Einfärben von Graustufenbildern mit Convolutional Neural Networks

Christian Bargmann

Hochschule für Angewandte Wissenschaften
Berliner Tor 5, 20099 Hamburg, Germany
christian.bargmann@haw-hamburg.de
https://www.haw-hamburg.de

Abstract. In dieser Ausarbeitung wird das Einfärben von Graustufenbildern mit von Convolutional Neural Networks thematisiert. Dazu werden Grundlagen zur Problemstellung und Arbeiten mit semi-, als auch mit vollautomatischen Verfahren als Lösungsansätze vorgestellt. Anschließend wird eine Methodik für das Training eines CNNs zum Einfärben von Grautufenbildern erläutert und die praktische Umsetzung präsentiert. Die Ergebnisse werden anschließend vorgestellt und zuletzt werden Möglichkeiten für weitere Experimente aufgezeigt.

Keywords: Image Colorization · Deep Learning · Convolutional Neural Networks · TensorFlow · Keras

1 Einleitung

Die automatisierte Einfärbung von Graustufenbildern ist ein Forschungsgebiet, dass in den letzten Jahren auf ein großes Interesse bei vielen Computer-Vision und Machine-Learning-Communities gestoßen ist. Das Einfärben von Graustufenbildern ist nicht nur aus ästhetischen Gründen interessant, sondern lässt sich praktisch in unterschiedlichsten Anwendungsbereichen einsetzen. Sei es die Restauration von historischem Bild- und Videomaterial, bis hin zur Verbesserung der Interpretierbarkeit von Bildern.

Für den menschlichen Verstand ist das Einfärben eines Graustufenbildes eine einfache Aufgabe. Schon als Kinder lernen wir, fehlende Farben in Malbüchern zu ergänzen, indem wir uns daran erinnern, dass Bäume grün sind oder der Himmel tagsüber blau und mit weißen Wolken bestückt ist. Schon durch das Hinzufügen von Farbe, kann die Menge an Informationen, die sich aus einem Bild gewinnen lassen, erhöht werden.

Aus dem Forschungsbereich der Computer-Vision, wurden in den letzten Jahren unterschiedliche semiautomatische Ansätze vorgestellt, welche in Abschnitt 2 detailierter erläutert werden. Lange wurden diese semiautomatischen Ansätze gegenüber vollautomatischen Ansätzen favorisiert, da zum Einfärben von Bildern ein umfangreiches Verständnis von der Bildkomposition, der Perspektive und den

Objektbeziehungen in dem zu färbenden Motiv wichtig sind. Dieses Verständnis ließ sich oft nur durch manuelle, menschliche Unterstützung herstellen. Durch den technischen und wissenschaftlichen Fortschritt im Bereich des maschinellen Lernens, hat sich ein großes Interesse an vollautomatischen Ansätzen entwickelt. Dieser Fortschritt führte dazu, dass heute hauptsächlich vollautomatische Lösungen mithilfe von Convolutional Neural Networks (CNNs) gegenüber semiautomatischen, von Menschen unterstützten Methoden zum Einfärben von Graustufenbildern eingesetzt werden [9].

1.1 Grundlagen der digitalen Darstellung von Graustufen- und Farbbildern

Für die Darstellung von Graustufen- und Farbbildern ist ein grundlegendes Verständnis über Farbräume notwendig. Der Farbraum bestimmt, wie sich die Farben, auf Grundlage der Anzahl der Farbkanäle in einem Farbmodell, kombinieren lassen. Unterschiedliche Farbräume führen zu anderen Ebenen, mit wiederum anderen Farbdetails. Es existieren verschiedene Farbräume für unterschiedliche Anwendungsbereiche. Beispielsweise wird der CMYK-Farbraum für Bilder in einer vollfarbigen Druckbroschüre und der RGB-Farbmodus für Bilder im Web oder in E-Mails verwendet, um die Dateigröße zu reduzieren und gleichzeitig die Farbintegrität zu erhalten.

Graustufenbilder Ein Graustufenbild ist ein Bild, bei dem der Wert jedes Pixels die jeweilige Lichtmenge an dessen Position repräsentiert, d.h. sie trägt nur Intensitätsinformationen. Der Kontrast reicht von Schwarz bei der schwächsten Intensität bis hin zu Weiß bei der stärksten Intensität. Ein Graustufenbild verwendet verschiedene Graustufen. In 8-Bit-Bildern können bis zu 256 Graustufen vorhanden sein. Jedes Pixel eines Graustufenbildes hat somit einen Helligkeitswert zwischen 0 (Schwarz) und 255 (Weiß). Bei 16- und 32-Bit-Bildern hingegen ist die Anzahl der Schattierungen in einem Bild größer als bei 8-Bit-Bildern.

