STYLE TRANSFER

Prepared for: MLND, Capstone Project

Prepared by: Himanshu Babal

12 December 2016

Domain Background

Style Transfer ideally refers to synthesising a texture from a source image while constraining the texture synthesis in order to preserve the semantic content of a target image. For texture synthesis, there existed a large range of algorithms that can synthesis photorealistic natural textures. Although these algorithms achieved remarkable results, they all suffered from same limitation, they used only low-level image features for texture transfer.

To separate content from style in natural images is still a very difficult task, however due to advancement in Deep Learning, Deep Convolutional Neural Networks has learned to extract high-level semantic information form image.

Problem Statement

For Style Transfer, we need to extract style features from an image (style image) and extract content features from another image (content image), and then redraw texture using extracted style and content.

The goal is to synthesise a texture form the source image while constraining the texture synthesis in order to preserve the semantic content of the target image.

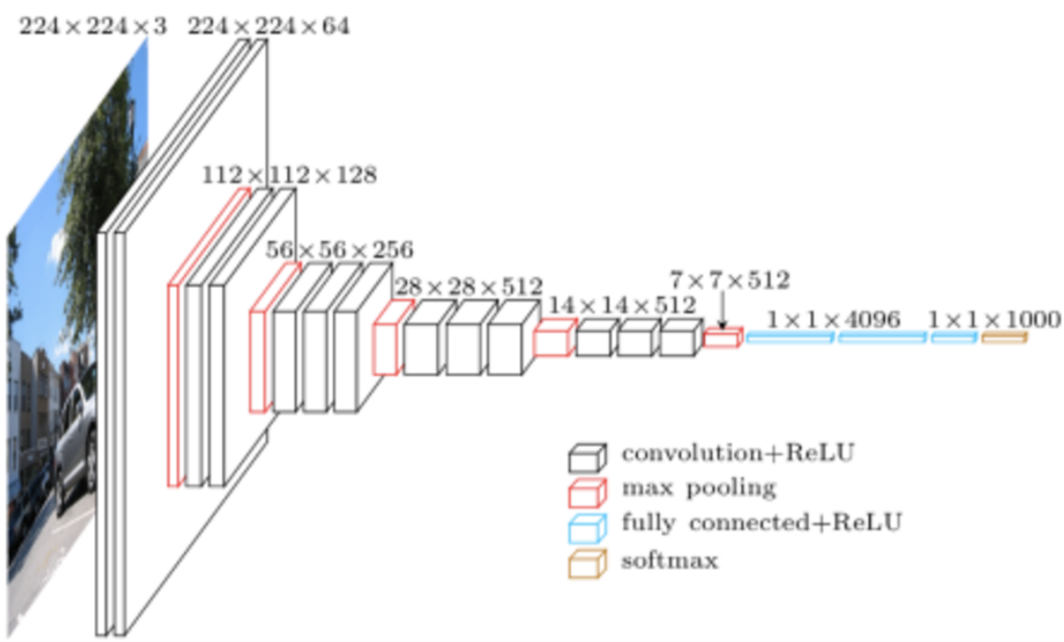
Dataset and Inputs

I will be using pertained VGG-16 Model, the original research paper on Style Transfer used VGG-19 (https://arxiv.org/abs/1508.06576) and Tensorflow to implement it.

VGG-16 Model

VGG is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition" . The model achieves 92.7% top-5 test accuracy in ImageNet , which is a dataset of over 14 million images belonging to 1000 classes.

Architecture of VGG-16 Model is shown below.



Solution Statement

A Neural Algorithm of Artistic Style

Given image is represented as set of filtered images at each processing stage in CNN. In high layers of the network, detailed pixel information in lost, while the high level content of the image is preserved.

On the top of original network, we use a feature space that capture texture information of input image. Style representation computes correlation between different features in different layers of the network.

Genetic features representations are learned by Convolutional Networks and can be used to independently process and manipulate the content and style of a given image.

To transfer the style of an artwork $\vec{\mathbf{a}}$ onto a photograph $\vec{p}$ we synthesise a new image that simultaneously matches the content representation of $\vec{p}$ and the style representation of $\vec{\mathbf{a}}$. Thus we jointly minimise the distance of the feature representations of a white noise image from the content representation of the photograph in one layer and the style representation of the painting defined on a number of layers of the Convolutional Neural Network.

$L_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha L_{content}(\vec{p}, \vec{x}) + \beta L_{style}(\vec{a}, \vec{x})$

Benchmark Model

Since, there can be no quantised evaluation for our work, our goal is that it should look amazing, but still we can compare our results to online sites for Style Transfer, such as

http://prisma-ai.com/

https://deepart.io/

https://www.instapainting.com/ai-painter

Evaluation Metrics

As I have already described above, we don't need any kind of evaluation. We can simply compare our produced image to professional software services given same images.

