



# UNIVERSITÄT LEIPZIG

Institut für Informatik  
Fakultät für Mathematik und Informatik  
Abteilung Datenbanken

## Kolorieren Historischer Fotos der Stadt Leipzig mit Deep-Learning-Techniken

Bachelorarbeit

vorgelegt von:  
Daniel Grohmann

Matrikelnummer:  
3710183

Betreuer:  
Prof. Dr. Erhard Rahm  
Lucas Lange

© 2023

Dieses Werk einschließlich seiner Teile ist **urheberrechtlich geschützt**. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Autors unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen sowie die Einspeicherung und Verarbeitung in elektronischen Systemen.

---

## Abstract

Im Bereich der digitalen Kolorierung schwarz-weißer Bilder mithilfe von Deep-Learning-Strukturen existieren unzählige Projekte, die das Thema bereits im Detail bearbeitet haben. Mit der immer wachsenden Reichweite und sich stetig entwickelnden Technologie der künstlichen Intelligenz ist es schwer, einen Überblick über bereits Bestehendes zu erlangen. Dieser Überblick soll dem Leser mit der vorliegenden Arbeit geboten werden, indem sich mit dem Kolorieren monochromer Fotos der Stadt Leipzig mithilfe von fünf ausgewählten Verfahren befasst wird, die verschiedene Deep-Learning-Modelle nutzen. Der vorliegende Datensatz von 17 Bildern wurde aus dem Buch "Leipzig - Gestern und Heute" von Dagmar Agsten-Aures selbst digitalisiert und wird gemeinsam mit den kolorierten Ergebnissen online über das angehängte GitHub-Verzeichnis und im digitalen Anhang zur Verfügung gestellt. Dabei werden die Projekte DeOldify (artistic) und Colorful Image Colorization eigens implementiert und - zusammen mit den gefärbten Ergebnissen - mit drei weiteren Onlineanbietern verglichen. Hierbei wird die einzigartige NoGAN-Architektur von DeOldify detailliert vorgestellt und dessen Bedeutung auf nachfolgende Projekte der Bildbe- und verarbeitung hervorgehoben.

Die Ergebnisse zeigen, dass ein plausibles Kolorieren mit DeOldify bei der richtigen Wahl des Renderfaktors möglich, wenn auch bei der Farbsättigung Luft nach oben ist. Auch wenn es sich bei den Schwarz-Weiß-Bildern meist um historische Straßen und Gebäude Leipzigs handelt, sind die Ergebnisse etwas farbenarm. Für Nutzer, die leuchtende Koloriererergebnisse bevorzugen, ist das beste Online-Tool MyHeritage In Color™ zu empfehlen, dessen Autoren jedoch nach der 14-tägigen Testversion beachtliche Geldsummen für das weitere Nutzen des Modells verlangen.

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>II</b>
<b>Tabellenverzeichnis</b>	<b>III</b>
<b>1. Einleitung</b>	<b>1</b>
1.1. Motivation . . . . .	1
1.2. Zielsetzung . . . . .	2
1.3. Aufbau der Arbeit . . . . .	2
<b>2. Grundlagen</b>	<b>3</b>
2.1. Historie . . . . .	3
2.1.1. Anfänge der Fotografie . . . . .	3
2.1.2. Handkolorierung . . . . .	4
2.1.3. Digitales Kolorieren . . . . .	6
2.2. Maschinelles Lernen . . . . .	6
2.2.1. Supervised Learning . . . . .	7
2.2.2. Unsupervised Learning . . . . .	7
2.2.3. Reinforcement Learning . . . . .	8
2.3. Deep Learning . . . . .	8
2.3.1. Künstliche Neuronale Netze . . . . .	9
2.3.2. Convolutional Neuronale Netze . . . . .	10
2.3.3. Generative Adversarial Networks . . . . .	11
<b>3. Projekte</b>	<b>13</b>
3.1. Verwandte Arbeiten . . . . .	13
3.1.1. Zhang et al.: Colorful Image Colorization (2016) . . . . .	13
3.1.2. Hotpot.ai und VanceAI . . . . .	14
3.2. DeOldify . . . . .	14
3.2.1. Technische Details . . . . .	15
3.3. Implementation . . . . .	17
3.3.1. DeOldify und MyHeritage . . . . .	18
3.3.2. Colorful Image Colorization (2016) . . . . .	19
3.3.3. Hotpot.ai und VanceAI . . . . .	19
<b>4. Evaluation</b>	<b>20</b>
4.1. Technische Auswertung . . . . .	20
4.2. Visuelle Auswertung . . . . .	22
<b>5. Fazit und Ausblick</b>	<b>27</b>
<b>Literatur</b>	<b>29</b>
<b>Erklärung</b>	<b>33</b>

## Abbildungsverzeichnis

1.1. Altes Rathaus (1986), Mendebrunnen (1988) und Fregehaus (1988) koloriert . . . . .	1
2.1. Blick aus dem Arbeitszimmer in Le Gras von Nicéphore Niépce, 1826 . . . . .	3
2.2. Daguerrotypien, entstanden um 1850 in den USA und England . . . . .	5
2.3. Deep Learning im Zusammenhang mit maschinellem Lernen und künstlicher Intelligenz	8
2.4. McCulloch-Pitts-Neuron . . . . .	9
2.5. Allgemeiner Aufbau eines vorwärtsgerichteten KNN-Modells . . . . .	10
2.6. Generative Adversarial Network [47] . . . . .	12
3.1. Zhang et al.'s Netzwerkstruktur aus CIC [48] . . . . .	13
3.2. U-Net Netzwerkstruktur . . . . .	15
4.1. Quantitativer Vergleich von DeOldify und Zhang's CIC (2016) auf ImageNet, CO- COStuff und Places205 aus "Instance-aware image colorization" [67] . . . . .	21
4.2. Barfußgässchen (1983) (1) aus "Leipzig - Gestern und Heute" [63] koloriert mit ver- schiedenen Projekten und (6) als geschossenes Foto . . . . .	23
4.3. Dimitroffmuseum (1988) koloriert mit DeOldify (links, A) und MyHeritage (rechts, B)	24
4.4. Renderfaktoren von links nach rechts: 10, 19, 28 und 37 von DeOldify, kolorierte Schillerstraße (1984) . . . . .	25

## **Tabellenverzeichnis**

4.1. Grundlegende Eigenschaften der Verfahren . . . . . 21

# 1. Einleitung

## 1.1. Motivation

Das Kolorieren alter Schwarz-Weiß-Fotos ist für viele, die sich damit beschäftigen, ein spaßiges Gimmick, etwas zum Probieren oder wird zur Befriedigung der eigenen Neugier genutzt. Man fragt sich, wie das Elternhaus, welches über Jahrzehnte von Generation zu Generation weitervererbt wurde, vor vielen Jahren ausgesehen haben kann. Wenn man als Student in eine neue Stadt zieht, wundert man sich, welche Geschichte in den bekannten Gebäuden und Straßen der Stadt liegen könnte. Als Zugezogener (und auch als Einheimischer) in Leipzig kommt so eventuell beim Anblick des prächtigen Neuen Rathauses die Frage auf, ob es schon immer so ausgesehen hat, oder welche Gebäudestrukturen den Markt mit dem Alten Rathaus vor 40 Jahren umgeben haben. So beginnt die Suche nach historischen Fotos der Stadt und man erkennt schließlich Unterschiede in der Anzahl der Gebäude der Schillerstraße oder Abweichungen im Erscheinungsbild des Mendebrunnens auf dem Augustusplatz. Doch oft ist das, was zum Vergleichen der alten Bilder mit den Bauwerken der Gegenwart fehlt, die Farbe. Bei der Forschung nach der zu anwendenden Koloriermethode ist man aber bei der Vielzahl der Angebote schnell überfordert. Dabei scheint das händische, anstrengende Färben von Bildern aus der Vergangenheit von den digitalen Verfahren längst eingeholt. Niemand, der neben des gewohnten Alltags Fotos kolorieren möchte, würde beim heutigen Stand der Technik (außer im Kunstunterricht in der Schule) auf die Idee kommen, einen Pinsel oder Stift in die Hand zu nehmen und alte Familienfotos per Hand zu färben. Somit führt die erweckte Neugier meist auf die Verwendung digitaler Methoden. Im Bereich des Deep Learning gibt es eine riesige Auswahl an Projekten und Experimenten, die allein im vergangenen Jahrzehnt entstanden sind.



Abbildung 1.1.: Altes Rathaus (1986), Mendebrunnen (1988) und Fregehaus (1988) koloriert

Während in der Vergangenheit das Kolorieren schwarz-weißer Fotos mit einer Menge von Nutzereingaben, wie bei SANGKLOY ET AL's "Controlling Deep Image Synthesis with Sketch and Color" [1], bei dem Farbhinweise vom Nutzer mit dem Originalbild verarbeitet werden und ein gefärbtes Ergebnis entsteht, abhängig waren, sind die vollautomatischen Ansätze der jungen Vergangenheit deutlich benutzerfreundlicher und effektiver. Das Lesen dieser Arbeit soll erleichtern, ein bestimmtes Verfahren vorrangig zu nutzen oder Interesse für eine bestimmte Modellarchitektur zu wecken, mit der sich der Leser näher beschäftigen möchte und die letztendlich Neugier für das Hantieren mit Deep-Learning-Techniken entfacht.

Diese Arbeit ist motiviert durch die rasante Entwicklung künstlicher Intelligenz (KI) im letzten Jahr, durch welche die Erstellung neuer GPT's (Generative Pretrained Transformer) folgte und die Welt der KI wesentlich veränderte. Die daraus resultierende Begeisterung für Modelle basierend auf Deep Learning, der Arbeit mit diesen und der Ortsbezug des Themas - der Umgang mit Bildern der Stadt Leipzig - sind weiterer Ansporn dieser Arbeit.

### 1.2. Zielsetzung

Das Ziel dieser Arbeit besteht darin, eine Auswahl historischer Fotos der Stadt Leipzig mit verschiedenen Deep-Learning-Methoden plausibel zu kolorieren, die Ergebnisse zu analysieren und digital zur Verfügung zu stellen. Es wird ein Einblick in verschiedene Verfahren des digitalen Kolorierens mittels Deep Learning gegeben und diese miteinander verglichen, sodass eine Aussage getätigt werden kann, welche der untersuchten Methoden am ehesten für diverse Ansprüche des Nutzers geeignet ist. Im Fokus steht hierbei das Projekt DeOldify [2], dessen Architektur und besondere Arbeitsweise im Detail betrachtet wird. Dabei wird die Bedeutung dieses Projektes auf nachfolgende Koloriermodelle und -ansätze erwähnt.

### 1.3. Aufbau der Arbeit

Im Abschnitt 2 werden Grundlagen zum besseren Verständnis und Einordnung des Themas präsentiert. Nachdem in 2.1 kurz die Anfänge der Fotografie und ersten Methoden der analogen Färbung alter Bilder vorgestellt wurden, wird in 2.2 auf die digitale Kolorierung im Detail eingegangen. Dabei werden Supervised, Unsupervised und Reinforcement Learning des maschinellen Lernens erklärt. Im darauf folgenden Abschnitt 2.3 wird der Begriff des Deep Learning in Bezug auf KI und ML eingeordnet und KNNs, CNNs und GANs detailliert vorgestellt. Schließlich werden in Punkt 3 die verwendeten Projekte betrachtet und mit besonderer Ausführlichkeit die Architektur von DeOldify eruiert. Danach wird in Kapitel 3.3 die Implementation dieser Projekte beschrieben, bevor letztlich in Abschnitt 4 die Ergebnisse der Arbeit technisch und visuell evaluiert werden. Den Abschluss der Arbeit bildet das Fazit im Punkt 5, welches die gewonnenen Erkenntnisse zusammenfasst und einen Ausblick auf zukünftige Herangehensweisen an das Thema der digitalen Kolorierung bietet.

## 2. Grundlagen

### 2.1. Historie

Bevor in die technischen Details der digitalen Fotografie und Bildbearbeitung der Gegenwart eingegangen wird, werden kurz die ersten erfolgreichen Versuche der Bilderfassung dargestellt.

#### 2.1.1. Anfänge der Fotografie

Die Camera obscura, welche ursprünglich nur ein Raum mit einem lichtdurchlässigen Loch war, beschrieb den ersten Mechanismus, ein vom menschlichen Auge erkanntes Bild (zumindest temporär) zu projizieren. Erste Überlieferungen von Anwendungen dieses Mechanismus gibt es laut LEFÉVRE [3] erst im 17. Jahrhundert, wobei vermutet wird, dass dieses Phänomen bereits in der Antike entdeckt wurde. Die ersten physischen Exemplare in verschiedenen Formen beinhalteten nur eine einzige konvexe Linse. Die Bilderzeugung erfolgte lediglich mithilfe von extern einströmendem Sonnenlicht, welches durch Brechung in der Linse ein reales, invertiertes Abbild auf einer dahinterliegenden Oberfläche erzeugte. Dieses konnte daraufhin von Künstlern in einem umständlichen Prozess des Rotierens und Wendens (analog zur Umkehrung eines Negativs mit einer aktuellen Kamera) abgepaust oder abgezeichnet werden. Neben der großen Wertschätzung für die Welt der Kunst war die Camera obscura in der Astronomie als Observationsmittel, in der Anatomie zum Verstehen des menschlichen Auges und in der Ergründung der wissenschaftlichen Optik von wesentlicher Bedeutung. [3, 4]



Abbildung 2.1.: Blick aus dem Arbeitszimmer in Le Gras von Nicéphore Niépce, 1826

Erst im Jahr 1826 gelang es, die flüchtigen Bilder in der Dunkelkammer zu fixieren und somit das erste "richtige" Foto herzustellen. Der Franzose Joseph Nicéphore Niépce gilt als der Erfinder des ersten Verfahrens, welches dauerhafte Bilder erzeugen konnte, der sogenannten Heliografie. Ihm gelang es, mithilfe von einer dünnen Asphaltsschicht, welche in Lavandelöl gelöst und anschließend auf einer versilberten Kupfer- oder Zinnplatte angebracht wurde, den Blick aus seinem Arbeitszimmer als Fotografie bei einer Belichtungszeit von ungefähr acht Stunden zu entwickeln. Diese erste Art der Momenterfassung gilt als die älteste erhaltene Fotografie (Abb. 2.1). Gemäß EDER [5] war Niépces größte Herausforderung nun, seine bahnbrechende Entdeckung anwendungsreif und für eine breitere Masse zugänglich zu machen. Deshalb schloss er sich 1829 mit dem französischen



Maler Louis Daguerre zusammen, jedoch gelang es den beiden während Niépces Lebenszeiten nicht, das anfangs sehr materiell aufwändige Verfahren kommerziell nutzbar zu machen. [5]

Daguerre schaffte es 1839, monochrome (schwarz-weiße) Fotos reproduktiv anfertigen zu können. Aufbauend auf den Grundlagen Niépces entdeckte er, dass das Silberiodid allein, was beim Behandeln der versilberten Kupferplatten mit Ioddampf entsteht, wegen seiner hohen Lichtempfindlichkeit reicht, um Bilder herzustellen. Eine Alternative stellte Quecksilberdampf dar und sorgte für ein ähnliches Ergebnis. Für seine sogenannte Daguerreotypie verwendete er eine verbesserte und kleinere Version der Camera obscura (oder Lochkamera) als Fotoapparat. [6] Die Daguerreotypie wurde durch darauf aufbauende Methoden wie die Kalotypie, Ambrotypie und Tintypie, welche erste Versuche der Farbfotografie repräsentierten, ergänzt und verfeinert [7]. Jedoch blieb die Mehrheit der fotografischen Aufnahmen bis zur Mitte des 20. Jahrhunderts monochrom [8].