Farbbilder Farbbilder können auf unterschiedliche Arten dargestellt werden. Die wohl bekannteste Darstellung von Farbbildern ist der RGB-Farbraum. Sei I_{rgb} ein Farbbild, dann lässt sich dieses in die drei Farbkanäle I_r, I_g und I_b, unterteilten, wobei das Farbbild durch Kombination aller Kanäle auf dem Bildschirm dargestellt werden kann ($I_{rgb} = (I_r, I_g, I_b)$. Bei Bildern mit 8 Bit pro Kanal reichen die Intensitätswerte von 0 (Schwarz) bis 255 (Weiß) für jede der RGB-Komponenten (Rot, Grün, Blau) in einem Farbbild. Wenn die Werte aller drei Komponenten gleich sind, ist das Ergebnis ein neutraler Grauton. Wenn die Werte aller Komponenten 255 betragen, ist das Ergebnis reines Weiß; wenn die Werte 0 sind, ein reines Schwarz.

Eine andere Darstellungsform ist der Lab-Farbraum. Dieser basiert auf der menschlichen Farbwahrnehmung. Die numerischen Werte im Lab-Farbraum beschreiben alle Farben, die ein normalsichtiger Mensch wahrnehmen kann. Der Lab-Farbraum ist ein geräteunabhängiges Farbmodell. Anders als RGB beschreibt

er, wie eine Farbe aussieht, und nicht, wie viel eines bestimmten Farbstoffs ein Gerät (wie z.B. ein Monitor, ein Drucker oder eine Digitalkamera) benötigt, um Farben zu erzeugen.
Sei I also ein Bild im Lab-Farbraum, so lässt es sich in die zwei Komponenten Luminanz und Chrominanz zerlegen, sodass gilt $I = (I_l, I_c)$. Das Bild kann vollständig rekonstruiert werden, wenn man die beiden Komponenten miteinander kombiniert werden. Fehlt eine der beiden Komponenten, lässt sich das Bild nur unvollständig wiederherstellen. Bilder, bei denen die Chrominanzinformationen vermisst werden, lassen sich dennoch natürlich mit dem Auge wahrnehmen, da die Luminanzkomponente visuelle Merkmale wie Objektkanten und Kontrastunterschiede vermittelt.

Der Vollständigkeit halber ist zu erwähnen, dass es noch viele weitere Farbräume für die Darstellung von Farbbildern gibt, jedoch hier nur der RGB- und Lab-Farbraum vorgestellt wurden, da diese im Verlauf dieser Ausarbeitung relevant sind.

1.2 Problemstellung

Bei dem Einfärben von Graustufenbildern wird ein Mapping von einem Pixel an einer Position in dem zugrundeliegenden Graustufenbild zu den farbigen Äquivalenten der Farbkanäle aus dem Farbraum des Ausgangsbildes gesucht.

$$map : P \rightarrow \tilde{P} \tag{1}$$

wobei $P \in \mathbb{R}^{H \times W}$ ein Pixel aus dem Graustufenbild und $\tilde{P} \in \mathbb{R}^{H \times W \times Channels}$ ein Pixel aus dem Farbbild ist. Die Schwierigkeit des korrekten Einfärbens von Graustufenbildern liegt in der multidimensionaler Natur der Problemstellung. Statt einer "korrekten" Farbe für einen Pixel im Ausgangsbild, kann es mehrere Möglichkeiten geben. In anderen Worten: das Übertragen einer Farbe aus dem Graustufenbild in das zu färbende Ausgangsbild, lässt sich nicht objektiv auf sämtliche Motive anwenden. Für das Mapping spielen zusätzliche subjektive Informationen, wie beispielsweise die Beziehungen zwischen den dargestellten Objekten und der Szene selbst, eine wichtige Rolle für die endgültige Färbung.

Diese Erkenntnis erscheint zunächst trivial, entwickelt allerdings schnell ihren eigenen, komplexen Charakter. Beispielsweise kann in einer historischen Aufnahme das Kleid einer Frau, durch analytische Ansätze oder durch ein trainiertes neuronales Netzwerk, blau gefärbt worden sein. Tatsächlich weiß man aber, dass aus historischen Gründen das Kleid rot gewesen sein muss. Diese Art von Kontextinformationen sind nicht im zu färbenden Graustufenbild enthalten.

2 Ähnliche Arbeiten

Diese Ausarbeitung baut auf wissenschaftlichen Arbeiten auf, die sich mit der Generierung von Farbbildern aus Graustufenbildern auseinander gesetzt haben.

