Neuronale Netze in der Videoproduktion

Laura Anger
Technische Hochschule Köln
Institut für Medien- und Phototechnik
laura.anger@th-koeln.de

Vera Brockmeyer
Technische Hochschule Köln
Institut für Medien- und Phototechnik
vera.brockmeyer@smail.th-koeln.de

Zusammenfassung

In der Videoproduktion können die Prozesse sehr zeit- und kostenintensiv sein. Aus diesem Grund gilt es Wege zu finden, einzelne Teilprozesse, vor allem in der Produktion und der Postproduktion, zu automatisieren und zu optimieren. Diese Teilbereiche beschäftigen sich überwiegend mit der Erstellung und Bearbeitung von Bild- und Audiodaten. In den letzten Jahren konnten enorme Erfolge in der Bild- und Audiodatenverarbeitung mit Neuronalen Netzen (NN) erzielt werden. Im besonderen erwiesen sich Faltungsnetze (CNN) als zuverlässige und robuste Automatisierungsalgorithmen, die zukünftig Anwendung in der Videoproduktion finden könnten. Beispielsweise können mit CNN Szenendynamiken aus einem Einzelbild vorhergesagt werden. Zum jetzigen Zeitpunkt ist dies nur mit sehr geringen Auflösung und Algorithmus bedingten Bildartefakten möglich. Die beiden Verfahren *Deep Dream* und *Deep Style* können die Stilsemantik eines Bildes vom Inhalt trennen und auf ein unabhängiges Bild übertragen. *Deep Style* ist ein vielversprechendes Verfahren zur automatischen Stilsynthese, während das *Deep Dream* Verfahren auf Grund mangelnder Reproduzierbarkeit ungeeigneter scheint. Alle vorgestellten Verfahren sind nicht speziell für die Videoproduktion entwickelt worden.

Schlüsselwörter

Faltungsnetze, Videoproduktion, Stilsynthese, Bewegtbildgenerierung, Deep Writing

1. Einleitung

Im Zuge der Digitalisierung hat die Videoproduktion einen enormen Wandel durchlebt. Jeder der vier Produktionsabschnitte (vgl.d Abbildung 1) konnte verbessert und nachhaltig erleichtert werden. In der Konzeptionsphase konnten die Arbeitsprozesse mit Hilfe des Internets erleichtert werden durch den beschleunigten weltweiten Austausch von Skripten oder den standort-unabhängigen Zugriff auf cloudbasierte Projektmanagement Systeme. Während der Produktion des Videomaterials unterstützen moderne digitale Kamera- und Produktionssysteme aktiv den Kameramann. Die darauffolgenden Arbeitsprozesse, wie das Schneiden und Editieren des Videomaterials währen der Postproduktionsphase, wurden in den letzten Jahren vereinfacht oder können teilweise durch zuverlässige Algorithmen automatisch durchgeführt werden. Mittlerweile können durch *Computer Generated Imaging* (CGI) Produktionen fast ausschließlich im Studio produziert werden und selbst aufwendige Fantasiewelten oder aufwendige Stunts mit geringeren Kosten und weniger Aufwand umgesetzt werden.

Doch gerade qualitativ hochwertige Videoproduktionen erfordern immer noch einen sehr hohen Arbeitsaufwand mit einer große Anzahl an professionellen Mitarbeitern und kostenaufwändigen Materialien. Einen großen Anteil daran hat die Postproduktion in der jede Szene separat editiert und an das Gesamtbild angepasst werden muss. Dieses Gesamtbild muss vorab genau festgelegt werden und erlaubt keine Experimente in der Stilfindung. Aber auch die Generierung von Bildmaterialien für kurze Schnittszenen oder Webvideos ist sehr zeitaufwändig und kostenintensiv. Ein mehrköpfiges Team mit dem umfangreichen Equipment muss zum Drehort transportiert und untergebracht werden. Auch äußere Einflüsse, wie zum Beispiel das Wetter, können den Zeitplan verzögern und zusätzliche Kosten verursachen. Für Studioaufnahmen müssen in der Regel entsprechende Ressourcen angemietet oder in Stand gehalten werden.

In der Zukunft gilt es diesen Arbeits- und Kostenaufwand weiter zu reduzieren indem die einzelnen Arbeitsschritte automatisiert oder teilautomatisiert werden. Eine andere Vision ist es kurze Videoszenen für Webvideos oder Schnittszenen künstlich auf Basis von einzelnen Photographien am Computer zu erstellen. Dies erfordert Methoden die komplexe Zusammenhänge und Erfahrungen ähnlich wie das menschliche Gehirn zusammensetzen und anwenden können. Sie sollten kreativ sein, Bewegungen und Abläufe voraussagen, bekannte Eigenschaften sinnvoll

kombinieren oder erlernte Informationen übertragen und anwenden können. Diese Anforderungen erfüllen künstlichen Intelligenzen, wie NN (vgl. Abschnitt 2.2), die dem menschlichen Gehirn nachempfunden sind [17]. In den letzten Jahren wurden sie stetig weiterentwickelt und es konnten vor allem im Bereich der Medienproduktion vielversprechende Erfolge erzielt werden. Die größten Erfolge konnten mit CNN (vgl. Abschnitt 2.3) erzielt werden. Diese ermöglichen orts- und skalierungs-unabhängige Operationen und sind somit ideal für mehrdimensionale digitale Signale geeignet.

Zu Beginn wurden CNN zur Objektklassifizierung eingesetzt um unter anderem automatisch Metadaten von Bild- oder Videodaten zu generieren und in Datenbanken oder Suchmaschinen einzupflegen. In den letzten Jahren wurden sie auch verstärkt für die Generierung oder Fortsetzung von bekannten Daten oder Signalen eingesetzt. Es konnten klassische Musikstücke sinnvoll beliebig verlängert werden [30] oder bewegte Sequenzen aus einzelnen Bildern generiert werden [32]. Auch in der Postproduktion konnten Bildern einer Stil aufgeprägt werden [18].

In den folgenden Kapiteln wird in den Grundlagen (vgl. Kapitel 2) auf den allgemeine Ablauf in der Videoproduktion, sowie detailliert die Funktionsweise der NN und CNN, beschrieben. Im Anschluss werden in den folgenden Kapiteln verschiedene Entwicklungen von Videoproduktionsmitteln vorgestellt, welche verschiedene Formen von NN und im besonderen von CNN nutzen. Drei Ansätze werden detailliert beschrieben und bewertet. Der erste Ansatz [32] beschreibt in Anschnitt 4.2 die Generierung von einer bewegten Bildsequenz aus einem Einzelbild. Die anderen Ansätze [18] und [7] erläutern die Übertragung eines Bildstils auf eine andere Videosequenz.

2. Grundlagen

Um gezielter Ansatzpunkte für den Einsatz von NN in der Videoproduktion aufzeigen zu können, wird diese in Kapitel 2.1 kurz beschrieben. Der Schwerpunkt dieses Kapitels liegt jedoch auf den Grundlagen von NN und insbesondere CNN, sowie deren Training.