Das erste offiziell erfolgreich entwickelte Farbfoto wurde von Thomas Sutton für eine Vorlesung von James Clerk Maxwell, der vor allem bekannt in der Physik durch seine Ausarbeitung der Maxwell-Gleichungen ist, kreiert. Maxwell erfand außerdem im Jahr 1861 die Drei-Farben-Methode (Rot, Grün und Blau), von welcher sich Sutton in den 1890er Jahren Gebrauch macht, um mit ihr das erste Farbfoto zu erschaffen. [9]

Dies ist nur ein sehr kleiner Überblick über die ersten Meilensteine der monochromen und Farbfotografie. Hauptaugenmerk dieser Arbeit soll jedoch die Kolorierung der schwarz-weißen Exemplare sein, sodass im Folgenden auch auf die ersten Versuche der händischen Kolorierung eingegangen wird.

### 2.1.2. Handkolorierung

Im Allgemeinen wird das Kolorieren als das Färben monochromer (schwarz-weißer) Bilder, seien sie bewegt oder unbewegt, definiert [10]. Im Englischen wird der Begriff "colorization" auch vom Begriff "coloring" (oft mit "hand coloring" spezifiziert) unterschieden. Ersterer beschreibt meist dabei den digitalen Vorgang, letzterer ist älter und an den händischen Prozess angelehnt. In der deutschen Sprache wird diese Differenzierung nicht vorgenommen. Neben der mittelalterlichen Buchmalerei, die den Ursprung der händischen Illustrationsfärbung darstellt, soll in dieser Arbeit aber vor allem auf das Färben von Fotos eingegangen werden.

Die Handkolorierung von Fotos, wie wir sie heute kennen, wurde direkt nach der Entwicklung erster Daguerrotypien mit verschiedenen Ansätzen in Angriff genommen und ist somit erstmals auf diese Zeit zurückzuführen. Für eine Zusatzgebühr konnte man beim Künstler verlangen, dass das schwarz-weiße Foto nachkoloriert wird. Der Beauftragte notierte sich die Farbe der Kleidung, Augen und Haare des abgebildeten Kunden, falls es sich um ein Portrait handelte [11]. Dieses Angebot der nachträglichen Einfärbung war notwendig, da laut HENISCH ET AL. [12] bei Weitem nicht alle Menschen von der "neuen Magie" der ersten monochromen Fotos überzeugt waren. Sie vermissten die Farbenfroheit der Gemälde, die in ihrem Zuhause hingen, und trachteten danach, ein Gegenmittel zur Monotonie der Abbildungen zu finden. Diese Bestrebungen nach mehr Realismus wurden zu Beginn mit einem Gemisch aus verschiedenen Pigmenten und Gummi Arabicum als Bindemittel erreicht. Unter Hitzeeinfluss wurden die Farbstoffe anschließend auf den Daguerrotypien fixiert.

Johann Baptist Isenring war 1839 der erste Künstler, der diese Technik nutzte. In den Daguerrotypien in Abb. 2.2 ist eine solche Methode verwendet wurden. Dabei handelt es sich im linken Bild um ein Daguerreotypieporträt eines Mädchens (USA um 1850), in der Mitte sieht man eine subtil kolorierte Daguerreotypie aus dem Atelier William Edward Kilburn (London um 1850-55) und auf der rechten Seite ein Porträt eines Jungen (USA um 1850). Die Künstler der amerikanischen Exemplare sind unbekannt, da gerade in den USA um diese Zeit diese Daguerrotypien in Massen angefertigt wurden und deshalb eine Nachverfolgung des Ursprung sehr schwierig ist.[13, 12]



Abbildung 2.2.: Daguerrotypien, entstanden um 1850 in den USA und England

Seit der Mitte des 19. Jahrhunderts ist mit den Methoden des frühen Kolorierens stetig weiterexperimentiert wurden, jedoch ohne nennenswerte Fortschritte zu erreichen. Erst einige Jahre später, mit der Erfindung der Solarkamera und damit einhergehend das Vorhandensein einer ausreichend starken Lichtquelle während des Vorgangs, konnten Fotos vergrößert auf Papier oder Leinwänden dargestellt werden [14]. So konnten beispielsweise Portraits für das Bearbeiten leichter zugänglich gemacht und in Lebensgröße abgebildet werden. Das durch diese und weitere Errungenschaften vereinfachte Handkolorieren war bis zur Mitte des 20. Jahrhunderts und zur Erfindung des Kodachrome Farbfilms die beliebteste Methode, um gefärbte Fotografien herzustellen. Trotz der Existenz der seitdem deutlich unkomplizierteren Methode und deren Nachfolgern, war das händische Kolorieren von Fotos in den folgenden Jahren immer noch von großer Bedeutung und erlebte in den 1970ern eine Art Wiedergeburt, weil viele Menschen sich nach dem Antiken und Alten sehnten. Bis heute ist die Handkolorierung eine beliebte Tätigkeit, sei es wegen visueller Ästhetik oder der Beständigkeit der beim Vorgang verarbeiteten Farbpigmente. [15]

FERGUSON [13] behauptet, dass die gewaltigsten Hürden, die es zu bezwingen galt, neben den materiellen Kosten vor allem der körperliche und zeitliche Aufwand, monochromen Aufnahmen mit Farben einen neuen Glanz zu verleihen, waren. Das Foto musste erst mit den vorhandenen Praktiken geschossen, entwickelt und gedruckt werden, bevor das Ergebnis von einem - meist demselben - Künstler in einem oft langwierigen Prozess handgefärbt werden konnte. Die nachgefragtesten Fotografen und Koloristen besaßen deshalb eine traditionelle künstlerische Ausbildung. Kolorierte Fotos waren deutlich erschwinglicher als echte Gemälde und erwiesen sich in der Öffentlichkeit schnell als lukrativ. [13, 16]

### 2.1.3. Digitales Kolorieren

Mit dem technischen Fortschritt kam jedoch die starke Vereinfachung vieler aufwändiger Prozesse, so auch in der Aufarbeitung alter Bilder. Deshalb ist heutzutage das Kolorieren hauptsächlich ein rechnergestützter Prozess, wobei der Aufwand des Färbens praktisch komplett von Programmen und darin verwendeten Algorithmen übernommen wird. Moderne Technologien und der Fortschritt im Bereich der künstlichen Intelligenz (KI) haben diesen Prozess revolutioniert und deutlich erleichtert. Trotz alledem ist es noch lange kein trivialer Vorgang. Besonders wenn man ausgezeichnete Resultate erwartet, sind eine gewisse Expertise und Erfahrung, gerade wenn es um Fotorestoration geht, fundamental.

Wenn man als Laie vorhat, selbst Kolorierungen durchzuführen, lehnt man sich in erster Linie an Programme, die mithilfe künstlicher Intelligenz vollautomatisch diesen Prozess übernehmen. Allein im letzten Jahrzehnt wurde (nicht nur in der digitalen Färbung) ein Meilenstein nach dem anderen gelegt, wenn es darum geht, zeitaufwändige Abläufe effektiv und ressourcensparend mittels KI zu simplifizieren. Um einen kleinen Einblick in die technische und wissenschaftliche Bedeutung zu erlangen, werden im Folgenden die Auswirkungen der KI vor allem in Bezug auf das maschinelle und tiefe Lernen aufgezeigt.

## 2.2. Maschinelles Lernen

Maschinelles Lernen (ML) ist inzwischen aus vielen Bereichen der Informationstechnik nicht mehr wegzudenken. Die Anwendungsweite reicht von der Filterung von Inhalten in sozialen Netzwerken, über Verfeinerung von Web-Suchmaschinen bis hin zum Zuschneiden von Werbe- und Nachrichteneinblendungen des Nutzers. ML-Systeme werden desweiteren in der digitalen Spracherkennung, -verarbeitung und zur Analyse von Bildmaterial (beispielsweise in der medizinischen Diagnostik) eingesetzt. Bei letzterem spielt vor allem die Erkennung und Färbung spezifischer pathologischer (möglich krankhafter) Bestandteile wie Tumoren [17] und Myomen [18] eine große Rolle. [19]

Im Allgemeinen ist maschinelles Lernen ein Teilgebiet der KI. Es sorgt dafür, dass Systeme mithilfe bestimmter Lernalgorithmen der KI gewisse Muster oder Regeln in meist großen Datenmengen erkennen, sie beschreiben, aus ihnen lernen und sogar Ergebnisse voraussagen können. ML ist dazu da, auch ohne explizites Programmieren komplexe Aufgaben zu lösen. Dabei ist es wichtig, ein geeignetes Modell zu entwerfen, damit ein gewünschtes Resultat erreicht werden kann. [20]

Laut MOHAMMED ET AL. [21] ist die fundamentalste Frage dieses Themengebiets, wie man Computer dazu bekommt, sich selbst (mit gesammelter Erfahrung und etwas Starthilfe von außen) zu programmieren. Deshalb ist ein weiterer wichtiger Punkt zur Bedeutung von ML-Programmen deren Adaptivität. Durch das ständige Trainieren mit angemessenen Datensätzen werden von den Programmen neue Lösungsstrategien erarbeitet und Resultate gefunden. Anders als konventionelle Computerprogramme, die einmal geschrieben nur schwer oder gar nicht veränderbar sind, sind sie durch das Anwenden der gelernten Erfahrungen deutlich flexibler und vor allem zu variierendem Input adaptiver. Um die genannten Ziele zu erreichen, werden beim maschinellen Lernen verschiedene Techniken eingesetzt, die im Folgenden kurz vorgestellt werden. [21, 22, 23]

### 2.2.1. Supervised Learning

Supervised learning (oder überwachtes Lernen) bedeutet, dass dem Programm neben der Eingabedatenmenge die gewünschten Ausgaben mitgegeben werden. Dabei sind die Eingaben mit spezifischen Attributen "gelabelt" oder beschriftet und beschreiben deren Inhalt, um das zu trainierende ML-Modell auf den richtigen Pfad zu lenken. Ziel dieses Vorgangs ist es, die Muster in den übergebenen Daten zu erkennen und ein Mapping von den übergebenen Eingaben zu den Ausgaben zu erstellen. Dies geschieht zum Beispiel bei der herkömmlichen Klassifizierung von Tierbildern, bei der die Programme mit jedem Input (Bild einer Katze) ein Output (Katze) assoziieren sollen, bei einem Hundebild soll ein Hund erkannt werden. Dabei soll mithilfe der bereits richtig gelabelten Bilder gelernt werden. Desweiteren soll nach Erstellen des Mappings neue Inputs mit bekannten Outputs verknüpft werden. Dieser Ablauf belegt letztendlich das Lernen des Programms, das größte Ziel der Arbeit mit (überwachten) Lernalgorithmen. Nach einigen Vorgängen und wiederholtem Trainieren des Programms kann sogar sein Verhalten unter Verwendung neuer Datensätze vorausgesagt werden. Der Überwacher (oder supervisor) ist dabei meist ein Mensch, der die Daten labelt und zur Verfügung stellt. Jedoch können Maschinen auch Überwacher sein, was ressourcensparender aber oft (noch) ungenauer als das menschliche Pendant ist. Überwachtes Lernen ist deshalb laut GOODFELLOW ET AL. [24] zum einen durch die Abhängigkeit menschlichen Inputs und Kontrolle, zum anderen durch den noch unoptimierten Lernprozess der Programme eingeschränkt, der noch nicht mit dem des Menschen konkurrieren kann. Beispielsweise müssen die Programme oft mehrere Millionen Male an Datensätzen trainiert werden, um die menschliche Performance zu übertreffen, wobei der Mensch ausreichende Ergebnisse bei deutlich geringerem Trainingsaufwand liefert. Natürlich ist dabei die Erwartungshaltung an die trainierte KI und der gelegte Maßstab sehr hoch, um den größtmöglichen technischen Fortschritt zu erlangen und bestimmte Abläufe stets zu optimieren. [21, 24]

Neben der bereits erwähnten Klassifikationsalgorithmen werden häufig Regressionsalgorithmen beim überwachten Lernen eingesetzt.

### 2.2.2. Unsupervised Learning

Beim unsupervised learning (zu Deutsch etwa unüberwachtes Lernen) fehlen sowohl Überwacher als auch die gewünschten Ausgaben, die erzeugt werden sollen. Dabei soll das Programm selbst die Struktur und Zusammengehörigkeit der eingegebenen Daten erschließen, ohne deren Bedeutung zu kennen. Unüberwachtes Lernen ist im Vergleich zum überwachten Lernen weniger präzise definiert, jedoch gibt es in diesem Strang des maschinellen Lernens sehr viele verschiedene Algorithmen und Ziele, die mit diesen verfolgt werden können [24]. Die Technik des unüberwachten Lernens wird vor allem beim Ermitteln neuer riesiger Informationsmengen im Bereich Big Data benutzt, um diese später eventuell zu labeln. Desweiteren wird sie bei Algorithmen wie der Clusteranalyse verwendet, bei der die eingegebenen Daten in Gruppen mit ähnlichen Eigenschaften (sog. Cluster) gegliedert werden. Die Clusteranalyse wird im Marketing und in der Produktentwicklung zur Kundengruppierung eingesetzt, um beispielsweise Newsletter zu personalisieren [25]. [21]

Ein wichtiger weiterer Ansatz liegt im "generative modeling" oder in der erzeugenden/generativen Modellierung, auf die im Abschnitt 2.3.3 im Detail eingegangen wird, da sie die Basis des Modells bildet, welches hauptsächlich in dieser Arbeit zum Färben monochromer Bilder genutzt wird.