Die Recherche nach Veröffentlichungen beschränkte sich nicht ausschließlich auf die Implementierung mit neuronalen Netzen, sondern hat analytische Ansätze gleichermaßen betrachtet. Grundlegend lassen sich zwei Kategorien mit Ansätzen für das Einfärben von Graustufenbildern unterscheiden. Semiautomatische Ansätze versuchen, von Menschen vorgegebene Informationen zur Färbung des Graustufenbildes durch Algorithmen zu unterstützen, während vollautomatische Ansätze die Problemstellung ohne menschliche Unterstützung versuchen zu lösen.

2.1 Semiautomatische Ansätze

Levin et al. präsentieren einen analytischen Ansatz, bei dem der Benutzer Farbhinweise in Form von sogenannten "Scribbles" auf dem Graustufenbild bereitstellt, welche den Algorithmus bei der Wahl endgültigen Farbgebung im Ausgangsbild untersützen. Ausgehend von der Idee, dass benachbarte Bildpixel mit der selben Belichtungsintensität womöglich auch dieselbe Farbe teilen, nutzen Levin et al. eine konvexe, quadratischen Kostenfunktion, um die Intensitätsunterschiede zwischen benachbarten Pixeln zu berechnen [6]. Mit der Ergänzung um eine verbesserte Kantenerkennung, die Luan et al. in ihrer Abeit [8] präsentieren um Color Bleeding zu reduzieren, sind Scribbles heutzutage ein gängiges Verfahren zur semiautomatischen Einfärbung von Graustufenbildern.

Andere Ansätze sind ebenfalls auf manuelle Benutzereingaben angewiesen. Hier werden allerdings keine Farbinformationen von einem Benutzer zur Verfügung gestellt, sondern ein oder mehrere Referenzbilder, die für die Einfärbung des Graustufenbildes unter Berücksichtigung von statistischer Farbverteilung oder von Bildmerkmalen verwendet werden sollen. Liu et al. [7] führen weiterhin eine Nachverarbeitung des Ausgangsbildes durch, um unerwünschte Effekte wie starke Belichtungsunterschiede zu entfernen oder Deshpande et al. [4], wo das Histogramm des gefärbtes Ausgangsbildes mit der relativen Farbverteilung der zugrundeliegenden Referenzbilder verglichen wird. Die meisten Implementierungen nutzen für die Übertragung von Farben aus Referenzbildern Bilddeskriptoren aus dem Bereich der Computer Vision wie Patch, GABOR oder Daisy. Hier werden zunächst gemeinsame Bildmerkmale wie Formverläufe oder Kontrastunterschiede in dem Graustufenbild und den Referenzen identifiziert und anschließend die Farbe auf Grundlage der dieser Bildmerkmale übertragen.

Welsh et al. beispielsweise präsentieren in ihrer Arbeit ein Verfahren zur Übertragung von Farben aus einem vom Benutzer bereitgestellten Referenzbild auf ein zu färbendes Graustufenbild [11]. Für das Einfärben werden Bildmerkmale anhand des lokalen Belichtungswertes eines Pixels und statistischen Eigenschaften der lokalen Nachbarschaftsbeziehung zu umliegenden Pixeln identifiziert und anschließend der passende Farbwert ermittelt. Die automatische Suche nach Referenzbildern setzen Chia et al. in ihrer Arbeit um [3]. Dort wird zunächst das zu färbende Objekt auf dem Graustufenbild von dem Motivhintergrund freigestellt

und gelabelt. Anschließend erfolgt auf Basis der zur Verfügung gestellten Informationen eine automatische Websuche nach Referenzbildern.
Im Vergleich zu Ansätzen, bei denen Farbhinweise durch den Benutzer notwendig sind, erfordern Ansätze die mit der Bereitstellung von Referenzbildern arbeiten weniger manuellen Aufwand. Allerdings fällt die Einflussnahme auf die Farbauswahl bei Referenzbildern deutlich schwerer, da sich die Farbauswahl vollständig auf das zur Verfügung gestellte Datenmaterial stützt und die Farbwahl durch die Implementierung mit Bilddeskriptoren nicht immer auf den ersten Blick interpretierbar ist.

2.2 Automatische Ansätze mit CNNs

Das manuelle Ergänzen von Informationen zum Einfärben eines Graustufenbildes durch den Benutzer, lässt sich aufgrund des zeitlichen Aufwandes nicht auf große Datenbestände skalieren. Mit dem wachsenden Interesse und dem Einzug von Maschine Learning in diverse Anwendungsbereiche, haben sich heutzutage CNNs als Standard für das Einfärben von Graustufenbildern etabliert. Der Einsatz von CNNs ähnelt dem zuvor beschriebenen, semiautomtischen Ansatz, bei dem Referenzbilder für das Einfärben hinzugezogen werden. Die Entscheidung über die Auswahl der zu nutzenden Referenzbilder, wird allerdings nicht von einem Menschen getroffen, stattdessen lernt das neuronale Netzwerk während des Trainingsprozesses mit großen Bildbeständen, diese Auswahl von sich aus treffen. Hierdurch wird es möglich, selbst große Bildbestände ohne menschliches Eingreifen, vollautomatisch zu verarbeiten. Durch die Verwendung von Trainingssets (wie z.B. der ImageNet[1] Datensatz), die eine große Vielfalt an unterschiedlichen Szenen mit gleichartigen Objekten enthalten, wird sichergestellt, dass das Lernen der passenden Farbübertragung gelingt.

