DEEP NEURAL STYLE TRANSFER

Project Report By
16D100018, 16D070064, 18305R002, 184123011
under supervision of
Prof. Sunita Sarawagi

Report submitted
as part of CS725 course project
Autumn,2019

Contents

1 Abstract

The project aims at implementation of style transfer on still images. Given an image and the style in which it has to be pastiched, the model successfully portrays the content image in the style of reference image. Implemented a deep learning based approach to achieve the objective. The set of algorithms generally used to achieve goal of style transfer is generally categorized under broader category called as Neural Style Transfer. We implemented the model in python using PyTorch. Incorporated style and content loss and appropriate regularizer to minimize the distortions and the transformation more realistic.

2 Introduction

Neural Style Transfer (NST) refers to a class of software algorithms that manipulate digital images, or videos, to adopt the appearance or visual style of another image. NST algorithms are characterized by their use of deep neural networks in order to perform the image transformation. Common uses for NST are the creation of artificial artwork from photographs, for example by transferring the appearance of famous paintings to user supplied photographs. Several notable mobile apps use NST techniques for this purpose, including DeepArt and Prisma.

3 Related Literature

3.1 A Neural Algorithm of Artistic Style[1]

The paper was published in year 2015 by Leon A. Gatys, Alexander S. Ecker and Matthias Bethge. The key finding of this paper is that the representations of content and style in the Convolutional Neural Network are separable. That is, we can manipulate both representations independently to produce new, perceptually meaningful images.To demonstrate this finding, they generated images that mix the content and style representation from two different source images. In particular, matched the content representation of a photograph depicting the "Neckarfront"in Tubingen, Germany and the style representations of several well-known artworks taken from different periods of art (Fig 2).The model details of their work is shown below:

Used the feature space provided by the 16 convolutional and 5 pooling layers of the 19 layer VGGNetwork. Not used any of the fully connected layers.

Found that replacing the max-pooling operation by average pooling improves the gradient flow and one obtains slightly more appealing results and generated images with average pooling.

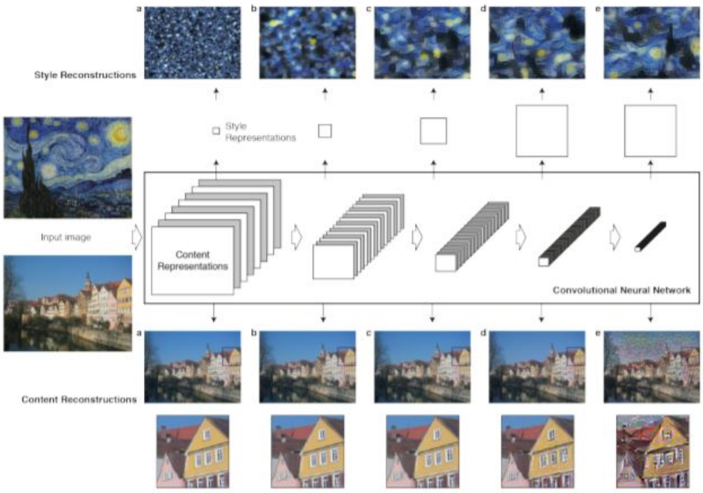


Figure 1: Model Structure

Figure 2: Model output

3.2 Deep Photo Style Transfer[2]

The paper was published in year 2017 by Fujun Luan, Sylvain Paris, Eli Shechtman and Kavita Bala. The paper builds on previous paper[1]. The paper claims to decrease the distortions created during transformation by constraining the transformation from the input to the output to be locally affine in colorspace, and expressing this constraint as a custom fully differentiable energy term. They show that this approach successfully suppresses distortion and yields satisfying photorealistic style transfers in a broad variety of scenarios, including transfer of the time of day, weather, season, and artistic edits. The paper incorporates style loss, content loss and Photorealism regularization term to minize the distortions. The mathematical expression of the regularzation term is mention below in section 4.2.

4 Description of the Set of Approaches

We tried two approaches. First one is same as implemented in the paper "A Neural Algorithm of Artistic Style[1] " where only content loss and style loss are used. In second approach, we are extending first approach by adding regularization term. However the output post incorporating regularization doesn't improved the output significantly, the results are discussed in section 5.3(Experiments section).