### 2.2.3. Reinforcement Learning

Im Vergleich zu den beiden anderen Arten werden beim reinforcement learning (zu Deutsch etwa verstärkendes Lernen) keine Eingabe- und Ausgabedaten benötigt. Stattdessen wird die Methode des "Trial-and-Error" genutzt, um gewisse Probleme in einer Simulationsumgebung mit vorgeetzten Größen iterativ zu lösen. Diese Größen beinhalten den aktuellen Stand des Systems, das gewünschte Ziel des Vorgangs, eine Reihe von erlaubten Schritten und deren Einschränkungen in der Umgebung für das verwendete ML-Modell [26]. Sinn der Aufgabenlösung mittels verstärkenden Lernens ist es, optimale Richtlinien- bzw. Wertefunktionen zu finden. Anwendung findet diese Technik mit großem Erfolg vor allem in geschlossenen Simulationsumgebungen wie z.B Schach oder anderen strategischen Spielen, bei denen Taktiken und Spielzüge bei bestimmten Spielsituationen optimiert werden können [27]. Desweiteren sind sie relevant in der Anwendung von Multiagentensystemen in beispielsweise elektronischen Marktplätzen [28]. [29]

## 2.3. Deep Learning

Das Deep Learning (DL) oder "tiefe Lernen" ist eine Teilmenge des maschinellen Lernens und somit auch eine Untergruppe des großen Begriffes der künstlichen Intelligenz (Abb. 2.3). Der größte Vorteil des Deep Learning besteht vor allem darin, riesige Datenmengen nicht nur von einer Quelle, sei es zur Strukturierung oder zur tiefgründigen Analyse, sondern auch gesammelte Daten mehrerer Quellen zu bearbeiten. Dies ermöglichen enorme Ketten von mehreren Schichten nichtlinear arbeitender Einheiten, um bestimmte Merkmale zu extrahieren und zu verwerten. Generell lernen hierbei die Schichten, welche näher an den Inputdaten liegen (sog. "lower layers"), einfache Eigenschaften und von diesen abgeleitet lernen weiter entfernte Schichten ("higher layers") komplexere Eigenschaften. [30] [31]

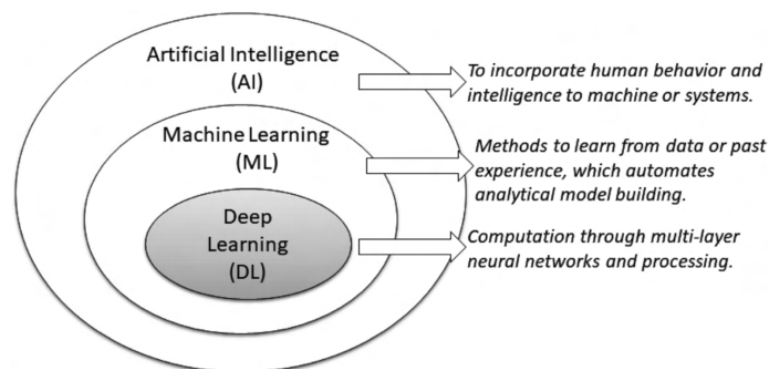


Abbildung 2.3.: Deep Learning im Zusammenhang mit maschinellem Lernen und künstlicher Intelligenz

Wichtige und bekannte Applikationen der letzten Jahre, die auf DL basieren, sind im Bereich Big Data unter anderem Microsofts Spracherkennung (MAVIS), bei der die Suche von Video- und Audiodateien mittels menschlicher Stimme und Sprache ermöglicht wird, aber auch Googles Bildsuche funktioniert mithilfe von Strukturen des Deep Learning, um Abbildungen mit Anmerkungen zu versehen oder zu taggen und sie somit für weitere Bildsuchmaschinen vorzubereiten. Weiterhin beschäftigten sich viele Autoren und Forscher mit den Vorhersagequalitäten (Prediction) von DL-Strukturen, sei es in der Bioinformatik [32], im Wirtschafts- und Finanzwesen [33] und in der Analyse und Aufarbeitung von Bilddateien [34]. Die meisten DL-Methoden basieren auf der Verwendung künstlicher neuronaler Netzarchitekturen und werden im Folgenden näher beleuchtet. [19] [30]

### 2.3.1. Künstliche Neuronale Netze

Die Basis für künstliche neuronale Netze (engl. artificial neural networks, hier KNN) orientiert sich an der Physiologie des Menschen, genauer gesagt am Neuronennetzwerk des menschlichen Gehirns. Diese Neuronen, welche aus Axonen, Dendriten und Synapsen bestehen, kommunizieren mittels elektrischen Impulsen im neuronalen Geflecht untereinander [35]. Das daran angelehnte Modell des künstlichen Neurons (Abb. 2.4) von MCCULLOCH und PITTS beschreibt, wie es eine gewichtete Summe (weighted sum) von  $n$  Eingangssignalen ( $I_j$  mit  $j = 1, 2, \dots, n$ ) in einer Schwellenfunktion (threshold function) verarbeitet und Ausgaben liefert. Bei dieser sogenannten "binären Schwelleneinheit" wird 1 ausgegeben, wenn die Summe eine gewisse Schwelle erreicht und 0 sonst. Die Gewichtskoeffizienten bestimmen die Relevanz der Neuronenkette im Netzwerk, also wie wichtig die Verbindung zweier oder mehrerer Neuronen für das neuronale Netz sind.

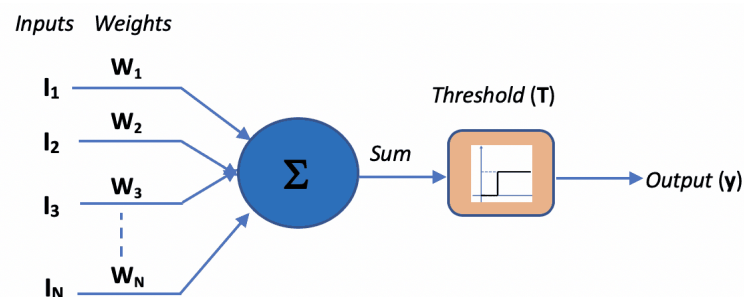


Abbildung 2.4.: McCulloch-Pitts-Neuron

Die Eingangssignale kommen dabei von anderen Neuronen im Netzwerk und aus externen Quellen. Positive Signale wirken exzitatorisch (erregend) und negative inhibierend (hemmend) auf das System. Die Auswertung der Eingaben ist nicht beschränkt auf eine Schwellenfunktion, sondern kann auch teilweise linear, eine Gaußsche Funktion oder sigmoid (S-Funktion) sein, wobei letztere laut JAIN ET AL. [36] die bei Weitem meistbenutzte in KNNs ist. McCulloch und Pitts bewiesen mithilfe dieses Modells, dass eine synchrone Anordnung der Neurone in einem Netz mit wohlgeählten gewichteten Eingangssignalen allgemeine Berechnungen lösen können. [37, 38]

Der Hauptvorteil neuronaler Netze liegt in der Verarbeitung *a priori* unbekannter Informationen, die in Datenmengen versteckt sind. Mithilfe des gezielten Anpassens der Gewichtskoeffizienten wer-

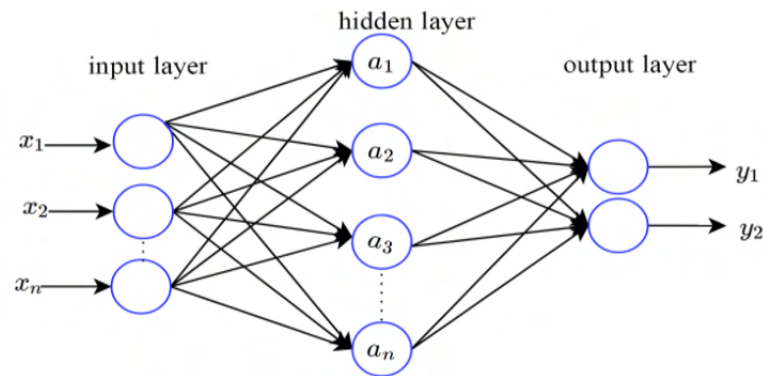


Abbildung 2.5.: Allgemeiner Aufbau eines vorwärtsgerichteten KNN-Modells

den gezielte Ausgaben erzeugt und somit bestimmte Bedingungen erfüllt. Diese Größen und Abläufe bilden so die Grundlage des Lernens neuronaler Netze, eine fundamentale Eigenschaft der Intelligenz, welche im Abschnitt ML bereits erklärt wurde. Ein Anwendungsbeispiel des überwachten Lernens bildet das "feed forward" oder vorwärtsgerichtete NN, welches eine Kategorie von KNNs bezüglich ihrer Architektur darstellt (Abb. 2.5). [38]

Hierbei sieht man die bereits vorgestellten Neurone in Verbindung durch Schichten, wobei die Eingaben  $x_1, x_2, \dots, x_n$  die Eingabeschicht,  $a_1, a_2, a_3, \dots, a_n$  die verborgene Schicht und schließlich  $y_1$  und  $y_2$  die Ausgangschicht bilden. Dabei sind die Eingabe- und Ausgangschicht in der Regel jeweils ein Mal vertreten, jedoch können mehrere verborgene Schichten vorkommen. Wenn das der Fall ist, spricht man von einem "tiefen" neuronalen Netz oder "deep neural network". Diese Netze sind namensgebend für ihre Anwendung im DL [35]. Bei der obigen Abbildung handelt es sich um ein zweischichtiges NN, weil die Eingabeschicht keine Berechnungen durchführt und somit nicht gezählt wird. Vorwärtsgerichtet ist es, weil die Kommunikation zwischen den Neuronenschichten nur zur jeweils folgenden Schicht geschieht. [38]

Als Pendant dazu ist das "recurrent/feedback" oder rückgekoppelte NN (RNN) zu erwähnen, bei welchem die Neuronen mit sich selbst (direkte Rückkopplung), mit einem Neuron einer vorherigen Schicht (indirekte Rückkopplung) oder mit Neuronen derselben Schicht (laterale Rückkopplung) kommunizieren können. Durch diese Rückkopplungsschleifen werden permanent Informationen zurück ins Netz gesendet, wodurch Eingaben miteinander verbunden werden und Rechenprozesse optimiert werden können. Besonders für Aufgaben mit sequentiellen Eingaben wie z.B. in der Sprachmodellierung [39], Spracherkennung [40] und maschinengestützten Übersetzung [41] finden RNNs ihre Anwendung. [42]

Als eine weitere Art neuronaler Netze sind die "convolutional" oder gefaltete NNs zu nennen, die vor allem ihren Nutzen im Verarbeiten von Audio-, Bilder- oder Daten mit räumlicher Struktur und damit einen großen Anteil an den in dieser Arbeit vorgestellten Modellen haben.

### 2.3.2. Convolutional Neuronale Netze

Die Eingabedaten, die in dieser Arbeit von Relevanz sind, haben die Form von zweidimensionalen Arrays oder Matrizen von Grauwerten, bei der jeder Pixelwert im Bild beschreibt, wie hell

oder dunkel der entsprechende Punkt ist. Das Problem, bestimmte Muster in diesen Bilddateien zu erkennen (und später farbige Ausgaben zu liefern), lösen gefaltete neuronale Netze (CNN's) mithilfe von sogenannten Faltungsschichten (convolutional layers), die eine Menge von Filtern (kernels) besitzen. Diese Filter analysieren nacheinander kleine Bildausschnitte, indem sie diese mit einer festen Pixelgröße (kernel size) und Schrittweite (step size) abarbeiten. Mit einem festgelegten Padding wird dabei beschrieben, wie der Filter reagieren soll, wenn er zum Beispiel an den Rand einer Pixelmatrix gelangt. Für jeden Punkt im Sichtfenster des Filters wird ein festes Gewicht mitgegeben, mithilfe dessen und - unter Einbeziehung der Pixelwerte - Ergebnismatrizen für alle Sichtfenster berechnet werden. Die Größe der Matrizen ist dabei von der Schritt-, Pixelgröße und dem gewählten Padding abhängig. Durch diese Vorgehensweise ist nicht jedes Pixel mit dem Filter verbunden, sondern z.B bei einer 2x2-Filtergröße jeweils vier Pixel gleichzeitig, sodass die Ergebnismatrix um ein Vielfaches kleiner als die Inputmatrix ist. Die Eingaben werden somit "gefaltet". Auf diesen - aus (meist mehreren) Faltungsschichten aufgebauten - ersten Teil eines CNNs folgt eine sogenannte Pooling-Schicht. Sie ist dazu da, die Anzahl der Parameter im Netzwerk zu reduzieren, indem sie die Zwischenergebnisse wiederum mithilfe einer Aggregatfunktion, die beispielsweise das Maximum oder den Durchschnitt der übergebenen Signale bildet, filtert. In der Bildverarbeitung hat dies ein "downsampling", also ein Herabsetzen der Auflösung, zur Folge. Dadurch gehen meist feinere Details der eingegebenen Bilder verloren, jedoch werden durch die Ausrichtung des Netzwerks auf robustere Merkmale gröbere Muster erkannt. Nach den Faltungs- und Pooling-Schichten, die sich in vielen CNN's je nach Größe der Datenmenge und gewählten Architektur wiederholen, schließt sich eine vollständig verknüpfte Schicht (fully connected layer) an. Diese ist oft der letzte Teil eines CNN's und ist (gemeinsam mit der letzten Ausgabeschicht) für die Ausgabe der von den vorherigen Schichten weitergegebenen Ergebnisse zuständig. Dabei sind alle Neuronen dieser Schicht direkt mit allen Neuronen vorangegangener (und möglicher nachfolgender Schichten) verbunden, was dem Aufbau eines traditionellen neuronalen Netzes entspricht. Vollständig verknüpfte Schichten operieren mit einer Vielzahl von Parametern, deren Auswertung und Berechnung die meiste Zeit im Training von CNN-basierten Modellen in Anspruch nehmen. [43, 44, 45]

### 2.3.3. Generative Adversarial Networks

Eine weitere Art neuronaler Netze sind die "Generative Adversarial Networks" (GAN), zu Deutsch etwa "Generierende konkurrierende Netzwerke", die zum ersten Mal von GOODFELLOW ET AL. [24] im Jahr 2014 vorgestellt und seitdem in vielen ML- und Bildverarbeitungsprojekten angewendet werden. Auf einer etwas abgewandelten Form der GAN-Architektur basiert ein wichtiger Teil dieser Arbeit, sodass diese Netzwerke hier von großer Bedeutung sind und im Detail beschrieben werden.

Die generative Modellierung ermöglicht im Allgemeinen das Erzeugen neuer Daten aus bereits bestehenden, ähnlichen Trainingsdaten, indem die Modelle die Verteilung jener lernen. Das Ziel besteht also darin, eine Sammlung von Trainingsbeispielen zu untersuchen und die Wahrscheinlichkeitsverteilung zu erlernen, die diese erzeugt hat. Generatives Modellieren steht im Gegensatz zum diskriminativen Modellieren, welches existierende Daten klassifiziert oder Vorhersagen auf Grundlage der Eingabevariablen trifft. Das Ziel ist es hier, eine Entscheidungsregel zu finden, die es dem jeweiligen Modell ermöglicht, Daten direkt zu klassifizieren, ohne sich auf eine bestehen-



de Verteilung zu konzentrieren. Das Wechselspiel dieser beiden Modellierungsarten wird bei der GAN-Architektur ausgenutzt, sie "konkurrieren".