2.1. Videoproduktion

Mit der Bezeichnung Videoproduktion oder auch Filmproduktion wird die Herstellung von Kino-, Werbe- und Fernsehfilmen zusammengefasst. In Abbildung 1 ist ein Ablaufplan einer typischen Videoproduktion zu sehen. Da es alleine in Deutschland über 850 Produktionsformen gibt (Stand 2014) [25], kann der Ablaufplan nur einen sehr allgemeinen Überblick über die notwendigen Arbeitsschritte

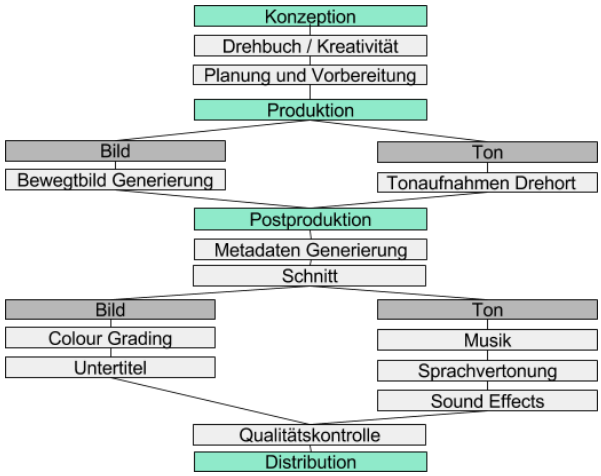

Abbildung 1: Grundlegende Arbeitsschritte einer Videoproduktion.

bieten.
Der erste Schritt, die Konzeption soll sowohl die Projektentwicklung, als auch die Vorproduktion zusammenfassen. Die sich anschließende Produktionsphase kann grob, wie im Schaubild zu dargestellt, in Bild und Ton unterteilt werden. Diese beiden Bereiche sollten nicht immer getrennt voneinander betrachtet werden. Die Postproduktion besteht aus vielen verschiedenen Arbeitsschritten, deren Schwerpunkt auf dem Schnitt und der digitalen Bildnachbearbeitung liegt. Der Schritt der Distribution ist hier der Vollständigkeit halber erwähnt und beschreibt die Filmverwertung.
Der Aufbau der folgenden Ausführungen orientiert sich an Abbildung 1 (vgl. Kapitel 3, 4 und 5).

2.2. Neuronale Netze

NN finden unter anderem Anwendung bei der Steuerung von Robotern, Börsenkursanalysen, Medizintechnik oder Fahrzeugsteuerung. In der Bildverarbeitung werden NN vor allem zur Klassifizierung genutzt.
Sie sind vom menschlichen Gehirn inspiriert, welches laut [9] ein nicht-lineares, komplexes und hoch paralleles System zur Verarbeitung von Informationen darstellt. Ähnlich wie dieses bestehen künstliche NN aus einer Menge an simulierten Neuronen, die untereinander verbunden sind und in Schichten organisiert sind. Es gibt verschiedene Arten der Vernetzung, die, wie in [9] und [24] beschrieben, in rück- und vorwärts gekoppelte Modelle unterteilt werden können.
Am häufigsten kommen sogenannte *Multilayer Perceptrons* (MLP) [2][19][24] zum Einsatz, welche eine einfache Systemarchitektur haben. Der generalisierter Aufbau ist beispielhaft in Abbildung 2 zu sehen. Dieses MLP besteht aus einer Eingabe- und Ausgabeschicht mit M bzw. K Neuronen und einer versteckten Schicht mit N Neuro-

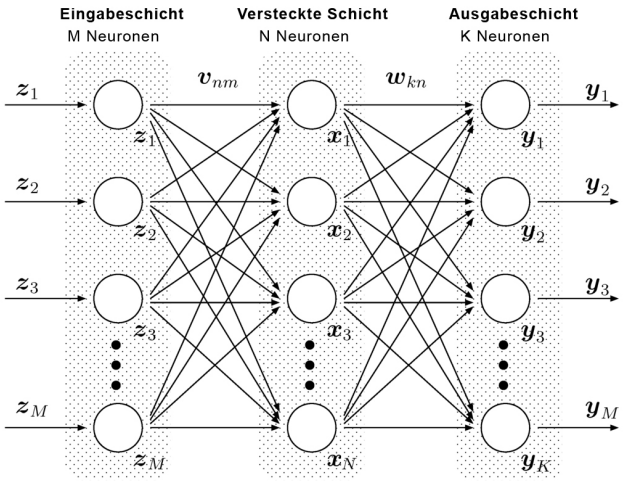

Abbildung 2: Prinzipieller Aufbau MLP nach [10].

nen. Es handelt sich um ein vorwärtsgekoppeltes Modell, bei welchem jedes Neuron einer Schicht mit jedem Neuron der darauffolgenden Schicht verbunden ist. Dies nennt man volle Verbindung. Im Bereich der Audiodatenverarbeitung finden jedoch immer häufiger rückgekoppelte Neuronale Netze (RNN) Anwendung.
Bei einem MLP dient die Eingangsschicht zum Verteilen der Daten z_m mit $m = 1, ..., M$. Die Ausgabe eines jeden Neurons in der versteckten Schicht, dargestellt durch x_n mit $n = 1, ..., N$, lässt sich durch Formel 1 berechnen. Hierbei steht v_{nm} für die jeweilige Gewichtung der Verbindungen zwischen den Neuronen der Eingabe- und der versteckten Schicht und f für die Aktivierungfunktion [24][9] des jeweiligen Neurons.

$$x_n = f\,(\sum_{m=1}^{M} v_{nm} z_m) \tag{1}$$

Die Ausgangswerte y_k, mit $k = 1, ..., K$, lassen sich äquivalent unter Berücksichtigung der Werte x_n und der Gewichte w_{kn}, sowie einer Aktivierungsfunktion g berechnen, und gelten als Vertrauenswerte. Sie müssen gemäß der Aufgabenstellung interpretiert werden.

2.3. Faltungsnetze

Im Folgenden wird detaillierter auf CNNs eingegangen, da diese die Grundlage für die meisten der hier vorgestellten Ansätze bilden. Vereinfacht ausgedrückt besteht ein CNN aus einer Vernetzung von Faltungsoperationen mit unterschiedlichen Filtermasken. CNNs kommen, bedingt durch ihre Architektur, oft zum Einsatz, wenn große Datenmengen von einem NN verarbeitet werden sollen. Ein schematischer Aufbau ist in Abbildung 3 zu sehen.
Jedes Pixel eines Feldes, das auf der Abbildung zu sehen ist, wird durch ein Neuron repräsentiert. Die Felder sind in Schichten organisiert. Die Eingangsschicht fungiert, vergleichbar mit den MLP aus Kapitel 2.2, als Verteiler der

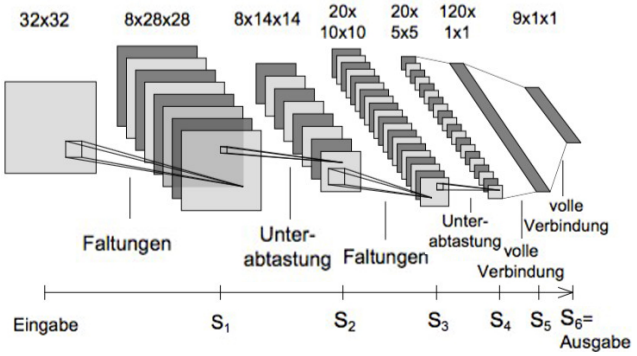

Abbildung 3: Prinzipieller Aufbau Faltungsnetz nach [21].

Information an die Neuronen der nächsten Schicht S_1. Die Besonderheit eines CNN sind die sich abwechselnd durchgeführte Faltung und anschließende Unterabtastung. Zwischen den Schichten S_4 und S_6 ähnelt das Modell einem MLP, da die Neuronen schichtweise voll verbunden sind.
Im Allgemeinen wird für eine Faltung eine Filtermaske h, also ein endlicher, zweidimensionaler Koeffizientensatz, wie in Formel 2 zu sehen, verwendet. Hierbei stehen x und y jeweils für die horizontale bzw. die senkrechte Bildkoordinate. Die Anzahl der Koeffizienten a_{xy}, wird in der Horizontalen mit N_{hx} und im Vertikalen mit N_{hy} bezeichnet.

