

Aktuelle Themen: Style Transfer

1. Convolutional Neural Networks:

Der Image Style Transfer, welche von Gatys, Ecker und Bethge 2016 in Tübingen entwickelt wurde, basiert auf einem Convolutional Neural Network welches auch zur Objekterkennung in Bildern benutzt wird. Anschließend wurde ein neuraler Algorithmus eingesetzt welcher das Bild in Inhalt und Style teilen und anschließend wieder zusammensetzen kann. Um den Vorgang dieses Style Transfer grundsätzlich zu verstehen wird daher im folgenden Abschnitt der Aufbau eines Convolutional Neural Networks für die Objekterkennung bei Bildern erklärt.[1]

Convolutional Neural Networks sind ein Teil des Deep Learnings. Unter Deep Learning versteht man bestimmte Netzwerke welche neben dem Input und dem Output Layers noch eine Vielzahl von Layers (Hidden Layers) dazwischen besitzt. Eine Durchbruch von Convolutional Neural Networks zur Objekterkennung gelang LeCun et al. 1998. Hierbei benutzten Sie ein Convolutional Neural Network welchem Sie lernten handgeschriebene Zahlen zu erkennen. Dieser Erfolg kann als grundlegender Meilenstein der Objekterkennung in Bild und Video gesehen werden.[2][3]

Ein Convolutional Neural Network besteht dabei aus zwei Teilen. Dem Feature Extractor, welcher den Convolution layer und den Pooling layer beinhaltet und dem Classifier welcher meist einen fully connected layer besitzt. Der standardmäßige Ablauf eines Convolutional Neural Networks geschieht dabei über 3 verschiedenen Phasen. Auf den Input layer folgt zuerst ein Convolutional layer, anschließend folgt auf diesen ein Pooling bzw. auch Subsampling layer genannt. Diese beiden Layers können dabei beliebig oft hintereinander wiederholt werden, wodurch das Netzwerk mit jedem neuen Convolutional Layer komplexere Strukturen im Bild erkennen kann. Anzumerken ist dabei jedoch dass durch mehrer Layers die Komplexität des Netzwerks zunimmt gleichzeitig aber auch die benötigte Rechenleistung. Am Ende des Networks folgt in der Regel eine fully connection, welches jedes Neuron mit jedem Neuron verbindet und anschließend der Output.[2][6]

1.1 Convolutional layer

Der convolutional layer setzt dabei einen Filter für ein bestimmtes Feature auf einen festgelegten Bereich im Bild.

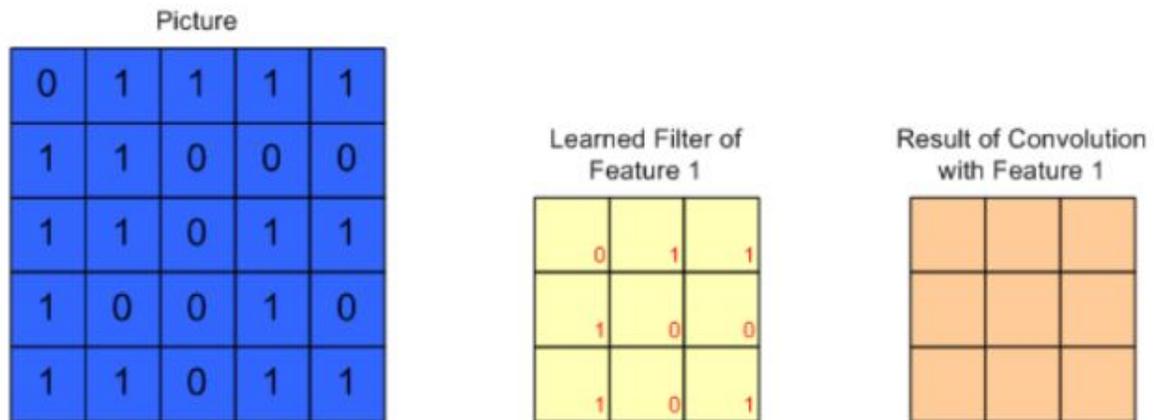


Abbildung 1: *Convolution Layer*

Quelle: Maucher, 2016, S. 18

Im Falle der dargestellten Abbildung handelt es sich um eine 5x5 Matrix welche das eingegebene Bild darstellt und eine 3x3 Matrix welche den Filter darstellt und durch die Zahlenwerte in der Matrix ein bestimmtes Feature angibt, beispielsweise eine Horizontale Linie.