Die Arbeit von Zhang et al. hat sehr viel Aufmerksamkeit für die dort vorgestellte Netzwerkarchitektur erhalten. Sie präsentieren ein CNN bestehend aus 22 Faltungsschichten, die mit einer Teilmenge des ImageNet Datensatzes trainiert worden sind [12]. Außerdem stellen sie eine eigene, polynomialverteilte Cross-Entropy Verlustfunktion vor, welche die Unterschiede in der Farbverteilung an einer Pixelposition im Ausgangsbild auf Basis von Histogrammen bestimmt. Andere Ansätze mit CNNs versuchen zunächst ein höheres semantisches Verständnis des Graustufenbildes zu erlangen, um später eine konsistentere Färbung vorzunehmen. So stellen Izuka et al. ein Netzwerkmodell vor, dass zwei Berechnungswege miteinander kombiniert [5]. Ein Teil des Gesamtmodells wird auf das Identifizieren von lokalen, der andere Teil auf die Identifizierung von globalen Bildmerkmalen mithilfe von Bildklassifizierung trainiert. Globale Bildmerkmale werden anschließend stückweise mit den Lokalen verknüpft und das Einfärben mit einer euklidischen Verlustfunktion trainiert.

[1] https://www.image-net.org/

Andere Ansätze, wie beispielsweise in der Arbeit von Cao et al. vorgestellt, verwenden sogennante Generative Adversarial Networks (GANs) für das vollautomatische Einfärben von Graustufenbildern [2]. Hier werden zwei unterschiedliche Netzwerke trainiert, ein Generator und ein Diskriminator, die anschließend gegeneinander antreten. Das Generator-Netzwerk wird verwendet, um zu einem Graustufenbild ein zugehöriges Farbbild zu erzeugen. Das Diskriminator-Netzwerk entscheidet anschließend darüber, ob das generierte Farbbild mit der wirklichen Farbgestaltung übereinstimmt. Stimmen der generierte Output des Generators und Wirklichkeit nicht miteinander überein, werden die Gewichtungen der Neuronen des Generators dahingehen aktualisiert, dass der neue generierte Output den Erwartungen des Diskriminators entspricht. In anderen Worten: das Generator-Netzwerk verwendet das Diskriminator-Netzwerk als adaptive, dynamische Verlustfunktion zum Lösen des Färbungsproblems.

3 Methodik

Im Rahmen dieser Ausarbeitung wurde ein Convolutional Neural Network für das Einfärben von Graustufenbildern trainiert. Die Methodik und die Implementierung sind inspiriert von den Arbeiten von Baldassarre et al. [1] und Zhang et al. [12].

3.1 Datensatz

Bei der Erstellung eines Bilddatensatzes für das Einfärben von Graustufenbildern entfällt das aufwendige Labeling der Trainingsdaten. Es können Farbbilder verwendet werden, wobei die jeweilige Graustufenversion als Trainingsinput und die Farbversion als gelabelte Wahrheit genutzt werden kann.

Wie zuvor in Abschnitt 2 erläutert, existieren bereits große Bilddatensätze wie ImageNet oder das Scene Categorization Benchmark[1] (SUN), die für das Training eines eigenen Netzmodells genutzt werden können. Auch besteht die Möglichkeit, ein vortrainiertes Netzmodell zu verwenden und zu erweitern. Für das Training des CNNs wurde jedoch auf beide Möglichkeiten verzichtet und ein eigener Datensatz auf Basis von freiem Bildmaterial der Bildcommunity Unsplash[2] erstellt. Hierzu wurden 9500 Bilder heruntergeladen und zu einem Bilddatensatz zusammengefügt.

Die Bilder im Trainings- und Testdatensatz wurden während des Downloads auf eine einheitliche Bildgröße von 256×256 tranformiert. 256×256 bzw. 224×224 sind gängige Bildgrößen, die für den Trainingsprozess bei CNNs verwendet werden. Das Verringern der Bildauflösung hat den Vorteil, dass die Geschwindigkeit des Trainingsprozesses beschleunigt wird und hilft gleichzeitig den benötigten Festplatten- und Arbeitsspeicher zu reduzieren.