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

Formel 3 beschreibt die Faltung eines Eingangssignals s mit einer Filtermaske h, wobei I das Ausgangssignal in Abhängigkeit von x und y beschreibt.

$$I(x,y) = (s * h)(x,y) = \sum_{m_x=-\infty}^{\infty} \sum_{m_y=-\infty}^{\infty} s(m_x, m_y) \cdot h(x - m_x, y - m_y) \tag{3}$$

Im Fall eines CNN wird die Faltung, wie in Abbildung 3, beispielsweise zwischen der Eingangsschicht und S_1 vollzogen und durch die Verbindung zwischen den Neuronen zweier Felder modelliert. Dabei entsprechen die Gewichte der Neuronen genau den Filterkoeffizienten a_{xy}. Für ein jedes Feld sind diese Koeffizienten konstant, was bedeutet, dass alle Neuronen eines Feldes mit nur einem Gewicht auskommen. Dieses Prinzip nennt man geteilte Gewichte.
Im CNN wird nach jeder Faltung eine Unterabtastung durchgeführt um zu gewährleisten, dass die Dimension der Eingangsdaten schrittweise an die Dimension des Ausgangsvektors angepasst wird. Hierzu wird meist eine bilineare Unterabtastung um den Faktor 2 vorgenommen. Allgemeiner betrachtet werden nxn Werte zu einem Wert zusammengefasst.
Wie zu Beginn des Kapitels erwähnt, haben CNNs gegenüber den MLPs den Vorteil, dass sie nahezu beliebig

hochskaliert werden können und somit gut geeignet für große Datenmengen sind. Dies liegt vor allem daran, dass die Neuronen nur lokal verbunden sind und sich somit das Prinzip der geteilten Gewichte zu Nutze gemacht werden kann. Ein weiterer Vorteil von CNNs, der vor allem in der Bildverarbeitung genutzt wird, ist das sie translationsinvariant sind.

2.4. Training Faltungsnetze

Meistens werden CNN mittels der *back-propagation* Methode trainiert. Bei dieser überwachten Lernmethode bedarf es einer großen Menge an vorab klassifizierten Trainingsdaten [15].
In den Faltungsschichten kann der Fehler der vorangegangen Schicht nach Formel 4 berechnet werden. Dabei steht E für den Fehler in der jeweiligen Schicht l gemacht wird. Während x^l für die Eingabe in die Schicht steht, bezeichnet y^l die Ausgabe der entsprechenden Schicht. Die Größe der Eingabe wird der Einfachheit halber als quadratisch, also mxm-groß angenommen. Die Gewichtung wird mit w bezeichnet. Um die Formel in der Realität anzuwenden, muss die linke und obere Grenze de Eingabeinhaltes, z.B. eines Bildes, mit Nullen ergänzt werden. Ansonsten wäre es nicht möglich den Fehler für Pixel zu berechnen, welche näher als m an den entsprechenden Rändern liegen.

$$\begin{aligned}\frac{\delta E}{\delta y_{ij}^{l-1}} &= \sum_{a=0}^{m-1}\sum_{b=0}^{m-1}\frac{\delta E}{\delta x_{(i-a)(j-b)}^{l}}\frac{\delta x_{(i-a)(j-b)}^{l}}{\delta y_{ij}^{l-1}} \\ &= \sum_{a=0}^{m-1}\sum_{b=0}^{m-1}\frac{\delta E}{\delta x_{(i-a)(j-b)}^{l}} w_{ab}\end{aligned} \tag{4}$$

Die Schichten, in denen die Unterabtastung stattfindet, leisten kaum Beitrag zum eigentlichen Lernprozess des CNN. Hier wird das Problem allerdings reduziert, da nxn Werte in einem einzigen resultieren.
Weil alle Gewichtungen w mittels des *back-propagation* Algorithmus während des Trainings angepasst werden, können CNN laut LeCun als Erzeuger ihrer eigenen Merkmalsextraktion gesehen werden [14].

3. Konzeption

Vor der eigentlichen Produktion von Videomaterial muss das Projekt zunächst konzipiert und detailliert geplant werden. Dies bezieht sich vor allem auf die kreativen Prozesse des Drehbuchschreibens und die darauffolgende gesamte Projektplanung und -vorbereitung. Naturgemäß sind NN weniger sinnvoll für die Planung und das Management von Projekten. Doch in den letzten Jahren wurde versucht mit Hilfe von NN kreative Prozesse umzusetzen, wie sie für das Schreiben von Drehbüchern benötigt werden. Es gibt bereits erste Versuche mit Hilfe von NN automatisch sinnvolle Texte und Dialoge zu erstellen, die auf bekannten Texten und Storylines basieren [27]. Dieses Verfahren wird in der Literatur auch *Deep Writing* genannt. Es können auch Romane, Dialoge oder Songtexte automatisch mit einem RNN erstellt werden [4].

3.1. Aktueller Stand

Ein solches RNN wurde mit allen bekannten Episoden der Serie *Silicon Valley* trainiert [5]. Zu Beginn des Schreibprozesses wird ein beliebiges Wort genutzt um den ersten Satz zu initialisieren. Das NN sucht im Anschluss das Wort, welches am häufigsten nach dem Startwort in den Trainingsdaten genannt wurde. Mit diesem Wortpaar wird nach dem selben Prinzip das dritte Wort des Textes ermittelt. Dieser Prozess wird solange wiederholt bis der generierte Text die gewünschte Länge erreicht hat [4]. Die generierten grammatikalisch korrekten Sätze beziehungsweise Dialoge ergeben keine sinnvollen Zusammenhänge und folgen keiner ersichtlichen Storyline. Der Film *Sunspring* [20] wurde nach einem ähnlichen Prinzip erstellt und im Anschluss von einem professionellen Filmteam realisiert. Der größte Unterschied zu [5] ist, dass das NN nicht nur Wörter unterscheidet. Stattdessen wird zunächst alles in Buchstaben zerlegt und dann in neue Wörter und Sätze zusammengesetzt. Das verwendete NN nannte sich selber *Benjamin* und wurde mit einer großen Anzahl an Drehbüchern von Science Fiction Filmen aus den 80er und 90er Jahren trainiert. Heraus kam ein Drehbuch, welches qualitativ mit den Resultaten des ersten Ansatz vergleichbar ist. Ein auffälliges Problem war der korrekte Umgang mit Namen. Das CNN konnte nur Charaktere unterscheiden, welche nur mit einem Buchstaben benannt wurden.

Selbst mit einer sehr großen Anzahl von Trainingsdaten konnten keine kohärenten Dialoge generiert werden und keine konstante und neue Storyline verfolgt werden. Daraus lässt sich ableiten, dass zum jetzigen Zeitpunkt NN große Schwierigkeiten haben neue Ideen zu entwickeln und Erlerntes innerhalb größerer Zusammenhänge sinnvoll zu verknüpfen. Deshalb kann derzeit der Aufwand des Drehbuchschreibens durch NN nicht merklich reduziert werden.

4. Produktion

Die Produktion ist von allen drei besprochenen Kategorien diejenige, die bisher am wenigsten automatisiert durchgeführt werden kann. Moderne Kameras- und Tonaufnahmesysteme bieten zwar eine einfachere Handhabung und Fehlerkontrolle, als es noch bei der analogen Aufzeichnung der Fall war, dennoch braucht eine Filmproduktion viele menschliche Arbeitsstunden.
Im Folgenden soll geprüft werden, ob es heutzutage realisierbare Automatiersungsansätze gibt. Dabei wird ober-

flächlich auf die Generierung von Musik eingegangen, die später in der Postproduktion unter die einzelnen Szenen gelegt werden kann. Daran schließt sich eine ausführlichere Betrachtung eines Ansatzes zur automatisierten Generierung von Szenendynamiken an.

4.1. Aktueller Stand Musikgenerierung

Es gibt verschiedene Ansätze NN zu nutzen, um Musik automatisiert generieren zu lassen. Im Folgenden werden drei Ansätze kurz vorgestellt, welche alle RNN benutzen.

