

# DeepStyle

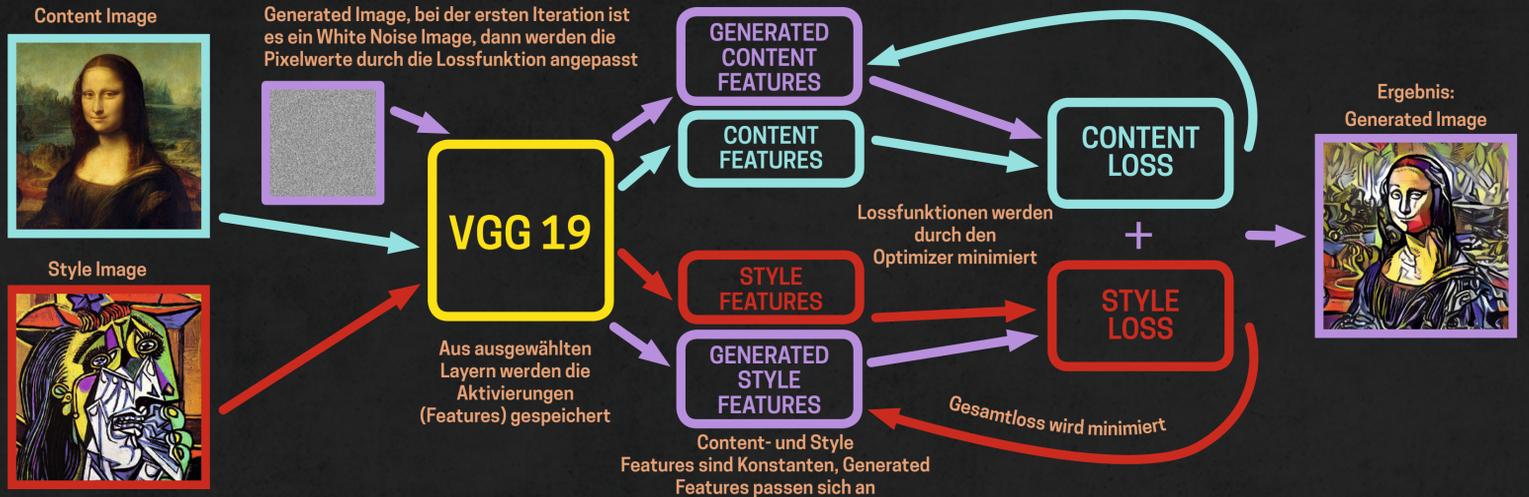
Projekt: Ute Orner-Klaiber  
 Betreuer: Prof. Johannes Maucher,  
 Johannes Theodoridis

„Neural Style Transfer“ beschreibt den Vorgang des Übertragens des Stiles eines Bildes auf den Inhalt eines anderen Bildes mit Hilfe von „Machine Learning“ Algorithmen. In diesem Projekt wurden zwei tiefe neuronale Netzwerke mit Python implementiert, um verschiedene

„Neural Style Transfer“ Verfahren durchzuführen und um diese im Hinblick auf Performance und Qualität zu vergleichen. Der Schwerpunkt lag dabei auf der Performance. Die zwei Algorithmen sind in ausgewählten wissenschaftlichen Publikationen beschrieben.

## Erster Algorithmus: „A Neural Algorithm of Artistic Style“, Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, 2. September 2015