[1] https://groups.csail.mit.edu/vision/SUN/
[2] https://unsplash.com

3.2 Farbraumkonvertierung

Wie bereits in Abschnitt 1.2 vorgestellt, wird für das Einfärben von Graustufenbildern ein Mapping gesucht. Das CNN wird trainiert, das Mapping $map : P \rightarrow \tilde{P}$ zu erlernen. Ausgehend von der Idee, dass das zu färbende Graustufenbild bereits viele korrekte Bildinformationen wie Kontrast- und Belichtungwerte enthält, wird mit den Bilddaten der Größe $H \times W$ im Lab-Farbraum gearbeitet.

Als Input für das Training des CNNs dient die Belichtungskomponente $I_l \in \mathbb{R}^{H \times W \times 1}$. Von dem trainierten CNN wird erwartet, die verbleibenden Komponenten $\tilde{I}_a$ und $\tilde{I}_b \in \mathbb{R}^{H \times W \times 3}$ zu schätzen. Gesucht ist also ein Mapping in der Form

$$map : I_l \rightarrow (\tilde{I}_a, \tilde{I}_b) \tag{2}$$

wobei $\tilde{I}_a$ dem a-Farbkanal und $\tilde{I}_b$ dem b-Farbkanal im Lab-Fabraum entspricht. Gemeinsam mit dem der Belichtungskomponente I_l ergeben sie in Kombination das vollständig eingefärbte Bild $\tilde{I} = (I_l, \tilde{I}_a, \tilde{I}_b)$. Durch die Entscheidung für den Lab-Farbraum lässt sich die Anzahl der vom CNN zu schätzenden Outputs, im Vergleich zum RGB-Farbraum, von Drei auf Zwei verringern.

3.3 Verlustfunktion

Für die Berechnung des Fehlers wird eine mittlere, quadratische Fehlerfunktion (Mean Square Error) verwendet, um den quadratischen Abstand zwischen dem Farbwert, den das trainierte CNN schätzt, und dem originalen Farbwert zu minimieren. Sei I ein Bild, dann wird der mittlere, quadratische Fehler berechnet wie folgt berechnet:

$$MSE(I) = \frac{1}{2HW} \times \sum_{k \in \{a,b\}} \sum_{x=1}^{H} \sum_{y=1}^{W} (I_{k_x,y} - \tilde{I}_{k_x,y})^2 \tag{3}$$

wobei $I_{k_x,y}$ und $\tilde{I}_{k_x,y}$ einen Pixelwert an einer Position innerhalb der zweidimensionalen Bildmatrix im k-ten Farbkanal darstellt. Zu berücksichtigen ist, dass der berechnete mittlere, quadratische Fehler von der gewählten Batchgröße abhängt. Der durchschnittlichen Verlustwert aller Batches β kann somit bestimmt werden durch

$$\frac{1}{\beta} \times \sum_{I \in \beta}^{n} MSE(I_n) \tag{4}$$

Während des Trainingsprozesses wird der hierdurch berechnete Verlust für das Aktualisieren der Netzparameter während der Backpropagation verwendet.

4 Implementierung

In diesem Abschnitt wird auf die konkrete Implementierung der zuvor erläuterten Methodik eingegangen. Der Sourcecode für die Implementierung ist auf der Projektseite[1] zu finden.

4.1 Netzwerkarchitektur

Die Netzwerkarchitektur orientiert sich an dem Modell von Zhang et al.. Das Modell erhält als Eingabe die Luminanzkomponente und soll die verbleibenden beiden Farbkomponenten a und b schätzen und zu einem Farbbild kombinieren.

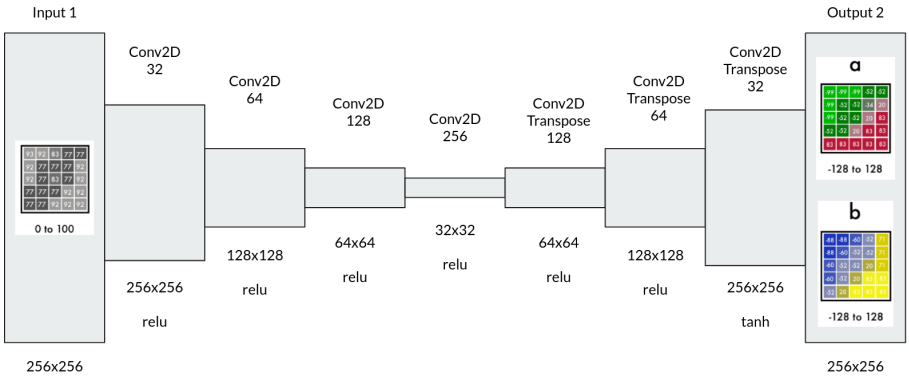

Fig. 1. Darstellung des implementierten Netzwerkmodells. Verwendet wurde ein reines CNN-Modell mit sieben Faltungsschichten.

