Depth Preserving Neural Style Transfer

Group 25
Aditya Somani - 2020A7PS2049H
Deep Chordia - 2020A7PS2073H
Kavyanjali Agnihotri - 2020A7PS0185H
Parth Tulsyan - 2020A7PS1883H

Abstract

Neural style transfer is an optimization technique which takes in two images—a content image and a style reference image (such as an artwork by a famous painter)—and blend them together so the output image looks like the content image, but "painted" in the style of the style reference image. This is implemented by optimizing the output image to match the content statistics of the content image and the style statistics of the style reference image. These statistics are extracted from the images using a deep residual CNNs.

1 Introduction

The goal of this project is to investigate current state of the art style transfer techniques and suggest improvements. Initially we use Gatys paper implementation which make use of image style loss and content loss to achieve style transfer. Following the baseline experiments we propose per-pixel mean squared depth loss function to improve upon the obtained results.

2 Baseline Experiments

Despite the impressive results some of the other algorithms produce, they all share the same fundamental flaw: they can only transfer textures using the low-level picture attributes of the destination image. However, in a perfect world, a style transfer algorithm could extract the semantic image content from the target image (such as the objects and general scenery). Moreover, the Gatys paper uses this information to guide a texture transfer procedure to render the semantic content of the target image in the source image's style.

The paper proposes a texture transfer algorithm that limits the texture synthesizing technique by feature representations. The style transfer method elegantly simplifies an optimization issue inside a single neural network. New images are created by performing a pre-image search to match feature representations of example photos. Moreover, the paper uses the feature space produced by a normalized version of the 19-layer VGG network. The squared-error loss between these two feature representations is then defined. Higher layers in the network capture high-level information about the objects and their placement in the input image. However, they do not significantly limit the reconstruction's actual pixel values. These layers are the content representation. Using a feature space that can collect texture data from numerous levels, the paper derives a representation of the style of an input image. By creating a matching image, the Gram matrix layers of the network provide these feature correlations. To visualize the data gathered by these style feature spaces, minimized mean-squared distance between the entries of the Gram matrices from the original image and the Gram matrices of the created image are trained using gradient descent from a white noise image.

Finally a new image is created which matches both the style and content representations by jointly minimizing the distance between the feature representations of a white noise image and the style representations of the painting.

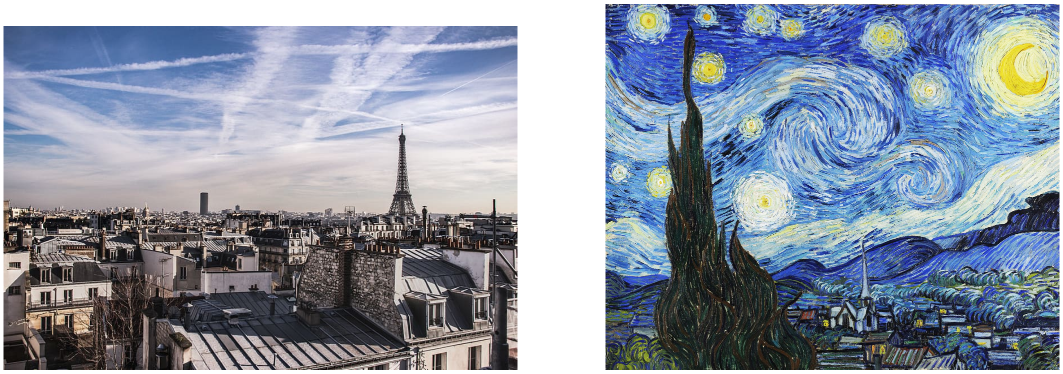

Figure 1: These are the input content(left) and style(right) images passed to the style network