Wie in Abb. 2.6 zu sehen ist, besteht ein GAN konkret aus zwei Teilnetzwerken, dem Generator und dem Diskriminator. Diese stehen im ständigen Konflikt, indem der Generator versucht, mit seinen generierten Daten den Diskriminator zu überzeugen, dass diese real sind. Der Diskriminator wiederum versucht, gefälschte (vom Generator erzeugte) Daten von den realen zu unterscheiden (klassifizieren). Durch regelmäßigen Austausch von Informationen und Anpassungen über viele Iterationen gelingt es bestenfalls dem Generator, Daten mit gültigem Muster zu erstellen, sodass der Diskriminator diese nicht mehr von realen unterscheiden kann. Der Aufbau von Generator und Diskriminator ist je nach Anwendung unterschiedlich. Neben CNN und RNN können sie auch als Autoencoder u.a. vorkommen. [24, 46]

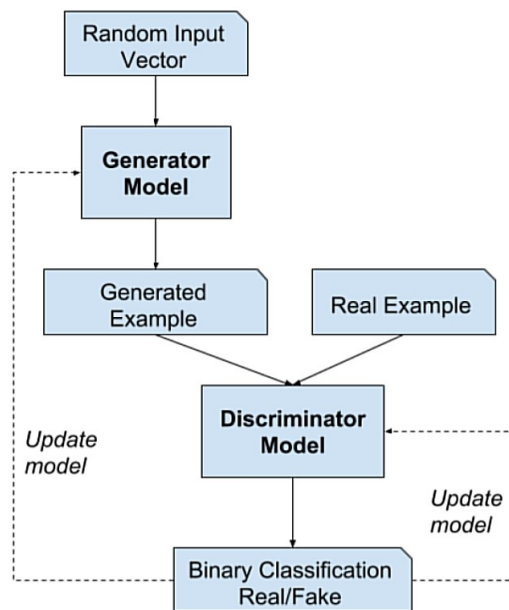


Abbildung 2.6.: Generative Adversarial Network [47]

Im Detail bekommt der Generator einen Rauschvektor (Zufallsvektor mit Gaußscher oder gleichmäßiger Verteilung) als Input, woraufhin er basierend auf der Standardsignalverteilung der Trainingsdaten ein "Fake Image" erstellt, welches in den Diskriminator gespeist wird. Dieser fungiert als binärer Klassifikator und wird so trainiert, dass er mit maximierter Wahrscheinlichkeit den echten und generierten Daten die richtigen Labels zuweist. Mithilfe von Verlustfunktionen (loss functions) werden bestimmte Parameter beider Modelle meist nach jedem Trainingsdurchlauf aktualisiert, so wie auch in Abb. 2.6 beschrieben ist.

## 3. Projekte

### 3.1. Verwandte Arbeiten

Bevor das Hauptprojekt DeOldify, mit dem sich diese Arbeit insbesondere beschäftigt und auf dessen Architektur viele andere Projekte basieren, im Detail erklärt wird, folgt eine kurze Vorstellung der anderen verwendeten Verfahren zum Kolorieren der Schwarz-Weiß-Bilder der Stadt Leipzig.

#### 3.1.1. Zhang et al.: Colorful Image Colorization (2016)

“Colorful Image Colorization” (CIC) [48] ist ein Forschungspapier von Richard Zhang, Phillip Isola und Alexei A. Efros, das 2016 veröffentlicht wurde. In diesem Papier greifen die Autoren das Problem an, eine plausible Farbversion eines Graustufenfotos zu erzeugen. Während frühere Ansätze auf starke Benutzerinteraktion angewiesen waren oder zu entsättigten Kolorierungen führten, wird hier ein vollautomatischer Ansatz vorgeschlagen, der lebendige und realistische Kolorierungen erzeugt. ZHANG ET AL. packen das Problem bei der Wurzel, indem sie es als Klassifikationsaufgabe formulieren und beim Training eine Klassen-Neubalancierung verwenden, um die Vielfalt der Farben im Ergebnis zu erhöhen. Das System wird als Vorwärtsdurchlauf in einem CNN zur Testzeit implementiert und auf über einer Million Farbbildern trainiert. Die Autoren bewerten ihren Algorithmus mit einem “Kolorierungs-Turing-Test”, bei dem menschliche Teilnehmer zwischen einem generierten und einem echten Farbbild wählen müssen. Ihre Methode täuscht Menschen erfolgreich in 32% der Versuche, deutlich höher als bei früheren Methoden. [48]

In späteren Projekten, die sich mit dem Kolorierungsproblem auseinandersetzen, wird dieser Test vollautomatisch mittels GAN-Architekturen (wie in 2.3.3 beschrieben) abgelöst.

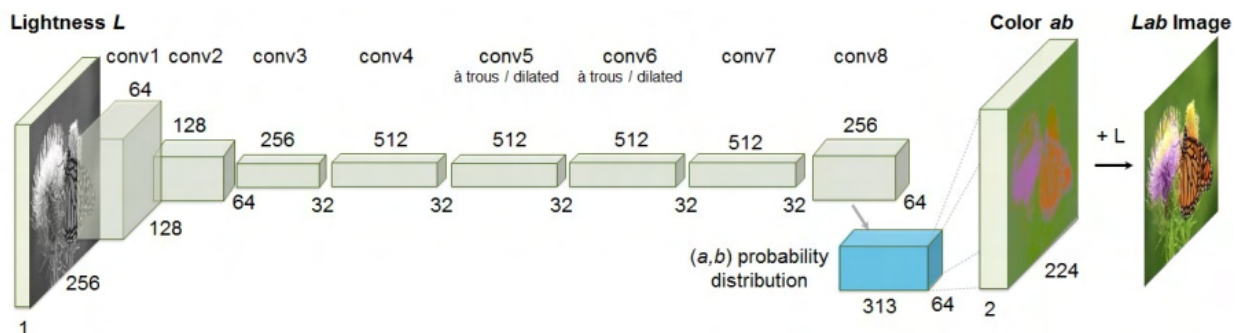


Abbildung 3.1.: Zhang et al.'s Netzwerkstruktur aus CIC [48]

In der dargestellten Netzwerkstruktur beinhaltet jede conv-Schicht einen Block aus zwei bis drei sich wiederholenden Faltungs- und ReLU-Schichten, die von einer BatchNorm-Schicht (in 3.2.1 erklärt) gefolgt werden, wobei das Netz keine Pooling-Schichten besitzt.

### 3.1.2. Hotpot.ai und VanceAI

In den Weiten der Internetwelt werden mittlerweile unzählige KI-Bildgeneratoren, -restaurierer, -kolorierer und -animierer angeboten, die sehr viele Aufgaben mit nur wenigen Klicks bewerkstelligen sollen. Mit Hotpot.ai und VanceAI werden hier zwei herausgepickt, um deren Qualität an eigenen Bildern zu testen und mit selbst implementierten Kolorierern zu vergleichen.

Hotpot.ai ist eine Webseite, die verschiedene KI-Tools für experimentierfreudige Nutzer anbietet. Neben einem KI-Kunstgenerator, Hintergrund- und Objektentferner, Bildskalierer und Fotorestaurierer stellt Hotpot auch einen "Photo Colorizer" zur Verfügung. Dieser Kolorierer basiert auf dem anfänglichen DeOldify-Modell, welches laut eigenen Angaben "proprietäre Verbesserungen basierend auf den neuesten Techniken im Bereich DL" [49] verwendet. Hotpot's Stable Diffusion API wird auf Anfrage und gegen einen gewissen Obolus zur Verfügung gestellt, jedoch wird hier lediglich das kostenlose Online-Angebot wahrgenommen.

VanceAI [50] ist ebenfalls ein Anbieter von KI-Fotoverbesserungs- und -bearbeitungstools, die auf tiefen neuronalen Netzen fußen. Dabei werden Mittel zur Hochskalierung, Schärfung, Rauschunterdrückung, Hintergrundentfernung und mehr angeboten. VanceAI ist neben der Onlineverfügbarkeit auch als Applikation für Windows, macOS, Android und iOS herunterzuladen. Wie bei dem anderen Onlineanbieter ist der Zugriff auf die verwendete API möglich, jedoch nur auf Anfrage und gegen den Abschluss eines kostenintensiven Abonnements. Ebenfalls wird in dieser Arbeit die kostenfreie Onlinevariante getestet.

Laut eigenen Angaben der Autoren dieser Toolanbieter basieren sowohl Hotpot.ai's als auch VanceAI's Kolorierer auf der Architektur einer Anfangsversion (stable) von DeOldify, die (wie schon angerissen) eigene Kniffe und Änderungen ihrer Autoren involvieren.

## 3.2. DeOldify

Jason Antics "DeOldify" [2] ist ein Open-Source-Projekt für vollautomatisches Kolorieren monochromer Bilder und Videos, welches 2018 mit Unterstützung vom kalifornischen Start-up-Unternehmen Fast.ai ins Leben gerufen wurde. Seit dem Erscheinen dieses Projektes gab es in den letzten Jahren unzählige Updates, einige Kollaborationen mit anderen Ansätzen und viele neue Ideen basierend auf der Struktur von DeOldify (z.B Teile von Hotpot.ai und VanceAI). Da die am meisten fortgeschrittene Version hinter einer Paywall steckt und nur über die Webseite myheritage.com zugänglich ist, wird zur Erklärung der technischen Details hauptsächlich auf die ursprüngliche Version eingegangen. Diese beinhaltete drei Varianten: Artistic, welche das Hauptaugenmerk auf den Farbenreichtum legt, video, bei der vor allem auf räumliche und zeitliche Stabilität geachtet wird, und stable, die nur räumliche Stabilität priorisiert, jedoch farbenfroher als die video-Version ist. Räumliche Stabilität ist in der Bildverarbeitung die Fähigkeit eines Algorithmus oder Modells, konsistente Ergebnisse im Hinblick auf die räumliche Struktur (korrekte Positionierung, Proportionierung und Textur) des Bildes zu erzielen. Zeitliche Stabilität bezieht sich auf stetige Ergebnisse in der temporalen Abfolge einzelner Frames oder Bilder, hauptsächlich wichtig für die Verarbeitung von Videomaterial, aber auch interessant für mehrere Bilder der gleichen Szene, bei der sich einige Objekte oder Personen

bewegen. Kurz gesagt, geht es also darum, störende Artefakte im Bild zu vermeiden [51]. Diese Varianten besitzen auch leichte Unterschiede in Training und Architektur, wobei sich hier auf den Typ artistic fokussiert wird. [2]

DeOldify nutzt eine modifizierte GAN-Architektur, die Antic selbst als NoGAN bezeichnet, bei der die Vorteile des GAN-Training beibehalten werden, ohne dabei viel Zeit mit dem eigentlichen Training zu verschwenden, diese Zeit also zu minimieren. Im Folgenden wird detailliert auf die Struktur und Arbeitsweise des Projektes eingegangen.

### 3.2.1. Technische Details

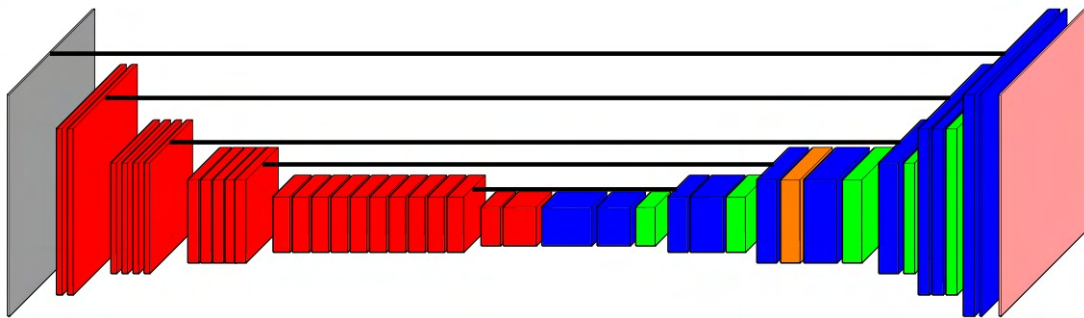


Abbildung 3.2.: U-Net Netzwerkstruktur

Die Struktur des Netzwerkes basiert auf einem sogenannten U-Net, dessen Namensherkunft in Abb. 3.2 anhand seiner U-Form nachvollzogen werden kann. Am Anfang des Netzes steht das Inputbild in Graustufen, welches im RGB-Farbraum codiert ist, also für jeden individuellen Pixel die gleichen Werte für Rot, Grün und Blau besitzt und benachbarte Pixel sich somit nur in der Helligkeit ihres Grautones unterscheiden. Nur mit dieser RGB-Kodierung kann das vortrainierte ResNet arbeiten.

Dieses "Residual Network" in der Version ResNet101 (in der Abbildung in rot dargestellt) ist ein in der Welt der Bildverarbeitung sehr bekanntes tiefes Lernmodell, das erstmals 2015 von HE ET AL. [52] vorgestellt wurde, vor allem um das bekannte "vanishing gradient" -Problem zu lösen. Dabei handelt es sich um die Tatsache, dass bei einer höheren Anzahl an Netzwerkschichten auch seine Leistung abnimmt bzw. seine Trainings- und Testfehlerrate zunimmt. Das Problem wird mithilfe des Konzeptes der "Residual Blocks" gelöst, welche durch sogenannte Skip-Verbindungen (schwarze Linien in Abb 3.2) gebildet werden. Diese Verbindungen verknüpfen Aktivierungen von Schichten mit tieferen und ermöglichen damit, Schichten, die die Leistung des Netzes beeinträchtigen würden, durch Netzwerkregulierung zu überspringen. Anstatt die Layer immer das bestehende Mapping der vorherigen lernen, wird dem Netzwerk ermöglicht, das "Rest-Mapping" mithilfe eines individuell errechneten Restwerts anzupassen. [52]

Wie bei den meisten ML-basierten Anwendungen ist es sehr hilfreich, die Inputdaten vor dem Durchlaufen des Netzwerkes so vorzubereiten, dass die Inputs ähnliche strukturelle und statistische Eigenschaften wie die Daten des ImageNet-Datensatzes [53], mit denen das ResNet trainiert wurde, besitzen und somit die Fehlerwahrscheinlichkeit reduziert wird. Anbei ist es generell wichtig, Daten, die man miteinander vergleichen möchte, auf eine gemeinsame Skala herunterzubrechen, um

diese gleichwertig gewichtet zu verarbeiten. Diese Normalisierung geschieht hier vor allem durch Standardisierung der Daten mittels mittlerer- und Standardabweichung.

