Ingenieurinformatik B.Sc.

I761 Computergrafik und Bildverarbeitung

Seminararbeit zum Thema:

“Neural Style Transfer”

Lukas Evers
MatrNr: 570976
Abgabedatum: 30.09.2021

Betreuer: Prof. Dr. Rodner

Inhaltsverzeichnis

Abbildungsverzeichnis

Einleitung

Problemstellung und Zielsetzung

Der Transfer des künstlerischen Stils von Bildern und Gemälden auf andere, beliebige Bilder stellt Algorithmen vor hohe Anforderungen hinsichtlich der Erhaltung des ursprünglichen Bildinhaltes bei gleichzeitiger Anpassung an den gewünschten künstlerischen Stil. Methoden ohne den Einsatz von Neuronalen Netzen fokussieren sich auf die künstliche Textursynthese [1]. Diese Methoden finden jedoch kaum noch Einsatz, da neuere Algorithmen, welche neuronale Netze einsetzen bessere Ergebnisse erzielen können. [2]

Ziel dieser Arbeit ist den Stiltransfer mittels Deep Convolutional Neural Networks nach Gaetys et al. 2015(CNN) zu erläutern und die Wirkmechanismen dahinter zu verdeutlichen. Dabei wird insbesondere auf die Methodik der Optimierung der Bildsynthese auf Basis von Gram-Matrizen eingegangen.

Material und Methoden

Convolutional Neural Networks (CNNs)

Bei einem CNN handelt es sich um einen speziellen Typ von neuronalen Netzen, welches in der Bild- und Videoverarbeitung zum Einsatz kommt. Ein CNN besteht aus einer Input-Schicht (Input Layer), mehreren versteckten Schichten (Hidden Layers) und einer Output Schicht (Output Layer). In den Hidden Layers befinden sich Layer, welche Faltungen (englisch: Convolutions) mithilfe eines Filterkernels berechnen. Mithilfe des Filterkernels wird der Input in eine Feature Map abstrahiert. Die Feature Map, auch Activation Map, ist eine 2D Matrix aus einzelnen Neuronen, welche die Aktivierung für den Filter an jeder Stelle des Inputs darstellt. Beispiele für solche Filterkernel sind unter anderem Kantenerkennung.
Neben den Convolutional Layern kann ein CNN auch sogenannte Pooling Layer beinhalten. Diese Layer abstrahieren die Outputs der Convolutional Layer weiter und verringern die Informationen auf das Wesentliche. Oft folgen auf mehrere Convolutional Layer ein Pooling Layer. Das führt zu schnelleren Berechnungsgeschwindigkeiten und erlaubt die Erzeugung von tieferen Netzwerken. [3]

Das für den Stiltransfer verwendete CNN (VGG-19) besteht aus 5 Blöcken mit unterschiedlich vielen Convolutional Layern und je einem Pooling Layer (vgl. Abb. 1). Das VGG-19 ist ein CNN zur Objekterkennung. [4] Für den Stiltransfer wird ein vortrainiertes Model verwendet, welches mit dem Imagenet-Datensatz trainiert wurde.

