufschluss.

Beide Datensätze (RaFD und AffectNet) zeigen ein ähnliches Verhalten, auch wenn die relative Veränderung voneinander abweicht. So erreicht ein Modell, das mit der nativ geringen Auflösung von 48 × 48 Pixeln trainiert wird, jeweils die schlechteste Accuracy mit 49,56 % bei AffectNet und 76,88% bei RaFD. Diese Accuracys werden als Basis für die relative Veränderung der nachfolgenden Versuche genommen. Werden die kleinen Dimensionen von 48 × 48 Pixeln durch Zero-Padding auf 224 × 224 Pixel erweitert, steigert sich die Accuracy bei AffectNet leicht um 1,67 % auf 50,39 %, bei RaFD deutlicher um 7,13 % auf 82,36 %. Das bikubische Upsampling erreicht, wie aus den vorherigen Ergebnissen in den Abschnitten 4.1 und 4.2 bereits ersichtlich, die höchste Accuracy mit 56,09 % (+13,18 %) bei AffectNet und 90,07 % (+17,16 %) bei RaFD. Das zeigt einerseits, dass Zero-Padding allein durch das Erweitern der Dimensionen mit Nullwerten bereits in einem Netzwerk wie EfficientNet-B0 für einen

Tabelle 4.10: Ergänzende Ergebnisse zum Vergleich von Zero-Padding und Interpolation in EfficientNet-B0 anhand der Daten von AffectNet und RaFD. Trainings- und Testdaten wurden jeweils unter denselben Bedingungen (Ausgangsauflösung & Verfahren) erzeugt.

	Auflösung[1]	**Ursprung**[2]	**Verfahren**	**Accuracy, %**	**Veränderung, %**
AffectNet	48 × 48	224 × 224	Downsample Bicubic	49,56	Basis
	224 × 224	48 × 48	Zero-Padding	50,39	+1,67
	224 × 224	48 × 48	Upsample Bicubic	56,09	+13,18
RaFD	48 × 48	310 × 310*	Downsample Bicubic	76,88	Basis
	224 × 224	48 × 48	Zero-Padding	82,36	+7,13
	224 × 224	48 × 48	Upsample Bicubic	90,07	+17,16

[1] Trainings- und Testauflösung

[2] Ausgangsauflösung, wovon neue Trainings- und Testauflösung erzeugt wurde

* Durchschnittliche Auflösung der Ausgangsdaten, siehe Unterabschnitt 3.2.2

Leistungszuwachs sorgt. Andererseits werden die Aussagen von Hashemi in [57], dass Zero-Padding keinen Effekt auf die Accuracy hat ggü. der bikubischen Interpolation, anhand dieser Ergebnisse nicht bestätigt. Bei der Frage, wie Bilder in ihrer Größe verändert werden sollten, sofern dies zur Eingabe in ein Netzwerk erforderlich ist, deuten die Werte aus Tabelle 4.10 eindeutig auf die bikubische Interpolation (gegenüber Zero-Padding) hin.

5 Diskussion

Das Ziel der vorliegenden Arbeit war es, empirisch den Einfluss von Super-Resolution in der Emotionserkennung zu untersuchen, die durch Klassifikation von Bildern mit Gesichtsausdrücken der Basisemotionen mithilfe von Deep Neural Networks realisiert wird. Hierzu wurden Versuche durchgeführt, die zunächst allgemein den Effekt unterschiedlicher Bildauflösungen auf die Leistung solcher Netzwerke beleuchtet haben. Dies geschah unter der Annahme, dass eine höhere Pixelanzahl aufgrund des höheren Informationsgehalts auch zu besseren Erkennungsraten führen würde, um in anschließenden Experimenten zu untersuchen, ob sich die durch eine höhere Auflösung erwartete bessere Klassifikationsleistung auch durch das Upsampeln gering aufgelöster Bilddaten insbesondere durch Super-Resolution replizieren lässt.

5.1 Einfluss der Bildauflösung

Beim Vergleich von den durch bikubisches Downsampling verschieden erstellten Bildauflösungen der verwendeten Datensätze verdeutlichen die Ergebnisse, dass eine höhere Bildauflösung tendenziell auch für bessere Erkennungsraten sorgt. So erzielt eine Auflösung von 224×224 Pixeln sowohl bei AffectNet als auch RaFD die höchste Accuracy (vgl. Tab. 4.1) und übertrifft die als Richtwert herangezogenen Leistungswerte von EffiecientNet-B0 in den Arbeiten [122] und [152]. Auch die Ergebnisse der Modelle, deren Trainingsbilder durch HAT, GFPGAN oder bikubisch upgesampelt wurden zeigen, dass die Leistung von höheren Auflösungen profitiert. So sind die Accuracys durch das Upsampling der Bilder von 48×48 zu 224×224 Pixeln höher als zu denen mit 112×112 Pixeln. Zusätzlich kann das Upsampling von GFPGAN von 48×48 zu 448×448 die Leistung gegenüber der Bilder in 224×224 Pixeln desselben Verfahrens noch um 0,5 Prozentpunkte übertreffen. Letzteres kann zwar auch auf die unterschiedlich verwendete Batch-Size zurückzuführen sein, insgesamt bestätigt dies aber die einleitend aufgestellte Hypothese, dass eine höhere Bildauflösung sowie mehr Bildinformationen (ein höherer Detailgrad) bis zu einem gewissen Grad vorteilhaft für die Erkennungsleistung in einem Deep Neural Network sind, da dem Modell mehr Bildinformationen zur Verfügung stehen. Dennoch zeigt die leichte Abnahme der Accuracy bei den etwas höher aufgelösten Bildern von RaFD mit 280×280 Pixeln ggü. der optimalen Auflösung von 224×224 Pixeln,

dass ab einem gewissen Punkt möglicherweise kein Leistungszuwachs mehr zu erwarten ist oder zumindest die recht geringe Steigerung von 224 auf 280 Pixel je Seite nicht ausreicht, um für eine weitere Verbesserung zu sorgen. Der Test mit der höchsten RaFD-Auflösung ist jedoch auch kritisch zu betrachten, da einige der Daten dieses Versuchs eine geringere native Auflösung als 280×280 Pixel hatten, dieser Teil also nach dem Zuschneiden der Gesichter des Datensatzes durch Interpolation upgesampelt werden musste, womit die Ergebnisse leicht verfälscht sein könnten. Es kann somit nicht ausgeschlossen werden, dass eine deutlich höhere Auflösung nicht doch für eine höhere Modelleistung sorgt. Hierbei könnte auch die Wahl eines tieferen Netzwerks eine Rolle spielen, das evtl. durch weitere Faltungsschichten oder andere zusätzliche Operationen in der Lage wäre, noch mehr Informationen aus höheren Auflösungen zu extrahieren. Gleichzeitig besteht bei sehr hoch aufgelösten Bildern vermutlich auch eine erhöhte Gefahr von schnellem Overfitting, wenn sich dies nicht durch den verstärkten Einsatz von Regularisierungstechniken eindämmen lässt, sodass ein hoher Informationsgehalt durch noch mehr Pixel nicht unbedingt zu einer verbesserten Netzwerkleistung führen muss.
Zu beachten bei der Verarbeitung und Eingabe der kleineren Bilder ist, dass das verwendete EfficientNet-B0 im Laufe der Verarbeitung durch seine Schichten die Dimensionen fünf Mal halbiert. Der Tensor eines mit 224×224 Pixeln aufgelösten Bildes hat anfangs so die Dimensionen $(224, 224, c)$ und nach dem letzten Convolutional-Block (Stage 8 bzw. Block 7, vgl. Tab. 3.11 und Abb. 3.7) $(7, 7, c)$, wobei c für die Anzahl der Kanäle bzw. Filter steht. Tensoren mit den Dimensionen $(112, 112, c)$ und $(48, 48, c)$ haben am Ende die Form $(4, 4, c)$, aufgerundet von $112 \div 2^5 = 3{,}5$ bzw. $(2, 2, c)$, aufgerundet von $48 \div 2^5 = 1{,}5$. Die Faltungsfenster haben je nach Block eine Größe von 5×5 bzw. 3×3, was bedeutet, dass im Fall von 112×112 Pixeln in Stage 8 gerade noch zusätzliche Filterhierarchien gebildet werden können. Bei 48×48 Pixeln wird hingegen bereits ab Stage 6 immer nur derselbe Ausschnitt betrachtet, da der Tensor kleiner als das Faltungsfenster ist und somit durch Zero-Padding erweitert werden muss, damit dieser überhaupt weiter verarbeitet werden kann. Dies könnte eine Erklärung dafür sein, dass die Leistung von 112×112 zu 224×224 Pixeln deutlich schwächer abfällt, als die des Modells mit den in 48×48 Pixeln aufgelösten Bildern. In wie weit ein weniger tiefes Netzwerk bessere Ergebnisse mit gering aufgelösten Daten erzielen bzw. ob die mehrmaligen Betrachtungen der Fenster des gleichen Bereichs eine negative Auswirkung auf das Ergebnis haben könnte, lässt sich damit nicht beantworten und bedarf weiterer Untersuchung. Die fehlende Möglichkeit von EfficientNet-B0 bei Bildern einer Auflösung von 48×48 Pixeln in späteren Stufen zusätzliche Filterhierarchien zu schaffen würde außerdem erklären, warum das Upsamplen unabhängig vom Verfahren besser funktioniert, als das beibehalten dieser Auflösung als Eingabegröße. Wie sich zeigt, kann dafür, wenn auch nicht sonderlich effektiv, bereits initiales Zero-Padding einen kleinen Unterschied machen, obwohl die Pixelinformationen des eigentlichen Motivs nicht verändert werden.