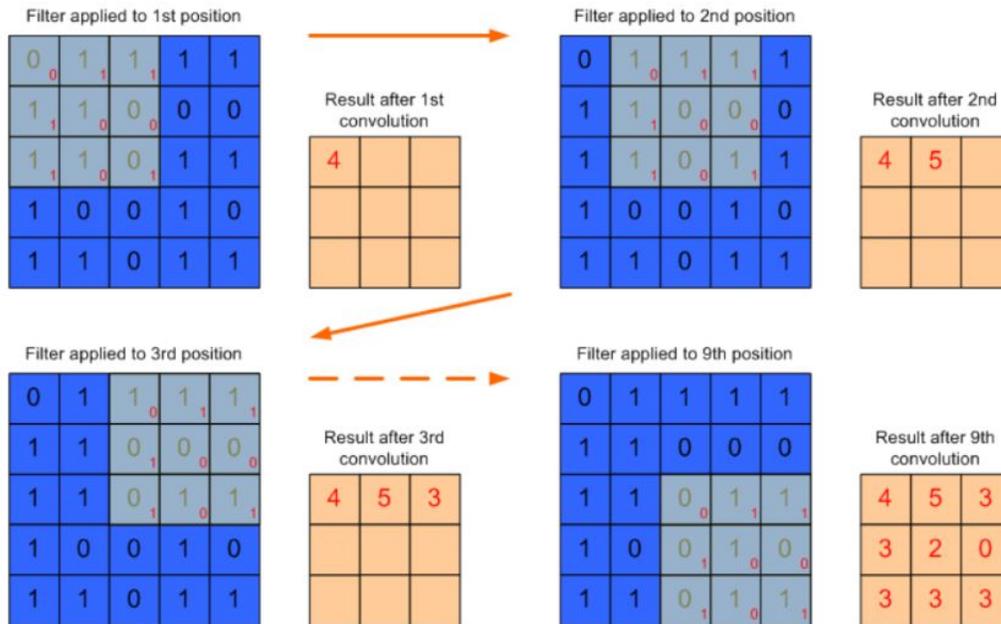


Abbildung 2: *Concept of Convolution*

Quelle: Maucher, 2016, S. 19

Nun wird diese Filter-Matrix zeilenweise auf einen gleich großen Bereich des Bildes gesetzt und miteinander verrechnet (siehe Abbildung). Dieses Ergebnis wird dann in eine neue Matrix geschrieben, welche als Ergebnis des Convolutional layer weitergegeben wird. Die Zahlen in der Ergebnismatrix geben dabei an, wie stark das Feature in dem bestimmten Bereich vorkommt. Eine hohe Zahl weist dabei auf eine starke Übereinstimmung mit dem vorgegebenen Feature hin, während eine niedrige Zahl darauf hinweist, dass das Feature nur sehr schwach oder gar nicht in diesem Bereich vorkommt. Die Ergebnismatrix wird im Allgemeinen als Feature Map bezeichnet. Das Bild wird dabei im ersten Convolutional layer nicht nur auf einen, sondern auf viele verschiedene Filter angewandt, wodurch das Bild auf verschiedene Strukturen untersucht wird, welche mit jedem Convolution layer komplexer werden.[2][4]