Die in Abb. 3.2 blau gekennzeichneten Abschnitte sind die "convolutional blocks", deren Arbeitsweise dem eines CNN's (wie im Abschnitt 2.3.2 beschrieben) ähneln. Basierend auf dem VGG-16-Netzwerk [54], welches eine hohe Testgenauigkeit am ImageNet-Datensatz ausweisen kann, beträgt die Filtergröße der einzelnen Konvolutionen  $3 \times 3$ . Besonders hierbei ist, dass auf die Konvolutionen meist immer eine Batch-Normalisierung [55] und Aktivierung mittels einer ReLU-Funktion folgt. Die Batch-Normalisierung ist eine Methode zur Umparametrisierung von neuronalen Netzen, um die Koordination zwischen verschiedenen Schichten zu erleichtern [56]. Es ist nichts anderes als die Normalisierung der Inputs für folgende Schichten in kleinen Batches, anstatt jeden Input einzeln zu normalisieren. Dadurch wird unter anderem die Leistung des Netzwerks positiv beeinflusst. ReLU oder "Rectified Linear Unit" ist die am häufigsten verwendete Aktivierungsfunktion in neuronalen Netzen mit Klassifizierungsproblemen, die erstmals von FUKUSHIMA in [57] veröffentlicht wurde. Ursprünglich ebenfalls zur Lösung des vanishing-gradient-Problems eingeführt, behält sie eingespeiste positive Werte bei und setzt negative gleich null. Der Lernprozess des Netzwerkes wird beschleunigt und robuster gemacht, da etwaige Fehler durch die Linearität der Funktion alle Schichten durchlaufen. Eine weitere Normalisierungsmethode, die Spektralnormalisierung [58], wird im U-Netz angewendet. Diese wird vor allem in GAN's zum Trainieren und Stabilisieren des Diskriminators eingesetzt. [55, 57, 59]

Die Upsampling-Schichten, die in Abb. 3.2 grün dargestellt sind, arbeiten mit einer Sub-Pixel-Faltung, wie in [60] beschrieben wird. Dabei werden den heruntergebrochenen Bildern in niedriger Auflösung nullwertige Pixel an den Außenseiten angehängt. Mithilfe dieses Paddings kann der Upsampling-Prozess durchgeführt werden. Eine Besonderheit des U-Nets ist die Self-Attention-Schicht, die bereits von ZHANG ET AL [61] in Bezug auf Self-Attention-GAN's (SAGAN's) beschrieben wird. Hierbei können spezielle Details in der Verarbeitung von Bilddateien aus allen Positionen der Feature-Maps generiert werden, nicht nur auf lokale Feature-Maps beschränkt. Desweiteren kann der Diskriminator des Netzwerkes überprüfen, ob hochdetaillierte Merkmale in entfernten Teilen des Bildes konsistent miteinander sind.

Die Ausgabe des Netzes erfolgt über die in rosa gefärbte (Abb. 3.2) Sigma-Schicht, die ihren Namen vom Sigma-Operator hat, welcher zur Rückskalierung der verarbeiteten Bildwerte in den normalisierten RGB-Farbraum nötig ist.

Das Netzwerk ist supervised mit ImageNet trainiert. Dabei ist der Trainingsprozess in zwei Phasen geteilt. Zum einen wird das Netzwerk mittels einer Optimierung der eigens für das Problem der Bildverarbeitung erstellten Verlustfunktion (loss function) "konventionell" trainiert. Anschließend wird ein binärer Klassifikator trainiert, um zu beurteilen, ob ein eingegebenes Bild mit dem Netzwerk koloriert, oder es bereits ein farbiges Bild (nicht gefärbt) ist. Zum anderen wird nach dem konventionellen ein GAN-Training angewendet, bei dem der obige binäre Klassifikator nun als Diskriminator und das Netzwerk zum Kolorieren als Generator eingesetzt wird. Antic hat nun herausgefunden, dass der nützliche Teil dieses GAN-Trainings nur in einem sehr kleinen Zeitfenster abläuft, bis zu einem gewissen "inflection point", bei dem der Diskriminator alle für den Generator nützlichen Informationen bereits übertragen hat. nach diesem Punkt liege kein produktives

Training des Modells mehr vor, sodass dieser Teil nach dem inflection point weggelassen werden kann und nur unnötige Zeit verschwendet. Laut dem Autor ist der schwierige Teil, genau diesen Punkt im Training zu finden, sodass der Prozess optimiert werden kann. Dies erreicht er durch setzen von gewissen Checkpoints und ständiges Prüfen der Ergebnisse, wann eine offensichtliche Falschfärbung (orangefarbene Haut, zu rote Lippen, etc.) vorliegt. Noch wurde keine Automatisierung dieses Prozesses erreicht, sodass dies ein Ziel für die Zukunft darstellt und eventuell mit der weiteren rasanten Evolution der KI möglich sein wird. Das Training der artistic-Version besteht aus einem 5-zyklischen NoGAN-Prozess mittels Diskriminator-Vor-Training und konventionellem GAN-Training. [2]

Die Verlustfunktion des Trainings ist eine abgeänderte Form des "perceptual loss", bei der die Besonderheit ist, dass bei der Übergabe der einzelnen Feature-Mappings das generierte Zwischenergebnis nicht exakt der Farbe, Textur und Form des gewünschten Ergebnisbildes entspricht, sondern nur grob der Struktur und des Inhaltes, was pauschal betrachtet ein Widerspruch darstellt, da Farbinformationen wichtig für den Kolorierungsprozess scheinen. Da das Kolorieren hauptsächlich über semantische Informationen der vorherigen Schichten geschieht, ist dieser Widerspruch nichtig. Diese Perceptual-Verlustfunktion geht auf JOHNSON ET AL. [62] zurück und nutzt ein VGG16-Netzwerk wie beschrieben in [54]. Bei Interesse kann der genaue Aufbau und die Arbeitsweise dort nachgelesen werden.

Die Bedeutung des ursprünglichen Projektes von Jason Antic ist in der Welt der Bildverarbeitung nicht zu unterschätzen. Er bedient sich bei alten bewährten Strukturen, die andere Autoren erfolgreich in der Vergangenheit entwickelt bzw. umgesetzt haben und schafft seinen eigenen Ansatz mittels der NoGAN-Idee. Er selbst erzählt in mehreren Interviews, dass diese Idee noch lange nicht ihr volles Potential ausschöpfen kann, da es noch kein automatisierter Prozess ist. Durch das manuelle Experimentieren und Finden des inflection points ist das Verfahren fehleranfällig und umständlicher als das Potential vermuten lässt. Die Verwendung von SAGAN's in Verbindung mit seinem NoGan-Training ist von Autoren oft zitiert und in ihre eigenen Projekte implementiert worden. Mit der Weiterentwicklung von DeOldify in den letzten Jahren ist er mit MyHeritage In Color™ an einem vielversprechenden Projekt beteiligt, welches erstaunliche Ergebnisse, wie in Abschnitt 4 diskutiert, in der Bildfärbung erzielt.

### 3.3. Implementation

Bevor die Colorizer implementiert und zum Laufen gebracht werden, müssen die Eingabedaten digitalisiert und vorbereitet werden. Die ursprünglichen Bilder stammen aus Dagmar Agsten-Aures' Buch von 1991 mit dem Namen "Leipzig - Gestern und Heute" [63]. In einer (aus diesem Buch) ausgewählten Sammlung von insgesamt 17 monochromen Bildern verschiedener Standorte Leipzigs basiert die Einschätzung der gefärbten Ergebnisse im Abschnitt 4. Die Daten wurden mithilfe der Scansoftware "Notebloc" so eingescannt, dass die Pixeldichte von 96 DPI ("dots per inch" oder Punkte pro Zoll) einheitlich auch bei unterschiedlichen Abmessungen bleibt, sodass die originalen und kolorierten Bilder miteinander verglichen werden können, ohne Skalieren zu müssen. Durch die Wahl einer hohen Auflösung (zwischen 1920x1080 und 2560x1440 Pixeln) wird die Fehleranfälligkeit und eventuelle Artefaktbildung beim späteren Kolorieren in der Regel reduziert. Eine

Umwandlung der Rohdaten aus dem RGB- in den YUV- oder L\*a\*b-Farbraum ist nicht nötig, da alle referenzierten Projekte den Farbraum der Inputdaten automatisch in ihren umwandeln. Alle Kolorierungsprozesse dieser Arbeit werden mit einer NVIDIA Geforce RTX 2080 SUPER Grafikkarte, einem 12th Gen. Intel Core i5-12600KF CPU mit 16GB DDR4-RAM internem Speicher vorgenommen.

#### 3.3.1. DeOldify und MyHeritage

DeOldify kann auf verschiedene Weisen implementiert und verwendet werden. Deoldify.NET [64] ist eine C#-Anwendung von ColorfulSoft, welches ermöglicht, die Standardversionen artistic und stable auf Windows 7-11 und Linux (getestet mit Mint) mithilfe des .NET-Frameworks über das referenzierte GitHub-Verzeichnis auf dem eigenen System zu bauen. Nach der Auswahl des gewünschten Modells, was sich in der Präzision der vortrainierten Gewichte (halbe oder einfache Präzision) und damit in der Dateigröße und im Leistungsverbrauch von anderen unterscheidet, kann nach dem Kompilieren mittels vorgegebenen Skriptes die Anwendung über eine .exe-Datei ausgeführt werden. Neben dieser und weiterer Möglichkeiten, wird in dieser Arbeit auf die Implementation mittels Anaconda auf Windows 11 zurückgegriffen.

```
1 git clone https://github.com/jantic/DeOldify.git DeOldify
2 cd DeOldify
3 conda env create -f environment.yml
```

Nach dem Klonen des Git-Verzeichnisses, welches Antic zum eigenen Implementieren und Experimentieren teilt, wird mittels GitBash (oder einer beliebigen Kommandozeile) eine Anaconda-Umgebung erstellt, mit der durch ein bereits erstelltes Kommando ein DeOldify-Jupyter-Notebook kreiert werden kann.

```
1 source activate deoldify
2 jupyter lab
```

Dieses Notebook kann man über den von Anaconda erstellten Link mittels Jupyter Lab ausführen. Nun ist es möglich, mit den verschiedenen Modellen DeOldify's in Python (hier Python 3) zu arbeiten. Es werden bereits vortrainierte Generator- und Diskriminatorgewichte angeboten, die man herunterladen kann, ohne selbst trainieren zu müssen. Nach dem Hochladen der Inputdaten wird das gewünschte Modell gewählt (hier artistic, welches wie schon erwähnt etwas ressourcenintensiver arbeitet als die stable Version, aber bessere Ergebnisse liefern soll). Neben dem Ausgabeordner ist der Renderfaktor (RF) anzugeben. Ein hoher Renderfaktor (zwischen 10 und 40 sind möglich) ermöglicht stabilere Kolorierungs-Entscheidungen des Modells, da mit einer höheren Auflösung gearbeitet wird. Vorausgesetzt wird hierbei jedoch eine gute bis sehr gute Grafikkarte, um den Renderprozess zu stämmen. Der Autor empfiehlt hierbei einen RF von 35, wenn es das System bewerkstelligen kann.

Die neueste Variante von DeOldify ist auf myheritage.com zu finden und wird mit "MyHeritage in Color™" vermarktet. Das verwendete Modell ist seit Juli 2021 implementiert und der Standard, um monochrome Bilder auf dieser Seite zu kolorieren. Zu dem Modell, was auf dem ursprünglichen

DeOldify-Modell von Jason Antic basiert, gibt es aber keinerlei Informationen, welche Änderungen seit dem kostenlosen Projekt vorgenommen wurden. Genaue Aussagen zur benutzten Architektur und zum Aufbau des Trainings können also nicht getätigt werden, ohne Vermutungen anzustellen. Das liegt vor allem daran, dass sich hinter MyHeritage ein kommerzielles Projekt befindet, welches ein jährliches Abonnement voraussetzt, um die neueste Technologie zu verwenden. Zum Kolorieren der Bilder Leipzigs wird hier die 14-tägige Testphase verwendet.

#### 3.3.2. Colorful Image Colorization (2016)

Ähnlich wie DeOldify wird hier mithilfe von Anaconda eine Python-Umgebung kreiert, mit der der Ursprungscode von ZHANG ET AL., welcher auf der offiziellen GitHub-Seite [65] zu finden ist, zum Laufen gebracht werden kann. Nach der Installation der vorausgesetzten Software, kann der Kolorierer mit dem zu färbenden Bild ausgeführt werden.

```
1 python demo_release.py -i imgs/barfussgaesschen1983.jpg
```

Dabei ist es wichtig zu erwähnen, dass bei dieser Demo auch das neuere Projekt aus 2017 eingebaut ist, welches auf der sogenannten "Real-time user-guided image colorization with learned deep priors" [66] basiert. Dieses baut stark auf dem 2016er Projekt auf und ist damit von der verwendeten Architektur sehr ähnlich zum älteren Modell.

#### 3.3.3. Hotpot.ai und VanceAI

Da diese beiden Projekte lediglich online über hotpot.ai und vanceai.com oder als fertige Desktopanwendung verfügbar sind und keinerlei Code veröffentlicht ist, ist eine klassische Implementation nicht möglich. Sie dienen neben den anderen Verfahren als benutzerfreundliche Online-Alternativen, die jedoch beide nicht unbegrenzt kostenlos nutzbar sind.



## 4. Evaluation

### 4.1. Technische Auswertung

Bevor die Ergebnisse des Kolorierens vorgestellt werden, wird die Zugänglichkeit der benutzten Verfahren kurz erläutert und auf deren technischen Möglichkeiten eingegangen. Die eigens implementierten DeOldify (artistic) und Zhang's CIC sind die einzigen unbegrenzt kostenfreien Methoden, die hier verwendet werden, wenn auch bei CIC nur die Demo verfügbar ist.

Bei Hotpot.ai gibt es sogenannte Credits, die man für Geld kaufen muss, wenn man Ausgabebilder in höherer Auflösung möchte. Hierbei liegen die Kosten (Stand Juli 2023) für 1000 Credits bei 12\$ als einmalige Zahlung, oder 10\$ monatlich (ebenfalls für 1000 Credits). Für das Kolorieren eines Bildes in der vollen Auflösung werden jeweils 25 Credits gebraucht, sodass man für 12\$ insgesamt 40 Bilder färben kann. Für andere Features von Hotpot.ai werden ebenfalls Credits benötigt. Der "Colorization Factor" (CF), der mit dem Renderfaktor von DeOldify zu vergleichen ist, ist hier von 12 bis 25 verfügbar. Da das Maximum hier CF 25 ist, können die Eingaben mit einer geringeren Auswahl von unterschiedlichen Pixeldichten koloriert werden, als es bei den anderen möglich ist. Jedoch läuft der Prozess sehr schnell ab und die Ergebnisse sind bei einem CF von 25 mit den anderen Verfahren relativ gut (in niedrigen Auflösungen) zu vergleichen. Hier muss zusätzlich angemerkt werden, dass es auf Hotpot.ai häufiger vorkommt, dass durch zu viel "Traffic" auf der Seite ein Färben der Bilder nicht möglich ist. Die Nutzer, die Credits für ihre Prozesse verwenden, werden hierbei bevorzugt und kostenlose Nutzer müssen abwarten. Das ist ein großer Nachteil gegenüber der selbst implementierten Verfahren, da diese jederzeit zugänglich sind.