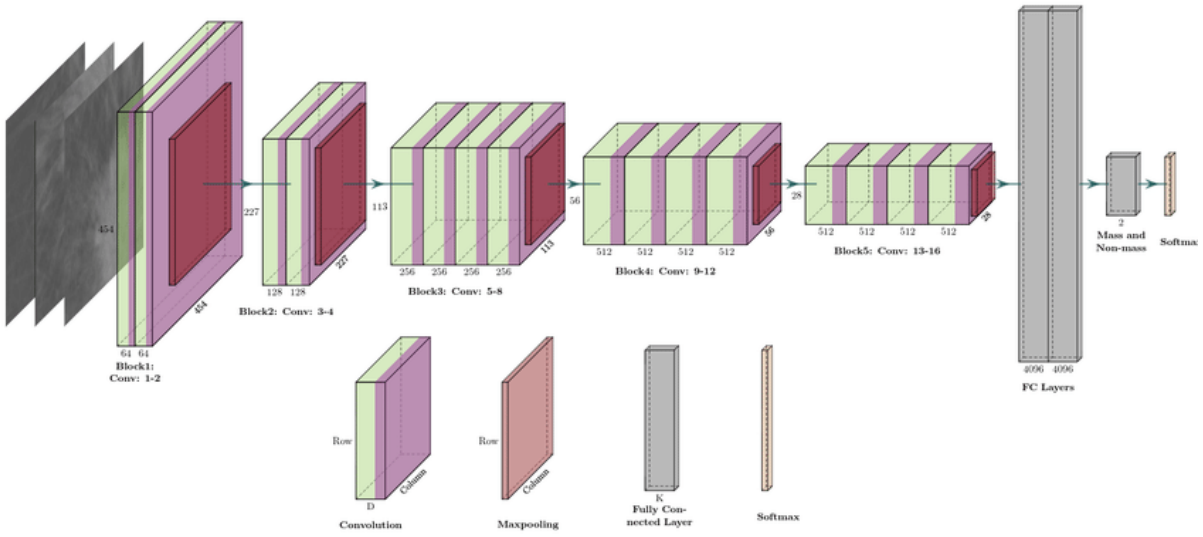

Abbildung 1: Struktur VGG-19 [5]

Stil und Inhalt im Convolutional Neural Network

In der Verarbeitungskette entlang eines CNN, das auf Objekterkennung trainiert wurde, wird das Eingabebild in Repräsentationen transformiert, welche zunehmend den tatsächlichen Inhalt des Bildes über den einzelnen Pixelwerten priorisieren. Höhere Layer im CNN

arbeiten also mit Feature Maps, welche den konkreten Inhalt im Bild, wie zum Beispiel Objekte und Szenen, repräsentieren. Im Gegensatz dazu sind die Feature Maps der niedrigeren Layer des CNN eine Repräsentation von tatsächlichen Pixelwerten/Kanten. Zur Repräsentation des Inhalt wird der zweite Convolutional Layer des fünften Blocks des VGG-19 Modells verwendet (Block5_Conv_2). [6]

Der Stil eines Bildes wie z.B. die Verwendung von bestimmten Farben, die Verwendung von klaren Kanten, Pinselstrichen wird im CNN jedoch anders als der Inhalt repräsentiert. Der Stil eines Bildes kann durch die Korrelation der Aktivierung von verschiedenen Filtern über alle Feature Maps eines Layers ermittelt werden. Auf diese Weise wird der Stil des Bildes holistisch dargestellt werden ohne konkret an die Orientierung oder das Auftreten des eigentlichen Inhalts gebunden zu sein. Zur Repräsentation des Stil werden von allem fünf Blocks der VGG19-Models der erste Convolutional Layer verwendet. [7]

Diese Trennung der Repräsentationen von Stil und Inhalt in einem CNN ermöglicht erst die Synthese neuer Bilder welcher den Inhalt von einem Bild mit der Stilrepräsentation eines anderen Bildes vermischt. [2]

Loss-Funktion

Um ein möglichst gutes stilisiertes Bild x^*aus einem Inhaltsbild x_c und einem Stilbild x_s zu erzeugen müssen Inhalts- und Stilrepräsentation von x^* so wenig wie möglich von x_c bzw. x_s abweichen. Hierfür wird eine Loss-Funktion für Inhalt und Stil verwendet, die es iterativ mit einem Gradient Descent Verfahren zu minimieren gilt.
Die Loss-Funktion ist wie folgt definiert:

$$L = \alpha L_{Inhalt} + \beta L_{Stil}$$

α und β sind die Gewichte für die einzelnen Loss-Funktionen.

Die Loss-Funktion des Inhalt ist durch die Summe der quadrierten Fehler zwischen den Feature Maps des Layers l (Block5_Conv2) von x^* und x_c:

$$L_{Inhalt} = \frac{1}{2} \sum_{i=1}^{N_l} \sum_{j=1}^{M_l} \left(F_{ij}^l - P_{ij}^l\right)^2$$

wobei F^l als die Feature Maps von x^* und P^l als die Feature Maps von definiert sind. N^l entspricht der Anzahl der Feature Maps in Layer l und M^l ist definiert als die Höhe mal die Breite der einzelnen Feature Maps.

