International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

NeuArt- Style Transfer Using VGG19 & RESNET50 and their Comparison

[1]Mamta Gahlan, [2]Tripti Sharma

IT Department, MSIT, New Delhi, India, mamtagahlan@msit.in
IT Department, MSIT, New Delhi, India, tripti_sharma@msit.in

ABSTRACT

Art has been part of humanity since the time of the cave paintings. Human beings have always tried to express themselves using colours. As they say a picture tells a thousand words. But art and science was always considered as two separate fields needing very different skill sets. However with the advancements of machine learning, this notion has been broken. How would the Mona Lisa look like if Vincent Van Gough decided to paint it in the same style as that of the starry night or how would the view outside your window look like in that style. These questions can be answered using the algorithm developed by Leon A. Gatys which was the first algorithm which could to a great extent separate the content and style of two images and merge them to produce a third image. Using this algorithm we can create new artworks using any image and transforming it using the style of well -known paintings. To implement this research paper, we use VGG-19 as that was proposed by the authors. We then tried to implement the same algorithm using Resnet-50 to see if the algorithm generalises well to other architectures.

Keywords: Neural Network, Image Processing, Gradient Descent

1. INTRODUCTION

Neural networks have been found to work very well in computer vision problems because of their ability to recognise simple features, combine them to recognize complex features and then aggregate them to make a prediction. They very efficiently convert pixel patterns into numbers in a matrix (representing features in the form of numbers). For our problem we are concerned with only the matrix values and not the final prediction. These numbers amazingly represent the content(objects) in the image and the style (colour patterns, brush strokes etc)[1]. This means that if we carefully select the matrices we can separate the content and style of an image to a large extent but not totally as our biological eyes can. Since these values can be differentiated with respect to their inputs we can use the gradient descent algorithm to generate a third image[2].

We run a content image, a style image, and a produced image (which was originally simply white noise) through a neural network that has already been trained to recognise numerous objects[3].We then calculate the loss functions that were given in the paper and then use gradient descent to reduce it. The loss functions are weighted so we can adjust them to make the final image look more like the content or the style image. Same weight combination does not give the best result for each combination of the content and style image but through trial and error we have used the values which generalises well.

We use the concept of transfer learning which means that the weight values of the connections are the same from the already trained network and are not updated. To reduce the loss function we only update the pixel value of the generated image. Unlike a classical deep learning problem which has a dataset, Neural Style Transfer only has two images and a third is generated using the algorithm.

Figure 1 shows an example of styling Mona Lisa by Picaso.

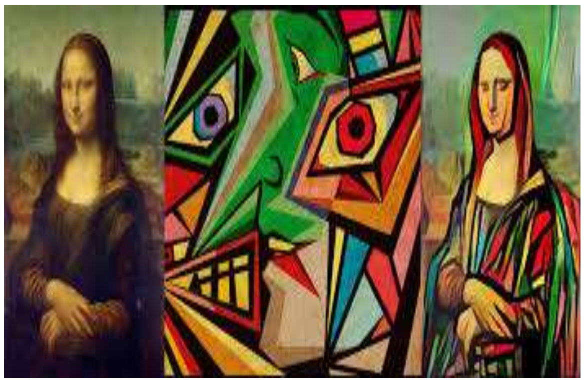


Figure 1. Example of styling Mona Lisa By Picasso

2. METHODOLOGY

VGG 19 MODEL

VGG-19 is a convolutional neural network that is about 19 layers deep. To use transfer learning we use the already trained version of it which is available on the internet (the weights of the connections are downloaded). This ensures that the model will be able to recognize the content of most of the images.

We don't make use of any of the fully connected layers. For our work we found that if we use max pool instead of average, the results are better.

2.1 CONTENT AND STYLE REPRESENTATIONS

We use the middle layers of the model to extract the image's content and style. Starting from the input layer of the network, the results of the initial few levels indicate low-level features like edges and textures, whereas the final few layers indicate higher-level features like wheels or eyeballs. To obtain the content and style of the image, one must obtain the output of these intermediary layers [4].

In this research work the middle layers used are:

```
content_layers = ['block5_conv2']

style_layers =  ['block1_conv1',

           'block2_conv1',

           'block3_conv1',

           'block4_conv1',

           'block5_conv1']
```

Selection of these layers is just by trial and error.

2.2 CALCULATING THE LOSSES

Loss function is a function that compute the deviation of the value given by the network from the expected value.