5.2 Effekte von Super-Resolution & Interpolation

Was das Replizieren der Accuracy einer hohen Auflösung von 224 × 224 Pixeln durch Upsampling betrifft, insbesondere durch Super-Resolution, sorgt dieses zwar für eine Annäherung an die Leistung mit den Originalbildern, kann aber, entgegen der in vorangegangenen Studien aufgeführten Erkenntnisse, die in Abschnitt 2.2 genannt werden, nicht für einen erwarteten Leistungszuwachs (Accuracy) gegenüber der einfachen Interpolation sorgen. Diese schneidet interessanterweise in den meisten Fällen besser ab, als die komplexeren SR-Verfahren HAT und GFPGAN. Das deutet darauf hin, dass das eigentlich vorhandene Potenzial dieser beiden Methoden zur Verbesserung von Auflösung und Bilddetails in einem Auflösungsbereich von 48 × 48 Pixeln unter den vorherrschenden Bedingungen nicht richtig eingesetzt werden kann. Die zur Auswahl der verwendeten SR-Methoden berechneten Metriken SSIM und LPIPS in Tabelle 3.7 scheinen somit auch keinen unmittelbaren Hinweis auf die zu erwartende Leistung eines Deep Neural Networks zur Klassifikation von Emotionen zu geben. Sie spiegeln nicht direkt wider, wie merkmalstreu eine Rekonstruktion im Bezug auf Gesichtsausdrücke stattfindet. Denn das Verfahren HAT übertrifft in beiden Werten SSIM und LPIPS die Interpolation, GFPGAN im LPIPS-Wert. Entsprechend wäre auf Grundlage dieser Werte zu erwarten gewesen, dass auch die FER-Leistung der SR-Methoden höher ausfällt. Das Leistungsverhältnis zwischen den Verfahren HAT und GFPGAN wird dennoch korrekt abgebildet, da HAT sowohl in den IQA-Metriken als auch in der FER-Leistung vor GFPGAN liegt, abgesehen von einem Ausreißer in bei 448 × 448 Pixeln. Was in dieser Arbeit nicht getestet wurde, ist SRFormer als einziges Verfahren, das den PSNR-Wert der bikubischen Interpolation übertrifft. Einerseits sind durch dieses Modell, das aus rein subjektiver Wahrnehmung schon schlechtere Ergebnisse als die verwendeten Verfahren erzielt (vgl. Tab. 3.8) und dessen PSNR-Wert sich von der Interpolation nur um 0,082 dB (vgl. Tab. 3.7) unterscheidet, keine signifikanten Leistungssteigerungen zu erwarten. Andererseits wäre in zukünftigen Untersuchungen zu überprüfen, ob ein SR-Modell mit allgemein höherer PSNR (zudem im Vergleich mit Interpolation) auch in FER-Tasks eine höhere Klassifikationsleistung bewirkt. Das ist grundsätzlich zu erwarten, da wie in Unterabschnitt 2.1.7 erläutert wird das PSNR einen absoluten Fehler zum Referenzbild ermittelt. Gleichzeitig ist aufgrund dieser Fehlersensitivität nicht ausgeschlossen, dass es Verfahren gibt, die die für FER entscheidenden Merkmale effektiv verbessern, dadurch aber von der Referenz abweichen und einen entsprechend schlechteren Wert haben. Des Weiteren könnte untersucht werden, inwiefern ein Unterschied zu einem klassisches Modell, das sich auf Degradierung bikubischer Interpolation fokussiert und keinen Gebrauch von bspw. GANs für Real-World-Super-Resolution macht, im Bezug auf den FER-Task feststellbar ist. Ebenso wirft das gute Abschneiden der bikubischen Interpolation die Frage auf, ob weitere positive Leistungsveränderungen durch Einsatz eines in Bezug auf Kantenschärfe optimierten Interpolationsverfahrens wie bspw. Catmull-Rom zu beobachten sind.
Bzgl. generierter Bildverfälschungen von SR-Verfahren in diesem Auflösungsbereich zeigt

Abbildung 4.3 vereinzelte Problemfälle des Upsamplings durch GFPGAN, die verdeutlichen, wie stark Merkmale teilweise verfremdet werden und dadurch eindeutig Einfluss auf die Erkennungsraten nehmen. Auch Tabelle 3.8 sowie Abbildung 4.4 lässt sich bereits entnehmen, dass GFPGAN einige Merkmale, die ggf. entscheidend für eine effektive Verbesserung der Leistung wären, glättet, abschwächt oder verfremdet. HAT hingegen kann in einer geringen Auflösung wie 48 × 48 Pixel die Augenpartie nur selten artefaktfrei generieren, was ebenfalls ein Nachteil ggü. der Interpolation sein kann, wobei HAT diese im Bereich von 112×112 Pixel übertrifft, in höheren Auflösungen wiederum nicht. Das könnte u. a. darauf zurückzuführen sein, dass HAT grundsätzlich Verbesserungen ggü. der Interpolation generieren kann, die von 48×48 Pixel upgesampelten Bilder aber durch die Vervierfachung zunächst nur 192×192 Pixel haben und zum Erreichen von 224 × 224 Pixeln weiter bikubisch Interpoliert werden müssen. Dadurch könnten eventuelle Merkmalsverbesserungen ggf. wieder abgeschwächt werden. Eine weitere Untersuchung mit Daten in einer Auflösung von 192×192 Pixeln würde ggf. noch mehr Klarheit schaffen können, in wie weit die nachträgliche Interpolation einen negativen Einfluss auf die Super-Resolution-Ergebnisse nimmt. Anhand der Ergebnisse zu den einzelnen Klassen lässt sich außerdem nicht feststellen, dass die Super-Resolution-Modelle bestimmte Merkmale von einer Emotion zu einer anderen verschieben, wie es anhand der Beispiele zu CodeFormer (vgl. Tab. 3.8) sichtbar wurde, bei denen die Bilder subjektiv wahrgenommen nach Verarbeitung eher der Klasse „happiness“ zuzuordnen gewesen wären.
Eine weitere Leistungssteigerung beim Vergrößern der Originaldaten von Affectnet von 224 × 224 auf 448 × 448 Pixel durch die verschiedenen Verfahren gegenüber der höchsten Accuracy mit 224 × 224 Pixeln kann ebenso nicht festgestellt werden. Mit diesen Tests wurde angestrebt, die in dem In-the-Wild-Datensatz eventuell bereits vorhandenen Degradierungen so weit zu reduzieren, dass durch den verbesserten Detailgrad eine weitere Leistungssteigerung möglich wäre. Dennoch wird auch hier die bikubische Interpolation von keinem der SR-Verfahren hinsichtlich der FER-Leistung übertroffen. Offen bleibt, ob ein noch tieferes CNN, das weitere Faltungen durchführt, ggf. mehr Merkmale aus diesen upgesampelten Bilddaten extrahieren und damit die Erkennungsleistung weiter steigern kann. Denn wie bereits festgestellt und in Abschnitt 5.1 diskutiert wurde, scheint die optimale Auflösung der verwendeten Architektur EfficientNet-B0 bei 224 × 224 Pixeln zu liegen. Entsprechende Versuche konnten aufgrund der Hardware- und damit einhergehenden Zeitlimitierungen bspw. mit einem der größeren EfficientNet-Netzwerke nicht durchgeführt werden, weshalb weitere Untersuchungen relevant sein könnten. Die von HAT in 448 × 448 ausgehend von 224 × 224 Pixeln generierten Bilder sorgen für die schwächste Accuracy aller Tests. Was bei dem damit trainierten Modell für einen so starken Leistungseinbruch verantwortlich ist, bleibt unklar. Hier lässt sich nur vermuten, dass dies evtl. an bestimmten neu generierten für SR typischen Pixelstrukturen oder Merkmalen liegt, die aufgrund der hohen Auflösung durch das GAN des Real-World-HAT-Modells vermehrt produziert werden oder vorhandene Muster so verstärken, dass diese von den eigentlich für FER wichtigen Merkmalen ablenken. Dass die Verwendung der ImageNet-Gewichte Einfluss darauf gehabt haben könnte, ist

auszuschließen, da die Ergebnisse der anderen beiden Verfahren diese Auffälligkeit nicht zeigen. So erzielt GFPGAN beim Upsampling von 224 × 224 auf 448 × 448 Pixel im Verhältnis deutlich bessere Ergebnisse als bei niedrigeren Auflösungen. Das ist nachvollziehbar, da hier die Originalbilddaten, mit denen die höchste Accuracy aller Versuche erzielt wird, als Ausgang verwendet wird, womit mehr Bildinformationen zur Verfügung stehen und somit auch eine dem Original treuere Rekonstruktion stattfinden kann. Dennoch zeigt Abbildung 4.4 (i) im Vergleich mit (g), dass GFPGAN auch beim Upsampeln einer höheren Auflösung weiterhin sichtlich Gesichtsmerkmale und damit ggf. für Facial Expression Recognition wichtige Merkmale verfremdet. Obgleich seiner hohen, subjektiv wahrgenommenen Bildqualität scheint ein Face-Restoration-Verfahren wie GFPGAN mit dem hier verwendeten Modell daher weniger geeignet für den FER-Tasks zu sein, was vermutlich mit der geringen Varianz an Merkmalen für unterschiedliche Emotionen in den Trainingsdaten zusammenhängt.