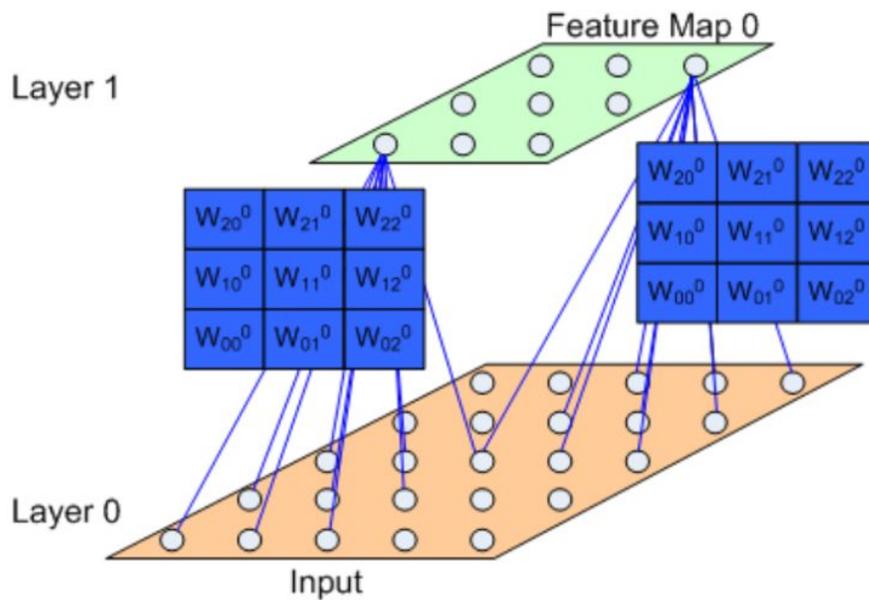


Abbildung 3: *Feature Map*
 Quelle: Maucher, 2016, S. 21

1.2 Sub-sampling / pooling layer

Das Sub-sampling oder pooling dient dazu das nur die relevanten Daten berücksichtigt werden. Dabei wird die eingegebene feature map in festgelegte Anzahl von nicht überschneidenden Unterregionen aufgeteilt. Jede dieser Unterregion enthält dabei die gleiche Anzahl an Neuronen.

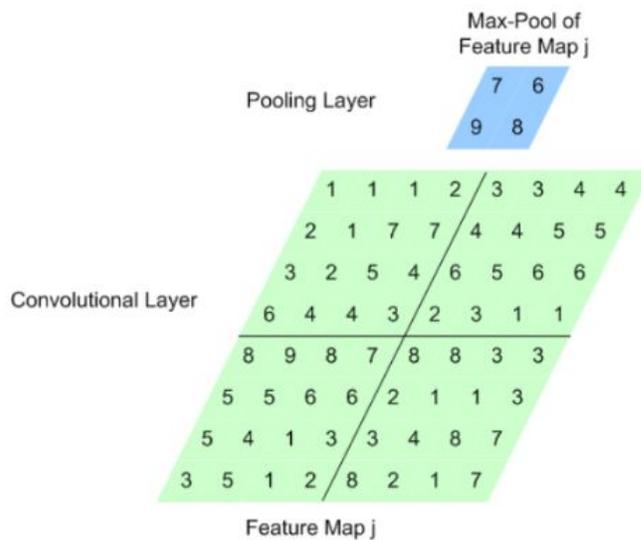


Abbildung 4: *Pooling Layer*
 Quelle: Maucher, 2016, S. 24

Nun gibt es verschiedene pooling Verfahren um die gewünschten Werte zu erhalten. Beim max-pooling wird der höchste Wert jeder Unterregion berechnet und gespeichert, anschließend werden diese in einer neuen feature map gespeichert. Min-Pooling berechnet hierbei nur den minimalen Werte einer jeden Unterregion und speichert diese in einer neuen feature map. Mean/ Average-Pooling berechnet den Durchschnittswert jeder Unterregion und speichert diesen anschließend in einer neuen feature map. Alle anderen Werte, welche nicht in der feature map gespeichert sind werden verworfen.[2]

1.3 Darstellung:

Aus diesen gewonnen Feature Maps setzt sich das Netzwerk neue Filter zusammen. Diese werden dann erneut auf ein convolutional layer angewandt, welche nun komplexere Features besitzen. Daher kann man allgemein sagen, je mehr Convolutional layers das Convolutional Neural Network besitzt umso komplexer Strukturen kann es in einem Bild erkennen.[5]

