Neural Artistic Style Transfer

K Vagdevi
IIIT Sricity
Chittoor
vagdevi.k15@iiits.in

M R Abhishek
IIIT Sricity
Chittoor
abhishek.m15@iiits.in

1. Introduction

Neural Artistic Style Transfer finds a wide range of applications to fancily modify images. When a content image and a style image are given as inputs, the output image is expected to contain the content image in the artistic style of the input style image. We implement these visual effects using the convolutional neural network.

2. Related Work

This field has so much influenced the technical world that many apps, such as Prisma, have received great craze amongst the users. In the recent days, a decent work has also been done in this area, which served as a holy grail to our project.

The heart of this capability is the convolutional neural network, often used for modern image processing. The work of Leon A. Gatys, et al on "A Neural Algorithm of Artistic Style"[1], has first come up with such an approach. They have mentioned all about their work, using the powerful VGG-19 newtork. They have used the pretrained weights of the VGG-19 model on the famous Imagenet database. With the millions of data samples available in the database, the model has attained a high power to represent the features of images. In order to generate an image of the stylized content image, the outputs of selective layers of the network have been used. This generated output image is the refined version of the input noise image. Such refinement is a result of the effect of decreasing losses between the two pairs - the content and the generated images, the style and the generated images. These updations aim to produce better versions and thus strive to output a good stylized content image. The method seems to be a bit tricky. The focus our project is to implement this work and give deeper sense of what is happening.

Furthermore, there are also other references we have taken help from, in order to build up our concpetual understanding about the working. [2][3]

Figure 1. Content Image

Figure 2. Style Image

Figure 3. Output Generated Image

3. Technical description

Technically, the work done by convolutional neural network off-screen is really astonishing. Convolutional neural networks outperform any other neural network architecture

for performing object recognition within images. The Imagenet Challenge poses an open challenge where teams work to come up with various methods to classifying objects contained within millions images into one of 1,000 different categories.In 2014, the VGG-19 architecture has stood as the winner with as low as 7.3% error rate. The network that have been trained on such vast datasets essentially learn and extract such important features which are independent and crucial for object classification amongst the various objects available. As mentioned above, the architecture used in the paper, the VGG-19. In their paper, Gatys et. al show that the already trained VGG-19 architecture for object detection will have developed some internal representations of the given input image. Our aim is to ensure that the content from the content image and the style for the style image are preserved. Hence we could use the internal representations related to the content from the content image and those related to the style from the style image to produce the brand new generated stylized image.

Each layer has several filters, which may be considered as the feature detectors of the input image, right from the low level features (in shallow layers) and high level features (in deeper layers) and using non-linear combinations of these feature detections to recognize objects.

The feature maps in the convolution layers of the network can be seen as the networks internal representation of the image content. The shallow layers represent simpler features whereas the deeper layers represent more complex features. Since the semantic orientations of the content image are supposed to be preserved, it is suggested to consider the feature maps (activations) which are of neither too shallow nor of too deep layers of the network. We have used the activations of the conv4_2 layer, as suggested in the paper by Gatys et. al, as the content feature maps.

The style of the style image, on the other hand, is taken in terms of the spacial correlation of the feature maps obtained at various layers. This is so, because, like for the content, the style could not be accounted for, by just considering a feature map. The style could be extracted out from an image by considering the spacial correlation of the values from the feature map, as found out by Gatys et. al. The layers from which the feature maps are to be considered may vary. Those from lower layers represent finer texture whereas those from higher layers represent more higher elements of the style image. The implementation has been found out to be give the best performance when the combination of feature maps from the shallow and deep layers were considered. We have considered the conv1_1, conv2_1, conv3_1, conv4_1, conv5_1, with which we are able to capture finer texture and higher elements as well, as experimented in the paper. The correlation matrix of a feature map F is drawn as follows, and is known to be the Gram