3 Improvement Attempts

Initially we tried to implement the idea by utilizing the state-of-art GANs. We were unsuccessful. It was tough to figure out how GANs work and how to use them adequately for our use case. Further we tried to implement the paper - A Learned Representation for Artistic Style, which trains on multiple style images and applies the style image of our choice to the content image. The attempt turned out to be unsuccessful due the lack of training resources as well as insignificant training results, wherein the image lost a significant portion of the content and exhibited the colouring of respective style images

Figure 2: Output obtained from multi-style transfer network

In VGGNet, each weight is slightly altered using the backpropagation method so as to reduce the model's loss. This value is multiplied by each local gradient as the gradient flows backward to the beginning layers. As a result, the gradient becomes smaller, which causes the updates to the initial layers to be very small and significantly increases training time. We use ResNet in order to solve the vanishing gradient issue. Shortcut connections are used in ResNet architecture to address the issue. These skip connections act as gradient superhighways, allowing the gradient to flow unhindered.

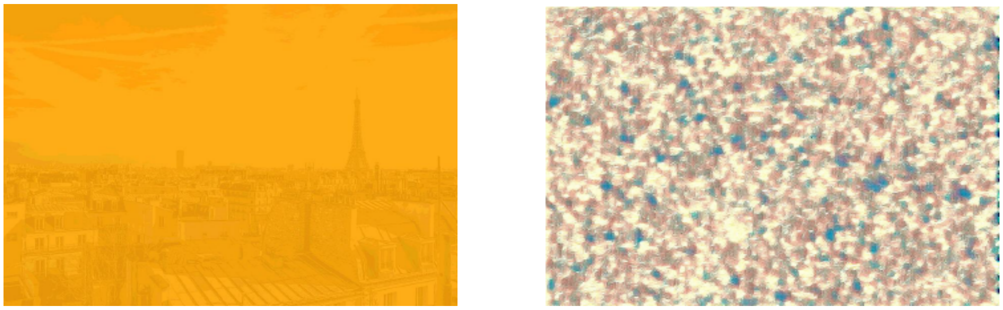

Figure 3: Outputs obtained by using ResNet and Efficient Net in the Gatys Neural Style Implementation

EfficientNet is a convolutional neural network architecture and scaling method that uniformly scales all dimensions of depth/width/resolution using a compound coefficient. The compound scaling method is justified by the intuition that if the input image is bigger, then the network needs more layers to increase the receptive field and more channels to capture more fine-grained patterns on the bigger image.

Inspite of replacing the VGGNet architecture and training the model using trial and error techniques such as increasing the number of training epochs, using different loss and optimization functions, changing the layers used to compute the style loss and content loss, we were unable to obtain any significant output.

4 Depth Preserving Function

The high-level features on pre-trained networks are primarily intended for object recognition; thus, they concentrate on the primary target and ignore extraneous aspects. The distinction between the foreground and background and various objects is lost. The depth map of the semantic content is kept when a picture is styled, effectively representing the spatial distribution in the image. As a result, the semantic content of the image is preserved. While transferring the style, incorporating depth preservation as a second loss preserves the overall image layout. The depth loss of is calculated using a pre-trained depth estimation network. The input context image and the image obtained from the transformation network are passed into the depth network to estimate depth of each image following which per pixel mean squared loss is calculated and added to the style and content loss while training.

$$l^{\phi_1}_{depth}(\hat{y}, x) = \frac{1}{C \times H \times W} \|\phi_1(\hat{y}) - \phi_1(x)\|_2^2$$

Figure 4: The depth loss function estimate the depth difference

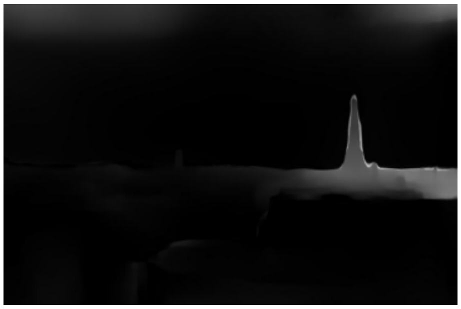

Figure 5: The depth difference map obtained for the content input image and the output image of transformation network