Die Ergebnisse widersprechen den Erwartungen, die auch durch die erwähnten Studien gestützt werden ([67], [91], [123], [137]), indem sie zeigen, dass die Leistung von Deep Neural Networks in der Aufgabe der Emotionserkennung anhand niedrig aufgelöster Bilddaten durch die Verwendung der in dieser Arbeit angewandten Super-Resolution-Verfahren nicht stärker verbessert werden kann als mit bikubischer Interpolation.
Indirekt werden damit die Aussagen von [130] und [110] bestätigt, dass Super-Resolution nur bis zu einem Downsampling-Faktor ×4 gute Leistungen hinsichtlich einer Steigerung der Klassifikationsleistung erzielt. Ein Downsampling von 224 × 224 auf 48 × 48 Pixel entspricht bereits einem Faktor von ~4,67. Hierzu muss jedoch angemerkt werden, dass der Downsampling-Faktor selbst kein informatives Maß ist, da vor allem die Ausgangsauflösung, von der upgesampelt wird, Auskunft darüber gibt, wie viel Bildinformationen anfänglich vorhanden sind.
Das Potenzial von Super-Resolution zur Anwendung in FER-Tasks lässt sich in dieser Arbeit mit den verwendeten Methoden nur in Ansätzen (siehe HAT Upsampling auf 112 × 112 Pixel oder Verbesserungen in vereinzelten Klassen) erkennen. Wie bereits in anderen Studien gezeigt werden konnte, ist dieses aber vorhanden. Die Ergebnisse bestätigen, dass das Upsampling gering aufgelöster Bilder für einen Leistungszuwachs bzw. die Annäherung an den Höchstwert mit optimal aufgelösten Daten sorgt. Jedoch deutet die begrenzte Effektivität der verwendeten Verfahren darauf hin, dass die Wahl der optimalen Super-Resolution-Methode und insbesondere eine feinere Anpassung dieser an den FER-Task entscheidend ist.

5.3 Limitationen

Da sich die Arbeit vor allem auf die Vergrößerung sehr kleiner Bildauflösungen durch Super-Resolution fokussiert, wurden nur die Extremfälle untersucht. So lassen sich mit den durch-

geführten Versuchen keine konkreten Aussagen darüber machen, ob Super-Resolution mit HAT und GFPGAN erst ab einer bestimmten Auflösung für eine effektive Bildverbesserung hinsichtlich eines Leistungszuwachses bei der Emotionserkennung sorgen kann.
Die Versuche mit den in 448×448 Pixeln aufgelösten Bildern wurden aufgrund von Hardware-Limitierungen mit einer kleineren Batch-Size (16 statt 32) durchgeführt, was u. U. durch eine ineffizientere Berechnung des Gradienten Einfluss auf die Leistung genommen haben könnte. Im Vergleich zwischen den Upsample-Verfahren innerhalb dieses Auflösungsbereichs ist dies aber vernachlässigbar.
Das verwendete Netzwerk EfficientNet-B0 ist u. U. aufgrund der vielen Dimensionsreduktionsschritte und der dadurch in den unteren Schichten ggf. ineffektiveren Feature-Extraktion ungeeignet für die Verarbeitung von sehr geringen Auflösungen. Umgekehrt ließen sich die Leistungswerte in einem tieferen Netzwerk bei höheren Auflösungen ggf. weiter erhöhen, was dann aber auch an der erhöhten Parameterzahl liegen kann. Die Grundidee von EfficientNet ist das gleichmäßige Skalieren der Dimensionen Breite, Tiefe und Auflösung eines Netzwerks, weshalb unterschiedliche Auflösungen ggf. auch mit unterschiedlichen Netzwerken getestet werden sollten. Außerdem erwartet EfficientNet-B0 in der Standardeinstellung eine Auflösung von 224 × 224 Pixeln [45], was ein weiterer Erklärungsgrund für die mit dieser Auflösung beste erreichte Leistung sein könnte. Der Test von weiteren Auflösungsstufen anhand eines deutlich höher aufgelösten Datensatzes für Facial Expression Recognition ließe noch klarere Aussagen über den optimalen Auflösungsbereich zu. Der RaFD-Datensatz bietet nach dem Zuschnitt nur eine geringe Steigerung ggü. der Auflösung von AffectNet.
Die größte Limitation der vorliegenden Arbeit stellen die verwendeten Super-Resolution-Netzwerke bzw. die zugehörigen vortrainierten Modelle dar. Diese wurden mit Datensätzen trainiert, die einerseits eine sehr breite Motivpalette abbilden, um möglichst universell SR-Aufgaben zu bewältigen und sind somit nicht allein auf Gesichter und deren Merkmale spezialisiert, abgesehen von den Face-Restoration-Verfahren CodeFormer und GFPGAN. Diese bringen aber durch ihre jeweilige Funktionsweise neue Herausforderungen wie das Verfremden von Gesichtsmerkmalen mit sich. Andererseits haben diese Trainingsdaten eine deutlich höhere Auflösung als die in dieser Arbeit verwendete Minimalauflösung von 48 × 48 Pixeln (vgl. Tab. 2.2 unter Berücksichtigung von Downsample-Faktoren von ×2, ×3, ×4 und in wenigen Fällen ×8), sind also nicht ausgelegt für das Upsampeln von solchen gering aufgelösten Daten und können damit voraussichtlich nicht ihr gesamtes Potenzial ausspielen. Dass die SR-Modelle nicht spezifisch auf die Anfordertungen des FER-Tasks angepasst sind, ist eine Erklärung dafür, warum die mit Super-Resolution upgesampelten Bilder entgegen den Erwartungen nicht für eine höhere Leistungssteigerung bei der Emotionserkennung sorgen. Damit sind die vorgestellten Erkenntnisse nicht allgemeingültig für Super-Resolution in der Emotionserkennung, wie auch vorangegangene Studien bereits zeigen konnten, und bedarf weiterer Untersuchung.

6 Fazit

Die vorliegende Arbeit untersucht, inwiefern die Vergrößerung von gering aufgelösten Bildern mithilfe von Super-Resolution-Verfahren einen positiven Einfluss auf die Erkennungsraten von Deep Neural Networks bei der Klassifikation von Emotionen anhand von Gesichtsausdrücken (Facial Expression Recognition) hat. Damit einhergehend werden die Fragen beantwortet, ob eine höhere Bildauflösung für eine bessere Klassifikationsleistung sorgt und welcher Auflösungsbereich eine optimale Erkennungsrate bewirkt. Zur Klärung dieser Fragestellungen wurden Versuche durchgeführt und ausgewertet.

Die Ergebnisse zeigen anhand der Netzwerkarchitektur EfficientNet-B0 und den für FER ausgelegten Datensätzen AffectNet sowie RaFD, dass der optimale Auflösungsbereich zum Erreichen der höchsten Accuracy bei 224×224 Pixeln liegt, was im Falle von AffectNet (60,34 % Accuracy) der Originalauflösung mit dem höchsten Informationsgehalt entspricht, bei RaFD (93,4 % Accuracy) leicht unter der nativen Auflösung von durchschnittlich 310×310 Pixeln liegt. Des Weiteren wurde festgestellt, dass gering aufgelöste Bilder mit einer Auflösung von 48×48 Pixeln und entsprechender Eingabegröße für das Netzwerk zu einer deutlichen Reduktion der Klassifikationsleistung (-17,87 % bei AffectNet, -17,69 % bei RaFD gegenüber dem Höchstwert) führen. Das lässt sich auf die fehlende Möglichkeit des verwendeten Netzwerks zurückführen, in tieferen Schichten weitere Filterhierarchien aufgrund der starken Dimensionsreduktion zu bilden. Aus den Ergebnissen wird ersichtlich, dass initiales Zero-Padding von dieser niedrigen Auflösung hin zur optimalen bereits in geringem Maße (+1,67 % AffectNet, +7,13 % RaFD) zu einer Leistungssteigerung beitragen kann, bikubisches Upsampling von 48 × 48 Pixeln die Verarbeitung aber so positiv beeinflusst, dass trotz der geringen Bildinformationen nur noch ein Leistungsabfall um 7,04 % (56,09 % Accuracy, AffectNet) gegenüber den nativ mit 224 × 224 Pixeln aufgelösten Daten (60,34 % Accuracy, AffectNet) stattfindet.
Aus diesen Erkenntnissen lässt sich schließlich ableiten, dass eine höhere Bildauflösung einerseits aufgrund des höheren Informationsgehalts (höherer Detailgrad) einen positiven Einfluss auf die Erkennungsraten hat, andererseits die damit einhergehenden größeren Dimensionen zu einer effektiveren Merkmalsextrahierung in Convolutional Neural Networks und damit einer höheren Leistung beitragen.

Bzgl. der Optimierung von niedrig aufgelösten Bildern durch Super-Resolution wurde mittels des downgesampelten AffectNet-Datensatzes ausgehend von einer Auflösung von 48 × 48 Pixeln festgestellt, dass das Upsampeln dieser Daten für Facial Expression Recognition mit

den verwendeten Super-Resolution-Modellen, darunter die aktuellen Verfahren HAT und GFPGAN, zwar zu einer grundsätzlichen Verbesserung und Annäherung an die Bestleistung des optimalen Auflösungsbereichs beitragen. Dennoch kann die deutlich weniger komplexe bikubische Interpolation, abgesehen von einem spezifischen Fall, bezogen auf die Gesamtleistung (Accuracy) nicht übertroffen werden: Während eine von 48 × 48 Pixeln auf 224 × 224 Pixel interpolierte Auflösung zu einer Accuracy von 56,09 % gegenüber den nativ entsprechend aufgelösten Daten mit 60,34 % führt (-7,04 %), erreichen die mit HAT generierten Bilder 55,29 % (-8,37 %) und GFPGAN 54,51 % (-9,66 %). Wird lediglich der als nicht optimal befundene Auflösungsbereich von 112 × 112 Pixeln betrachtet, zeigt das Verfahren HAT, dass es die Interpolation im Upsampling hinsichtlich der erreichten Accuracy leicht übertreffen kann (54,36 % gegenüber 53,54 %).
Weiter wird gezeigt, dass die höchste Accuracy von 60,34 % der mit 224×224 Pixel aufgelösten Bilder auch nicht durch die von 224×224 auf 448×448 Pixel upgesampelten Daten übertroffen wird.
Die verwendeten Super-Resolution-Modelle sind für die Vergrößerung von Bildern einer höheren Ausgangsauflösung und im Falle von HAT für ein breites Spektrum an Motiven ausgelegt. Somit waren diese nicht explizit an den speziellen FER-Task angepasst. Anhand der Ergebnisse und dieser Tatsache kann geschlussfolgert werden, dass nicht das gesamte Potenzial dieser Verfahren ausgeschöpft wurde. Abschließend lässt sich festhalten, dass durch diese Arbeit mit den verwendeten Methoden kein durchgehend positiver Effekt auf die Erkennungsleistung in Deep Neural Networks bei der Klassifikation von Emotionen in Gesichtsausdrücken gegenüber einfacheren Upsampling-Methoden wie der bikubischen Interpolation gezeigt werden kann.
Die spezifische Anpassung der SR-Modelle durch auf den FER-Task angepasstes Training mit sehr gering aufgelösten Bildern von Gesichtern stellt entsprechend einen Ansatz für zukünftige Forschungsarbeiten dar. Ebenso ist ein Ansatzpunkt die Verwendung eines noch leistungsstärkeren Netzwerks zur Analyse von höheren Auflösungen. Dass Super-Resolution eine vielversprechende und relevante Methode zur Verbesserung von Emotionserkennungssystemen sein kann, zeigen vorangegangene Studien.