2.3 CONTENT LOSS

Content loss is the difference between "content" of final image and the content image. This makes sure that the final image has a good representation of the content.

We pass the content image(p) and generated image(x) through the network and take the output of the layers we selected above[5]. Then we simply take the euclidean distance between the outputs for these two images as in equation 1

$$L^l_{content}(p,x) = \sum_{i,j}\left(F^l_{ij}(x) - P^L_{Ij}(p)\right)^2 \qquad (1)$$

Where 'i' and 'j' is used to specify the pixel value at ith row and jth column, 'F' and 'P' denote the output at layer 'l' for generated and content image respectively. We use gradient descent to minimize this function.

2.4 STYLE LOSS

Style loss is the difference between "style" of final image and the style image. To do this, we had to first calculate gram matrix of both the images and then input those values in the style loss function given as in equation(2)

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

Where 'i' and 'j' is used to specify the pixel value at ith row and jth column, 'G' and 'A' denote the output at layer 'l' for generated and style image respectively. Nl offers the number of matrices in the output of that layer and Ml is the product of the height and width of the image[5]. We will use gradient descent to cut back on this function as in equation(3).

$$L_{style}(a, x) = \sum_{l \in L} W_l E_l \qquad (3)$$

where we can modify the influence of each layer in the final loss function. For our work we set equal weight to each layer.

The final loss function we minimise is given in equation 4:

$$\mathrm{L}_{\text{total}}(\dot{p}, \dot{a}, \dot{x}) = \alpha \mathrm{L}_{\text{content}}(\dot{p}, \dot{x}) + \beta \mathrm{L}_{\text{style}}(\dot{a}, \dot{x}) \qquad (4)$$

where α and β the content and style reconstruction weighting factors, respectively[6]. The Adam optimizer is what we utilise.

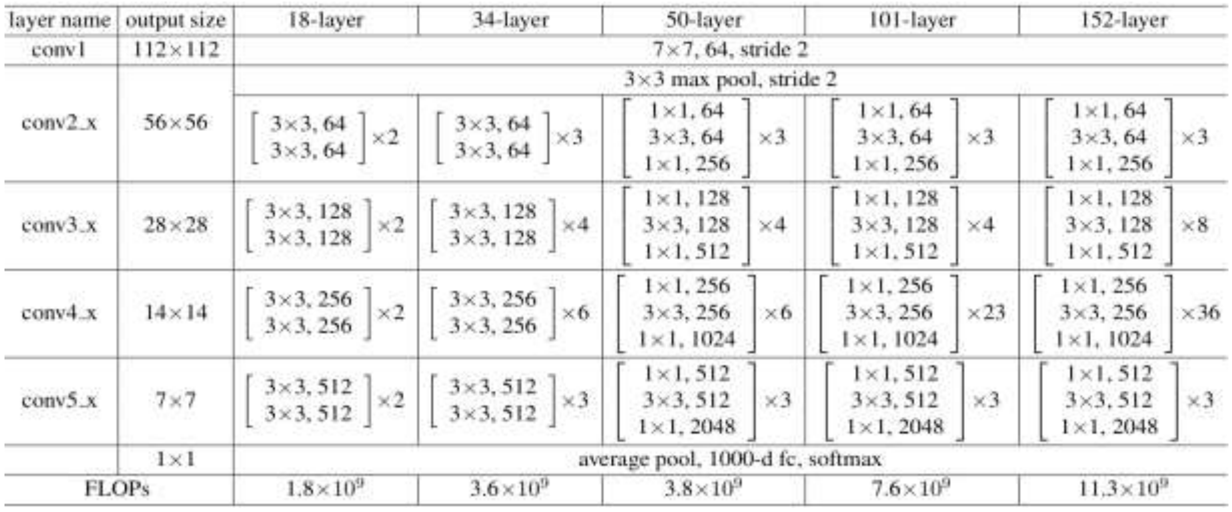
2.5 INITIALISATION OF GRADIENT DESCENT

The created image was started with white noise, but it could also be started with the content or the style image. It should be noted that even though initializing the final image with the content or style image reduces loss function at a faster rate thus reducing execution time but it makes the final image heavily biased to the initialized image.

