National Chiao Tung University
Spring 2019
Deep Learning
Instructor: Jen-Tsung Chien

Deep Learning Final Project Report

Neural Style Transfer to Make Photo Younger or Older

Group 13 members
Alfons Hwu (Group leader)
0416324
Pin-Jen Hunag
0416033
Chia-Yu Sun
0416045
Min-Xue Yang
0416022

Dept. of Computer Science
Composed with LaTeXon Overleaf

Deep Learning Final Project Report

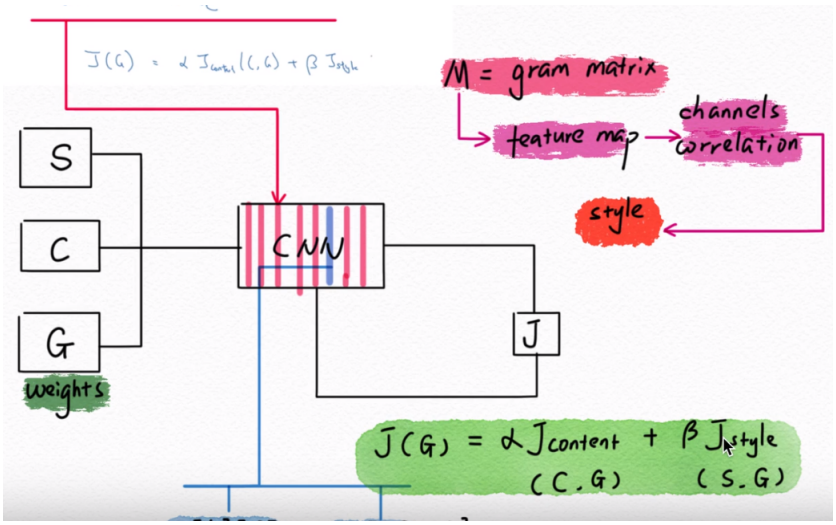
June 20, 2019

1 Introduction

Here we introduce an artificial system based on a Convolutional Neural Network that creates artistic images of high perceptual quality. The system uses neural representations to separate and recombine content and style of arbitrary images, providing a neural algorithm for the creation of artistic images. In this project, we hope to "translate" the image from "young face" to "old face" and vice versa by using the "Neuron Style Transfer" mentioned in `https://arxiv.org/abs/1508.06576` with vgg network and some self-designed model.

2 Model Architecture

The architecture is simple, we have three images, S represents the style image, C represents the content image and G represent the synthesized image to be generated.



(a) Model architecture

Figure 1: Model

The CNN model used is the pretrained VGG-19 network, which is a convolutional neural network that is trained on more than a million images from the ImageNet database. The network is 19 layers deep and trained on millions of images. Because of which it is able to detect high-level features in an image.

During the convolution procedure, features in images can be filtered out and they are called "feature maps". We may calculate the content loss and style loss respectively with p being the content image, a being the art style image, x being the generated, synthesized image.

Hence, overall loss to be

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

3 Understand the math behind the loss

Reference and revised from `https://arxiv.org/abs/1508.06576` page 10 to 12

3.1 The loss of content

A layer with N_l distinct filters has N_l feature maps each of size M_l , where M_l is the height times the width of the feature map (magnitude).

So the responses in a layer l can be stored in a matrix $F_l \in R^{N_l \times M_l}$ where F^l_{ij} is the activation of the i^{th} filter at position j in layer l and the same is true for P^l_{ij}

Let p and x be be the original image and the image that is generated. P^l, F^l being their respective feature representation in layer l after extracted by convolutional filter.

We may thus use the mean square error to compare the original content image and generated image **pixel-wisely.**

$$L_{content}(p, x, l) = \frac{1}{2} \sum_{i,j} (F^l_{ij} - P^l_{ij})^2$$

3.2 The loss of style

On top of the CNN responses in each layer of the network we built a style representation that computes the correlations between the different filter responses, where the expectation is taken over the spatial extend of the input image. These feature correlations are given by the Gram matrix $G_l \in R^{N_l \times N_l}$, where Gl_{ij} is the inner product between the vectorised feature map i and j in layer l:

$$G_{ij}^l = \sum_k (F_{ik}^l - F_{jk}^l)$$

Finally we use gradient descent to minimize the difference between original style image a and generated image x and A^l and G^l being their respective style representations in layer l

Total loss of style will be

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G_{ij}^l - A_{ij}^l)^2$$

Why does it work?, especially in the style loss.

First, try to optimize the content difference is quite in intuitive.

Yet concerning the style loss,in order to capture the style of an image we would calculate how "correlated" these filters are to each other meaning how similar are these feature maps.

Here the **Gram matrix** helps solve this problem, since it is defined as a set of vectors in an inner product space, which is the **Hermitian matrix** of inner products, whose entries are given by $G_{ij} = \langle V_i, V_j \rangle$

If the dot-product across the activation of two filters is large then two channels are said to be correlated and if it is small then the images are un-correlated.

(a) Reply

Figure 2: Zhihu reply for understanding the gram matrix

Based on the reply from `https://www.zhihu.com/question/49805962`

$G^l_{ij} = \sum_k (F^l_{ik} - F^l_{jk})$ will go through each combination of inner-producted feature maps, either in the original image or generated image, consequently all the all vital features of images are represented. Cost function between style and generated image is the **square of difference between the Gram Matrix of the style Image with the Gram Matrix of generated Image.** $E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (G^l_{ij} - A^l_{ij})^2$

Finally we will derive the total loss function $L_{total}(p, a, x) = \alpha L_{content}(p, x) + \beta L_{style}(a, x)$

By adjusting the weight of α and β, we may either emphasize the style or content for the generated image.

4 Dataset Description

We have collected some of the young face as the style image and old face as content image hoping to rejuvenile the old face.

5 Dataset Preprocessing

6 Result Evaluation

7 Group Work Distribution

0416324: NN architecture, testing, shell scripting to automatically organize and train data / IO, writing this LaTeX report.
0416033: Organize data, checking output result and feedback to 0416324 and 0416045 to revise.
0416045: NN architecture, testing and ppt.
0416022: Make demo video, ppt

8 Reference Data

Neuron Style Algorithm: `https://arxiv.org/abs/1508.06576`

Neuron Style Superposition(famous paper by Li Fei-Fei from Stanford University) `https://arxiv.org/abs/1508.06576`

Zhihu reply `https://www.zhihu.com/question/49805962`

Medium article 1 `https://hackernoon.com/how-do-neural-style-transfers-work-7bedaee0559a`

Medium article 2 `https://towardsdatascience.com/style-transfer-styling-images-with-convolutional-`

Bilibili video tutorial `https://www.bilibili.com/video/av22581633/?p=41`

Quora reply `https://www.quora.com/What-is-Gram-matrix-What-is-its-significance-in-linear-algebra`