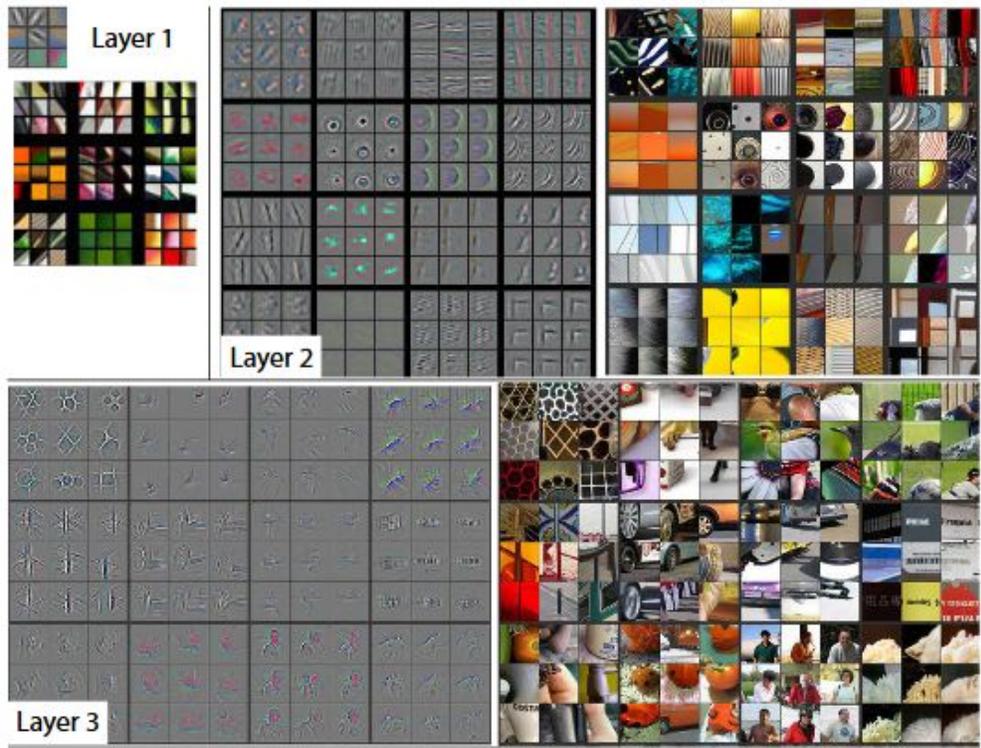


Abbildung 5: Darstellung Convolution layer

Quelle: Zeiler & Fergus, 2014, S. 7

In der dargestellten Abbildung kann man diesen Vorgang nachvollziehen. Während im ersten Layer nur sehr rudimentäre Formen erkannt werden, wie Linien oder Farbübergänge erkennt das Netzwerk bereits im zweiten Layer komplexere Strukturen. So erkennt das Netzwerk hier bereits Kreise oder Winkel. Durch die Abbildung kann man sehr gut nachvollziehen wie stark die Erkennung von komplexen Strukturen durch hinzufügen eines weiteren Convolutional layer steigt. So ist das Netzwerk aus der Abbildung bereits im dritten Layer in der Lage menschliche Gesichter oder Autoreifen zu erkennen.

Diese Abbildung der Darstellung des Netzwerks beruht auf einer Arbeit von Zeiler und Fergus aus dem Jahr 2014, welche es ermöglicht die gute Performance der Convolutional Neural Networks verständlich nachzuvollziehen.[5]

2. Anwendung Style Transfer

Style Transfer oder Neural Style Transfer bedient sich der grundlegenden Techniken eines CNNs zur Objekterkennung, um den Inhalt eines Bildes - üblicherweise einem Foto, dem sogenannten "Content Image" - in einem bestimmten Stil darzustellen. Um diesen Stil zu definieren, wird ein Bild benötigt, welches eine eindeutige stilistische Richtung vorgibt. Üblicherweise werden hierfür künstlerische Bilder aller Art verwendet, die sogenannten "Style Images".