Bei VanceAI wird es ähnlich gehandhabt. Während man hier theoretisch eine unbegrenzte Anzahl an Fotos kolorieren kann, wird für das Herunterladen des Ergebnisses jeweils ein Credit abgezogen. Wenn man sich auf der Webseite anmeldet, hat jeder kostenfreie (nicht-Premium) Nutzer drei Credits pro Tag zum Verwenden der Software. Man kann diese Einschränkung teilweise umgehen, indem man die Ergebnisse als Screenshot speichert, jedoch ist die Auflösung deutlich geringer als beim "richtigen" Nutzen der Software. Die Kosten liegen hier bei 24,99€ (Juli 2023) für 100 Credits, wobei jedes heruntergeladene Bild einen Credit kostet. Verglichen mit Hotpot.ai hat man ca. 10 Bilder mehr, die man mit dieser Software für 11€ bis 12€ (Schwankungen beim Dollar-Euro-Kurs sind zu beachten) färben kann. Ein selbst zu bestimmender Renderingfaktor ist ebenfalls vorhanden, der sogar bis RF 70 einstellbar ist.

MyHeritage ist in der Gruppe der benutzten Programme das teuerste. Es wird eine 14-tägige Testphase angeboten, danach muss man jedoch die kostenintensivste Stufe der angebotenen Abonnements auswählen, um unbegrenzt Fotos kolorieren zu können. Dieses Komplett-Abo liegt hierbei bei einem jährlichen Betrag von ca. 160€ (Stand Juli 2023). Grund dafür ist, dass MyHeritage's primäre Funktion das Erstellen eines Stammbaumes ist und man durch eine riesige Datenbank herausfinden kann, wer seine Vorfahren/Verwandten sind. Die Funktion MyHeritage In Color™ ist hierbei nur ein Zusatz, der ordentlich bezahlt werden will.

Tabelle 4.1.: Grundlegende Eigenschaften der Verfahren

Kolorierverfahren	Prozesszeit ( $\bar{x}$ in Sekunden)	RF oder CF	komplett kostenlos
CIC von Zhang et al.	8,2	N/A	ja (mit Implementation)
DeOldify (artistic)	9,8	10 - 40	ja (mit Implementation)
Hotpot.ai	2,2	12 - 25	nein (nur geringe Auflösung)
VanceAI	11,8	10 - 70	nein (nur 3 Fotos)
MyHeritage In Color™	5,3	16 - 96	nein (nur 1 Foto pro Tag)

Wie in Tab. 4.1 zu sehen ist, unterscheiden sich die verschiedenen Verfahren bereits in der Zeit, in der die Kolorierung vorgenommen wird. Dabei wurden alle ausgewählten Bilder des Datensatzes einmal (mit RF oder CF 25) gefärbt, um in etwa gleiche Bedingungen zu schaffen. Der Mittelwert der gemessenen Zeiten ist in der Tabelle abzulesen. Auffällig hierbei ist vor allem Hotpot.ai, das im Schnitt zum Kolorieren nur 2,2 Sekunden benötigt mit Abstand das schnellste der getesteten Kolorierverfahren ist. Jedoch werden hier die Bilder lediglich mit einer Auflösung von 512 x 386 ausgegeben (weniger als 480p), was auf Mobilgeräten zum schnellen Färben vertretbar, jedoch für ernsthaftes und professionelleres Kolorieren nicht relevant ist. Die Schnelligkeit ist hier somit der eingeschränkten Skalierbarkeit zu verdanken. Zeitlich folgt darauf der Kolorierer von MyHeritage In Color™, der auch bei hohen RF's ein schnelles, dennoch hochaufgelöstes Ergebnis liefert. Die Prozesszeiten von den implementierten Verfahren ist höher, was vor allem abhängig von dem System ist, auf welchem sie ausgeführt werden. Die hier verwendeten Spezifikationen (siehe 3.3) sind dabei zum Zeitpunkt dieser Arbeit im überdurchschnittlichen Bereich, sodass es auf älteren Maschinen sicherlich länger dauern kann, Bilder mit diesen Verfahren zu färben. Hier ist ein Vorteil in den online verfügbaren Methoden zu sehen, da diese meist über einen externen Server (nicht lokal) laufen und deren Performance nicht abhängig vom eigenen System ist.

Da die Statistiken des Trainingsprozesses der kostenpflichtigen Methoden nicht verfügbar sind, kann hier nur auf die implementierten Projekte eingegangen werden. Da aber sowohl Hotpot.ai, VanceAI als auch MyHeritage In Color™ auf Teilen von DeOldify basieren, kann nur vermutet werden, dass sie ähnliche Ergebnisse im Training ihrer Modelle erzielen.

Method	Imagenet ctest10k			COCOSTuff validation split			Places205 validation split		
	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑
<i>lizuka et al.</i>	0.200	23.636	0.917	0.185	23.863	0.922	0.146	25.581	0.950
<i>Larsson et al.</i>	0.188	25.107	0.927	0.183	25.061	0.930	0.161	25.722	0.951
<i>Zhang et al. (2016)</i>	0.238	21.791	0.892	0.234	21.838	0.895	0.205	22.581	0.921
<i>Zhang et al. (2017)</i>	0.145	26.166	0.932	0.138	26.823	0.937	0.149	25.823	0.948
<i>Deoldify</i>	0.187	23.537	0.914	0.180	23.692	0.920	0.161	23.983	0.939

Abbildung 4.1.: Quantitativer Vergleich von DeOldify und Zhang's CIC (2016) auf ImageNet, COCOStuff und Places205 aus "Instance-aware image colorization" [67]

Der in Abb. 4.1 dargestellte Vergleich zeigt die erreichten LPIPS-, PSNR- und SSIM-Trainingswerte auf verschiedenen Datensätzen. LPIPS oder "Learned Perceptual Image Patch Singularity" ist eine Metrik zur Messung der strukturellen Ähnlichkeit hochdimensionaler Bilder, die sich interne Aktivierungen von tiefen CNN's zu Nutze macht. Ein niedriger LPIPS-Wert bedeutet hierbei eine hohe wahrnehmungsbezogene Ähnlichkeit zwischen zwei Bildern oder Bildabschnitten. Das PSNR oder

"Peak Signal-to-Noise Ratio" beschreibt das Verhältnis zwischen der maximal möglichen Leistung eines Signals und der Leistung des störenden Rauschens, das die Treue der Wiedergabe beeinflusst. Es bestimmt somit die Bildkompressionsqualität in Dezibel. Je höher das PSNR ist, desto höher ist die Qualität der komprimierten oder rekonstruierten Eingabe. SSIM steht für "Structural Similarity Index" und dient ebenfalls zur Messung der Ähnlichkeit zwischen Bildern mittels Extrahieren der Eigenschaften Leuchtdichte, Kontrast und Struktur. Der Unterschied zu PSNR ist hier, dass absolute Fehler geschätzt werden. [68, 69, 70, 71]

Von den Autoren aus [67] wird von ImageNet die ursprüngliche Trainingsaufteilung (1,3 Millionen Bilder) für das Training aller Modelle verwendet und die Testaufteilung (ctest10k) mit 10.000 Bildern zur Bewertung benutzt. Im Gegensatz zu den objektzentrierten Bildern im ImageNet-Datensatz enthält der COCO-Stuff-Datensatz (ca. 118.000 Bilder) eine Vielzahl von natürlichen Szenen mit mehreren Objekten im Bild. Die Autoren verwenden die 5.000 Bilder im ursprünglichen Validierungsset zur Bewertung. Desweiteren werden 20.500 Testbilder aus 205 Kategorien aus dem Places205-Datensatz zur Evaluation, jedoch nur für die Übertragbarkeit, verwendet. Vom Trainingsset und den Szenenkategorie-Labels des Datensatzes wird von den Autoren nicht Gebrauch gemacht. [67]

In Abb. 4.1 ist hierbei zu sehen, dass DeOldify bessere Ergebnisse auf dem Places205-Datensatz als auf den anderen erzielt, sodass es Landschaften und Gebäude im allgemeinen vermeintlich präziser färbt als Personen oder kleinere Objekte. In der Tat kann man eventuell so die hervorstechende Färbung der Person in der Bildmitte von Abb. 4.2 (2) erklären, die anders als der restliche Teil des Bildes stark gefärbt in den Vordergrund tritt. Desweiteren performt DeOldify in allen aufgezeigten Metriken besser als CIC, wobei es wiederum vom 2017er Modell übertroffen wird, welches laut dieser Tests am besten von allen untersuchten Methoden in fast allen Metriken abschneidet. Bevor es in die visuelle Auswertung geht, soll noch gesagt werden, dass die Eingabedaten hauptsächlich Gebäude, Objekte, Pflanzen abbilden, Personen hingegen sind nur in weiter Entfernung zu betrachten. Deshalb sind die erreichten Trainingswerte am Places205-Datensatz wohlmöglichst die am ehesten für diese Arbeit relevantesten Werte von den getesteten Datensätzen.

### 4.2. Visuelle Auswertung

Nachdem nun die technischen Feinheiten abgearbeitet sind, werden einigen Orten des historischen Leipzigs mit Farbe Leben eingehaucht. In Abb. 4.2 sieht man einen ersten Vergleich am gefärbten Barfussgässchen aus dem Jahr 1983. Als Erstes fällt die Farbverteilung und -stärke der einzelnen Methoden auf. Während bei dem ursprünglichen DeOldify-Modell (2) eine sachte, leichte Färbung des in der Mitte gelegenen Gebäudes und der Straße zu sehen ist, interpretieren (3), (4) und (5) eine rot-bräunliche Färbung des Daches und Teilen der Fassade. Von allen benutzten Verfahren werden die Bäume als solche erkannt und dementsprechend gefärbt. Sehr auffällig ist bei (2) die starke Rotfärbung des Rockes der Dame im Vordergrund, als ob sie vom Modell als Hauptaugenmerk ausgewählt wird. Ob es sich dabei um eine Fehlinterpretation des Modells als kupfernen Gegenstand oder der Rock wirklich als Kleidungsstück erkannt und somit beliebig gefärbt wird, kann mit Sicherheit nicht gesagt werden. Als kurze Zusatzinformation ist zu erwähnen, dass der hier benutzte Renderfaktor 26 durch selbstständiges Experimentieren gefunden und als geeignetste Variante

gewählt wird. In (4) und vor allem in (5) wird der Rock in einem dunklen Blauton wie fast alle anderen Kleidungsstücke im Bild gefärbt. Dabei gilt es nur für den Betrachter zu bewerten, welche Färbung nun die plausible ist.

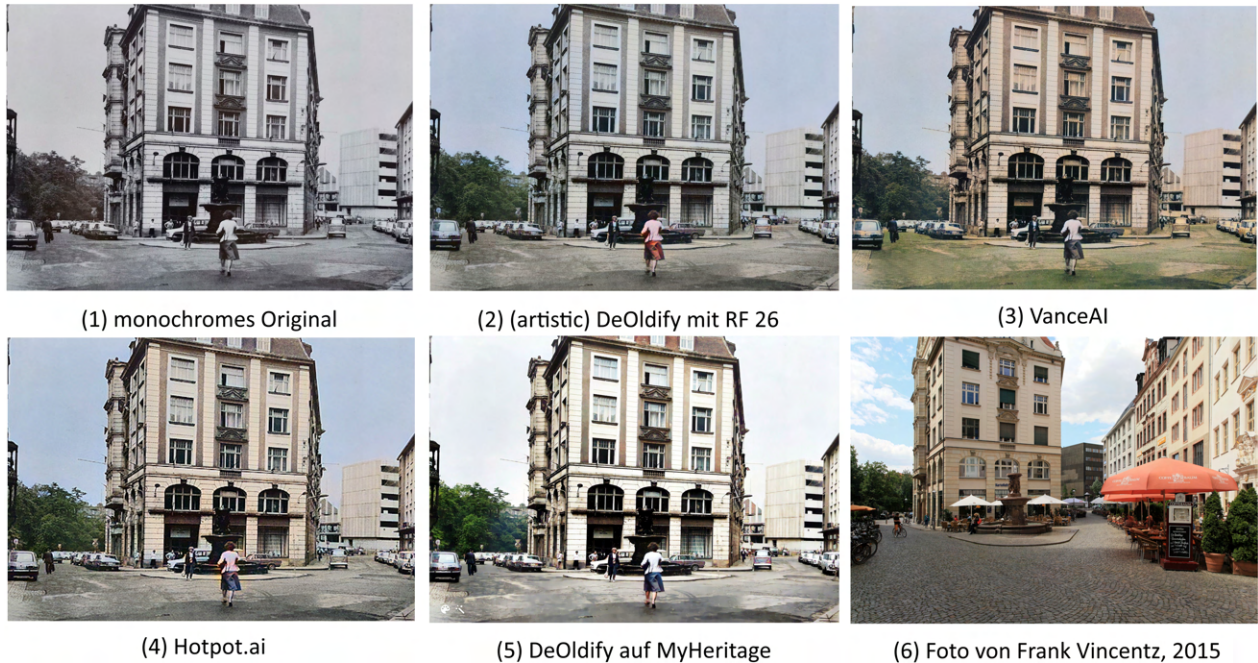


Abbildung 4.2.: Barfußgässchen (1983) (1) aus "Leipzig - Gestern und Heute" [63] koloriert mit verschiedenen Projekten und (6) als geschossenes Foto

Eine vermeintliche Falschfärbung ist in (3) bei der Farbgebung der Straße zu beobachten. Das Modell hat den Asphalt als Gras- bzw. Rasenfläche erkannt und dementsprechend grün eingefärbt. Hierzu muss gesagt werden, dass VanceAI durchgängig zufriedenstellende Resultate liefert. Dieses ist jedoch eine der wenigen unplausiblen Färbungen, die hier absichtlich gewählt wurde. Auf mögliche Fehlerursachen wird etwas später in diesem Abschnitt näher eingegangen. Eine weitere Auffälligkeit des Asphalts wird beim genauen Beobachten von (5) deutlich. Das Eingabebild wird hier bereits ohne Nutzerinput von MyHeritage In Color™ bearbeitet, sodass der Boden geglättet scheint und deutlicher nach Asphalt oder Beton aussieht. Der Himmel ist bei allen Verfahren mit leichtem Blauton versehen, lediglich in (5) ist er relativ farblos dargestellt. Wiederum kann hier nicht mit Sicherheit beurteilt werden, welche Farbwahl der Modelle die realistischere ist. Da auf dem Originalbild nicht zu erkennen ist, ob der Himmel wolkenlos oder komplett bewölkt ist, sind alle gewählten Varianten plausibel.