Literatur

[1] C. C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*. Springer International Publishing, 2018, ISBN: 9783319944630. DOI: 10.1007/978-3-319-94463-0.

[2] E. Agustsson und R. Timofte, „NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, Juli 2017. DOI: 10.1109/cvprw.2017.150.

[3] P. Arbeláez, M. Maire, C. Fowlkes und J. Malik, „Contour Detection and Hierarchical Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Jg. 33, Nr. 5, S. 898–916, Mai 2011, ISSN: 2160-9292. DOI: 10.1109/tpami.2010.161.

[4] S. Baker und T. Kanade, „Hallucinating faces," in *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, Ser. AFGR-00, IEEE Comput. Soc, 2000, ISBN: 0-7695-0580-5. DOI: 10.1109/afgr.2000.840616.

[5] C. F. Benitez-Quiroz, R. Srinivasan und A. M. Martinez, „EmotioNet: An Accurate, Real-Time Algorithm for the Automatic Annotation of a Million Facial Expressions in the Wild," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Juni 2016. DOI: 10.1109/cvpr.2016.600.

[6] M. Bevilacqua, A. Roumy, C. Guillemot und M.-l. A. Morel, „Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding," in *Procedings of the British Machine Vision Conference 2012*, Ser. BMVC 2012, British Machine Vision Association, 2012. DOI: 10.5244/c.26.135.

[7] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli und L. Zelnik-Manor, „The 2018 PIRM Challenge on Perceptual Image Super-Resolution," in *Computer Vision – ECCV 2018 Workshops*. Springer International Publishing, 2019, S. 334–355, ISBN: 9783030110215. DOI: 10.1007/978-3-030-11021-5_21.

[8] Bocsika. „BilinearInterpolation.svg." (Okt. 2009), Adresse: https://commons.wikimedia.org/wiki/File:BilinearInterpolation.svg (besucht am 30. 01. 2024).

[9] H. Borgli, V. Thambawita, P. H. Smedsrud *et al.*, „HyperKvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy," *Scientific Data*, Jg. 7, Nr. 1, Aug. 2020, ISSN: 2052-4463. DOI: 10.1038/s41597-020-00622-y.

[10] G. Bradski, „The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[11] K. Bredies und D. Lorenz, *Mathematische Bildverarbeitung*. Vieweg+Teubner, 2011, ISBN: 9783834898142. DOI: 10.1007/978-3-8348-9814-2.

[12] N. Buduma und N. Buduma, *Fundamentals of Deep Learning, Designing Next-Generation Machine Intelligence Algorithms*. O'Reilly Media, Incorporated, 2021, S. 450, ISBN: 9781492082187.

[13] W. Burger, *Digital image processing, An algorithmic introduction* (Texts in computer science), Third edition, M. J. Burge, Hrsg. Cham, Switzerland: Springer, 2022, 943 S., ISBN: 9783031057434.

[14] J. Cai, H. Zeng, H. Yong, Z. Cao und L. Zhang, „Toward Real-World Single Image Super-Resolution: A New Benchmark and a New Model," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, IEEE, Okt. 2019. DOI: 10.1109/iccv.2019.00318.

[15] W. Cai und Z. Wei, „Diversity-Generated Image Inpainting with Style Extraction," Dez. 2019. DOI: 10.20944/preprints201912.0028.v1.

[16] A. Cheikh Sidiya und X. Li, „Toward extreme face super-resolution in the wild: A self-supervised learning approach," *Frontiers in Computer Science*, Jg. 4, Nov. 2022, ISSN: 2624-9898. DOI: 10.3389/fcomp.2022.1037435.

[17] C. Chen, Z. Xiong, X. Tian, Z.-J. Zha und F. Wu, „Camera Lens Super-Resolution," 6. Apr. 2019. DOI: 10.48550/ARXIV.1904.03378. arXiv: 1904.03378 [cs.CV].

[18] X. Chen, X. Wang, J. Zhou, Y. Qiao und C. Dong, *Activating More Pixels in Image Super-Resolution Transformer*, 2022. DOI: 10.48550/ARXIV.2205.04437.

[19] X. (Chen. „HAT GitHub Readme." Commit-Hash (verkürzt): c7e0b2b. (2024), Adresse: https://github.com/XPixelGroup/HAT/blob/main/README.md (besucht am 22. 03. 2024).

[20] Z. Chen, Y. Zhang, J. Gu, L. Kong, X. Yang und F. Yu, *Dual Aggregation Transformer for Image Super-Resolution*, 2023. DOI: 10.48550/ARXIV.2308.03364.

[21] S. Cheng, I. Kotsia, M. Pantic und S. Zafeiriou, „4DFAB: A Large Scale 4D Database for Facial Expression Analysis and Biometric Applications," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Juni 2018. DOI: 10.1109/cvpr.2018.00537.

[22] F. Chollet *et al.*, *Keras*, https://keras.io, 2015.

[23] F. Chollet. „Transfer learning & fine-tuning." (2020), Adresse: https://keras.io/guides/transfer_learning/ (besucht am 25. 03. 2024).

[24] F. Chollet, *Deep Learning with Python, Second Edition*. New York: Manning Publications Co. LLC, 2021, 1470 S., ISBN: 1617296864.

[25] J. A. Clark. „Pillow (PIL Fork), 10.2.0 Documentation." (2024), Adresse: https://pillow.readthedocs.io/_/downloads/en/stable/pdf/ (besucht am 21. 02. 2024).

[26] J. Cleve und U. Lämmel, *Data Mining* (De Gruyter Studium), 3. Auflage. Berlin: De Gruyter, 2020, 1320 S., ISBN: 9783110677294.

[27] J. Cleve und U. Lämmel, *Künstliche Intelligenz*, 5. Auflage. Carl Hanser Verlag München, 2020, 340 S., ISBN: 978-3-446-46363-9.

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li und L. Fei-Fei, „ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Juni 2009. DOI: 10.1109/cvpr.2009.5206848.

[29] N. S. Detlefsen, J. Borovec, J. Schock *et al.*, *TorchMetrics - Measuring Reproducibility in PyTorch*, Feb. 2022. DOI: 10.21105/joss.04101. Adresse: https://www.pytorchlightning.ai.

[30] J. Devlin, M.-W. Chang, K. Lee und K. Toutanova, *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, 2019. DOI: https://doi.org/10.48550/arXiv.1810.04805. arXiv: 1810.04805 [cs.CL].

[31] A. Dhall, R. Goecke, S. Ghosh, J. Joshi, J. Hoey und T. Gedeon, „From individual to group-level emotion recognition: EmotiW 5.0," in *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, Ser. ICMI '17, ACM, Nov. 2017. DOI: 10.1145/3136755.3143004.

[32] A. Dhall, O. Ramana Murthy, R. Goecke, J. Joshi und T. Gedeon, „Video and Image based Emotion Recognition Challenges in the Wild: EmotiW 2015," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, Ser. ICMI '15, ACM, Nov. 2015. DOI: 10.1145/2818346.2829994.

[33] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding und J. Sun, *RepVGG: Making VGG-style ConvNets Great Again*, 2021. DOI: 10.48550/ARXIV.2101.03697.

[34] J. Donahue, P. Krähenbühl und T. Darrell, „Adversarial Feature Learning," 31. Mai 2016. DOI: 10.48550/ARXIV.1605.09782. arXiv: 1605.09782 [cs.LG].

[35] C. Dong, C. C. Loy, K. He und X. Tang, „Learning a Deep Convolutional Network for Image Super-Resolution," in *Lecture Notes in Computer Science*. Springer International Publishing, 2014, S. 184–199, ISBN: 9783319105932. DOI: 10.1007/978-3-319-10593-2_13.

[36] C. Dong, C. C. Loy, K. He und X. Tang, „Image Super-Resolution Using Deep Convolutional Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Jg. 38, Nr. 2, S. 295–307, Feb. 2016, ISSN: 2160-9292. DOI: 10.1109/tpami.2015.2439281.

[37] C. Dong, C. C. Loy und X. Tang, „Accelerating the Super-Resolution Convolutional Neural Network," in *Lecture Notes in Computer Science*. Springer International Publishing, 2016, S. 391–407, ISBN: 9783319464756. DOI: 10.1007/978-3-319-46475-6_25.

[38] A. Dosovitskiy, L. Beyer, A. Kolesnikov *et al.*, *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, 2021. arXiv: 2010.11929 [cs.CV].