An der *University of Washington* ist im Rahmen einer Projektarbeit ein Musik Generator namens *Algo Rythm* entstanden [28]. Für die Umsetzung haben die Studierenden mehrere RNN mit klassischer Musik abgespeichert im XML-Datenformat trainiert. Der Quellcode kann auf *Github* [29] eingesehen werden.
Ein weiterer Ansatz, der in [6] beschrieben wird, arbeitet ebenfalls mit einem RNN, welches mit einem Autokorrelation-basierten Prädiktor kombiniert wird. Dabei soll die Struktur von Musikstücken erlernt werden, indem zunächst die folgende Note einer Tonreihenfolge vorausgesagt wird.

Der letzte Ansatz kann ebenfalls auf Grundlage einer Notensequenz ein Musikstück komponieren [3]. Dabei wird die Eingabesequenz zunächst interpretiert. Anschließend sorgen zwei RNN-basierte Algorithmen für die Produktion von sowohl Rhythmus, als auch die Vorhersage der nächsten Note.

Alle drei Ansätze weisen hohes Potential auf, wenn es darum geht Musikstücke automatisch zu generieren. Über das Genre des zugeführten Musikmaterials lässt sich die gewünschte Ausgabe begrenzt steuern. **Selbsterklärend ist der persönliche Einfluss auf das Resultat hat wesentlich geringer, als bei selbst produzierter Musik.** Diese Methoden können also nur dann zum Einsatz kommen, wenn die zu unterlegende Szene noch in Planung ist und zumindest geringfügig auf die Musik angepasst werden kann. Denkbar wäre es die automatisch generierten Musikstücke in einer Musikdatenbank zugänglich zu machen, sodass Ausschnitte nach belieben Verwendung in Postproduktion finden könnten.

4.2. Generierung von Videos mit Szenendynamiken

Mit dem Ansatz von Vondrick et al. [32] können aus Einzelbildern ganze Szenendynamiken erstellt werden, welche für die Klassifizierung oder für die Vorhersage von Bewegungen genutzt werden können. Das Resultat sind kleine

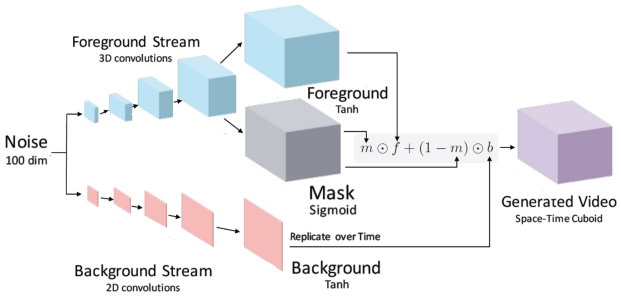

Abbildung 4: Aufbau Generator nach [32].

Bildsequenzen mit einer Auflösung von 64x64 Pixeln und 32 Einzelbildern.
Die Grundlage für dieses Verfahren bilden zwei CNN, die als *Generative Adversarial Networks* (GAN) [8] verstanden werden. Da beide Netze als Gegenspieler fungieren, trainieren sie sich selber und man spricht, anders als bei der *back propagation* (vgl. Kapitel 2.4), von unüberwachtem Lernen. Während der Generator G für die Generierung von einer Videosequenz auf Grundlage eines Standbildes zuständig ist, soll der Diskriminator D zwischen den so erzeugten und realen Videosequenzen differenzieren können. Um dies zu erreichen werden beide GAN, wie in Formel 1 des Papers [32] zu sehen gegeneinander trainiert. Das Minimierungs- bzw. Maximierungsproblem wird hierzu mittels Gradientenverfahren gelöst.
Da ähnliche Objekte in der Regel auch ähnliche Bewegungsmuster aufweisen, lassen die Autoren das zur Verfügung stehende Videomaterial in Kategorien (Strand, Babys, Golf und Züge) einteilen. Zudem sorgt ein Stabilisierungsalgorithmus dafür, dass bei dem gesamten Testmaterial von einer statischen Kamera ausgegangen werden kann und die zu erlernende Szenendynamik nicht durch eine Bewegung der Kamera verfälscht wird. Für jede dieser Kategorien wird dann ein eigenes Pärchen bestehend aus Generator und Diskriminator trainiert.
Bevor die beiden CNN G und D auf die beschriebene Art trainiert werden können, müssen sie erst einmal ihren Aufgaben entsprechend aufgebaut werden.
Den Aufbau des genutzten Generators [32] ist in Abbildung 4 zu sehen. Im Gegensatz zu dem in Kapitel 2.3 beschriebenen, gewöhnlichen CNN, wird hierbei keine Unterabtastung vorgenommen, sondern die Menge der Daten von Schicht zu Schicht erweitert. Zudem werden die Eingabedaten für die Verarbeitung in zwei Datenströme aufgeteilt. Während in dem einen Strom der statische Hintergrund entsprechend vergrößert wird, wird in dem anderen die Bewegung des sich im Vordergrund befindlichen Objektes vorhergesagt. Beide Teile werden dann am Ausgang nach Formel 5 zusammengefasst.

$$G(z) = m(z) \odot f(z) + (1 - m(z)) \odot b(z) \qquad (5)$$

Dafür ist die binäre Maske m so gewählt, dass sie überall den Wert eins hat, wo der Vordergrund übernommen wer-

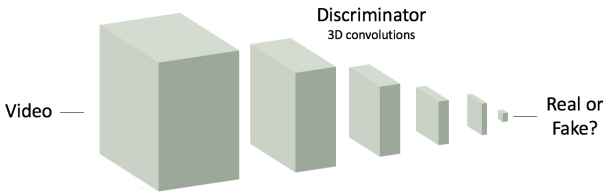

Abbildung 5: Aufbau Diskriminator nach [32].

den soll und ansonsten nur Nullen enthält.
Der Diskriminator hat eine typische Faltungsnetzarchitektur, die in Abbildung 5 zu sehen ist. Dabei muss seine Eingabeschicht eben so groß sein, wie die Ausgabeschicht des Generators.
Die Bedeutung für so erzeugte Bewegtbildsequenzen für die Automatisierung der Produktion von Filmen, wird im nächsten Kapitel diskutiert. Die Resultate können auf der Internetseite der Autoren eingesehen werden [31].

4.3. Bewertung des Ansatzes von Vondrick

Das in Kapitel 4.2 beschriebene Verfahren schafft es automatisiert aus einem Einzelbild eine kurze und niedrig aufgelöste Bewegtbildsequenz zu erstellen. Hierbei wird vor allem die Szenendynamiken in den meisten Fällen annähernd realistisch vorhergesagt, wobei die Modellierung des sich bewegenden Bildinhaltes eher mangelhaft ist. Hier kommt es zunehmend zu Artefakten, die sich in Traubenförmigen Pixelanhäufungen äußern.
Trotz des hohen Bedarfs an Trainingsmaterial und den damit verbundenen Zeitaufwand, kann das System ohne menschliche Hilfe eigenständig Kategorisieren, Trainieren und letztendlich Szenendynamiken generieren. Jedoch muss das Trainingsmaterial für jede Kategorie bewusst gewählt werden, da Dopplungen in den Trainingsdaten zu ungewollten Resultaten führen können. Es muss auch bedacht werden, dass für jede Kategorie ein eigenes Pärchen bestehend aus Generator und Diskriminator trainiert werden muss, was dank des unüberwachten Lernens lediglich einen hohen Zeit-, sowie Rechenbedarf bedeutet.
Gegen einen Einsatz in der Videoproduktion sprechen die geringe Auflösung und Zeitspanne die bisher generiert werden kann. Zudem kann in einer Szene nicht immer nur von einem sich bewegenden Objekt ausgegangen werden. Sieht man sich die Resultate in der Kategorie Golf an, wird schnell klar, dass Menschen, die sich für den Betrachter logisch in unterschiedliche Richtungen bewegen, nach Anwendung des Algorithmus oft miteinander zu verschmelzen scheinen. Die Individuen scheinen als ein ganzes wahrgenommen zu werden und die resultierende Szenendynamik ist nicht realistisch bzw. brauchbar.
Dennoch könnte der Ansatz als Grundlage für einen ausgereifteren Algorithmus dienen, der beispielsweise die Anweisungen des Drehbuchs mit einbezieht und somit bessere Orientierung entlang der Storyline gewährleitest. Kann man das so schreiben?? - Schwierig, kann man parallel Texte und Videodaten in ein Netz trainieren?- Sonst ist der Satz so okay.
Der jetzige Stand des Algorithmus könnte beispielsweise für die Generierung von GIFs oder Simulationen genutzt werden. Zudem könnte der Ansatz nach einer geeigneten Anpassung Bewegungsvektoren aus einem Standbild vorhersagen, welche in der Videocodierung Einsatz finden