Als Letztes soll in diesem Vergleich auf das leichte Artefakt des monochromen Originals (1), was wahrscheinlich beim Digitalisierungsprozess geschehen ist, Bezug genommen werden. An der linken, unteren Seite des zentralen Gebäudes ist ein Teil der Baumreihe etwas heller geraten. Dieses Artefakt ist deutlich in (3) und (5) zu sehen, wohingegen (2) und (4) weniger markante Kontrastunterschiede in der Baumpartie aufweisen. Zum teilweisen Vergleich ist ein von Frank Vincentz geschossenes Foto aus dem Jahr 2015 dargestellt. Teilweise muss hierbei gesagt werden, da im Laufe der Zeit (32 Jahre Unterschied) sowohl die Straße gepflastert und die Gebäude in Teilen restauriert wurden, sodass man die kolorierten Ergebnisse nicht mit 100%iger Sicherheit mit dem selbst aufgenommenen Foto gegenüberstellen kann, obwohl die Perspektive sehr ähnelt.



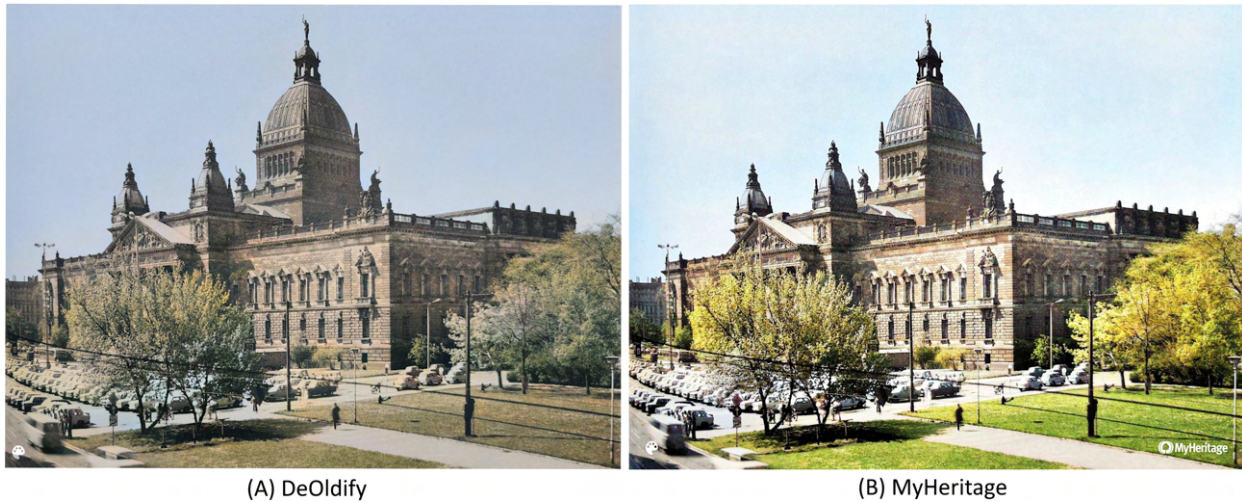


Abbildung 4.3.: Dimitroffmuseum (1988) koloriert mit DeOldify (links, A) und MyHeritage (rechts, B)

Um den Unterschied der originalen (artistic) Version von DeOldify mit der neuesten, ausgereiftesten und kommerziell vermarkteten von MyHeritage In Color™ zu verdeutlichen, wird hier das alte Gebäude des Dimitroffmuseums von 1988 (Abb. 4.3) gefärbt, was heute das Bundesverwaltungsgericht beherbergt. Dabei fällt der Kontrast- und Helligkeitsunterschied der beiden Ergebnisse als Erstes auf. Bei (A), welches mit RF 40 koloriert wurde, liegt ein matter, sandiger Ton auf dem Bild. Die abgebildeten Bäume deuten eine leichte Grünfärbung an, jedoch verläuft die Farbgebung auf der linken Seite des Bildes mit der geparkten Autos. Daraus folgt, dass die Struktur des Baumes nicht komplett von den dahinterliegenden Autos abzugrenzen ist. In (B) ist der Übergang zwischen Autos und Bäumen besser zu erkennen, gerade auch weil die leuchtende Grünfärbung heraussticht und somit Übergänge zwischen verschiedenen Objekten deutlich besser zu sehen sind. Diese sind vor allem zwischen Dach und Fassade sehr gut zu erkennen. Ebenfalls interessant bei diesem Vergleich ist die Interpretation des Himmels. Während bei (A) eine durchgehend gleichmäßige Blaufärbung erkennbar ist, färbt (B) scheinbar in einem helleren (Himmelblau) ungleichmäßig mit vereinzelt kleinen hellen Stellen, sodass der Anschein erweckt wird, es seien Wolken vorhanden. Dies scheint eine "künstlerische Interpretation" des neuen Modells zu sein, da im originalen schwarz-weiß Foto ähnlich wie in Abb. 4.2 nicht zu erkennen ist, ob Wolken vorhanden sind. Erstaunlicherweise ist in beiden Ergebnissen die sandsteinartige Färbung der Fassade vorzufinden, welche dem gegenwärtigen Gebäude entspricht. Ob das Gebäude jedoch im Jahr 1988 auch diesen Sandstein-Look hatte, ist mit Sicherheit nicht zu sagen. Plausibel ist die Färbung des Museums aber in beiden Fällen. Auch in der Deutlichkeit der Strommasten, -kabel und Laternen sind beide konsistent.

Im Allgemeinen wirkt (B) heller, freundlicher und lebendiger als (A), was aber (neben den Unterschieden im Koloriermodell) hauptsächlich am verwendeten RF 40 von DeOldify liegt. Bei niedrigeren Renderfaktoren sind schnellere Ergebnisse zu erwarten und die Farben sind deutlich leuchtender als bei hohen, jedoch sind die Ergebnisse inkonsistenter und mit mehr Artefakten bestückt. Um die Unterschiede zwischen einzelnen Renderfaktoren im selbstimplementierten artistic-Modell von DeOldify zu zeigen, werden in Abb 4.4 einige gegenübergestellt.

Die größten Unterschiede zwischen den Bildern sieht man im Wesentlichen an den Ecken von Objekten. Nimmt man beispielsweise den vordersten, weiß gefärbten PKW ins Auge, so sieht man eine



Abbildung 4.4.: Renderfaktoren von links nach rechts: 10, 19, 28 und 37 von DeOldify, kolorierte Schillerstraße (1984)

sehr unsaubere, verwaschene Färbung mit RF 10 (links außen), wohingegen mit RF 37 (rechts außen) die Kanten sehr klar zu erkennen und deutlich sauberer koloriert sind. Auch in den schmalen Fensterbögen am rechten Bildrand ist die Genauigkeit der Färbung bei hohen RF's sehr gut zu sehen. Generell wirkt es bei RF 10 so, als würde sich die Grünfärbung der Bäume bis über die Bildmitte von links nach rechts ziehen. Man kann leichte grüne Schimmer in den abgebildeten Personen und im Turm des neuen Rathauses (Bildmitte) sehen. Diese Ungenauigkeiten treten bei niedrigeren RF's vorrangig in Sektionen auf, wo der Kontrast zwischen Hintergrund und beobachteten Objekt nur gering ausgeprägt ist, sodass das Modell die Objektübergänge vermischt.

Generell muss gesagt werden, dass Fehler in den Färbungen vor allem dann vorkommen, wenn bereits Artefakte durch einen fehlerhaften Digitalisierungsprozess der Eingaben geschehen, die Auflösung des zu färbenden Bildmaterials zu gering ist oder der Renderfaktor zu niedrig bzw. zu hoch gewählt wurde. Letzteres erzeugt eine Überkolorierung des Bildes, da Pixel oder Pixelmengen unnötigerweise koloriert werden, nachdem sie vom Modell falsch oder als eigenes Objekt interpretiert worden sind. Natürlich ist die Qualität des Eingabebildes ebenfalls von großer Bedeutung, jedoch sind die hier ausgewählten Quelldaten aus "Leipzig - Gestern und Heute" hochwertig und sehr gut für den Färbeprozess geeignet. Einige Fotos in diesem Buch sind nicht schwarz-weiß, sondern in einem leichten Sepia-Look gedruckt oder durch den Digitalisierungsprozess entstanden. Das Kolorieren dieser Bilder hat in den meisten Fällen keine zufriedenstellenden Ergebnisse mit den benutzten Methoden erzeugt, dennoch sind diese auch im digital angehängten GitHub-Verzeichnis [72] dieser Arbeit und somit im Quelldatensatz vorhanden, um auch fehlerhafte Kolorierungen zur Schau zu stellen.

Im Großen und Ganzen generieren alle verwendeten Methoden befriedigende Ergebnisse, wobei nach technischer und visueller Auswertung MyHeritage In Color™ und VanceAI am meisten überzeugen konnten, zumindest, wenn man kontrastreiche und farbenfrohe Bilder bevorzugt. Wenn es rein um die Fähigkeit des Kolorierens alter Bilder geht und die finanziellen Kosten irrelevant sind, liegen diese beiden Verfahren vor den anderen, da sie ansehnliche und plausible Resultate in nur wenigen Sekunden ohne großen Aufwand liefern. Durch das unkomplizierte Hochladen seiner Eingabedatei und das schnelle Bearbeiten dieser, sind die Onlineanbieter (Hotpot.ai, VanceAI und MyHeritage)

generell die effektiveren Kolorierer. Jedoch muss man ganz klar festhalten, dass die Kosten zur Nutzung dieser Verfahren nicht zu unterschlagen sind. Hotpot.ai ist hierbei mit einmalig 12\$ die günstigste Variante, während der Preis vom MyHeritage-Abonnement allein für das Kolorieren in keinsten Weise zu rechtfertigen ist. Hier ist wirklich nur die 14-tägige Testversion zu empfehlen. Wer diese bereits genutzt hat, sollte auf einen anderen Kolorierer zurückgreifen. Somit ist als bestes Preis-Leistungs-Verhältnis der hier getesteten, kostenpflichtigen Anbieter VanceAI hervorzuheben, da es für einen einmaligen, im Rahmen liegenden Preis ermöglicht, bis zu 100 Bilder zu bearbeiten oder zu färben. Die Ergebnisse, die durch das Erstellen mehrerer Accounts und das Nutzen der 3 freien Credits pro Account entstanden sind, weisen eine äußerst zufriedenstellende Färbung auf. Bei Hotpot.ai scheint es, als würden die Autoren des Programms eine sehr kommerziell getriebene Fahrtrichtung bezüglich der Nutzung dieses Verfahrens vorgeben. Sowohl der Kolorierer als auch die anderen KI-Tools verbrauchen bei allen Optionen, die ansatzweise in ordentlicher Auflösung zu verwerten sind, Credits und damit Geld. Die anderen kostenpflichtigen Vertreter bieten mit ihrem begrenzten kostenlosen Angebot dennoch Programme, mit denen hochqualitative Resultate erzeugt werden können. Wenn man kein Geld für das Kolorieren ausgeben, etwas Eigenes kreieren oder bereits Bestehendes modifizieren möchte, so sind die in dieser Arbeit implementierten Modelle eine gute Alternative, obwohl Zhang's CIC mittlerweile ein sehr in die Jahre gekommenes Projekt ist und eher auf das neuere Modell zurückgegriffen werden sollte.

Wie schon in Abschnitt 3.2.1 angerissen, gilt DeOldify in der digitalen Bildkolorierung als ein bedeutendes Projekt, da Antic mit seinen Ideen neue Perspektiven für Forscher und Hobbyentwickler bietet. VanceAI und Hotpot.ai sind bei Weitem nicht die einzigen Programme, deren Architektur auf einer alten Iteration von DeOldify basieren. Viele neuartige KI-Tools nutzen einen Kolorierer, der auf DeOldify zurückzuführen ist.

## 5. Fazit und Ausblick

Mit dem fast schon überwältigendem Angebot an Software, das mittlerweile in den Weiten des Internets präsentiert wird, ist es nicht leicht, das richtige Tool für seine Bedürfnisse zu wählen. Wie schon in der Einleitung dieser Arbeit angedeutet, ist es mittlerweile eine Frage der geeigneten Auswahl der Methode, mit welcher man Bilder nach eigenem Ermessen am besten färben kann. So ist diese Arbeit einerseits als Stütze für Unschlüssige, die einen Einblick in bestehende Verfahren zur digitalen Kolorierung mittels Deep Learning erhaschen wollen, zu sehen. Andererseits werden hier mit diesen Verfahren gefärbte, historische Bilder Leipzigs zur Verfügung gestellt, um das "alte Leipzig" in neuer Farbe zu betrachten. Dabei wurden von den selbst implementierten Methoden DeOldify und ZHANG ET AL.'S "Colorful Image Colorization" alle 17 Bilder des Eingabedatensatzes und ausgewählte Fotos mit den teilweise kostenintensiven Vertretern Hotpot.ai, VanceAI und MyHeritage In Color™ koloriert und zur Verfügung gestellt. Jedoch wurden hier jeweils die kostenlosen Angebote der webbasierten Verfahren benutzt und bewertet, sodass im Zuge dieser Arbeit kein Geld für das Kolorieren der Bilder ausgegeben wurde. Alle Ergebnisse sind online über das referenzierte GitHub-Verzeichnis [72] und im digitalen Anhang dieser Arbeit zur Verfügung gestellt.

Nach dem Prozess des Färbens wurde erschlossen, dass MyHeritage (bezogen auf die Qualität der Resultate) die zufriedenstellendsten Bilder für Leute, die eine starke Sättigung und hohen Farbenreichtum schätzen, erzeugt. Seine schnelle und unkomplizierte Arbeitsweise, die auch auf die anderen Onlineanbieter bezogen werden kann, sorgt für eine effektive Färbung sogar von mehreren Bildern zur gleichen Zeit. Die enormen Kosten, die durch das Angebot etlicher weiterer Features der Webseite entstehen, sind für das Kolorieren im kleinen Rahmen aber nicht zu rechtfertigen. Selbst für das professionelle Färben sollten andere Verfahren bevorzugt werden, wenn die 14-tägige Testphase beendet ist. Zum Beispiel ist hier VanceAI eine gute zweite Wahl, das von den kostenpflichtigen Projekten ganz klar vor Hotpot.ai liegt. Da die kostenfreien Optionen von letzterem in keinsten Weise an die untersuchten Konkurrenten heranreichen, ist Hotpot.ai nur zum schnellen Kolorieren auf mobilen Endgeräten zu empfehlen.