[39] S. Du, Y. Tao und A. M. Martinez, „Compound facial expressions of emotion," *Proceedings of the National Academy of Sciences*, Jg. 111, Nr. 15, März 2014, ISSN: 1091-6490. DOI: 10.1073/pnas.1322355111.

[40] P. Ekman und W. V. Friesen, „Constants across cultures in the face and emotion.," *Journal of Personality and Social Psychology*, Jg. 17, Nr. 2, S. 124–129, 1971. DOI: 10.1037/h0030377.

[41] P. Ekman und W. V. Friesen, *Facial Action Coding System: A Technique for the Measurement of Facial Movement*, Palo Alto, CA: Consulting Psychologists Press, 1978.

[42] P. Ekman und W. V. Friesen, „A new pan-cultural facial expression of emotion," *Motivation and Emotion*, Jg. 10, Nr. 2, S. 159–168, Juni 1986, ISSN: 1573-6644. DOI: 10.1007/bf00992253.

[43] P. Ekman, W. V. Friesen und J. C. Hager, *Facial action coding system* (A research Nexus eBook). Salt Lake City, Utah: A Human Face, 2002, 1 S., ISBN: 9780931835018.

[44] European Data Protection Supervisor, K. Vemou, A. Horvath und T. Zerdick, *EDPS TechDispatch: facial emotion recognition. Issue 1, 2021.* T. Zerdick, Hrsg. Publications Office, 2021. DOI: 10.2804/014217.

[45] Y. Fu. „Image classification via fine-tuning with EfficientNet." (30. Juni 2020), Adresse: https://keras.io/examples/vision/image_classification_efficientnet_fine_tuning/ (besucht am 31. 03. 2024).

[46] A. Fujimoto, T. Ogawa, K. Yamamoto, Y. Matsui, T. Yamasaki und K. Aizawa, „Manga109 dataset and creation of metadata," in *Proceedings of the 1st International Workshop on coMics ANalysis, Processing and Understanding*, Ser. MANPU '16, ACM, Dez. 2016. DOI: 10.1145/3011549.3011551.

[47] M.-I. Georgescu, R. T. Ionescu und M. Popescu, „Local Learning With Deep and Handcrafted Features for Facial Expression Recognition," *IEEE Access*, Jg. 7, S. 64 827–64 836, 2019, ISSN: 2169-3536. DOI: 10.1109/access.2019.2917266.

[48] A. Ghildyal und F. Liu, *Shift-tolerant Perceptual Similarity Metric*, 2022. DOI: 10.48550/ARXIV.2207.13686.

[49] E. Goeleven, R. De Raedt, L. Leyman und B. Verschuere, „The Karolinska Directed Emotional Faces: A validation study," *Cognition & Emotion*, Jg. 22, Nr. 6, S. 1094–1118, Sep. 2008, ISSN: 1464-0600. DOI: 10.1080/02699930701626582.

[50] R. C. Gonzalez und R. E. Woods, *Digital image processing*, 3. Aufl. Upper Saddle River, NJ: Pearson/Prentice Hall, 2008, 954 S., ISBN: 9780131687288.

[51] I. J. Goodfellow, D. Erhan, P. L. Carrier *et al.*, „Challenges in Representation Learning: A Report on Three Machine Learning Contests," in *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2013, S. 117–124, ISBN: 9783642420511. DOI: 10.1007/978-3-642-42051-1_16.

[52] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza *et al.*, *Generative Adversarial Networks*, 2014. DOI: 10.48550/ARXIV.1406.2661.

[53] G. Görz, U. Schmid und T. Braun, Hrsg., *Handbuch der Künstlichen Intelligenz*, 6. Auflage. Berlin: De Gruyter Oldenbourg, 2021, 1956 S., ISBN: 9783110659955.

[54] R. Gross, I. Matthews, J. Cohn, T. Kanade und S. Baker, „Multi-PIE," in *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*, IEEE, Sep. 2008. DOI: 10.1109/afgr.2008.4813399.

[55] K. Gu, H. Liu und C. Zhou, „Quality Assessment of Super-Resolution Images," in *Advances in Computer Vision and Pattern Recognition*. Springer Nature Singapore, 2022, S. 217–242, ISBN: 9789811933479. DOI: 10.1007/978-981-19-3347-9_8.

[56] B. Hasani, A. Mollahosseini und M. A. Mahoor, *AffectNet Database: Documentation*, März 2021.

[57] M. Hashemi, „Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. interpolation," *Journal of Big Data*, Jg. 6, Nr. 1, Nov. 2019, ISSN: 2196-1115. DOI: 10.1186/s40537-019-0263-7.

[58] K. He, X. Zhang, S. Ren und J. Sun, *Deep Residual Learning for Image Recognition*, 2015. DOI: 10.48550/ARXIV.1512.03385.

[59] H. Herold, *Grundlagen der Informatik* (it - Informatik), 3., aktualisierte Auflage, B. Lurz, J. Wohlrab und M. Hopf, Hrsg. Hallbergmoos: Pearson, 2017, ISBN: 9783863268039.

[60] A. G. Howard, M. Zhu, B. Chen *et al.*, *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*, 2017. DOI: 10.48550/ARXIV.1704.04861.

[61] J. Hu, L. Shen, S. Albanie, G. Sun und E. Wu, *Squeeze-and-Excitation Networks*, 2017. DOI: 10.48550/ARXIV.1709.01507.

[62] G. Huang, Z. Liu, L. Van Der Maaten und K. Q. Weinberger, „Densely Connected Convolutional Networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Juli 2017. DOI: 10.1109/cvpr.2017.243.

[63] J.-B. Huang, A. Singh und N. Ahuja, „Single image super-resolution from transformed self-exemplars," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Juni 2015. DOI: 10.1109/cvpr.2015.7299156.

[64] H. Iqbal. „PlotNeuralNet." (17. Jan. 2020), Adresse: https://github.com/HarisIqbal88/PlotNeuralNet (besucht am 27. 03. 2024).

[65] R. E. Jack, O. G. B. Garrod, H. Yu, R. Caldara und P. G. Schyns, „Facial expressions of emotion are not culturally universal," *Proceedings of the National Academy of Sciences*, Jg. 109, Nr. 19, S. 7241–7244, Apr. 2012, ISSN: 1091-6490. DOI: 10.1073/pnas.1200155109.

[66] B. Jähne, *Digitale Bildverarbeitung, und Bildgewinnung*, 7. Aufl. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, 1711 S., ISBN: 9783642049521.

[67] X. Jin, J. Liu und D. Yue, „The Research and Improvement of Facial Expression Recognition Algorithm Based on Convolutional Neural Network“ in *2023 26th ACIS International Winter Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD-Winter)*, IEEE, Juli 2023. DOI: 10.1109/snpd-winter57765.2023.10224044.

[68] G. Jinjin, C. Haoming, C. Haoyu, Y. Xiaoxing, J. S. Ren und D. Chao, „PIPAL: A Large-Scale Image Quality Assessment Dataset for Perceptual Image Restoration“ in *Lecture Notes in Computer Science*. Springer International Publishing, 2020, S. 633–651, ISBN: 9783030586218. DOI: 10.1007/978-3-030-58621-8_37.

[69] Jonaorle. „A Woman Wearing a Black Hijab.“ Pexels License. (März 2020), Adresse: https://www.pexels.com/photo/a-woman-wearing-a-black-hijab-4029925/ (besucht am 01. 02. 2024).

[70] T. Karras, S. Laine und T. Aila, „A Style-Based Generator Architecture for Generative Adversarial Networks,“ *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Jg. 43, Nr. 12, S. 4217–4228, Dez. 2021, ISSN: 1939-3539. DOI: 10.1109/tpami.2020.2970919.

[71] M. Kettunen, E. Härkönen und J. Lehtinen, „E-LPIPS: Robust Perceptual Image Similarity via Random Transformation Ensembles,“ 10. Juni 2019. DOI: 10.48550/ARXIV.1906.03973. arXiv: 1906.03973 [cs.CV].

[72] J. Kim, J. K. Lee und K. M. Lee, „Accurate Image Super-Resolution Using Very Deep Convolutional Networks,“ in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Juni 2016. DOI: 10.1109/cvpr.2016.182.

[73] P. W. Kim, „Image super-resolution model using an improved deep learning-based facial expression analysis,“ *Multimedia Systems*, Jg. 27, Nr. 4, S. 615–625, Okt. 2020, ISSN: 1432-1882. DOI: 10.1007/s00530-020-00705-1.

[74] D. P. Kingma und J. Ba, *Adam: A Method for Stochastic Optimization*, 2014. DOI: 10.48550/ARXIV.1412.6980.

[75] S. Koelstra, C. Muhl, M. Soleymani *et al.*, „DEAP: A Database for Emotion Analysis ;Using Physiological Signals,“ *IEEE Transactions on Affective Computing*, Jg. 3, Nr. 1, S. 18–31, Jan. 2012, ISSN: 1949-3045. DOI: 10.1109/t-affc.2011.15.

[76] I. Kotsia, S. Zafeiriou und S. Fotopoulos, „Affective Gaming: A Comprehensive Survey,“ in *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, IEEE, Juni 2013. DOI: 10.1109/cvprw.2013.100.

[77] A. Krizhevsky, I. Sutskever und G. E. Hinton, „ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. Burges, L. Bottou und K. Weinberger, Hrsg., Bd. 25, Curran Associates, Inc., 2012, ISBN: 9781627480031. Adresse: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[78] O. Langner, R. Dotsch, G. Bijlstra, D. H. J. Wigboldus, S. T. Hawk und A. van Knippenberg, „Presentation and validation of the Radboud Faces Database," *Cognition & Emotion*, Jg. 24, Nr. 8, S. 1377–1388, Dez. 2010, ISSN: 1464-0600. DOI: 10.1080/02699930903485076.