Style Transfer wurde zum ersten Mal im September 2015 von Leon A. Gatys, Alexander S. Ecker und Matthias Bethge von der Universität Tübingen vorgestellt.

2.1 Funktionsweise:

Wie genau der Prozess des Style Transferings abläuft, hängt natürlich von der jeweiligen Implementierung ab. Es gibt viel Spielraum für individuelle gestalterische Anpassungen und verschiedene Implementierungen funktionieren für unterschiedliche Daten natürlich auch nicht immer gleich gut. Grundlegend für die Funktionsweise ist aber das Ziel, ein Bild zu erzeugen, das im Bezug auf den Inhalt nahe am Content Image ist und im Bezug auf den Stil nahe am Style Image.

Bei Style Transfer wird dafür zunächst ein zufällige Rauschen für das resultierende Bild gewählt. Dieses wird anschließend hinsichtlich ihrer Übereinstimmungen zum Content Image und zu Style Image analysiert, mit den Ergebnissen der Analyse verrechnet und das resultierende Bild erneut diesem Prozess unterzogen. Je öfter dieser Prozess wiederholt wird, desto genauer sind in der Regel die Ergebnisse.

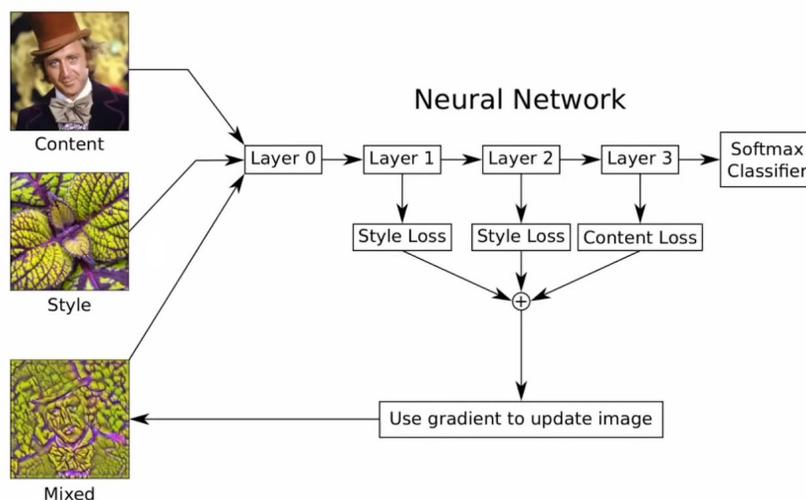


Abbildung 6: Konzeption Style Transfer, Quelle: Pedersen

Was einen Stil ausmacht, kann unterschiedlich definiert werden. Häufig wird aber nach wiederkehrenden Mustern im Style Image gesucht. Das bezieht sich sowohl auf Strukturen und Kanten als auch auf Farben und Texturen. Im Falle der Sternennacht von Vincent van Gogh aus dem Jahre 1889 können das zum Beispiel das Farbspektrum von schwarz über blau zu gelb und weiß sein, die wellenartigen langezogenen Texturen oder die als gelbliche ausgefranste Kugeln dargestellten Sterne.



Abbildung 8: *Sternennacht*

Diese Informationen können anhand von Korrelationen zwischen den Ergebnissen der verschiedenen Feature Maps des CNNs berechnet werden. So wird eine repräsentative Darstellung der Texturen des Style Images erreicht. [11]

3. DeepArt.io

DeepArt.io ist ein Online-Service, der von den Entwicklern der ersten Version des Neural Style Transfers betrieben wird. Die Website ermöglicht das Anwenden von Style Transfer mit beliebigen Bildern und ist eine günstige Gelegenheit, zu überprüfen, welche Eigenschaften Style Image und Content Image aufweisen sollten, um ein gutes Resultat zu erzielen.

Wird als Content Image ein Bild verwendet, in dem kleine detaillierte Strukturen abgebildet sind, muss damit gerechnet werden, dass diese Details beim Style Transfer verloren gehen und schlimmstenfalls zu Artefakten führen. Ähnliches gilt für auch für große strukturlose Flächen im Bild. Ob ein Content Image brauchbar ist, hängt ansonsten vor allem von dem korrespondierenden Style Image ab.