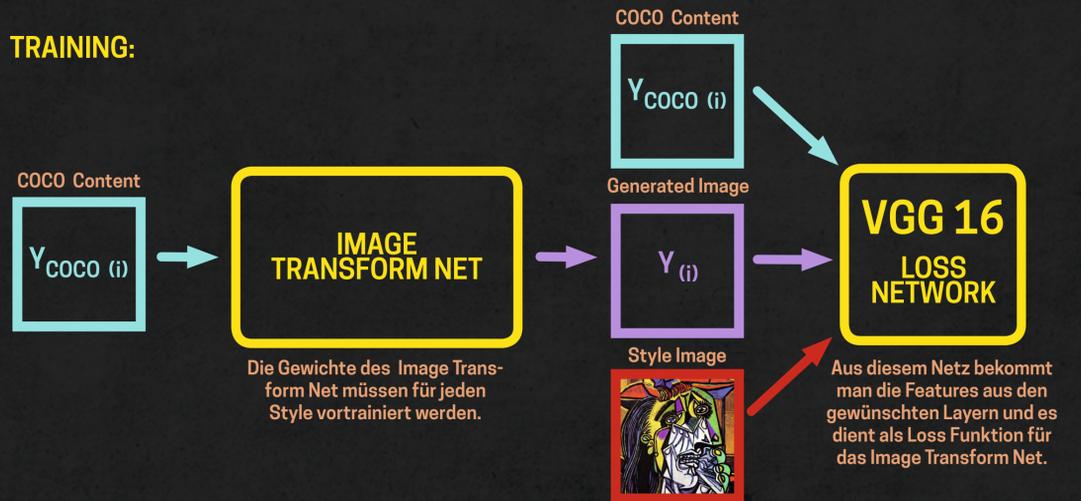
Als Grundlage für den Neural Style Transfer wird ein vortrainiertes Modell benutzt. Dieses Modell ist ein CNN das darauf spezialisiert ist, Objekte zu erkennen (z.B. VGG). In den niedrigen Layern werden Linien und Formen erkannt und in den höheren Layern werden Objekte und komplexere Formen erkannt. Diese Eigenschaft wird für den Neural Style Transfer genutzt. Die entscheidende Idee ist es, zwei Lossfunktionen zu nutzen. Eine repräsentiert den Style (Textur, Farbe vom Style Image) und eine repräsentiert den Inhalt (Struktur und Formen vom Content Image). Da beide Lossfunktionen minimiert werden, kommt ein guter Kompromiss zwischen beiden Bildern heraus.



## Zweiter Algorithmus: „Perceptual Losses for Real-Time Style Transfer and Super-Resolution“, Justin Johnson, Alexandre Alahi, Li Fei-Fei, 27. März 2016

Bei diesem Algorithmus werden zwei trainierte Netze benutzt. Ein „Image Transform Net“ und ein „Loss Network“. Das „Loss Network“ funktioniert wie das Netz vom ersten Algorithmus, nur das VGG16 benutzt wird. Dieses Netz wird als Lossfunktion für das „Image Transform Net“ eingesetzt. Für das Training benötigt man ein Style Image und das COCO-Datenset. Das Set beinhaltet über 40.000 Bilder. Das Style Image wird einmal am VGG16 angelegt um die Style Features zu bekommen und auch jedes COCO-Bild wird angelegt um die Content Features zu bekommen. Jedes COCO-Bild wird auch am „Image Transform Net“ angelegt um das Generated Image zu bekommen. Das Generated Image wird dann am VGG16 Netz ange-

legt, so bekommt man die Generated Style-, und Content Features. Nun kann man den Loss berechnen und durch den Optimizer werden die Gewichte des „Image Transform Net“ trainiert. Es ist speziell auf diesen einen Style trainiert. Wenn man jetzt den Style Transfer mit einem eigenen Content Image durchführt benötigt man nur noch eine Iteration. Dadurch ist dieser Algorithmus viel schneller, wie der erste, allerdings können nicht beliebige Stylebilder eingesetzt werden, sondern nur Stylebilder, die schon mit dem Netz trainiert wurden.