5 Final Goals & Evaluation

We broadly observed that increasing the complexity of the architecture wasn't giving better results. But the Depth Aware loss function was increasing the output considerably. So, In order to preserve the details of the foreground as well as the background, we added a function for depth. We added it to the implementation of the chosen paper using VGGNet and got the following results. Along with this, we were able to reduce the training time considerably. This is indicated by the number of iterations needed by the models. In our implementation, the model provides a much better output at 1500 iterations, whereas the original model requires almost double the time.

Furthermore, the model had a completely apt amount of transfer from the two input images around 800 - 1500 iterations. After 5000 iterations, the model was over-training as we can see from the image below.

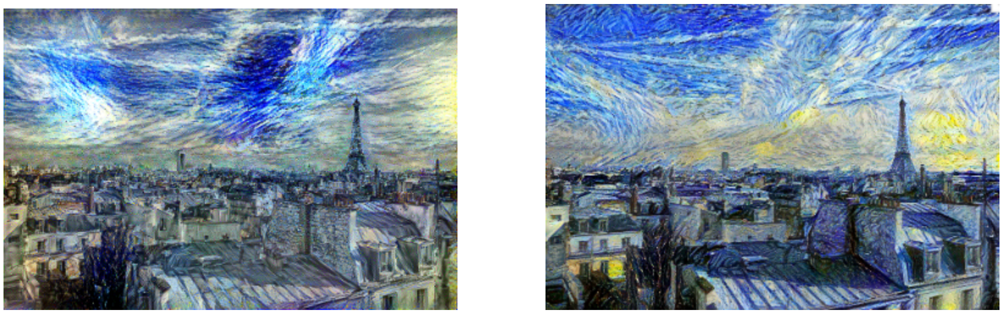

Figure 6: Output obtained Gatys Neural Style Network(left) after 3000 iterations and the one obtained with Depth Preserving Neural Style Network (right) after 1500 iterations

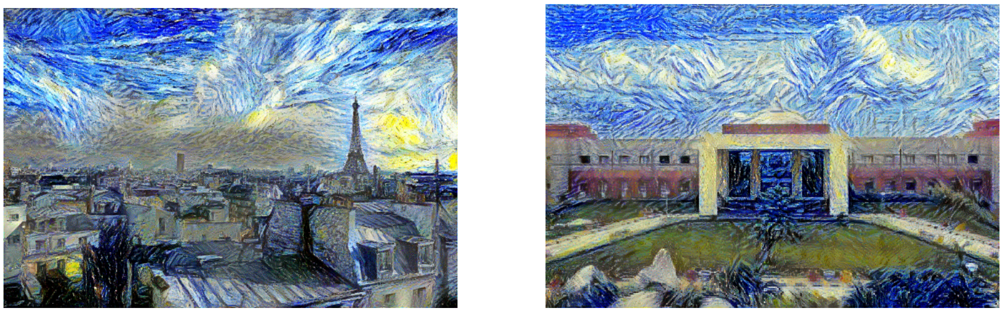

Figure 7: Output obtained from Depth Preserving Neural Style Network(left) after 5000 iterations. It is clearly visible that excess style is being transferred to the content image. The same network applied on the BITS Hyderabad Academic Block building after 500 iterations. The style image considered training all the above images is starry night shown in Figure 1

6 CONTRIBUTIONS

Name	**Contribution**
Aditya Somani	Reading paper, Code for base model, Proposed improvement, Implementing Improvement, Report
Deep Chordia	Reading paper, Proposed Improvement, Implementing Improvement, Report
Kavyanjali Agnihotri	Initial papers selection, Reading paper, Base model selection, Trying Improvement, Report
Parth Tulsyan	Initial papers selection, Reading paper, Initial model selection, Proposed improvement