5. Postproduktion

In diesem Kapitel wird die Verwendung von NN in der Postproduktion beschrieben. Gerade in diesem Produktionsabschnitt finden NN eine breites vorstellbares Anwendungsfeld, da die CNN im Bereich der Bildverarbeitung in den letzten Jahren sehr erfolgreich eingesetzt werden konnten. Im ersten Abschnitt werden zunächst die interessantesten auf NN-basierenden Ansätze für die Postproduktion vorgestellt und bewertet. Im Anschluss werden zwei Verfahren zur Stilsynthese detailliert beschrieben und in Hinsicht auf den Einsatz in der professionellen Videoproduktion analysiert. Zuletzt wird ein Ausblick auf Einsatzmöglichkeiten gegeben.

5.1. Aktueller Stand

Für einige Produktionen, wie Dokumentationen oder Nachrichtensendungen, werden *Voice-Over* benötigt. Teilweise werden diese bereits mit sogenannten *Text-to-Speech*-Verfahren produziert, doch die Stimmen klingen meist sehr künstlich.
Ein vielversprechender Ansatz ist *WaveNet*, welcher im Gegensatz zu den meisten anderen NN-basierten Ansätzen ein CNN verwendet. Dieses wird mit Text- und Audiodatenpaaren von realen Sprechern trainiert und die künstlich erzeugten Stimmen klingen sehr natürlich und menschlich [30]. Das *Deep Voice* Verfahren [1] entwickelt das *WaveNet* Verfahren weiter. Während *WaveNet* mehrere Minuten Rechenzeit benötigt um eine Sekunde an Audiodaten zu generieren, ist *Deep Voice* Realtime-fähig und ein eigenständiges System. Dadurch ist es für die Nutzung in der Videoproduktion interessanter. Einen Vergleich bezüglich der Klangqualitätsunterschiede zwischen den beiden Verfahren gibt es in der Literatur leider nicht.
Nachdem alle notwendigen Daten produziert wurden sollten diese gekennzeichnet und in eine entsprechende Mediendatenbanken eingepflegt werden. Die Kennzeichnung der Daten kann mit Hilfe von NN zuverlässig automatisiert werden, da in der Objektklassifikation mit CNN in den letzten Jahren eine sehr geringe Fehlerquote

und hohe Robustheit erreicht werden konnte. Eine bekannte Bibliothek ist *Clarifai* [22], welche konfigurierte CNN anbietet um optimierte Bild- oder Videodatenbanken anzulegen und zu verwalten. Weitere aktuelle Veröffentlichungen [34][33][12] konzentrieren sich auf die Verbesserung der Leistungsfähigkeit um auch große Datenmengen robust zu kennzeichnen. Dies ist für die große Menge an Videodaten einer Filmproduktion ein wichtiges Kriterium. Zum jetzigen Zeitpunkt erfordert die Erkennung dieser Mengen noch sehr große Rechenkapazitäten und ist entsprechend kostenintensiv.

Neben der verbreiteten Klassifikation können CNN auch Bilddaten generieren und den Stil eines Bildes übertragen. Zwei vielversprechende Stilsynthese-Ansätze sind die bereits erwähnten *Deep Dream* [18] und das *Deep Style* Verfahren [7]. Im Gegensatz zu anderen bekannten Stilsynthese-Verfahren können diese CNN-basierten Verfahren die gesamte Bildsemantik eindeutig vom Inhalt trennen und mit einem anderen Bildes verschmelzen ohne dessen Inhalt zu verändern [17]. Diese beiden Verfahren werden in den nächsten Abschnitten genauer beschrieben.

5.2. Faltungsnetze zur Stilsynthese

Das Verständnis der folgenden Algorithmen in Abschnitt 5.3 und 5.4 setzt ein grundlegendes Verständnis der Prozesse innerhalb eines CNN voraus. In Abbildung 3 wird deutlich, dass mit jeder Schicht das Bild stärker unterabgetastet wird und die Größe der Merkmalsvektoren stetig steigt. Mit jeder weiteren Schicht wird ein größer Ausschnitt des Bildes erfasst und dessen Merkmale extrahiert. Die Merkmalsvektoren werden größer, da ein größerer Bereich mehr Merkmale enthält und diese folglich immer komplexere Merkmale enthalten.
Im oberen Bereich der Abbildung 6 wird veranschaulicht, dass die stilgebenden Merkmale immer mehr der Semantik des Stilbildes annähern. Während in der ersten Schicht die Stilmerkmale sehr klein sind, können in den höherwertigen Schichten Stilelemente aus dem gesamten Bild gefunden werden, wie zum Beispiel die gelben Kreise und die Strudel im Himmel. Im untereren Abbildungsbereich sind die inhaltsbezogenen Merkmale dargestellt. Hier ist die Unterabtastung klar zu erkennen, doch das Gebäude bleibt stets identifizierbar. Die Inhaltsinformationen bleiben trotz der Unterabtastung erhalten.

Für ein besseres Verständnis der verborgenen Schichten wurde der Prozess zur Objekterkennung invertiert. Das trainierte CNN wurde mit einem Eingangsbild initialisiert, welches ausschließlich zufälliges Rauschen enthält. Dieses Eingangsbild wurde iteriert bis ein Objekt klassifiziert werden konnte. In diesem Bild konnten teilweise eindeutige Merkmale dieses Zielobjektes ausgemacht werden. An dieser Stelle sei hinzugefügt, dass das CNN ein Objekt nicht immer präzise bestimmen kann, da die Klassifikation stark von den Trainingsdaten abhängt. Zum Beispiel bei der Bestimmung von Hanteln kann es dazu kommen, dass auch Arme zum Merkmalsvektor hinzugefügt werden, da diese überdurchschnittlich häufig auf den Trainingsdaten mit den Hanteln zu sehen waren. Trotzdem lässt sich eindeutig nachweisen, dass CNN auch für die Generierung von Bilddaten genutzt werden und nicht ausschließlich für die Klassifizierung [18][16].

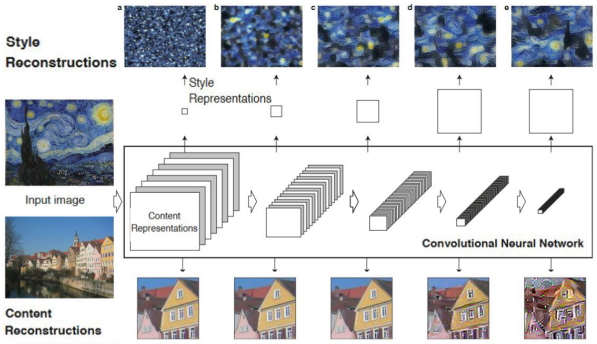

Abbildung 6: Merkmalsextrahierung in einem Faltungsnetz[7].