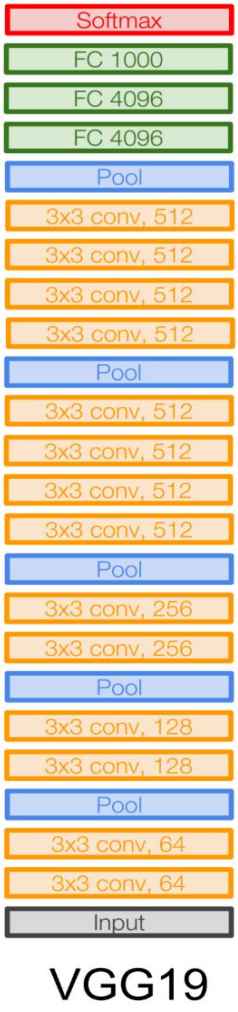

Figure 4. The VGG-19 architecture

Matrix:

$$G_{ij} = \sum_{k} F_{ik} F_{jk}$$

Since we have the content and style representations ready with us, its time for us to discuss the process of generating the stylized output image. The generated image is the resultant of it being refined through many updations. It is important to note is that, unlike any other conventional problems of computer vision where the weights of the network are updated, we update the generated image for achieving better refinement to form the stylized image gradually. Hence, this problem of style transfer could be viewed as an optimization problem. Initially, the generated image is the noise version of the content image.

Initially, the generated image is the noise version of the content image. The generated image is passed to the network as the input. Its feature map F at the conv4_2 are re-