[79] C. Ledig, L. Theis, F. Huszar *et al.*, „Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Juli 2017. DOI: 10.1109/cvpr.2017.19.

[80] D. C. Lepcha, B. Goyal, A. Dogra und V. Goyal, „Image super-resolution: A comprehensive review, recent trends, challenges and applications," *Information Fusion*, Jg. 91, S. 230–260, März 2023, ISSN: 1566-2535. DOI: 10.1016/j.inffus.2022.10.007.

[81] J. Li, Z. Pei und T. Zeng, *From Beginner to Master: A Survey for Deep Learning-based Single-Image Super-Resolution*, 2021. DOI: 10.48550/ARXIV.2109.14335.

[82] K. Li, S. Yang, R. Dong, X. Wang und J. Huang, „Survey of single image super-resolution reconstruction," *IET Image Processing*, Jg. 14, Nr. 11, S. 2273–2290, Juli 2020, ISSN: 1751-9667. DOI: 10.1049/iet-ipr.2019.1438.

[83] S. Li und W. Deng, „Reliable Crowdsourcing and Deep Locality-Preserving Learning for Unconstrained Facial Expression Recognition," *IEEE Transactions on Image Processing*, Jg. 28, Nr. 1, S. 356–370, Jan. 2019, ISSN: 1941-0042. DOI: 10.1109/tip.2018.2868382.

[84] S. Li und W. Deng, „Deep Facial Expression Recognition: A Survey," *IEEE Transactions on Affective Computing*, Jg. 13, Nr. 3, S. 1195–1215, Juli 2022, ISSN: 2371-9850. DOI: 10.1109/taffc.2020.2981446.

[85] S. Li, W. Deng und J. Du, „Reliable Crowdsourcing and Deep Locality-Preserving Learning for Expression Recognition in the Wild," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Juli 2017. DOI: 10.1109/cvpr.2017.277.

[86] S. Z. Li und A. Jain, Hrsg., *Encyclopedia of Biometrics* (SpringerLink). Boston, MA: Springer US, 2009, ISBN: 9780387730035.

[87] Y. Li, K. Zhang, R. Timofte *et al.*, „NTIRE 2022 Challenge on Efficient Super-Resolution: Methods and Results," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, Juni 2022. DOI: 10.1109/cvprw56347.2022.00118.

[88] Y. Li, Y. Zhang, R. Timofte *et al.*, „NTIRE 2023 Challenge on Efficient Super-Resolution: Methods and Results," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, Juni 2023. DOI: 10.1109/cvprw59228.2023.00189.

[89] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool und R. Timofte, *SwinIR: Image Restoration Using Swin Transformer*, 2021. DOI: 10.48550/ARXIV.2108.10257.

[90] B. Lim, S. Son, H. Kim, S. Nah und K. M. Lee, „Enhanced Deep Residual Networks for Single Image Super-Resolution," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, Juli 2017. DOI: 10.1109/cvprw.2017.151.

[91] B. Liu und D. Ait-Boudaoud, „Effective image super resolution via hierarchical convolutional neural network," *Neurocomputing*, Jg. 374, S. 109–116, Jan. 2020, ISSN: 0925-2312. DOI: 10.1016/j.neucom.2019.09.035.

[92] C. Liu, H.-Y. Shum und W. T. Freeman, „Face Hallucination: Theory and Practice," *International Journal of Computer Vision*, Jg. 75, Nr. 1, S. 115–134, Feb. 2007, ISSN: 1573-1405. DOI: 10.1007/s11263-006-0029-5.

[93] W. Liu, D. Anguelov, D. Erhan *et al.*, „SSD: Single Shot MultiBox Detector," 2015. DOI: 10.48550/ARXIV.1512.02325.

[94] Y. Liu, H. Dong, B. Liang *et al.*, „Unfolding Once is Enough: A Deployment-Friendly Transformer Unit for Super-Resolution," in *Proceedings of the 31st ACM International Conference on Multimedia*, Ser. MM '23, ACM, Okt. 2023. DOI: 10.1145/3581783.3612128.

[95] Z. Liu, Y. Lin, Y. Cao *et al.*, *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*, 2021. DOI: 10.48550/ARXIV.2103.14030.

[96] Z. Liu, P. Luo, X. Wang und X. Tang, „Deep Learning Face Attributes in the Wild," in *2015 IEEE International Conference on Computer Vision (ICCV)*, IEEE, Dez. 2015. DOI: 10.1109/iccv.2015.425.

[97] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar und I. Matthews, „The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, IEEE, Juni 2010. DOI: 10.1109/cvprw.2010.5543262.

[98] C. Lugaresi, J. Tang, H. Nash *et al.*, „MediaPipe: A Framework for Perceiving and Processing Reality," in *Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR) 2019*, 2019. Adresse: https://mixedreality.cs.cornell.edu/s/NewTitle_May1_MediaPipe_CVPR_CV4ARVR_Workshop_2019.pdf.

[99] M. Lyons, S. Akamatsu, M. Kamachi und J. Gyoba, „Coding facial expressions with Gabor wavelets," in *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, Ser. AFGR-98, IEEE Comput. Soc, 1998. DOI: 10.1109/afgr.1998.670949.

[100] C. Ma, C.-Y. Yang, X. Yang und M.-H. Yang, „Learning a no-reference quality metric for single-image super-resolution," *Computer Vision and Image Understanding*, Jg. 158, S. 1–16, Mai 2017, ISSN: 1077-3142. DOI: 10.1016/j.cviu.2016.12.009.

[101] K. Ma, Z. Duanmu, Q. Wu *et al.*, „Waterloo Exploration Database: New Challenges for Image Quality Assessment Models," *IEEE Transactions on Image Processing*, Jg. 26, Nr. 2, S. 1004–1016, Feb. 2017, ISSN: 1941-0042. DOI: 10.1109/tip.2016.2631888.

[102] D. Martin, C. Fowlkes, D. Tal und J. Malik, „A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, Ser. ICCV-01, IEEE Comput. Soc, 2001. DOI: 10.1109/iccv.2001.937655.

[103] Martín Abadi, Ashish Agarwal, Paul Barham *et al.*, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, Software available from tensorflow.org, 2015. Adresse: https://www.tensorflow.org/.

[104] B. Martinez und M. F. Valstar, „Advances, Challenges, and Opportunities in Automatic Facial Expression Recognition," in *Advances in Face Detection and Facial Image Analysis*. Springer International Publishing, 2016, S. 63–100, ISBN: 9783319259581. DOI: 10.1007/978-3-319-25958-1_4.

[105] D. Matsumoto, „More evidence for the universality of a contempt expression," *Motivation and Emotion*, Jg. 16, Nr. 4, S. 363–368, Dez. 1992, ISSN: 1573-6644. DOI: 10.1007/bf00992972.

[106] S. McCloud, *Making comics, Storytelling secrets of comics, manga and graphic novels*. New York: William Morrow, an imprint of HarperCollinsPublishers, 2006, 264 S., ISBN: 0060780940.

[107] T. M. Mitchell, *Machine learning* (McGraw-Hill international editions), Nachdruck. New York [u.a.]: McGraw-Hill, 2013, 414 S., ISBN: 0071154671.

[108] A. Mollahosseini, B. Hasani und M. H. Mahoor, „AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild," *IEEE Transactions on Affective Computing*, Jg. 10, Nr. 1, S. 18–31, Jan. 2019, ISSN: 2371-9850. DOI: 10.1109/taffc.2017.2740923.

[109] A. Mollahosseini, B. Hassani, M. J. Salvador, H. Abdollahi, D. Chan und M. H. Mahoor, „Facial Expression Recognition from World Wild Web," in *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, Juni 2016. DOI: 10.1109/cvprw.2016.188.

[110] F. Nan, W. Jing, F. Tian *et al.*, „Feature super-resolution based Facial Expression Recognition for multi-scale low-resolution images," *Knowledge-Based Systems*, Jg. 236, S. 107 678, Jan. 2022, ISSN: 0950-7051. DOI: 10.1016/j.knosys.2021.107678.

[111] M. Pantic, M. Valstar, R. Rademaker und L. Maat, „Web-Based Database for Facial Expression Analysis," in *2005 IEEE International Conference on Multimedia and Expo*, IEEE, 2005. DOI: 10.1109/icme.2005.1521424.

[112] Papers with Code (Meta AI Research). „Facial Expression Recognition (FER) on AffectNet." (2024), Adresse: https://paperswithcode.com/sota/facial-expression-recognition-on-affectnet (besucht am 25. 03. 2024).

[113] N. Ponomarenko, L. Jin, O. Ieremeiev *et al.*, „Image database TID2013: Peculiarities, results and perspectives," *Signal Processing: Image Communication*, Jg. 30, S. 57–77, Jan. 2015, ISSN: 0923-5965. DOI: 10.1016/j.image.2014.10.009.

[114] J. Posner, J. A. Russel und B. S. Peterson, „The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology," *Development and Psychopathology*, Jg. 17, Nr. 03, Sep. 2005, ISSN: 1469-2198. DOI: 10.1017/s0954579405050340.

[115] S. J. D. Prince, *Understanding deep learning*. Cambridge, Massachusetts: The MIT Press, 2023, 527 S., Literaturverzeichnis: Seite 462-511, ISBN: 9780262048644.

[116] A. Radford, K. Narasimhan, T. Salimans und I. Sutskever, „Improving language understanding by generative pre-training," 2018. Adresse: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

[117] S. Raschka, *Machine Learning with PyTorch and Scikit-Learn, Develop machine learning and deep learning models with Python*, 1. Auflage, Y. H. Liu, V. Mirjalili und D. Dzhulgakov, Hrsg. Birmingham: Packt Publishing Limited, 2022, 1770 S., ISBN: 9781801816380.