Ein anderer Ansatz zur Analyse von CNN lässt es selbst entscheiden, welche Merkmale verstärkt werden. Ein beliebiges Eingangsbild wird durch das CNN propagiert und die Merkmalsvektoren der einzelnen Schichten untersucht. In den niedrigeren Schichten können nur primitive Texturmerkmale ausgemacht werden, wie Linien oder Rechtecke, während der Bildinhalt unverändert bleibt. Mit steigender Schichtanzahl werden die verstärkten Muster immer komplexer und größer während der Bildinhalt teilweise modifiziert wird. So wird deutlich, dass CNN in den höheren Schichten die benötigten Informationen der Semantik enthalten und diese eindeutig vom Bildinhalt separiert.

5.3. Deep Dream

Der folgende Abschnitt stellt das Stilsynthese-Verfahren *Deep Dream* [18] vor, welches ein Nebenprodukt des Versuches den Inhalt eines CNN zu visualisieren ist. *Deep Dream* basiert auf einem *GoogLeNet* CNN mit 22 Schichten und fünf Unterabtastungsschichten [26]. Die Schichten haben eine besondere Architektur, welche *Inception Layer* genannt wird. Mit Hilfe dieser Architektur können auch große Merkmalsvektoren effizient verarbeitet werden. Dieses *GoogLeNet* wird im ersten Schritt mit dem *ImageNet* [13] Trainingsdatenset trainiert.
An dieser Stelle sei zum besseren Verständnis hinzugefügt, dass ein wichtiger Unterschied zu den bereits bekannten Faltungsnetzmodellen aus Abschnitt 2.3 besteht. Die

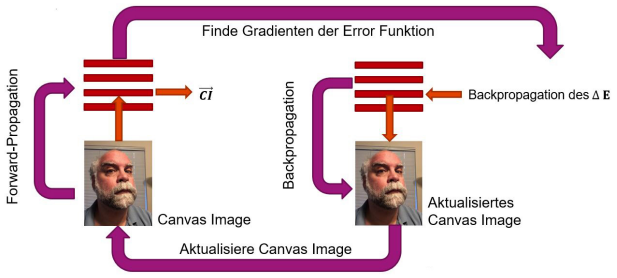

Abbildung 7: Ablaufdiagramm des Deep Dream Verfahrens.

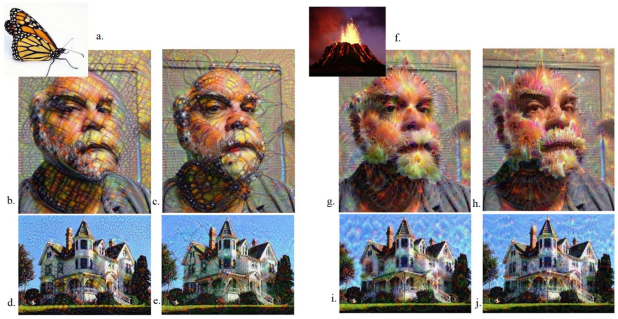

Abbildung 8: Resultate des Deep Dream Verfahrens [17].

Schichten zur Klassifizierung sind für die Stilsynthese nicht weiter relevant. Nur die Faltungsschichten werden verwendet, welche die benötigten Merkmalsvektoren extrahieren und speichern.
Zunächst wird das stilgebende Bild, im folgenden Style Source (SS) genannt, bis zu einer beliebigen Faltungsschicht vorwärtspropagiert und der entsprechende Merkmalsvektor als $\overrightarrow{GS}$ gespeichert. Dieser enthält die Merkmale der gewünschten Semantik und steuert die Transformation in Richtung des gewünschten Stils. Würde $\overrightarrow{GS}$ entfallen nähert das CNN die Transformation an die Trainingsdaten an. Bezogen auf Abschnitt 5.2 empfiehlt es sich höhere Schichten zu wählen um eine zufriedenstellendes Synthese zu generieren.
Den weiteren Transformationsprozesses stellt Abbildung 7 schematisch dar. Das Zielbild, im folgenden Canvas Image (CI) genannt, wird iterativ transformiert. Zu Beginn einer Iteration wird es bis zur selben Schicht wie $\overrightarrow{GS}$ vorwärtspropagiert und als $\overrightarrow{CI}$ gespeichert. Im Anschluss wird der Gradient der Fehlerfunktion ΔE aus Gleichung 6 berechnet. Der Fehler ΔE wird demnach maximal, wenn die beiden Vektoren in die selbe Richtung zeigen.

$$E = \overrightarrow{GS} * \overrightarrow{CI} \tag{6}$$

Der resultierende ΔE wird zurückpropagiert und die Gewichtungen innerhalb des Netzes in Richtung des SS korrigiert. Das daraus entstehende Bild wird als neues CI aktualisiert und für die nächste Iteration wiederverwendet. Dieser Ablauf terminiert, wenn der Fehler minimal ist und $\overrightarrow{CI}$ möglichst weit dem Ausgangsbild $\overrightarrow{GS}$ angenähert ist. In diesem Fall ist E maximal und der gesamte Prozess wird als Maximierung von Gleichung 6 interpretiert.

Einige Ergebnisse des *Deep Dream* Verfahrens sind in Abbildung 8 dargestellt. Die kleinen Bilder in der oberen Ecke sind die verwendeten SS. Auf der linken Seite eines Blockes sind jeweils die Ergebnisse mit einer niedrigen Schicht und rechts die einer höheren Schicht dargestellt. Auf den ersten Blick fällt direkt auf das die linken Bilder stärker kleinere Merkmale skalieren und übernehmen. Rechts werden eindeutig größere Texturen berücksichtigt, wie zum Beispiel die Form des Schmetterlings oder Vulkans am Hals. Gleichzeitig fällt rechts auf, dass markante Bildpunkte, wie die Augen oder die Nase, in der Form modifiziert wurden. Somit gehen teilweise Inhalte des ursprünglichen Bildes verloren. Die Nase wurde teilweise einer Hundeschnauze angenähert, da die Trainingsdaten von *ImageNet* [13] einen Überschuss an Hundemotiven enthalten. Besonders deutlich wird diese Tendenz, wenn kein SS verwendet wird [17]. Andere Ergebnisse zeigten auch deutlich, dass durchaus auch die Merkmale der Trainingsdaten gegenüber denen des SS überwiegen können.
Es ist demzufolge für den Nutzer nicht möglich die Endergebnisse zu beeinflussen. Die einzige Möglichkeit der Kontrolle bietet die Wahl des SS und der Trainingsdaten, welche einen nicht vorhersehbaren Einfluss auf das Endergebnis haben. Daher kann von keiner strikten Trennung von Semantik und Inhalt ausgegangen werden.
Weiter kann nicht gewährleistet werden die Ergebnisse zu reproduzieren und ähnliche Bilder auf kohärente Weise zu transformierten, da die Gewichte des CNN stetig während einer Iteration modifiziert werden. Diese Reproduzierbarkeit und Kohärenz ist ein wichtiges Kriterium für die Videoproduktion, da eine Sequenz von ähnlichen Bilder entsprechen gleichförmig transformiert werden muss. Aus diesem Grund ist dieses Verfahren mehr eine neue Kunstform und interessante Weise die Funktionalität von CNN besser zu verstehen. Für die Videoproduktion ist sie gänzlich ungeeignet.

5.4. Deep Style

Gerade die mangelnde Einflussnahme und die fehlende Reproduzierbarkeit des Nutzers beim *Deep Dream* Verfahren [18] verlangt nach einer Weiterentwicklung. Diese ist durch das *Deep Style* Verfahren [7] gegeben, welche ein ähnliches Schema verfolgt. Der große Unterschied ist die strikte Trennung und Berücksichtigung von stilgebenden und inhaltsbezogen Informationen, einer erweiterten Fehlerfunktion und der Initialisierung durch ein Bild mit zufälligem Rauschen.

