

# **Exposé - Bachelorabschlussarbeit: Untersuchungen von Style Transfer Methoden für die Verwendung mit leistungsarmer Hardware**

Christoph Stach

07.01.2019

## **Abstract**

Die Abschlussarbeit im Bereich der Angewandte Informatik handelt darum verschiedene Style Transfer Methoden zu implementieren, die Konzepte zu verstehen und Sie auf Tauglichkeit für Systeme mit leistungsarmer Hardware zu testen. Style Transfer ist ein Bereich des Deep Learning. Es werden dabei zwei, oder mehrere, willkürliche Bilder zu einem neuen Bild kombiniert. Beim ersten Bild wird der Inhalt aus dem Bild herausgefiltert, z.B. aus einem Foto auf dem ein Vogel zu sehen ist. Aus dem zweiten Bild wird der Stil herausgefiltert, z.B. aus einem abstrakt gezeichneten Gemälde. Das Ergebnis ist die Mischung aus Objekt und Style. Die Ausrichtung, ob das Ergebnis mehr Objekt oder eher Style ist, kann durch Parameter beeinflusst werden. Es sollen unterschiedliche Algorithmen untersucht werden und die Ergebnisse miteinander verglichen werden.

## 1 Motivation

Style Transfer ist ein interessantes Forschungsobjekt und erfordert die intensive Auseinandersetzung mit Convolutional Neural Networks [LeC98] und dem Backpropagation Algorithmus [LeC89]. Außerdem müssen benutzerdefinierte Loss-Functions, die nicht in Problemstellungen der Bildklassifizierung verwendet werden, benutzt werden.

## 2 Zielsetzung

Ziel des Projektes ist ein Softwaresystem zu erstellen das in der Lage ist Style Transfer auf Geräten mit leistungsarmer Hardware zu ermöglichen. Dabei sollen Mechanismen des Style Transfer untersucht werden und sukzessiv verbessert werden.

## 3 Vorgehen

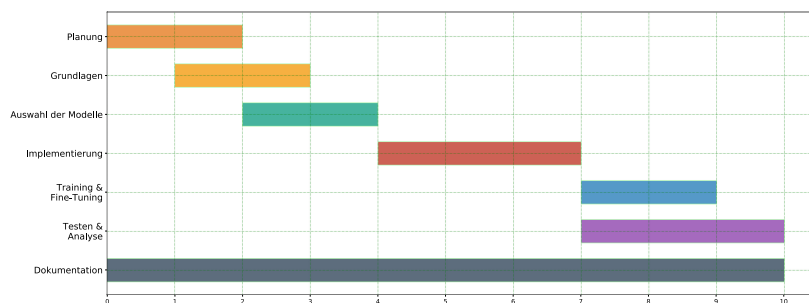
Nach der Einarbeitung in die Grundlagen von Style Transfer [GEB15] [JAL16] Methoden, müssen entsprechende Algorithmen gefunden werden. Da es sich um die Manipulation von Bildern handelt, werden Convolutional Neural Networks [LeC98] zum Einsatz kommen, die sich dazu eignen Features aus Bildern zu extrahieren. Im nächsten Schritt muss ein Datensatz wie Image Net [Den09] benutzt oder in entsprechender Größe generiert werden, mit dem ein solches Netzwerk trainiert werden kann. Auch möglich ist der Einsatz von Transfer Learning mit bereits bestehenden, vortrainierten Netzwerkarchitekturen [SZ14], um die Ergebnisse zu verbessern. Letztendlich muss das Netzwerk auf den verwendeten Datensatz durch Backpropagation [LeC89] optimiert werden.

Die Implementierung des Models in den Prototypen eines Softwaresystems soll der Veranschaulichung der Ergebnisse dienen.

## 4 Erfordernisse und Randbedingungen

Da das Berechnen von Deep Neural Networks und Convolutional Neural Networks sehr rechenintensiv sein kann, werden diese normalerweise auf Serversystemen mit eingebauter Graphic Processing Unit trainiert und ausgeführt. Das soll bei der Evaluation der Modelle berücksichtigt werden, damit Ergebnisse in einem vertretbaren Zeitaufwand auf Systemen ohne GPU zu berechnen sind.

## 5 Zeitplan



Das Gantt-Diagramm stellt den vorläufigen Zeitplan des Projektes da. Dieser kann sich jedoch während des Projektverlaufes noch ändern und wird bei Bedarf angepasst. Besonders zu beachten ist, dass das Training eines Neural Networks in Verbindung mit großen Datenmengen viel Zeit in Anspruch nehmen kann. Deswegen wurden für das Training vorläufig zwei Wochen veranschlagt.

## 6 Erwartete Ergebnisse

Das Ergebnis des Projektes ist der lauffähige Prototyp eines Softwaresystems mit der dazugehörigen Dokumentation und Erklärung aller verwendeten Algorithmen und Modelle. Der Prototyp soll eine Vermischung von einem Style mit einem Bild ermöglichen. Sollte das erfolgreich implementiert werden können, soll er erweitert werden um die Möglichkeit mehrere verschiedene Styles zu wählen. Ziel ist es außerdem den performantesten Ansatz des Algorithmus in den Prototypen zu implementieren.

## 7 Vorläufige Gliederung

1. Zusammenfassung (Abstract)
2. Einleitung
3. Grundlagen
4. Analyse
5. Konzeption
  - 5.1. Suche und Auswahl geeigneter Modelle
  - 5.2. Implementierung einer Basisversion der Modelle
  - 5.3. Verbesserungen und Erweiterungen an Modellen
  - 5.4. Hyperparameter-Tuning und Training
  - 5.5. Implementierung der Modelle in den Prototypen eines Softwaresystems
6. Implementierung
7. Test
8. Evaluation
9. Fazit / Ausblick

## Literaturangaben

- [Den09] J. Deng u. a. „ImageNet: A Large-Scale Hierarchical Image Database“. In: *CVPR09*. 2009.
- [GEB15] Leon A. Gatys, Alexander S. Ecker und Matthias Bethge. „A Neural Algorithm of Artistic Style“. In: *CoRR* abs/1508.06576 (2015). arXiv: 1508.06576. URL: <http://arxiv.org/abs/1508.06576>.
- [JAL16] Justin Johnson, Alexandre Alahi und Fei-Fei Li. „Perceptual Losses for Real-Time Style Transfer and Super-Resolution“. In: *CoRR* abs/1603.08155 (2016). arXiv: 1603.08155. URL: <http://arxiv.org/abs/1603.08155>.
- [LeC89] Y. LeCun u. a. „Backpropagation Applied to Handwritten Zip Code Recognition“. In: *Neural Computation* 1.4 (1989), S. 541–551. DOI: 10.1162/neco.1989.1.4.541. eprint: <https://doi.org/10.1162/neco.1989.1.4.541>. URL: <https://doi.org/10.1162/neco.1989.1.4.541>.
- [LeC98] Yann LeCun u. a. „Gradient-Based Learning Applied to Document Recognition“. In: *Proceedings of the IEEE*. Bd. 86. 11. 1998, S. 2278–2324. URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.42.7665>.
- [SZ14] Karen Simonyan und Andrew Zisserman. „Very Deep Convolutional Networks for Large-Scale Image Recognition“. In: *CoRR* abs/1409.1556 (2014). arXiv: 1409.1556. URL: <http://arxiv.org/abs/1409.1556>.