I will make a separate section to compare images from my Network to the other two described above.

Project Design

To extract image information on comparable scales, we always resized the style image to the same size as the content image before computing its feature representations First content and style features are extracted and stored.

The style image $\vec{\mathbf{a}}$ is passed through the network and its style representation on all layers included are computed and stored. The content image $\vec{p}$ is passed through the network and the content representation in one layer is stored.

We initialise all images with white noise.

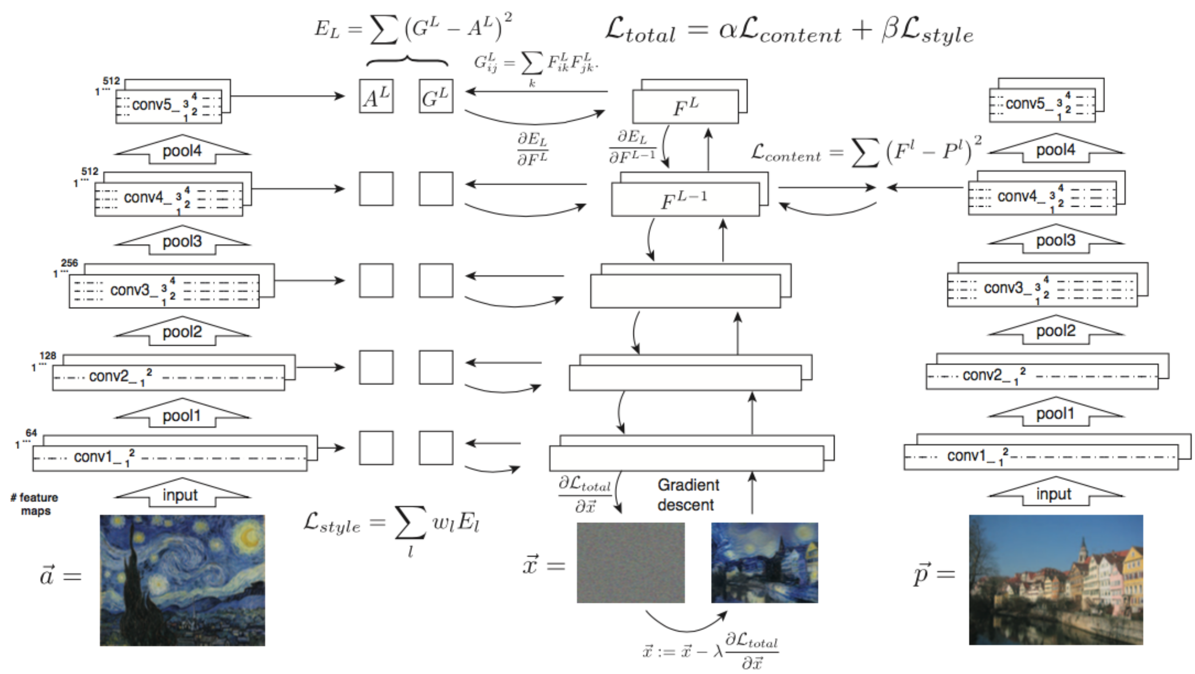
$$\mathcal{L}_{\text{content}}(\vec{p},\vec{x},l) = \frac{1}{2}\sum_{i,j}\left(F^l_{ij} - P^l_{ij}\right)^2 . \qquad \frac{\partial \mathcal{L}_{\text{content}}}{\partial F^l_{ij}} = \begin{cases} \left(F^l - P^l\right)_{ij} & \text{if } F^l_{ij} > 0 \\ 0 & \text{if } F^l_{ij} < 0 , \end{cases}$$

Let $\vec{p}$ and $\vec{\mathbf{x}}$ be the original image and the image that is generated, and P^l and F^l their respective feature representation in layer l. We then define the squared-error loss between the two feature representations by the above Equation (1st one)

The derivative of this loss with respect to the activations in layer l is shown in Equation (2nd one)

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk} .$$

To obtain a representation of the *style* of an input image, we use a feature space designed to capture texture information. These feature correlations are given by the Gram matrix $G^l \in R^{N_l \times N_l}$, where G^l_{ij} is the inner product between the vectorised feature maps i and j in layer l.
Final Structure of the Convolutional Neural Network is given in the following image.



References -

- A Neural Algorithm of Artistic Style - https://arxiv.org/abs/1508.06576
- Image Style Transfer Using Convolutional Neural Networks - http://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Gatys_Image_Style_Transfer_CVPR_2016_paper.pdf
- Blog Post - http://www.anishathalye.com/2015/12/19/an-ai-that-can-mimic-any-artist/