Das Modell besteht aus insgesamt sieben Faltungsschichten. Die Anzahl an Filtern wird anfangs bei jeder Faltungsschicht verdoppelt, sodass die mittlere Faltungsschicht die höchste Filterdichte besitzt. Durch die ersten vier Faltungsschichten wird die Auflösung des Bildes von den ursprünglichen 256×256 Pixeln auf ein Achtel der Eingangsgröße verkleinert. Die Verwendung von niedrigen Auflösungen bei tieferen Faltungsschichten mit einer hohen Filterdichte, führt nach Aussage von Zhang et al. dazu, dass ein Overfitting des Modells verhindert wird. Filter auf tieferen Faltungsschichten müssen einen kleineren Bildausschnitt mit wenigen Pixeln interpretieren, wodurch diese sich auf abstraktere Bildstrukturen spezialisieren.

Eine weitere Besonderheit des Modells ist, dass auf Pooling-Schichten verzichtet wurde. Pooling-Layer stellen eine Methode zur Reduzierung der Bildauflösung bei CNNs dar. Durch sie wird der von einem Filter zu interpretierende Bildausschnitt auf einer Faltungsschicht im Netzwerk vergrößert. Der Vorgang des

[1] https://gitlab.informatik.haw-hamburg.de/abz616/image-colorization

Poolings gleicht einer mathematischen Operation, bei dem aus einer Matrix ein numerischer Wert (Maximum, Durchschnitt, ...) gebildet wird. Generell haben Pooling-Schichten vordefinierte Filter und keine trainierbaren Parameter. Mit anderen Worten: Beim Einsatz von Pooling-Schichten werden Bildinformationen verworfen. Nach Erkenntnissen von Springenberg et al. [10] ist dieses Vorgehen für das Einfärben von Graustufenbildern nicht sinnvoll, da jede Bildinformation für das Ergebnis des Farbbildes wertvoll ist. Stattdessen wird die Reduzierung der Bildauflösung durch das Erhöhen des Kernel-Strides auf den ersten Faltungsschichten realisiert. Hierdurch werden mehr Bildinformationen bewahrt und können tiefere Netzwerkschichten erreichen. Auf allen Faltungsschichten im Modell wird mit einer Kernel-Größe von 3×3 gearbeitet.

Um die Fehlerrate während des Trainingsprozesses zu berechnen, wurden die Farbbilder zunächst auf einen Wertebereich zwischen -1 und 1 normiert. Die letzte Faltungsschicht im Modell verwendet als Aktivierungsfunktion die Sigmoidfunktion *Tangens hyperbolicus* (tanh), welche die vom Netzwerk geschätzten Farbwerte für die a- und b-Farbkomponenten ebenfalls auf diesen Wertebereich normiert. Mit der in Abschnitt 3.3 erläuterten Verlustfunktion wird anschließend der Fehler ermittelt.

4.2 Trainingsprozess

Von dem insgesamt im Datensatz enthaltenen 9500 Bilddaten wurden 500 Bilder als Validierungsdatensatz und weitere 500 Bilder als Testdatensatz vorgehalten. Der Ablauf des Trainingsprozesses gliedert sich in drei Abschnitte:

> ***Encoding*** - Ein Batch an Bilddaten wird in den Arbeitsspeicher geladen. Der Farbraum der Bilddaten wird von dem RGB- zum Lab-Farbraum konvertiert, wobei die Luminanzkomponente als Trainingsinput und die a- und b-Farbkomponenten als Referenzwert für den den geschätzten Output dienen. a- und b-Farbkomponente werden auf einen Wertebereich zwischen -1 und 1 normiert.
>
> ***Training*** - Das Netzwerk wird mit den Datensätzen im Batch trainiert über mehrere Epochen trainiert.
>
> ***Decoding (bei Testdaten)*** - Die Bildinformationen im Lab-Farbraum werden zusammengefügt. Für das Resultat wird die als Input verwendete Luminanzkomponente mit den geschätzten, denormalisierten a- und b-Farbkomponenten kombiniert, wie in Abschnitt 3.2 beschrieben. Anschließend erfolgt eine Farbraumkonvertierung von dem Lab- in den ursprünglichen RGB-Farbraum.

Das Netzwerk wurde mit den GPU-Knoten der Informatik Compute Cloud[1] an der Hochschule für Angewandte Wissenschaften Hamburg trainiert und getestet.

[1] https://userdoc.informatik.haw-hamburg.de/doku.php?id=docu:informatikcomputecloud

Zur Beschleunigung der Berechnungen während des Trainingsprozesses wurden das Nvidia CUDA[1] Toolkit und eine Nvidia Tesla K80 GPU verwendet. Die Bilddaten wurden in Batches mit einer Größe von jeweils 50 Datensätzen in den Arbeitsspeicher geladen. Nach jeder Trainingsepoche wurden die Trainingsdaten zuvor nach dem Zufallsprinzip gemischt, um ein Overfitting des Modells zu minimieren.