Die Loss-Funktion des Bildstils wird berechnet aus der Summe der quadrierten Fehler der Feature Korrelationen von mehreren Layern von x^* und x_s. Die Feature Korrelation kann durch eine Gram Matrix ausgedrückt werden. Die Gram Matrix entspricht dem inneren Produkt der vektorisierten Feature Maps.

$$L_{Stil} = \sum_l w_l \, L_{Stil}^l$$

wobei w_l die Gewichtung des Losses für den Layer l ist.

Die Loss-Funktion für einen einzelnen Layer ist durch die quadratische Abweichung der Gram-Matrizen von x^* und x_s definiert:

$$L_{Stil}^l = \frac{1}{4N_l^2 M_l^2} \sum_{i=1}^{N_l} \sum_{j=1}^{M_l} \left(G_{ij}^l - A_{ij}^l\right)^2$$

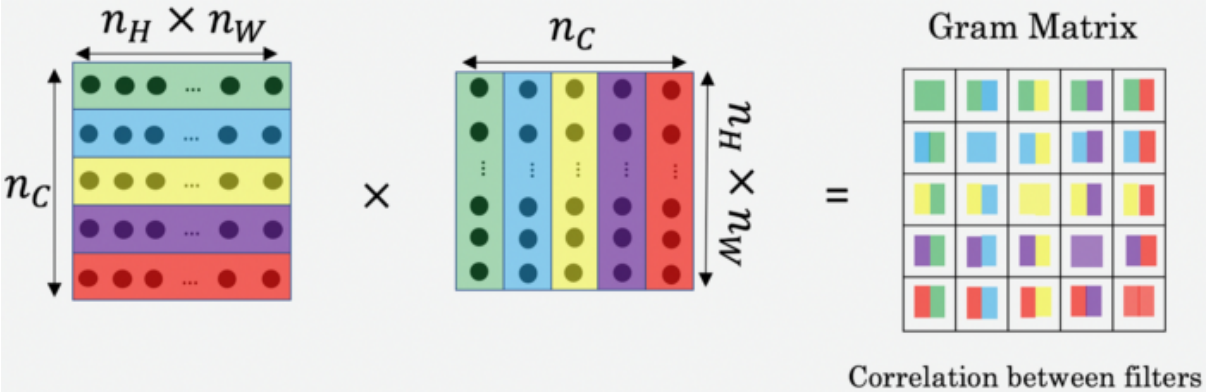

Abb. 2: Berechnung Gram-Matrix. Vektorisierte Feature Map wird transponiert und mit sich selbst multipliziert. Anhand der resultierenden Aktivierungen in der Gram Matrix werden korrelierende Filter identifiziert.

Die Gram - Matrix für eine einzelne Feature Map ist definiert als:

$$G_{ij}^l = \sum_{k=1}^{M^l} F_{ik}^l F_{jk}^l$$

wobei G^l die Gram Matrix für den Layer l im Bild x^*, respektive A^l für den Layer l im Bild x_s ist. [8], [9]

Total Variation Loss und Hochpassfilter

Bei dieser Art der Implementierung und einer hohen Gewichtung für den Stil des Bildes treten störende Artefakte auf, welche die Qualität des synthetisierten Bildes mindern (vgl. Abb.)Um diese Artefakte zu reduzieren wird ein weiterer Kostenterm hinzugefügt. Mittels dieses Kostenfunktionals auf Basis eines Hochpassfilters (Sobel-Kantendetektion) wird die Glättung und der räumliche Zusammenhang des neuen Bildes priorisiert. [10]

$$L_{Total\,Variation} = \sum_{i,j} \sqrt{\left|x_{i+1,\,j}^* - x_{i,j}^*\right|^2 + \left|x_{i,\,j+1}^* - x_{i,j}^*\right|^2}$$

Die gesamte Loss-Funktion setzt sich dann folgendermaßen zusammen:

$$L = \alpha L_{Inhalt} + \beta L_{Stil} + \gamma L_{TotalVariation}$$

mit γ als Gewichtung für den Einfluss der Variation auf die Kosten.

Gradient Descent

Um ein bestmögliches Bild zu synthetisieren wird iterativ ein lokales Minimum für die Loss-Funktion mittels einem Gradientenverfahren (Gradient Descent) gesucht. Hierfür wird der Adam - Algorithmus verwendet, welcher sich vor allem durch eine adaptive Lernrate von einem normalen Gradientenverfahren unterscheidet. [11]
Es wurde eine Grenze nach 1000 Iterationen gesetzt, da die Erfahrung zeigte, dass meist keine oder nur noch geringfügige Änderungen von Iteration zu Iteration zu beobachten waren.