[118] J. A. Russell, „A circumplex model of affect.," *Journal of Personality and Social Psychology*, Jg. 39, Nr. 6, S. 1161–1178, Dez. 1980, ISSN: 0022-3514. DOI: 10.1037/h0077714.

[119] C. F. Sabottke und B. M. Spieler, „The Effect of Image Resolution on Deep Learning in Radiography," *Radiology: Artificial Intelligence*, Jg. 2, Nr. 1, e190015, Jan. 2020, ISSN: 2638-6100. DOI: 10.1148/ryai.2019190015.

[120] M. Saleh, A. Yong, N. Marbukhari *et al.*, „Facial Expression Recognition: A New Dataset and a Review of the Literature," *Turkish Online Journal of Qualitative Inquiry*, Jg. 12, S. 9804–9811, Juli 2021.

[121] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov und L.-C. Chen, „MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018. DOI: 10.48550/ARXIV.1801.04381.

[122] A. V. Savchenko, L. V. Savchenko und I. Makarov, „Classifying Emotions and Engagement in Online Learning Based on a Single Facial Expression Recognition Neural Network," *IEEE Transactions on Affective Computing*, Jg. 13, Nr. 4, S. 2132–2143, Okt. 2022, ISSN: 2371-9850. DOI: 10.1109/taffc.2022.3188390.

[123] J. Shao und Q. Cheng, „E-FCNN for tiny facial expression recognition," *Applied Intelligence*, Jg. 51, Nr. 1, S. 549–559, Aug. 2020, ISSN: 1573-7497. DOI: 10.1007/s10489-020-01855-5.

[124] H. Sheikh, M. Sabir und A. Bovik, „A Statistical Evaluation of Recent Full Reference Image Quality Assessment Algorithms," *IEEE Transactions on Image Processing*, Jg. 15, Nr. 11, S. 3440–3451, Nov. 2006, ISSN: 1057-7149. DOI: 10.1109/tip.2006.881959.

[125] S. Son und K. M. Lee, „Image Super-Resolution," in *Computer Vision: A Reference Guide*, K. Ikeuchi, Hrsg. Cham: Springer International Publishing, 2021, S. 646–650, ISBN: 978-3-030-63416-2. DOI: 10.1007/978-3-030-63416-2_838.

[126] J. Susskind, A. Anderson und G. E. Hinton, „The Toronto face dataset," 2010.

[127] M. Taini, G. Zhao, S. Z. Li und M. Pietikainen, „Facial expression recognition from near-infrared video sequences," in *2008 19th International Conference on Pattern Recognition*, IEEE, Dez. 2008. DOI: 10.1109/icpr.2008.4761697.

[128] M. Tan, B. Chen, R. Pang *et al.*, „MnasNet: Platform-Aware Neural Architecture Search for Mobile," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Juni 2019. DOI: 10.1109/cvpr.2019.00293.

[129] M. Tan und Q. V. Le, „EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," 2019. DOI: 10.48550/ARXIV.1905.11946.

[130] W. Tan, B. Yan und B. Bare, „Feature Super-Resolution: Make Machine See More Clearly," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Juni 2018. DOI: 10.1109/cvpr.2018.00420.

[131] V. Thambawita, I. Strümke, S. A. Hicks, P. Halvorsen, S. Parasa und M. A. Riegler, „Impact of Image Resolution on Deep Learning Performance in Endoscopy Image Classification: An Experimental Study Using a Large Dataset of Endoscopic Images," *Diagnostics*, Jg. 11, Nr. 12, S. 2183, Nov. 2021, ISSN: 2075-4418. DOI: 10.3390/diagnostics11122183.

[132] R. Timofte, E. Agustsson, L. V. Gool *et al.*, „NTIRE 2017 Challenge on Single Image Super-Resolution: Methods and Results," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, Juli 2017. DOI: 10.1109/cvprw.2017.149.

[133] R. Timofte, S. Gu, J. Wu *et al.*, „NTIRE 2018 Challenge on Single Image Super-Resolution: Methods and Results," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, Juni 2018. DOI: 10.1109/cvprw.2018.00130.

[134] R. Timofte, R. Rothe und L. Van Gool, „Seven Ways to Improve Example-Based Single Image Super Resolution," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Juni 2016. DOI: 10.1109/cvpr.2016.206.

[135] M. F. Valstar und M. Pantic, „Induced Disgust, Happiness and Surprise: an Addition to the MMI Facial Expression Database," 2010.

[136] A. Vaswani, N. Shazeer, N. Parmar *et al.*, „Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Ser. NIPS'17, Long Beach, California, USA: Curran Associates Inc., 2017, S. 6000–6010, ISBN: 9781510860964.

[137] T.-H. Vo, G.-S. Lee, H.-J. Yang und S.-H. Kim, „Pyramid With Super Resolution for In-the-Wild Facial Expression Recognition," *IEEE Access*, Jg. 8, S. 131 988–132 001, 2020, ISSN: 2169-3536. DOI: 10.1109/access.2020.3010018.

[138] H. Wang, X. Chen, B. Ni, Y. Liu und J. Liu, „Omni Aggregation Networks for Lightweight Image Super-Resolution," 20. Apr. 2023. DOI: 10.48550/ARXIV.2304.10244. arXiv: 2304.10244 [cs.CV].

[139] K. Wang, X. Peng, J. Yang, D. Meng und Y. Qiao, „Region Attention Networks for Pose and Occlusion Robust Facial Expression Recognition," *IEEE Transactions on Image Processing*, Jg. 29, S. 4057–4069, 2020, ISSN: 1941-0042. DOI: 10.1109/tip.2019.2956143.

[140] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri und R. M. Summers, „ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases," *IEEE CVPR 2017, pp. 2097-2106 (2017)*, 5. Mai 2017. DOI: 10.1109/cvpr.2017.369. arXiv: 1705.02315 [cs.CV].

[141] X. Wang, Y. Li, H. Zhang und Y. Shan, „Towards Real-World Blind Face Restoration with Generative Facial Prior," 11. Jan. 2021. DOI: 10.48550/ARXIV.2101.04061. arXiv: 2101.04061 [cs.CV].

[142] X. Wang, L. Xie, C. Dong und Y. Shan, *Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data*, 2021. DOI: 10.48550/ARXIV.2107.10833.

[143] X. Wang, K. Yu, C. Dong und C. Change Loy, „Recovering Realistic Texture in Image Super-Resolution by Deep Spatial Feature Transform," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Juni 2018. DOI: 10.1109/cvpr.2018.00070.

[144] X. Wang, K. Yu, S. Wu *et al.*, *ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks*, 2018. DOI: 10.48550/ARXIV.1809.00219.

[145] X. (Wang. „GFPGAN GitHub Readme." Commit-Hash (verkürzt): bc5a5de. (2022), Adresse: https://github.com/TencentARC/GFPGAN/blob/master/README.md (besucht am 22. 03. 2024).

[146] Y. Wang, L. Wang, J. Yang, W. An und Y. Guo, „Flickr1024: A Large-Scale Dataset for Stereo Image Super-Resolution," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, IEEE, Okt. 2019. DOI: 10.1109/iccvw.2019.00478.

[147] Z. Wang, A. Bovik, H. Sheikh und E. Simoncelli, „Image Quality Assessment: From Error Visibility to Structural Similarity," *IEEE Transactions on Image Processing*, Jg. 13, Nr. 4, S. 600–612, Apr. 2004, ISSN: 1057-7149. DOI: 10.1109/tip.2003.819861.

[148] Z. Wang, J. Chen und S. C. H. Hoi, „Deep Learning for Image Super-Resolution: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Jg. 43, Nr. 10, S. 3365–3387, Okt. 2021, ISSN: 1939-3539. DOI: 10.1109/tpami.2020.2982166.

[149] E. Weitz, *Konkrete Mathematik (nicht nur) für Informatiker, Mit vielen Grafiken und Algorithmen in Python*. Springer Berlin / Heidelberg, 2021, ISBN: 9783662626177.

[150] J. Yang, J. Wright, T. S. Huang und Y. Ma, „Image Super-Resolution Via Sparse Representation," *IEEE Transactions on Image Processing*, Jg. 19, Nr. 11, S. 2861–2873, Nov. 2010, ISSN: 1057-7149. DOI: 10.1109/tip.2010.2050625.

[151] L. Ye und C. Zou, „A Survey of Image Super-Resolution Reconstruction Based on Deep Learning," in *Proceeding of 2022 International Conference on Wireless Communications, Networking and Applications (WCNA 2022)*. Springer Nature Singapore, 2023, S. 584–594, ISBN: 9789819939510. DOI: 10.1007/978-981-99-3951-0_64.

[152] C.-T. Yen und K.-H. Li, „Discussions of Different Deep Transfer Learning Models for Emotion Recognitions," *IEEE Access*, Jg. 10, S. 102 860–102 875, 2022, ISSN: 2169-3536. DOI: 10.1109/access.2022.3209813.

[153] L. Yin, X. Chen, Y. Sun, T. Worm und M. Reale, „A high-resolution 3D dynamic facial expression database," in *2008 8th IEEE International Conference on Automatic Face & Gesture Recognition*, IEEE, Sep. 2008. DOI: 10.1109/afgr.2008.4813324.

[154] L. Yin, X. Wei, Y. Sun, J. Wang und M. Rosato, „A 3D Facial Expression Database For Facial Behavior Research," in *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, IEEE, Apr. 2006. DOI: 10.1109/fgr.2006.6.

[155] G. Yu und G. Sapiro, „Image Enhancement and Restoration: Traditional Approaches," in *Computer Vision: A Reference Guide*, K. Ikeuchi, Hrsg. Cham: Springer International Publishing, 2021, S. 615–619, ISBN: 978-3-030-63416-2. DOI: 10.1007/978-3-030-63416-2_233.