4.3 Ergebnisse

Für die Evaluierung der Ergebnisse wurden im Rahmen dieser Ausarbeitung keine Umfragen durchgeführt oder Metriken eingesetzt, wie beispielsweise in den Papern von Zhang et al. [12] oder Baldassarre et al. [1]. Die Qualität der Ergebnisse bezieht sich auf die persönliche, subjektive und visuelle Beurteilung des erzeugten Outputs nach Empfinden des Autors dieser Arbeit.

Nach dem Trainingsprozess von 80 Epochen, wurden dem Netzwerk Bilddaten aus dem Testdatensatz zur Färbung zur Verfügung gestellt. Aufgrund des vergleichsweise kleinen Datensatzes, fallen die Ergebnisse jedoch abhängig vom jeweiligen Motiv stark unterschiedlich aus. Es ist zu beobachten, dass keine knalligen Farben im generierten Output erzeugt worden sind, sondern meist neutrale Brauntöne zur Färbung wurden.

In Fig. 2 ist festzustellen, dass Farbdetails wie das blaue Verkehrschild im rechten Bildausschnitt, in dem erzeugten Output verloren gegangen sind. Auch bei leeren Flächen, wie der Straße im linken Bildausschnitt, hatte das CNN Schwierigkeiten, vermutlich aufgrund fehlender Strukturen, eine korrekte Färbung vorzunehmen.

Fig. 2. (Links) Graustufenbild, (Mitte) Output des CNNs, (Rechts) Originales Farbbild

In Fig. 3 lässt sich ein ähnliches Ergebnis wie in Fig. 2 feststellen. Die Farbdetails am Himmel gingen bei der Färbung verloren. Das weiße Oberteil der Frau wurde in neutralen Brauntönen gefärbt. Der Output in Fig. 5 zeigte deutliche Farbfehler. Die Berggruppen und der Himmel wurden stellenweise bräunlich

[1] https://developer.nvidia.com/cuda-zone

Fig. 3. (Links) Graustufenbild, (Mitte) Output des CNNs, (Rechts) Originales Farbbild

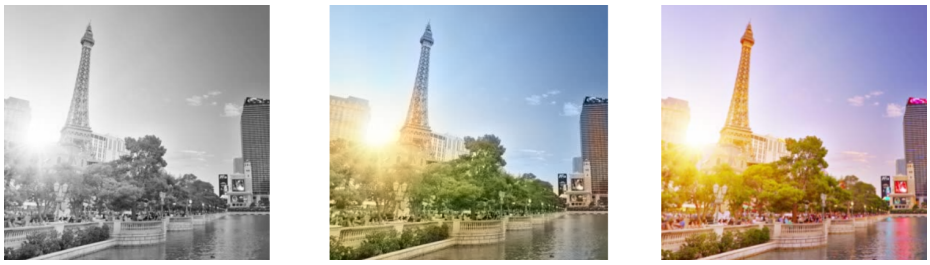

Fig. 4. (Links) Graustufenbild, (Mitte) Output des CNNs, (Rechts) Originales Farbbild

gefärbt. Zu erkennen ist, dass das CNN Schwierigkeiten bei der Interpretation der Ebenenstrukturen im Vordergrund hatte. Der in Fig. 4 vorgestellte Output, entsprach von der Farbgebung her sehr dem Original. Zu beobachten war, dass Details, wie die farblichen Spiegelungen auf der Wasseroberfläche, sowie Farbinformationen der Menschengruppen, verloren gingen. Stattdessen wurde ausgehend von der Sonneneinstrahlung ein entsättigtes Gelb verwendet.

Fig. 5. (Links) Graustufenbild, (Mitte) Output des CNNs, (Rechts) Originales Farbbild

Insgesamt sind die Ergebnisse optisch zufriedenstellend. Nach der Präsentation der Ergebnisse vor Kommilitonen, wurde oft angemerkt, dass die vom CNN generierten Bilder eine natürlichere Farbgebung als die Originale besäßen.