Ergebnisse und Ausblick

Bei der Anwendung des CNN und der Anpassung der Parameter der Loss-Funktion haben sich interessante Ergebnisse erzielen lassen.

Dauer des Trainings

Auf einer Nvidia GTX 1050Ti dauert eine Optimierung und damit die Synthese eines stilisierten Bildes in etwa 20 Minuten.
Die damit verbundene Wartezeit und Auslastung der Grafikkarte sind nicht mehr zeitgemäß. In den letzten Jahren seit Veröffentlichung der Methode von Gaetys 2015 wurden viele verschiedene Methoden gefunden um die benötigte Zeit für einen Stiltransfer auf wenige Sekunden und auch darunter zu senken. Diese Methoden nutzen unter anderem Feed-Forward-Encoder und Decoder Netzwerke auf Basis von stark abstrahierenden Pooling Layern. [12]

Parametereffekte

Den größten Einfluss auf die Qualität des Bildes haben die beiden Gewichtungen für den Content und Style Loss (vgl. Abb. 3).
Der Einfluss des Total Variation Loss ist erst in den letzten Iterationen deutlich präsent und vermindert das Rauschen und andere Artfakte (vgl. Abb.4)

Abbildung 3: α = 1, β = 10000

Rechts: α= 10000, β= 1

Abbildung 4: α = 1, β= 10000, γ=100

Literaturverzeichnis

Bibliography

[1] A. A. Efros and W. T. Freeman, “Image Quilting for Texture Synthesis and Transfer,” *University of California, Berkeley*, Accessed: Sep. 21, 2021. [Online]. Available: http://graphics.cs.cmu.edu/people/efros/research/quilting/quilting.pdf.

[2] L. A. Gatys, A. S. Ecker, and M. Bethge, “A Neural Algorithm of Artistic Style,” Sep. 2015, Accessed: Sep. 30, 2021. [Online]. Available: https://arxiv.org/pdf/1508.06576.pdf.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolu-tional neural networks,” *Advances in neural information processing systems*, Apr. 2012, Accessed: Sep. 30, 2021. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet.

[4] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” Sep. 2014, Accessed: Sep. 30, 2021. [Online]. Available: https://arxiv.org/abs/1409.1556.

[5] K. Hasan and T. A. Aleef, “Automatic Mass Detection in Breast Using Deep Convolutional Neural Network and SVM Classifier,” Jul. 2019, Accessed: Sep. 30, 2021. [Online]. Available: https://www.researchgate.net/publication/334388209_Automatic_Mass_Detection_in_Breast_Using_Deep_Convolutional_Neural_Network_and_SVM_Classifier.

[6] A. Mahendran and A. Vedaldi, “Understanding Deep Image Representations by Inverting Them,” Nov. 2014, Accessed: Sep. 30, 2021. [Online]. Available: http://arxiv.org/abs/1412.0035.

[7] L. A. Gatys, A. S. Ecker, and M. Bethge, “Texture synthesis and the controlled generation ofnatural stimuli using convolutional neural networks,” May 2015, Accessed: Sep. 30, 2021. [Online]. Available: https://arxiv.org/abs/1505.07376.

[8] A. Beutelspacher, *Lineare Algebra*. Springer-verlag, 2003, p. 310.

[9] Y. Li, N. Wang, J. Liu, and X. Hou, “Demystifying Neural Style Transfer,” Jul. 2017, Accessed: Sep. 30, 2021. [Online]. Available: https://arxiv.org/pdf/1701.01036.pdf.

[10] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” *Physica D: Nonlinear Phenomena*, vol. 60, no. 1–4, pp. 259–268, Nov. 1992, doi: 10.1016/0167-2789(92)90242-F.

[11] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” presented at the 3rd International Conference for Learning Representations, Dec. 2014.

[12] Y. Li, “Universal Style Transfer via Feature Transforms,” Nov. 2017, Accessed: Sep. 30, 2021. [Online]. Available: https://arxiv.org/pdf/1705.08086.pdf.