Abbildung 9: *Style Transfer Artefakte*

Die Grünpflanzen sind kaum noch zu erkennen und im Himmel sind deutlich Artefakte zu sehen in Farben, die gar nicht im Style Image enthalten sind.

Style Images sollten vergleichsweise simple und sich wiederholende Muster enthalten. Je weniger Strukturen einzigartig im Style Image sind, desto besser wird das Ergebnis. Daher funktionieren künstlerische Bilder meist besser als Fotos. Zudem ist es von Vorteil, wenn ein Großteil des Farbspektrums in diesen Mustern vertreten ist. So können beispielsweise helle Muster im Style Image auf entsprechend helle Flächen im Content Image projiziert werden.

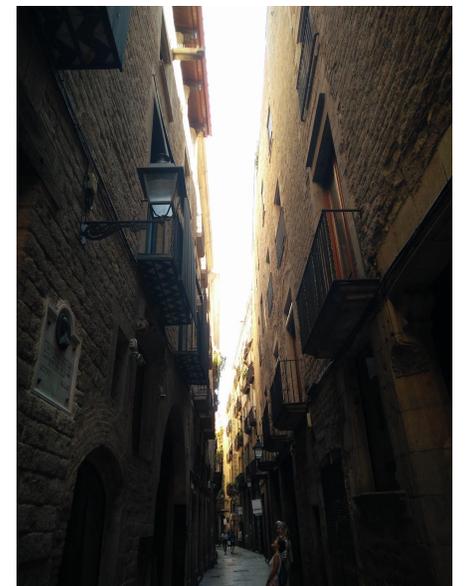


Abbildung 10: *Style Transfer Kontraste*
Kontrastreiches Content Image liefert in Kombination mit einem kontrastreichen Style Image ein gutes Ergebnis.

Bei Style Images gibt DeepArt.io mit einer Sternbewertung Auskunft über die Qualität von Style Images. Diese ist allerdings nur teilweise belastbar. Ein Bild des Farbspektrums ohne jegliche Muster wird beispielsweise mit der höchsten Sternbewertung versehen, da es sehr viele Farben und Helligkeitsstufen enthält, liefert als Style Image aber unbrauchbare Ergebnisse, da keine Muster erkennbar sind.

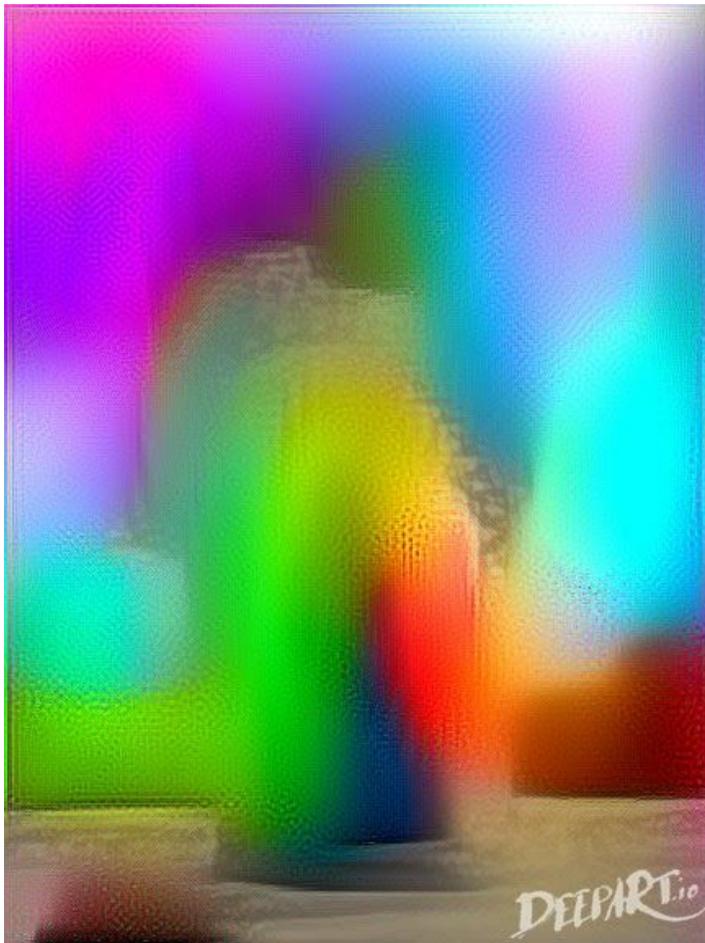


Abbildung 11: *Style Transfer Farbspektrum*

StyleBlit:

Eine sehr beeindruckende Anwendung von Style Transfer ist das Netzwerk StyleBlit welches 2019 von Sýkora et al. entwickelt wurde. StyleBlit wird hauptsächlich im Bereich des 3D Modelling zur texturierung angewendet und ist inspiriert durch Netzwerke wie "StyLit" oder "The Lit Sphere". Jedoch schafft es dieser Algorithmus durch Pixellevel Operationen sehr viel Rechenleistung einzusparen und benötigt zum berechnen des Style Transfers nur noch Millisekunden (siehe Abbildung). [7][8]

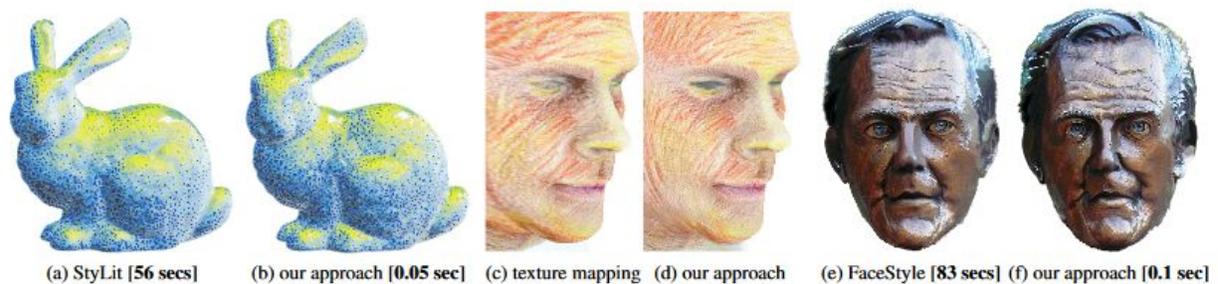


Abbildung 12: *StyleBlit*

Quelle: Sýkora et al., 2019, S. 1

Dadurch ist dieses Netzwerk besonders für Geräte mit wenig Rechenleistung interessant. Beispielsweise für Spiele auf weniger leistungsfähigen Computern oder Handys. Ebenfalls im Film bzw. Kurzfilmbereich könnte dieses Netzwerk interessant sein, da es im Vergleich mit den genannten Netzwerken ähnliche, wenn nicht sogar bessere Qualität liefert. [7][8]

Transports-Based Neural Style Transfer for Smoke:

Eine ebenfalls sehr beeindruckende Arbeit von Style Transfer ist die 2019 erschienene Arbeit über Style Transfer für Rauch. Dieser Algorithmus wurde von Byungsoo et al. in Zürich entwickelt und ermöglicht es den Style von einem eingegebenen Bildern auf Rauch zu übertragen und diesen Style so zu übernehmen. [9]



Abbildung 13: *Style Transfer for Smoke*

Quelle: Byungsoo et al., 2019, S. 2

Hierbei wird ebenfalls durch ein Convolutional Neural Network der Style eines Bild von dessen Inhalt getrennt und auf das 3D Modell des Rauches übertragen. Trotz der Veränderung des Styles behält der Rauch weiterhin seine physikalischen Eigenschaften wodurch teils sehr natürliche und teils sehr unnatürliche aussehende Rauchsimulationen entstehen. Ebenfalls ist es bei diesem Algorithmus möglich bereits transformierten Rauch erneut einen neuen Style zuzuweisen. Hierdurch entsteht wiederum eine noch komplexere Rauchsimulation. Der Algorithmus erlaubt es ebenfalls verschiedene Parameter anzupassen um den Rauch individuelle zu gestalten.[9][10]