4.1 First approach

Terms relevant to first approach:
Pastiche image : The image which are backpropogate into stylized image is called pastiche image. The pastiche image is initialised to be a random noise. Ihis initialised image ,content image and style image are passed through several layers of a network which is pretrained on image classification(Here we are using VGG network for the same.) We use the outputs of the various intermediate layers to compute two losses namely content loss and style loss.
Content loss: During image classification by VGG network , VGG understand the image. We uses these understanding of image to compute the content loss.Equation of the content loss is given by-

$$\mathcal{L}_{content}(\vec{(p)}, \vec{(x)}, l) = \frac{1}{2} \sum_{l} \sum_{i,j} (C_{ij}^{l} - P_{ij}^{l})^2 \tag{1}$$

The derivative of this loss with respect to the activations in layer l equals

$$\frac{\partial \mathcal{L}_{content}}{\partial F_{ij}^{l}} = \begin{cases} (F^{l} - P^{l})_{ij} & if F_{ij}^{l} > 0 \\ 0 & if F_{ij}^{l} > 0 \end{cases}$$

Style loss: Style loss is similar to content loss except here,we compare Gram matrices of the output at various intermediate layers.
Gram matrix: It is defined as multiplication of a matrix by its transpose so

$$G_{ik}^{l} = \sum_{j} F_{ij}^{l} F_{kj}^{l} \tag{2}$$

Gram matrix contains non-localized information,So it have information about style. The Euclidean distance between the Gram matrices of the intermediate representation of pastiche and style image is style loss which is defined as

$$\mathcal{L}_{style}(\vec{x}, \vec{a}) = \sum_{l} w_l E_l = \frac{1}{4} \sum_{l} \frac{w_l}{N_l^2 M_l^2} \sum_{i,j} (G_{ij}^{s,l} - G_{ij}^{p,l})^2 \tag{3}$$

The derivative of E_l with respect to the activations in layer l can be computed analytically:

$$\frac{\partial E_l}{\partial F_{ij}^{l}} = \begin{cases} \frac{1}{N_l^2 M_l^2} ((F^{l})^{T} (G^{sl} - G^{pl}))_{ij} & if F_{ij}^{l} > 0 \\ 0 & if F_{ij}^{l} > 0 \end{cases}$$

Hence total loss:

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{content} + \beta \mathcal{L}_{style} \tag{4}$$

4.2 Second approach:

regularization term is:

$$\mathcal{L}_m = \sum_{c=1}^{3} \vec{x}_c^{T} \mathcal{M}_I \vec{x}_c \tag{5}$$

Derivative of $\mathcal{L}_m$ with respect to output image is-

$$\frac{\partial \mathcal{L}_m}{\partial \vec{x}_c} = 2\mathcal{M}_I \vec{x}_c$$

Hence total loss in this approach is-

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{content} + \beta \mathcal{L}_{style} + \lambda \mathcal{L}_m \tag{6}$$

5 Experiments

5.1 Code Description

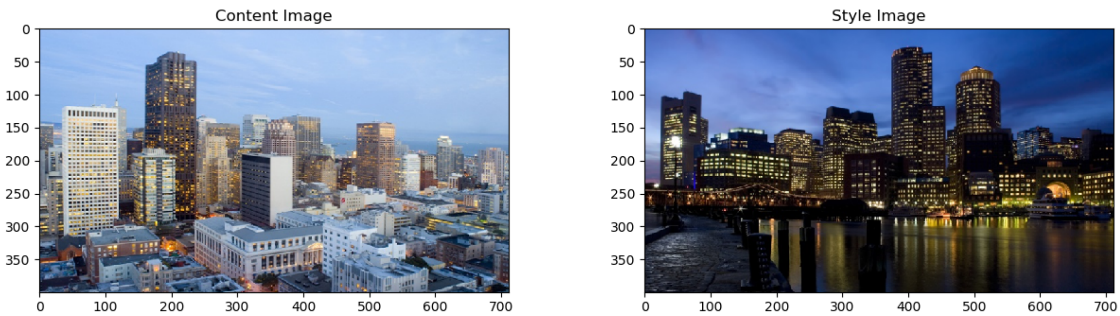
5.2 Experimental Platform

Code written in python. Used Pytorch Library. Run the code locally on NVIDIA gpu using CUDA.

Coding Language	python
DL Library	PyTorch
Machine Used	Local(using GPU)