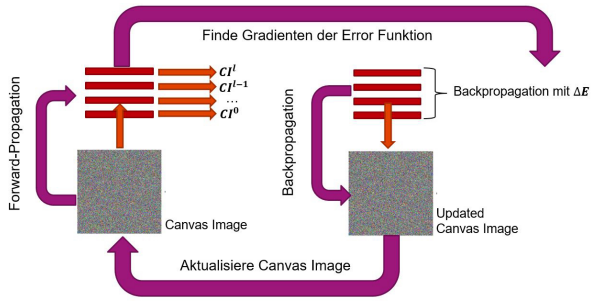

Abbildung 9: Ablaufdiagramm des Deep Style Verfahrens.

Zusätzlich wird eine andere CNN-Architektur verwendet. Es wird ein trainiertes CNN verwendet, dessen Architektur von der *Visual Geometry Group* entwickelt wurde [23]. Es besteht aus 16-19 Schichten, welche in erster Linie aus 3×3 Faltungskernen besteht. Dieses CNN wurde ebenfalls mit dem *ImageNet* Trainingsdatenset [13] trainiert.

Nach dem Training wird zunächst ein SS bis zur letzten Faltungsschicht vorwärtspropagiert. Für jede Schicht wird im Gegensatz zu *Deep Dream* der entsprechende Merkmalsvektor $\overrightarrow{GS}$ abgespeichert. Diese Vektoren beinhalten die Stilmerkmale aller Abstraktionsebenen, welche für die spätere Berechnung des Fehlergradienten benötigt werden. Im Anschluss wird das Zielbild, welches im folgenden Content Source Image (CS) genannt wird, bis zur letzten Schicht propagiert. An dieser Stelle wird nur der Merkmalsvektor $\overrightarrow{GC}$ der letzten Faltungsschicht berücksichtigt.

Abbildung 9 zeigt den schematischen Ablauf nach der erfolgreichen Initialisierung. Wie bereits erwähnt, beinhaltet das Canvas Image (CI) zufälliges Rauschen, welches während jeder Iteration in die Richtung von $\overrightarrow{GC}$ und $\overrightarrow{GS}$ transformiert wird. Das CI wird ebenfalls vorwärtspropagiert und mit den resultierenden Merkmalsvektoren $\overrightarrow{CI}$ aus jeder Schicht die Korrelation G zwischen den einzelnen Merkmalsvektoren innerhalb einer Schicht ermittelt, wie in Gleichung 7. In dieser Gleichung ist l die Anzahl Schichten im VGG. Dieser Schritt ist der erste zur Fehlerberechnung der Stilmerkmale in Gleichung 11.

$$G_{ij}^{l} = \frac{1}{2}\sum_{k=0}(\overrightarrow{CI_{ik}} - \overrightarrow{CI_{jk}})^2 \tag{7}$$

Im Anschluss wird mit diesen Korrelationen G_{ij}^{l} jeweils pro Schicht die quadrierte Fehlerfunktion F_l aus Gleichung 8 berechnet. In diesem Fall bezeichnet N die Anzahl der Merkmalsvektoren in einer Schicht und M deren Größe. Für den gesamten Fehler des Stils E_{style} werden alle F_l wie in Gleichung 11 gewichtet aufsummiert.

$$F_l = \frac{1}{2N_l^2 M_l^2}\sum_{k=0}^{N}(G_{ij}^{l} - \overrightarrow{GS_{ij}^{l}})^2 \tag{8}$$

$$E_{style} = \sum_{k=0} w_l F_k \tag{9}$$

Abbildung 10: Resultate des Deep Style Verfahrens [17].

Neben E_{style} wird auch $E_{content}$ benötigt, doch dessen Berechnung ist weniger komplex. $E_{content}$ ist als simple quadrierte Fehlerfunktion aus $\overrightarrow{GC}$ und den $\overrightarrow{CI}$ der letzten Schicht definiert.

$$E_{content} = \frac{1}{2}\sum_{i,j}(\overrightarrow{GC_{ij}} - \overrightarrow{CI}_{ij})^2 \tag{10}$$

Die gesamte Fehlerfunktion E ergibt sich aus der gewichteten Summe von $E_{content}$ und E_{style}. Die beiden Gewichte α und β ermöglichen dem Nutzer das Verhältnis zwischen Stil und Inhalt zu modifizieren und somit das Endergebnis seinen Vorstellungen anzupassen.

$$E = \alpha E_{content} + \beta E_{style} \tag{11}$$

Von E wird, wie beim *Deep Dream* Verfahren, auch der Gradient bestimmt um anschließend eine Rückwärtspropagation durchzuführen. Das weitere Ablaufschema gleicht dem des *Deep Dream* und terminiert, wenn die Merkmalsvektoren von CI sich an die Initalisierungsvektoren von SS und CS angenähert haben.

Generell ermöglicht das *Deep Style* Verfahren eine Stilsynthese ohne den Verlust von Bildinhalt (vgl. Abbildung 10). Die Semantik und der Bildinhalt werden gut getrennt, wobei es unter Umständen vorkommt, dass die Trainingsdaten einen Einfluss auf das Endergebnis haben. Dieser Einfluss ist dennoch deutlich geringer als beim *Deep*

Style Verfahren [17]. Die Ergebnisse auf der rechten Seite der Abbildung haben die Semantik kohärent adaptiert ohne die Gesichtszüge der Person zu modifizieren. Die Farbwelt des SS wurde sinnvoll in die des CS transformiert.

Die Gewichtung von Stil und Inhalt in E ist skalierbar und ermöglicht dem Nutzer eine bessere Kontrolle über das Endergebnis. Generell wirken die Ergebnisse weniger zufällig und unabhängiger von den Trainingsdaten. In [17] wurden neben den klassischen Gemälden auch photorealistische Bilder synthetisiert. Diese Resultate übertragen das Farbschema des SS nachvollziehbar auf das CS. Leider zur Zeit noch mit signifikanten Artefakten. An dieser Stelle besteht noch ein Optimierungsbedarf durch ein Pre- und Postprocessing oder einer geringeren Schichtanzahl. Dies könnte bei einer Übertragung des Farbschemas sinnvoll sein, da die Merkmalsvektoren weniger komplexe Strukturen enthalten.

Ein weiteres wichtiges Kriterium ist die Reproduzierbarkeit und die Kohärenz von ähnlichen Bildern. Um eine Reproduzierbarkeit zu gewährleisten muss stets von einem CNN mit der selben Initialisierung ausgegangen werden. Die Gewichtungen in einem CNN können durch ein erneutes Training nicht exakt reproduziert werden, deshalb ist es notwendig das CNN nach jeder erfolgreichen Synthese mit den selben Gewichtungen zu initialisieren. Bekannte Bibliotheken, wie zum Beispiel *Caffe* [11], bieten diese Möglichkeit an. Die Kohärenz von aufeinanderfolgenden Bilder einer Videosequenz sollte möglich sein unter der Bedingung dass die Trainingsdaten einen äußerst geringen Einfluss haben und die Parameter α und β konstant sind. Durch die naturgemäße konstante Beleuchtungssituation innerhalb einer Szenensequenz verfügen die CS über ähnliche Stilmerkmale und nur die Inhaltsmerkmale verändern sich teilweise. Letztere werden bekanntermaßen nur sehr gering modifiziert. Über die konkreten Unterschiede zwischen ähnlichen Bildern gibt es in der Literatur derzeit keine Aussagen.

Der Rechenaufwand steigt exponentiell mit der Bildauflösung. Derzeit ist eine Stilsynthese mit einer Auflösung von 512×512 Pixel für ein Einzelbild möglich, welche nicht ausreichend für die Videoproduktion ist. Aus diesem Grund ist die kostenintensive Nutzung von Renderfarms notwendig um eine ausreichende Qualität zu gewährleisten.