Dieser Algorithmus benötigt zwar eine hohen Rechenzeit von ca. 12 Minuten pro Frame dennoch ist er ein wichtiger Vorreiter im Bereich des Style Transfers für flüssige Materialien und es ist spannend zu sehen welche weiteren Arbeiten auf dieser Aufbau werden.[9]

Literaturverzeichnis

[1]

https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Gatys_Image_Style_Transfer_CVPR_2016_paper.pdf

[2] <https://gitlab.mi.hdm-stuttgart.de/maucher/ML/blob/master/Slides/V11DeepLearning.pdf>

[3] <http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf>

[4] https://deeplizard.com/learn/video/YRhxdVk_sls

[5] <https://matthewzeiler.com/mattzeiler/eccv2014.pdf>

[6] https://en.wikipedia.org/wiki/Convolutional_neural_network

[7] https://www.youtube.com/watch?time_continue=173&v=krQrDhestVA&feature=emb_logo

[8] <https://dcgi.fel.cvut.cz/home/sykorad/styleblit.html>

[9] https://www.youtube.com/watch?time_continue=87&v=67qVRhoOQPE&feature=emb_logo

[10] <http://www.byungsoo.me/project/neural-flow-style/paper.pdf>

[11] Adjustable Real-time Style Transfer, Mohammad Babaeizadeh and Golnaz Ghiasi, Nov 2018 <https://arxiv.org/pdf/1811.08560.pdf>

[12] A Neural Algorithm of Artistic Style, Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, Sep 2015 <https://arxiv.org/pdf/1508.06576.pdf>

Bildquellen

Abbildung 1: *Convolution Layer*

Quelle: Maucher, 2016, S. 18,

<https://gitlab.mi.hdm-stuttgart.de/maucher/ML/-/blob/master/Slides/V11DeepLearning.pdf>

Abbildung 2: *Concept of Convolution*

Quelle: Maucher, 2016, S. 19,

<https://gitlab.mi.hdm-stuttgart.de/maucher/ML/-/blob/master/Slides/V11DeepLearning.pdf>

Abbildung 3: *Feature Map*

Quelle: Maucher, 2016, S. 21,

<https://gitlab.mi.hdm-stuttgart.de/maucher/ML/-/blob/master/Slides/V11DeepLearning.pdf>

Abbildung 4: *Polling Layer*

Quelle: Maucher, 2016, S. 24,

<https://gitlab.mi.hdm-stuttgart.de/maucher/ML/-/blob/master/Slides/V11DeepLearning.pdf>

Abbildung 5: *Darstellung Convolution layer*

Quelle: Zeiler & Fergus, 2014, S. 7, <https://matthewzeiler.com/mattzeiler/eccv2014.pdf>

Abbildung 6: *Konzeption Style Transfer*

Quelle: Magnus Erik Hvass Pedersen, 2016,

https://www.zepl.com/viewer/github/Hvass-Labs/TensorFlow-Tutorials/blob/master/15_Style_Transfer.ipynb

Abbildung 7: *Style Transfer Content/Style Gewichtungen*

Quelle: Babaeizadeh & Ghiasi, 2018, S. 3 <https://arxiv.org/pdf/1811.08560.pdf>

Abbildung 8: *Sternennacht*

Quelle: Vincent van Gogh, 1889,

<https://demotix.com/vincent-van-goghs-most-famous-paintings/>

Abbildung 9: *Style Transfer Artefakte*

Quelle: Privatbilder

Abbildung 10: *Style Transfer Kontraste*

Quelle: Privatbilder

Abbildung 11: *Style Transfer Farbspektrum*

Quelle: Privatbilder

Abbildung 12: *StyleBlit*

Quelle: Sýkora et al., 2019, S. 1, <https://dcgi.fel.cvut.cz/home/sykorad/Sykora19-EG.pdf>

Abbildung 13: *Style Transfer for Smoke*

Quelle: Byungsoo et al., 2019, S. 2,

<http://www.byungsoo.me/project/neural-flow-style/paper.pdf>