Von den implementierten Modellen gelingt es DeOldify, die plausibelsten Ergebnisse zu erzeugen. Während es ein wenig an leuchtenden Farben mangelt, werden die meisten Bereiche der Schwarz-Weiß-Bilder (mit der korrekten Auswahl des Renderfaktors) befriedigend koloriert. Dadurch offenbart sich DeOldify als echte Alternative zu den neueren Onlineangeboten, vor allem, wenn man sich selbst mit dem Training des Modells beschäftigen oder einfach mit verschiedenen Versionen des Codes experimentieren möchte. Das besondere Training des Modells, was Antic selbst als NoGAN bezeichnet, verbindet klassisches GAN-Training mit reduziertem Training mittels manuellem Findens eines sogenannten "inflection points", nach welchem das Training nicht mehr spürbar besser im Verhältnis zur verstrichenen Zeit wird. Dieses NoGAN in Verbindung mit SAGAN-Strukturen bilden die Grundpfeiler der einzigartigen Architektur von Antic's Projekt, wodurch dessen Bedeutung in der digitalen Kolorierung beachtlich ist. Viele nachfolgende Projekte stützen sich auf die frühe Iteration des DeOldify-Algorithmus. Von daher ist es keine Überraschung, dass Hotpot.ai und VanceAI ebenfalls darauf basieren. Die neueste und ausgeklügeltste Iteration ist im MyHeritage-Modell seit Juli 2021 implementiert. Allein diese Tatsache lässt das Potential erahnen, das in Zukunft erreicht werden könnte, da innerhalb von 3 Jahren das anfängliche DeOldify-Modell in ein



kommerziell vermarktbares und scheinbar sehr erfolgreiches verwandelt wurde. Sonst würde der hohe Preis für das Abonnement in diesem Maße nicht vertretbar sein, auch wenn der Kolorierer nur ein Bonus-Feature der Marke MyHeritage ist.

Im Allgemeinen wirkt der ständige Drang von Interessierten nach immer besseren Ergebnissen positiv auf das Entstehen neuer Projekte im Bereich der digitalen Bildbe- und verarbeitung. Allein mit der rapiden Entwicklung neuer GPT's in den letzten Jahren, wird der Einfluss von KI auf den Alltag immer größer und mehr Prozesse werden automatisiert, wobei in vielen Gebieten eine Vollautomatisierung angestrebt oder bereits erreicht wird. Ähnlich wie ein Generator und Diskriminator eines GANs müssen Menschen mittlerweile hinterfragen, welche Bilder und Videos handgemacht oder von KI kreiert wurden. Die Faszination nach neuen Erkenntnissen und Errungenschaften motiviert, immer weiter zu forschen und die Grenzen weiter zu treiben. In Bezug auf das digitale Kolorieren ist das Ziel, bessere und schnellere Methoden als die bestehenden zu entwickeln, sodass die Metriken in Richtung Perfektion wandern. Wünschenswert wäre ein exzellenter Kolorierer, der für die breite Masse zugänglich ist und plausible Ergebnisse generiert, ohne ein kleines Vermögen zu kosten.

## Literatur

- [1] Patsorn Sangkloy u. a. „Scribbler: Controlling deep image synthesis with sketch and color“. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, S. 5400–5409.
- [2] Jason Antic. „Deoldify.(2019)“. In: *GitHub: github.com/jantic/DeOldify* (2019).
- [3] Wolfgang Lefèvre. „Inside the camera obscura–Optics and art under the spell of the projected image“. In: (2007).
- [4] Allan A Mills. „Vermeer and the camera obscura: Some practical considerations“. In: *Leonardo* 31.3 (1998), S. 213–218.
- [5] Josef Maria Eder. *Ausführliches handbuch der photographie*. Bd. 1. W. Knapp, 1884.
- [6] M Susan Barger und William B White. *The Daguerreotype: nineteenth-century technology and modern science*. JHU Press, 2000.
- [7] Bertrand Lavédrine u. a. *Photographs of the past: process and preservation*. Getty Publications, 2009.
- [8] Juliet Hacking. *Foto: Fotografie; die ganze Geschichte*. DuMont-Verlag, 2012.
- [9] Joseph Solomon Friedman. *History of color photography*. Joseph Solomon Friedman, 1947.
- [10] Qing Luan u. a. „Natural image colorization“. In: *Proceedings of the 18th Eurographics conference on Rendering Techniques*. 2007, S. 309–320.
- [11] John Hannavy. *Encyclopedia of nineteenth-century photography*. Routledge, 2013.
- [12] Heinz K Henisch und Bridget Ann Henisch. *The painted photograph, 1839-1914: origins, techniques, aspirations*. Penn State University Press, 1996, S. 21.
- [13] S Ferguson. „In living color: Process and materials of the hand-colored daguerreotype“. In: *The Daguerreian Annual* (2008), S. 13–18.
- [14] John Towler. *The Silver Sunbeam: A Practical and Theoretical Text-book on Sun Drawing and Photographic Printing*. E. & HT Anthony, 1873.
- [15] Michael Ivankovich und Susan Ivankovich. *Early Twentieth Century Hand-painted Photography: Identification & Values*. Collector Books, 2005.
- [16] Jonathan Hoppe. „Spectroscopic analysis of hand-colored photographs and photographic hand-coloring materials“. Diss. University of Delaware, 2010.
- [17] Rasool Fakoor u. a. „Using deep learning to enhance cancer diagnosis and classification“. In: *Proceedings of the international conference on machine learning*. Bd. 28. ACM, New York, USA. 2013, S. 3937–3949.
- [18] Holger R Roth u. a. „Improving computer-aided detection using convolutional neural networks and random view aggregation“. In: *IEEE transactions on medical imaging* 35.5 (2015), S. 1170–1181.
- [19] Yann LeCun, Yoshua Bengio und Geoffrey Hinton. „Deep learning“. In: *nature* 521.7553 (2015), S. 436–444.

- [20] Marc Peter Deisenroth, A Aldo Faisal und Cheng Soon Ong. *Mathematics for machine learning*. Cambridge University Press, 2020.
- [21] Mohssen Mohammed, Muhammad Badruddin Khan und Eihab Bashier Mohammed Bashier. *Machine learning: algorithms and applications*. Crc Press, 2016.
- [22] Shai Shalev-Shwartz und Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.
- [23] John Paul Mueller und Luca Massaron. *Machine learning for dummies*. John Wiley & Sons, 2021.
- [24] Ian Goodfellow u. a. „Generative adversarial networks“. In: *Communications of the ACM* 63.11 (2020), S. 139–144.
- [25] Kilian Semmelmann. *Der Clustering Guide: Definition, Methoden und Beispiele*. Kobold AI, 2022.
- [26] Christian Janiesch, Patrick Zschech und Kai Heinrich. „Machine learning and deep learning“. In: *Electronic Markets* 31.3 (2021), S. 685–695.
- [27] David Silver u. a. „A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play“. In: *Science* 362.6419 (2018), S. 1140–1144.
- [28] Markus Peters u. a. „A reinforcement learning approach to autonomous decision-making in smart electricity markets“. In: *Machine learning* 92.1 (2013), S. 5–39.
- [29] Christian Lemke. „Mit Reinforcement Learning auf dem Weg zur Allgemeinen KI“. In: *Digitale Welt* 5.3 (2021), S. 66–68.
- [30] Pramila P Shinde und Seema Shah. „A review of machine learning and deep learning applications“. In: *2018 Fourth international conference on computing communication control and automation (ICCUBEA)*. IEEE. 2018, S. 1–6.
- [31] Mehdi Gheisari, Guojun Wang und Md Zakirul Alam Bhuiyan. „A survey on deep learning in big data“. In: *2017 IEEE international conference on computational science and engineering (CSE) and IEEE international conference on embedded and ubiquitous computing (EUC)*. Bd. 2. IEEE. 2017, S. 173–180.
- [32] Daniele Ravi u. a. „Deep learning for health informatics“. In: *IEEE journal of biomedical and health informatics* 21.1 (2016), S. 4–21.
- [33] Yue Deng u. a. „Deep direct reinforcement learning for financial signal representation and trading“. In: *IEEE transactions on neural networks and learning systems* 28.3 (2016), S. 653–664.
- [34] Fuyong Xing u. a. „Deep learning in microscopy image analysis: A survey“. In: *IEEE transactions on neural networks and learning systems* 29.10 (2017), S. 4550–4568.
- [35] Anders Krogh. „What are artificial neural networks?“ In: *Nature biotechnology* 26.2 (2008), S. 195–197.
- [36] Anil K Jain, Jianchang Mao und K Moidin Mohiuddin. „Artificial neural networks: A tutorial“. In: *Computer* 29.3 (1996), S. 31–44.
- [37] Warren S McCulloch und Walter Pitts. „A logical calculus of the ideas immanent in nervous activity“. In: *The bulletin of mathematical biophysics* 5.4 (1943), S. 115–133.

- [38] Daniel Svozil, Vladimir Kvasnicka und Jiri Pospichal. „Introduction to multi-layer feed-forward neural networks“. In: *Chemometrics and intelligent laboratory systems* 39.1 (1997), S. 43–62.
- [39] Tomas Mikolov u. a. „Recurrent neural network based language model.“ In: *Interspeech*. Bd. 2. 3. Makuhari. 2010, S. 1045–1048.
- [40] Alex Graves, Abdel-rahman Mohamed und Geoffrey Hinton. „Speech recognition with deep recurrent neural networks“. In: *2013 IEEE international conference on acoustics, speech and signal processing*. Ieee. 2013, S. 6645–6649.
- [41] Nal Kalchbrenner und Phil Blunsom. „Recurrent continuous translation models“. In: *Proceedings of the 2013 conference on empirical methods in natural language processing*. 2013, S. 1700–1709.
- [42] Wojciech Zaremba, Ilya Sutskever und Oriol Vinyals. „Recurrent neural network regularization“. In: *arXiv preprint arXiv:1409.2329* (2014).
- [43] Yann LeCun u. a. „Gradient-based learning applied to document recognition“. In: *Proceedings of the IEEE* 86.11 (1998), S. 2278–2324.
- [44] Saad Albawi, Tareq Abed Mohammed und Saad Al-Zawi. „Understanding of a convolutional neural network“. In: *2017 international conference on engineering and technology (ICET)*. Ieee. 2017, S. 1–6.
- [45] Liang-Chieh Chen u. a. „Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs“. In: *IEEE transactions on pattern analysis and machine intelligence* 40.4 (2017), S. 834–848.
- [46] Ian Goodfellow. „Nips 2016 tutorial: Generative adversarial networks“. In: *arXiv preprint arXiv:1701.00160* (2016).
- [47] Jason Brownlee. *Generative adversarial networks with python: deep learning generative models for image synthesis and image translation*. Machine Learning Mastery, 2019.
- [48] Richard Zhang, Phillip Isola und Alexei A Efros. „Colorful Image Colorization“. In: *ECCV*. 2016.
- [49] *Hotpot.ai*. URL: <https://hotpot.ai/> (besucht am 10.06.2023).
- [50] *VanceAI*. URL: <https://vanceai.com/de/> (besucht am 15.06.2023).
- [51] Xiaohui Song und Jane WS Liu. „How well can data temporal consistency be maintained?“ In: *IEEE Symposium on Computer-Aided Control System Design*. IEEE. 1992, S. 275–284.
- [52] Kaiming He u. a. „Deep residual learning for image recognition“. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, S. 770–778.
- [53] Olga Russakovsky u. a. „Imagenet large scale visual recognition challenge“. In: *International journal of computer vision* 115 (2015), S. 211–252.
- [54] Karen Simonyan und Andrew Zisserman. „Very deep convolutional networks for large-scale image recognition“. In: *arXiv preprint arXiv:1409.1556* (2014).
- [55] Sergey Ioffe und Christian Szegedy. „Batch normalization: Accelerating deep network training by reducing internal covariate shift“. In: *International conference on machine learning*. pmlr. 2015, S. 448–456.

- [56] Ian Goodfellow, Yoshua Bengio und Aaron Courville. *Deep learning*. MIT press, 2016.
- [57] Kunihiko Fukushima. „A neural network for visual pattern recognition“. In: *Computer* 21.3 (1988), S. 65–75.
- [58] Takeru Miyato u. a. „Spectral normalization for generative adversarial networks“. In: *arXiv preprint arXiv:1802.05957* (2018).
- [59] Antoine Salmona, Lucia Bouza und Julie Delon. „DeOldify: A Review and Implementation of an Automatic Colorization Method“. In: *Image Processing On Line* 12 (2022), S. 347–368.
- [60] Wenzhe Shi u. a. „Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network“. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, S. 1874–1883.
- [61] Han Zhang u. a. „Self-attention generative adversarial networks“. In: *International conference on machine learning*. PMLR. 2019, S. 7354–7363.
- [62] Justin Johnson, Alexandre Alahi und Li Fei-Fei. „Perceptual losses for real-time style transfer and super-resolution“. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14*. Springer. 2016, S. 694–711.
- [63] Dagmar Agsten-Aures. *Leipzig - Gestern und Heute*. Agsten-Aures Verlag Bremen, 1991.
- [64] ColorfulSoft. *DeOldify.NET*. 2021. URL: <https://github.com/ColorfulSoft/DeOldify.NET>.
- [65] Richard Zhang. *Colorful Image Colorization*. 2016. URL: <https://github.com/richzhang/colorization>.
- [66] Richard Zhang u. a. „Real-time user-guided image colorization with learned deep priors“. In: *arXiv preprint arXiv:1705.02999* (2017).
- [67] Jheng-Wei Su, Hung-Kuo Chu und Jia-Bin Huang. „Instance-aware image colorization“. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, S. 7968–7977.
- [68] Richard Zhang u. a. „The unreasonable effectiveness of deep features as a perceptual metric“. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, S. 586–595.
- [69] Karen Egiazarian u. a. „New full-reference quality metrics based on HVS“. In: *Proceedings of the second international workshop on video processing and quality metrics*. Bd. 4. 2006, S. 4.
- [70] Zhou Wang u. a. „Image quality assessment: from error visibility to structural similarity“. In: *IEEE transactions on image processing* 13.4 (2004), S. 600–612.
- [71] The MathWorks Inc. *MATLAB version: 9.13.0 (R2022b)*. Natick, Massachusetts, United States, 2022. URL: <https://www.mathworks.com>.
- [72] Daniel Grohmann. *ColoringLeipzig*. 2023. URL: <https://github.com/MOP5N/ColoringLeipzig>.

## Erklärung

Ich versichere, dass ich die vorliegende Arbeit mit dem Thema:

*„Kolorieren Historischer Fotos der Stadt Leipzig mit Deep-Learning-Techniken“*

selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann.

Leipzig, den 31.07.2023

---

DANIEL GROHMANN