Mit vordefinierten Presets aus trainierten CNN und diversen CS ist auch eine Verwendung im Consumerbereich vorstellbar ohne dass der Nutzer detaillierte Kenntnisse über den Algorithmus besitzen muss. Generell ist das *Deep Style* Verfahren sehr vielversprechend für die Videoproduktion und könnte mit einigen Optimierungen bei der Stilsynthese mit photorealistischen Bildern und der Nutzung einer Renderfarm einige Prozesse in der Postproduktion vereinfachen.

5.5. Ausblick

In der Zukunft kann das *Deep Style* Verfahren für verschiedene Prozesse in der Postproduktion sinnvoll sein. Zunächst sei hier die qualitativ hochwertige und automatische Coloration und Texturierung von animierten Filmen ohne die aufwendige manuelle Texturierung von einzelnen 3D Objekten genannt.
Die Rekonstruktion und Restauration von verlorenen Filmmaterial, wie zum Beispiel *Metropolis* von Fritz Lang ist auch vorstellbar. Da verlorene Szenen nachgestellt und im Anschluss an den Stil der verbliebenen Szenen angepasst werden könnten.
Auch eine individuelle Anpassung an den Geschmack und Stimmung des Nutzers von Filmen ist für *Video-on-Demand* denkbar. Gerade die Farbgebung beeinflusst die Wahrnehmung des Zuschauers. So könnte zum Beispiel die Bedrohlichkeit eines Thrillers verschärft oder minimiert werden. Dasselbe gilt auch für Computerspiele, welche unter Umständen an Geschmack von jüngerem oder erwachsenem Publikum angepasst werden könnte.

6. Fazit

Die vorgestellten Verfahren befinden sich noch am Anfang ihrer Entwicklung und erfordern noch weitreichende Optimierungen. Gerade der Ansatz zur Generierung von Szenendynamiken [32] scheint noch ausbaufähig zu sein. Dennoch bietet der Ansatz eine Grundlage für weitere Bemühungen in Richtung automatisierter Bewegtbildgenerierung. Zudem kann der Ansatz viele wichtige Erkenntnisse für die Weiterentwicklung von Videokodierungsverfahren liefern.

Das *Deep Dream* Verfahren [18] scheint aufgrund nicht reproduzierbarer Ergebnisse und mangelnder Kontrolle durch den Nutzer für eine Verwendung in der Postproduktion uninteressant zu sein. Erfolgversprechender ist dagegen die Weiterentwicklung *Deep Style*, welche die Semantik eines Bilder vom Inhalt trennen und übertragen kann. Nur die Synthese von photorealistischen Bildern enthält noch sichtbare Artefakte.

Zusammengefasst zeigen die Verfahren, dass NN durchaus einen sinnvolle Verwendungszweck in der Videoproduktion haben können und es lohnenswert ist, diese weiterzuverfolgen. Mit ihrer Hilfe könnten unter Umständen einzelne Arbeitsschritte in der Produktion und Postproduktion automatisiert werden, wenn die Verfahren weiterentwickelt und die Rechenkapazität weiter gesteigert wird. Lediglich die Automatisierung von kreativen Prozessen und der Generierung von vollständig neuen Ideen, wie zum Beispiel das Drehbuchschreiben, scheint zum jetzigen Zeitpunkt wenig vielversprechend.

Literatur

[1] S. Ö. Arik, M. Chrzanowski, A. Coates, G. Diamos, A. Gibiansky, Y. Kang, X. Li, J. Miller, J. Raiman, S. Sengupta, and M. Shoeybi. Deep voice: Real-time neural text-to-speech. *CoRR*, abs/1702.07825, 2017.

[2] H. Braun. *Neuronale Netze Optimierung durch Lernen und Evolution*. Springer, 1997.

[3] C. Browne. System and method for automatic music generation using a neural network architecture, Oct. 2 2001. US Patent 6,297,439.

[4] M. Deutsch. How to write with artificial intelligence. https://medium.com/deep-writing/how-to-write-with-artificial-intelligence-45747ed073c. Abgerufen: 19.08.2016.

[5] M. Deutsch. Silicon valley: A new episode written by ai. https://medium.com/deep-writing/silicon-valley-a-new-episode-written-by-ai-a8f832645bc2. Abgerufen: 19.08.2016.

[6] D. Eck and J. Lapalme. Learning musical structure directly from sequences of music. (1300), 2008.

[7] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[9] S. Haykin. *Neural Networks A Comprehensive Foundation*. Macmillan College Publishing Company, 1994.

[10] T. Isokawa, H. Nishimura, and N. Matsui. Quaternionic multilayer perceptron with local analyticity. *Information*, 3(4):756, 2012.

[11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22Nd ACM International Conference on Multimedia*, MM '14, pages 675–678, New York, NY, USA, 2014. ACM.

[12] D. Kang, J. Emmons, F. Abuzaid, P. Bailis, and M. Zaharia. Optimizing deep cnn-based queries over video streams at scale. *CoRR*, abs/1703.02529, 2017.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[14] Y. LeCun and Y. Bengio. The handbook of brain theory and neural networks. chapter Convolutional Networks for Images, Speech, and Time Series, pages 255–258. MIT Press, Cambridge, MA, USA, 1998.

[15] Y. LeCun, K. Kavukcuoglu, and C. Farabet. Convolutional networks and applications in vision. In *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*, pages 253–256, May 2010.

[16] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. *CoRR*, abs/1412.0035, 2014.

[17] G. McCaig, S. DiPaola, and L. Gabora. Deep convolutional networks as models of generalization and blending within visual creativity. *CoRR*, abs/1610.02478, 2016.

[18] A. Mordvintsev. Inceptionism: Going deeper into neural networks. https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html. Abgerufen: 16.08.2016.

[19] D. Nauck, F. Klawonn, and R. Kruse. *Neuronale Netze und Fuzzy Systeme*. Vieweg, 1994.

[20] A. Newitz. Movie written by algorithm turns out to be hilarious and intense. https://arstechnica.com/gaming/2016/06/an-ai-wrote-this-movie-and-its-strangely-moving/. Abgerufen: 19.08.2016.

[21] M. Osadchy, Y. LeCun, and M. Miller. Synergistic face detection and pose estimation with energy-based models. *Journal of Machine Learning Research*, 8:1197–1215, 2007.

[22] T. Simonite. A startup's neural network can understand video. https://www.technologyreview.com/s/534631/a-startups-neural-network-can-understand-video/. Abgerufen: 16.08.2016.

[23] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[24] J. Stanley and E. Bate. *Neuronale Netze Computersimulation biologischer Intelligenz*. Systhema Verlag GmbH, 1991.

[25] statista. Anzahl der aktiven film- und fernsehproduktionsfirmen in deutschland in den jahren 1998 bis 2014. https://de.statista.com/statistik/daten/studie/243238/umfrage/anzahl-der-film-und-fernsehproduktionsfirmen-in-deutschland/. Abgerufen: 10.08.2016.

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[27] M. Thoma. Creativity in machine learning. *CoRR*, abs/1601.03642, 2016.

[28] G. Timmermann. Algo rhythm: Music composition using neural networks. https://medium.com/@granttimmerman/algo-rhythm-music-composition-using-neural-networks-f89897ff2df7. Abgerufen: 18.08.2016.

[29] G. Timmermann. Algorithmic music composition using artificial neural nets. https://github.com/grant/algo-rhythm. Abgerufen: 18.08.2016.

[30] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR*, abs/1609.03499, 2016.

[31] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating Videos with Scene Dynamics. http://carlvondrick.com/tinyvideo/. Abgerufen: 17.08.2016.

[32] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating Videos with Scene Dynamics. *CoRR*, abs/1609.02612, 2016.

[33] H. Ye, Z. Wu, R.-W. Zhao, X. Wang, Y.-G. Jiang, and X. Xue. Evaluating two-stream cnn for video classification. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, ICMR '15, pages 435–442, New York, NY, USA, 2015. ACM.

[34] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.