[156] J. U. Yun, B. Jo und I. K. Park, „Joint Face Super-Resolution and Deblurring Using Generative Adversarial Network," *IEEE Access*, Jg. 8, S. 159 661–159 671, 2020, ISSN: 2169-3536. DOI: 10.1109/access.2020.3020729.

[157] J. Zeng, S. Shan und X. Chen, „Facial Expression Recognition with Inconsistently Annotated Datasets," in *Lecture Notes in Computer Science*. Springer International Publishing, 2018, S. 227–243, ISBN: 9783030012618. DOI: 10.1007/978-3-030-01261-8_14.

[158] R. Zeyde, M. Elad und M. Protter, „On Single Image Scale-Up Using Sparse-Representations," in *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012, S. 711–730, ISBN: 9783642274138. DOI: 10.1007/978-3-642-27413-8_47.

[159] K. Zhang, S. Gu, R. Timofte *et al.*, *NTIRE 2020 Challenge on Perceptual Extreme Super-Resolution: Methods and Results*, 2020. DOI: 10.48550/ARXIV.2005.01056.

[160] K. Zhang, J. Liang, L. Van Gool und R. Timofte, *Designing a Practical Degradation Model for Deep Blind Image Super-Resolution*, 2021. DOI: 10.48550/ARXIV.2103.14006.

[161] L. Zhang, L. Zhang, X. Mou und D. Zhang, „FSIM: A Feature Similarity Index for Image Quality Assessment," *IEEE Transactions on Image Processing*, Jg. 20, Nr. 8, S. 2378–2386, Aug. 2011, ISSN: 1941-0042. DOI: 10.1109/tip.2011.2109730.

[162] R. Zhang. „PerceptualSimilarity, README.md." (2019), Adresse: https://github.com/richzhang/PerceptualSimilarity/blob/master/README.md (besucht am 06. 03. 2024).

[163] R. Zhang. „Google for Developers: MediaPipe Face Detection Guide." (2024), Adresse: https://developers.google.com/mediapipe/solutions/vision/face_detector (besucht am 07. 03. 2024).

[164] R. Zhang, P. Isola, A. A. Efros, E. Shechtman und O. Wang, *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric*, 2018. DOI: 10.48550/ARXIV.1801.03924.

[165] S. Zhang, Y. Zhang, Y. Zhang, Y. Wang und Z. Song, „A Dual-Direction Attention Mixed Feature Network for Facial Expression Recognition," *Electronics*, Jg. 12, Nr. 17, S. 3595, Aug. 2023, ISSN: 2079-9292. DOI: 10.3390/electronics12173595.

[166] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong und Y. Fu, *Image Super-Resolution Using Very Deep Residual Channel Attention Networks*, 2018. DOI: 10.48550/ARXIV.1807.02758.

[167] Y. Zhang, K. Zhang, Z. Chen *et al.*, „NTIRE 2023 Challenge on Image Super-Resolution (×4): Methods and Results," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, Juni 2023. DOI: 10.1109/cvprw59228.2023.00185.

[168] Z. Zhang, P. Luo, C. C. Loy und X. Tang, „From Facial Expression Recognition to Interpersonal Relation Prediction," *International Journal of Computer Vision*, Jg. 126, Nr. 5, S. 550–569, Nov. 2017, ISSN: 1573-1405. DOI: 10.1007/s11263-017-1055-1.

[169] Z. Zhang, Z. Wang, Z. Lin und H. Qi, „Image Super-Resolution by Neural Texture Transfer," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Juni 2019. DOI: 10.1109/cvpr.2019.00817.

[170] S. Zhou, K. C. K. Chan, C. Li und C. C. Loy, „Towards Robust Blind Face Restoration with Codebook Lookup Transformer," 22. Juni 2022. DOI: 10.48550/ARXIV.2206.11253. arXiv: 2206.11253 [cs.CV].

[171] Y. Zhou, Z. Li, C.-L. Guo, S. Bai, M.-M. Cheng und Q. Hou, *SRFormer: Permuted Self-Attention for Single Image Super-Resolution*, 2023. DOI: 10.48550/ARXIV.2303.09735.

Anhang

A1 Ergänzende Quellen

A1.1 Dateibezeichnung RaFD-Beispiele

Tabelle A1.1: Genaue Dateibezeichnung der gezeigten RaFD-Bilder

Nr.	Ausdruck	Dateiname
A1.1.1	Neutral	Rafd000_03_Caucasian_male_neutral_frontal.jpg
A1.1.2	Freude	Rafd045_55_Moroccan_male_happy_right.jpg
A1.1.3	Traurigkeit	Rafd090_04_Caucasian_female_sad_left.jpg
A1.1.4	Überrascht	Rafd090_05_Caucasian_male_surprised_frontal.jpg
A1.1.5	Angst	Rafd090_09_Caucasian_male_fearful_frontal.jpg
A1.1.6	Ekel	Rafd090_37_Caucasian_female_disgusted_right.jpg
A1.1.7	Wut	Rafd135_01_Caucasian_female_angry_frontal.jpg
A1.1.8	Verachtung	Rafd180_18_Caucasian_female_contemptuous_frontal.jpg

A1.2 Super-Resolution-Netzwerke

Tabelle A1.2: Verwendete Super-Resolution-Netzwerke

Nr.	Beschreibung	Quelle[1]	Version[2]
A1.2.1	CodeFormer	https://github.com/sczhou/CodeFormer/tree/master	8392d03
A1.2.2	DAT	https://github.com/zhengchen1999/DAT/tree/main	57637a3
A1.2.3	GFPGAN	https://github.com/TencentARC/GFPGAN/tree/master	2eac203
A1.2.4	HAT	https://github.com/XPixelGroup/HAT/tree/main	1b22ba0
A1.2.5	RealESRGAN	https://github.com/xinntao/Real-ESRGAN/tree/master	5ca1078
A1.2.6	SRFormer	https://github.com/HVision-NKU/SRFormer/tree/main	0562f7b
A1.2.7	SwinIR	https://github.com/JingyunLiang/SwinIR/tree/main	6545850

[1] Alle Quellen besucht am 07.03.2024
[2] Verkürzter Git-Commit-Hash

A1.3 Super-Resolution-Modelle

Tabelle A1.3: Verwendete Super-Resolution-Modelle

Nr.	verwendet mit	Modell Quelle[1]
A1.3.1	CodeFormer	codeformer.pth https://github.com/sczhou/CodeFormer/releases/download/v0.1.0/codeformer.pth
A1.3.2	CodeFormer	RealESRGAN_x2plus.pth https://github.com/sczhou/CodeFormer/releases/download/v0.1.0/RealESRGAN_x2plus.pth
A1.3.3	DAT	4xFaceUpSharpLDAT.pth[2] https://openmodeldb.info/models/4x-FaceUpSharpLDAT
A1.3.4	GFPGAN	GFPGANv1.3.pth https://github.com/TencentARC/GFPGAN/tree/master
A1.3.5	GFPGAN	RealESRGAN_x2plus.pth https://github.com/xinntao/Real-ESRGAN/releases/download/v0.2.1/RealESRGAN_x2plus.pth
A1.3.6	HAT	Real_HAT_GAN_sharper.pth https://drive.google.com/file/d/1EioFq5-mKmv1uqta_Byd9cgXp9SU3zjj/
A1.3.7	RealESRGAN	RealESRGAN_x4plus.pth https://github.com/xinntao/Real-ESRGAN/releases/download/v0.1.0/RealESRGAN_x4plus.pth
A1.3.8	SRFormer	SRFormer_SRx4_DF2K.pth https://drive.google.com/file/d/13_fpD4aDE1wbEYX8yGWA3mVLZOCRWkWv/
A1.3.9	SwinIR	003_realSR_BSRGAN_DFO_s64w8_SwinIR-M_x2_GAN.pth https://github.com/JingyunLiang/SwinIR/releases/download/v0.0/003_realSR_BSRGAN_DFO_s64w8_SwinIR-M_x2_GAN.pth
A1.3.10	SwinIR	003_realSR_BSRGAN_DFO_s64w8_SwinIR-M_x4_GAN.pth https://github.com/JingyunLiang/SwinIR/releases/download/v0.0/003_realSR_BSRGAN_DFO_s64w8_SwinIR-M_x4_GAN.pth
A1.3.11	SwinIR	003_realSR_BSRGAN_DFOWMFC_s64w8_SwinIR-L_x4_GAN.pth https://github.com/JingyunLiang/SwinIR/releases/download/v0.0/003_realSR_BSRGAN_DFOWMFC_s64w8_SwinIR-L_x4_GAN.pth
A1.3.12	SwinIR	001_classicalSR_DIV2K_s48w8_SwinIR-M_x4.pth https://github.com/JingyunLiang/SwinIR/releases/download/v0.0/001_classicalSR_DIV2K_s48w8_SwinIR-M_x4.pth

[1] Alle Quellen besucht am 07.03.2024
[2] Modell wurde in neuem Python-Dictionary mit dem Key `'params'` neu gespeichert, um mit dem Netzwerk zu funktionieren

Eigenständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit mit dem Titel

Super-Resolution in der Emotionserkennung:
Effekte auflösungsoptimierter Bilder auf die Klassifikation von Gesichtsausdrücken durch Deep Neural Networks

selbstständig und nur mit den angegebenen Hilfsmitteln verfasst habe. Alle Passagen, die ich wörtlich aus der Literatur oder aus anderen Quellen wie z. B. Internetseiten übernommen habe, habe ich deutlich als Zitat mit Angabe der Quelle kenntlich gemacht.

Hamburg, 7. Mai 2024