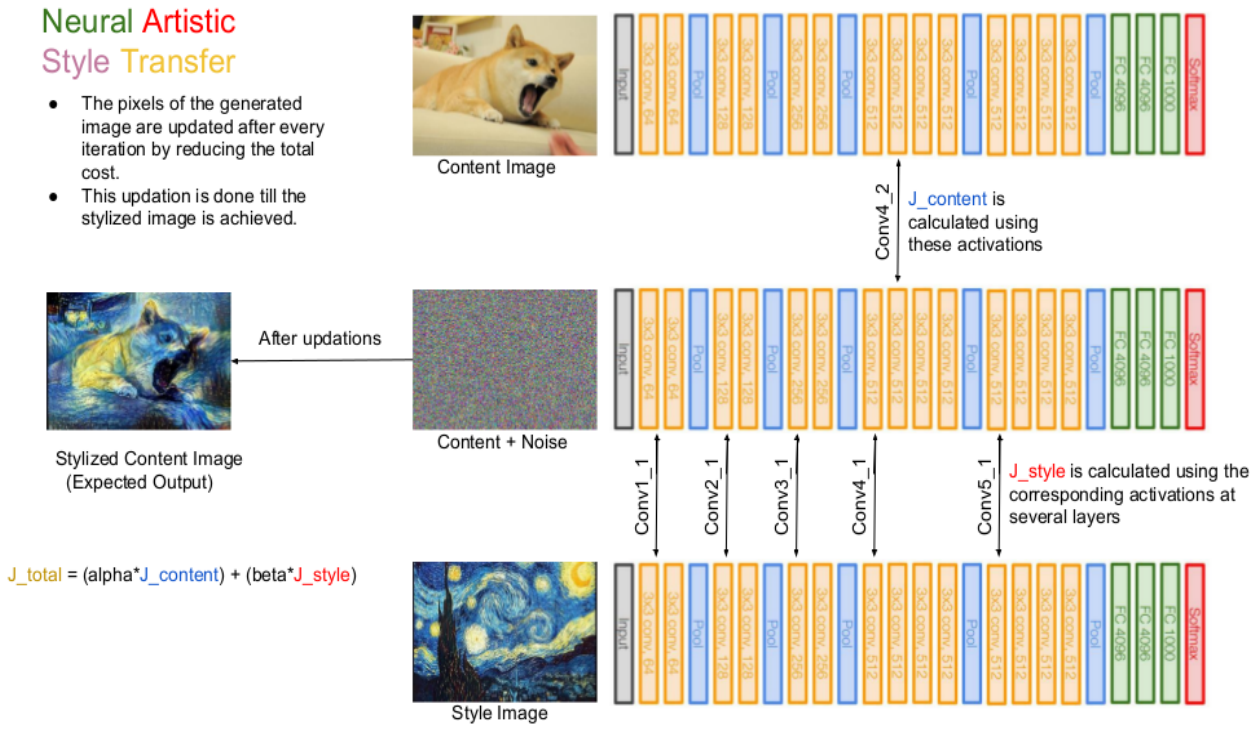

Figure 5. A brief work-flow

trieved. Let the feature map of the content image at conv4_2 be P. The content loss, $J_{content}$, between the feature map from layer l, F^l and that of the generated image from layer l, O^l, could be accounted as follows:

$$J_{content} = \frac{1}{2}\sum_{i,j}(F_{ij}^l - O_{ij}^l)^2$$

Similarly, the style loss, J_{style}, could be calculated using the Gram Matrices at various layers considered for capturing the style of the style image.

$$J_{style} = \frac{1}{2}\sum_{l=0}^{L}(G_{ij}^l - O_{ij}^l)^2$$

We can see that both $J_{content}$ and J_{style} contribute to the total loss. Hence, the joint loss is defined as follows:

$$J_{total} = \alpha J_{content} + \beta J_{style}$$

Therefore, in order to generate the stylized image, the combination of the content and style image, reduce the total loss needs to be minimized. We iteratively improve the generated image using gradient descent via back propagation. After we determine the gradient, the generated image is updated in the negative direction of the gradient so as to decrease the cost. This process is repeated till some number of iterations till we get satisfactory results.

Algorithm: Neural Style Transfer

*generated_image = noise * content_image*

F = conv4_2_feature_map(content_image)

layers = ['conv1_1','conv2_1','conv3_1','conv4_1','conv5_1']

gram_matrices=[]

***for** layer in layers:*

layer_gram_matrix=get_gram_matrix(layer, style_image)

gram_matrices.append(layer_gram_matrix)

end for

***for** iterartion in range(3000):*

J_content = compute_content_cost(F,generated_image)

J_style = compute_style_cost(gram_matrices,generated_image)

Figure 6. iteration 1 (noise + content)

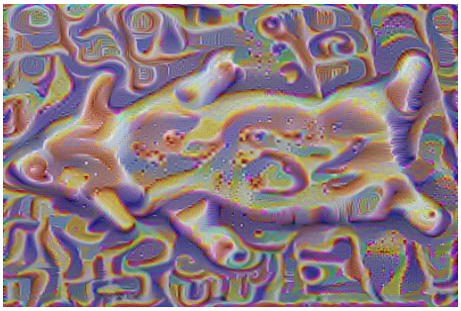

Figure 7. iteration 1065

Figure 8. iteration 2999

J_total = J_content + J_style

gradient = get_gradient(J_total)

update generated_image

end for

output generated_image

4. Experimental Results

The experiment has achieved satisfactory results at last. For a given pair of a content image and a style image , we can now output the stylized image. Although the layers other than the mentioned ones can be taken, the best results are produced with the specified set of activations. If other layers are considered, the depth of the content or style varies correspondingly.

5. Challenges

The main strength of this work is that the usage of a well pre-trained network. The ability and power of such a network has made it possible to extract the most important features.

In the beginning, we faced problems due to slower learning rate. We could observe little change howsoever the number of iterations may be. Then, we have taken care that a good working learning rate has been sorted out after a series of experiments with various learning rates.

6. References

1. https://arxiv.org/abs/1508.06576
2. https://medium.com/data-science-group-iitr/artistic-style-transfer-with-convolutional-neural-network-7ce2476039fd
3. https://medium.com/artists-and-machine-intelligence/neural-artistic-style-transfer-a-comprehensive-look-f54d8649c199