5 Weitere Experimente und Ausblick

Auf der Grundlage der vorgestellten Implementierung können nun weitere Experimente durchgeführt werden. Um die Qualität der Ergebnisse zu beeinflussen, lassen sich unterschiedliche Parameter konfigurieren. Darunter folgende:

- Konfiguration der Anzahl an Faltungsschichten
- Konfiguration der Anzahl der Filter pro Faltungsschicht
- Anpassung der Werte für Kernel-Größen, Stride und Padding
- Einsatz Pooling-Schichten entgegen der Empfehlung von Springenberg et al.
- Einsatz von Netzwerk-in-Netzwerk Schichten
- Einsatz von Objektklassifizierung, Lokalisierung oder Bildsegmentierung als Unterstützung für das Einfärben
- Nutzen von Transfer-Learning mit vortrainierten Netzwerk-Modellen
- Veränderungen an der in Abschnitt 3.3 vorgestellten Verlustfunktion
- Anpassung der Batch-Größe während des Trainingsprozesses

Wie in Abschnitt 1.2 erläutert, ist das Einfärben von Graustufenbildern eine multidimensionale Problemstellung. Auf dieser Erkenntnis aufbauend wäre interessant, inwiefern sich weitere Metainformationen über zu färbende Graustufenbilder, wie historische Hintergründe oder künstlerische Intention, in das Ergebnis einbeziehen lassen.

5.1 Zusammenfassung und Ausblick

In dieser Ausarbeitung wurde das Einfärben von Graustufenbildern mithilfe von Convolutional Neural Networks thematisiert. Zunächst wurde in Abschnitt 1 eine Einführung in die Thematik gegeben, Grundlagen der digitalen Darstellung von Graustufen und Farbbildern erläutert und die Problemstellung dargestellt. In Abschnitt 2 wurden verwandte Arbeiten zur Lösung der Problemstellung präsentiert. Hierzu wurden sowohl semiautomatische, als auch vollautomatische Ansätze mit CNNs vorgestellt. Abschnitt 3 beschäftigte sich mit der übergreifenden Methodik der in Abschnitt 4 durchgeführten Implementierung eines CNN-Modells zur Einfärbung von Graustufenbildern. Die Ergebnisse wurden anschließend vorgestellt und zuletzt Möglichkeiten für weitere Experimente aufgezeigt.

Diese Ausarbeitung dient als Grundlage für eine weiterführende Arbeit zur Einfärbung von Graustufenbildern. Zunächst soll das vorgestellte Modell mit einem anderen Datensatz (*ImageNet*, *SUN*) trainiert werden, um das Mapping von Farbinformationen durch einen größeren Datenbestand zu erlernen. Weiterhin ist der Einsatz von geeigneten Metriken für die Bestimmung der Qualität des Outputs notwendig für eine weitere Forschung. Auch ein Vergleich der eigenen Ergebnissen mit den Resultaten anderer Arbeiten, wie beispielsweise Zhang et al., Izuka et al. oder Cao et al. ist anzustreben. Abschließend lässt sich zusammenfassen, dass das Einfärben von Graustufenbildern ein sehr spannendes Forschungsgebiet mit rasantem Entwicklungsfortschritt ist, das gleichzeitig viel Potential für wissenschaftliche Experimente bietet.

References

1. Baldassarre, F., Morín, D.G., Rodés-Guirao, L.: Deep koalarization: Image colorization using CNNs and inception-ResNet-v2 http://arxiv.org/abs/1712.03400
2. Cao, Y., Zhou, Z., Zhang, W., Yu, Y.: Unsupervised diverse colorization via generative adversarial networks http://arxiv.org/abs/1702.06674
3. Cheng, Z., Yang, Q., Sheng, B.: Deep colorization http://arxiv.org/abs/1605.00075
4. Deshpande, A., Rock, J., Forsyth, D.: Learning large-scale automatic image colorization. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 567–575. https://doi.org/10.1109/ICCV.2015.72, ISSN: 2380-7504
5. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification **35**(4), 110:1–110:11. https://doi.org/10.1145/2897824.2925974, https://doi.org/10.1145/2897824.2925974
6. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. In: ACM SIGGRAPH 2004 Papers. pp. 689–694. SIGGRAPH '04, Association for Computing Machinery. https://doi.org/10.1145/1186562.1015780, https://doi.org/10.1145/1186562.1015780, event-place: Los Angeles, California
7. Liu, X., Wan, L., Qu, Y., Wong, T.T., Lin, S., Leung, C.S., Heng, P.A.: Intrinsic colorization **27**(5), 152:1–152:9
8. Luan, Q., Wen, F., Cohen-Or, D., Liang, L., Xu, Y.Q., Shum, H.Y.: Natural image colorization. pp. 309–320. https://doi.org/10.2312/EGWR/EGSR07/309-320
9. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge http://arxiv.org/abs/1409.0575
10. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.: Striving for simplicity: The all convolutional net http://arxiv.org/abs/1412.6806
11. Welsh, T., Ashikhmin, M., Mueller, K.: Transferring color to greyscale images **21**(3), 277–280. https://doi.org/10.1145/566654.566576, https://doi.org/10.1145/566654.566576
12. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization http://arxiv.org/abs/1603.08511