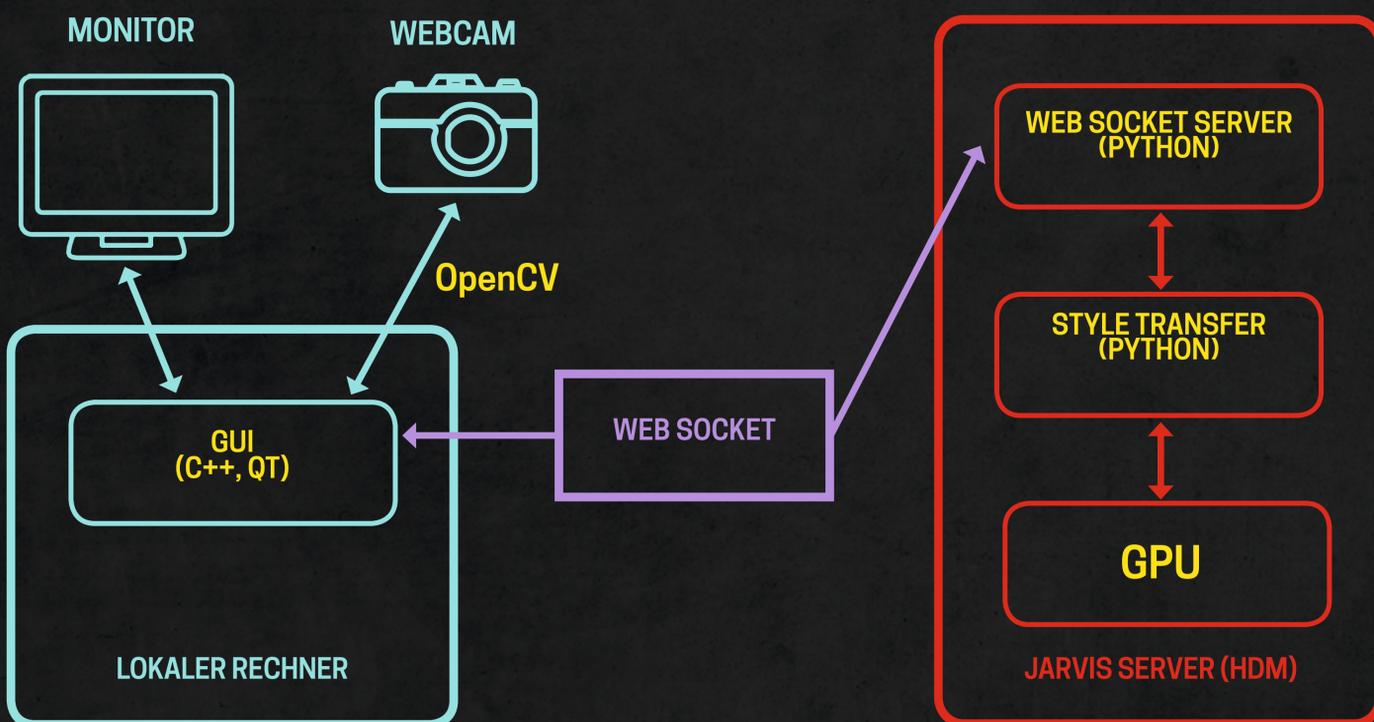
# DeepPortrait

Projekt: Ute Orner-Klaiber  
 Betreuer: Benjamin Binder,  
 Patrick Bader

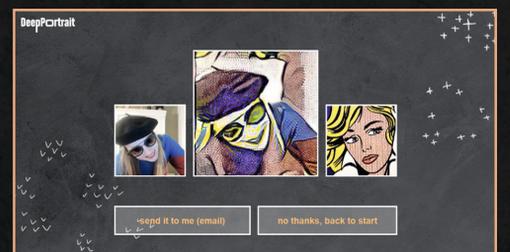
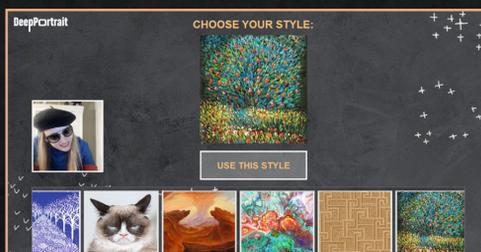
Es wurde mit C++ ein Programm implementiert, mit dem die Nutzer ein Portrait von sich in verschiedenen Stilen erstellen lassen können. Dazu wurde „QT“ genutzt. Über „OpenCV“ wurde eine Webcam angebunden, mit der die Nutzer direkt ein Portrait von sich aufnehmen können. Der Stil wird mit Hilfe von

künstlicher Intelligenz auf das Portrait übertragen. Dieser Algorithmus wurde in Python programmiert und läuft über einen Server auf einer GPU. Sobald das neue Portrait fertig ist, können die Nutzer sich das Portrait per E-Mail zuschicken lassen.

## Kommunikationswege innerhalb des Projekts:



## Screenshots:



# DeepPortrait



*It's no magic, just beautiful mathematics!*

**MAKE YOUR  
OWN PORTRAIT  
WITH  
ARTIFICIAL  
INTELLIGENCE**