5.3 Experimental Results

Sample Content image and sample style image are-



(a) content image

(b) style image

Figure 3: Model Input

5.3.1 Approach 1 Results

Loss term including only the style and content loss

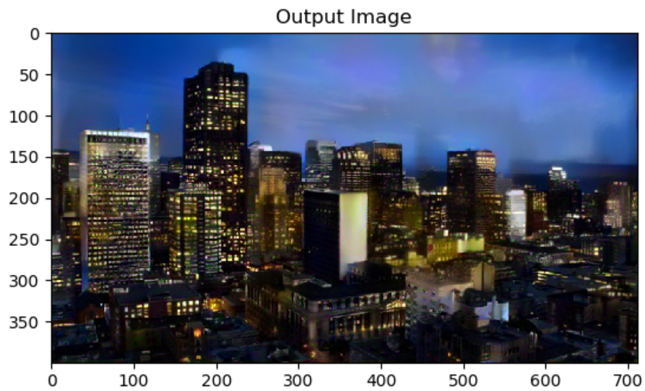
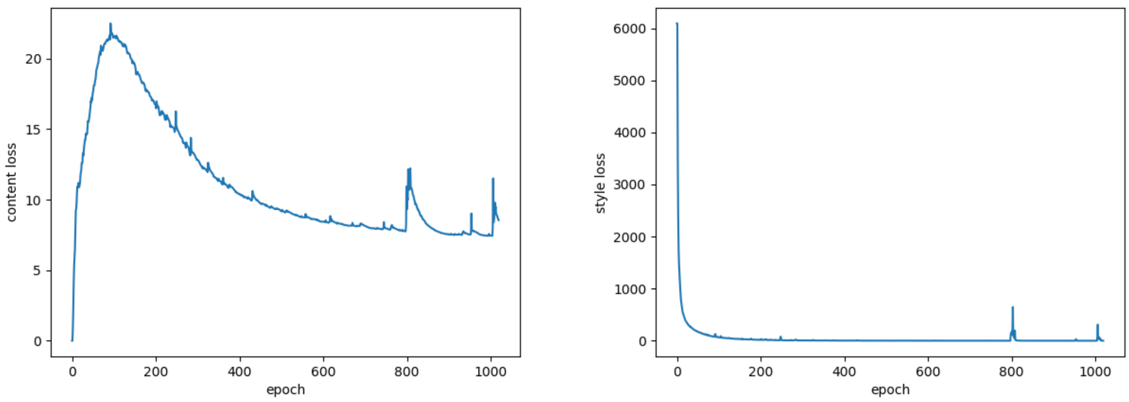


Figure 4: Approach 1 Output

Value of losses with epoch:



(a) Content Loss with epoch (b) Style Loss with epoch

Figure 5: Variation of losses with epoch

5.3.2 Approach 2 Results

Sample output:

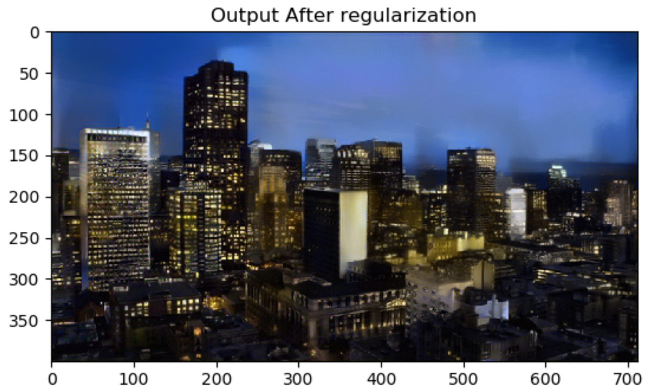


Figure 6: Approach 2 Output

Regularizer loss with epoch(content and style loss will be same as regularization term is added on the image obtained post minimising style and content loss).

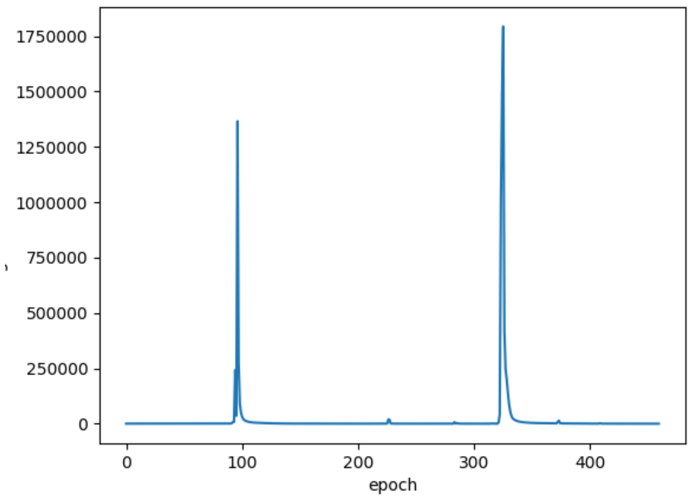


Figure 7: Regularizer loss with epoch

6 Effort Analysis

6.1 Fraction of Time Spent

Major part of our time was spent in writing and debugging basic code for the network. Rough time distribution:

Pytorch learning	2 hrs
Code writing	3 hrs
Debugging	8 hrs
Output Analysis	2 hrs
Report and ppt making	8 hrs

6.2 Most Challenging Part

When we introduced regularization term in approach2, we see that regularization loss was exploding at certain epoch which was not at all expected. The probable reason could be gradient explosion of the regularization term.

6.3 Fraction of Work Done by Individual Team Members

We feel that work was more or less equally shared between our team members, with special credit to Neeraj, contributed significantly in code debugging, making ppt and adding details in the report as well.

References

[1] A Neural Algorithm of Artistic Style Leon A. Gatys, Alexander S. Ecker, Matthias Bethge (https://arxiv.org/pdf/1508.06576.pdf)

[2] Deep Photo Style Transfer Fujun Luan, Sylvain Paris, Eli Shechtman, Kavita Bala (https://arxiv.org/pdf/1703.07511.pdf)