2.6 RESNET 50 MODEL

ResNet-50 is a 50-layer deep convolutional neural network [7]. As more layers are added to the neural network the loss function forms a plateau and then starts to increase which is unreasonable. This phenomenon is attributed to a problem called as vanishing gradient problem. To solve this Residual Neural Networks (RESNET) was introduced. Resnet uses a strange method known as skip connections. What skip connections do is they backpropagate through an identity function therefore eliminating the problem of vanishing gradient [3]. We sought to investigate a structure that has previously been shown to be effective for picture categorization.

The Architecture of RESNET 50 is given in figure 2.



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 2. Architecture of Resnet 50

3. IMPLEMENTING THE MODEL

The pre-processing techniques used in VGG are also applied here. We normalize the image (BGR format) using mean = [103.939, 116.779, 123.68]. We then select which layers to use to represent our content and style as we did in vgg-19[8].

3.1 CONTENT AND STYLE LAYERS

Content layers: In most cases, activating just one layer is enough to convey the image's content.

Style layers: To select style layers, lot of trial and error is needed. The execution time increases as we increase the number of selected layers. For our work we used 4 layers to represent the style of the image.

3.2 GRADIENT DESCENT

In our project, we used the Adam optimizer to increase the speed of execution. We iteratively change the pixel values of our final image to reduce our loss function. We don't update the weights of our network, but update the pixel values .

The values of α and β and learning rate we used to implement using this model is very different to those used in VGG-19. Here we use $\beta = 1e6$, $\alpha = 1$, and learning rate = 20. These values were finalized by trial and error.

4. COMPARISION OF VGG19 AND RESNET50

The pre-trained VGG-19 model worked very well. We could use any 4-5 layers and it worked but the only constraint was that those layers should be well spaced and not only the initial and final layers. Trying to implement the same algorithm using resnet-50 didn't yield good results as it was tedious to find sets of layers that worked and even if we found some, the images generated were not visually appealing.

The VGG-19 model is very big(575MB) and the number of weights are 143,667,240 which helps to clump lot of information which the other state of the art models with the aim of being faster and reducing the number of parameters leave out and hence it generalises well for style transfer problems[9]. Resnets on the other hand combine the output of different paths to make a prediction i.e. unlike VGG where there is only a single path from the input to the output layer, in Resnet due to the presence of skip connections there are multiple paths. For example for a Resnet which is 110 layers deep, during backpropagation the layers which give most to it are only 10-34 layers. This helps to solve the vanishing gradient problem but causes the features to be spread out across the model making it difficult to pull them out and causing the sequence of aggregation of features very difficult to determine.

In VGG since there are no skip connections, the aggregation of features happen in a predictable way i.e. the lower layers recognize simpler features like edges and the higher layers recognise complex features like eyes. Hence skip connections in Resnet which were the key to its exceptional performance for tasks such as object localization become the main reason for its inability to work well for style transfer problems.

Therefore, to make Resnets work well, lot of trial and error using hyperparameter tuning is required and it does not generalize well.

5. USE CASES AND APPLICATIONS

have a look at how style transfer can be used to the following scenarios:

- Social media
- Gaming
- Photo and video editors
- Virtual Reality
- Commercial art

5.1 PHOTO AND VIDEO EDITORS

Software for editing pictures and videos is one of the most essential forms of style transfer. These kinds of tools are going to possess lots of power as they will be able to add iconic art styles to pictures and videos[10]. This will allow users to share stylish selfies, add to user-generated music videos, and do much more.

Style transfer models are easily integrated on edge devices because they possess the performance and flexibility of modern deep learning algorithms. Applications on mobile phones, for instance, enable real-time processing and conversion of images and videos. This implies that better than ever, high-quality photo and video editing software will be easier to find and use.

Many excellent tools already exist that incorporate style transfer into their toolkits. Here are some applications and demos [10]:

• Video Star (iOS, Art Studio)

• The Painter's Lens on iOS

• Instant Painting's AI Painter (Desktop)

• Looq (iOS)

• Transferring Styles Arbitrarily in the Desktop Browser

5.2 COMMERCIAL ART

An era of AI-generated poetry, art, and music has just begun. Style transfer is one of the computer vision techniques that makes this wave of AI-powered artwork creation possible. Whether it is art that is sold at a high-end auction or emerging artists finding new ways to share their appeal with the public, style transfer guarantees that we change the way we think about art, what originality means, and how we portray art in the real world.

Another application for style transfer would be to create dependable, high-quality prints for commercial spaces or advertising campaigns.

Figure 3 shows the Neural Network Powered Photos to Painting (AI PAINTER)

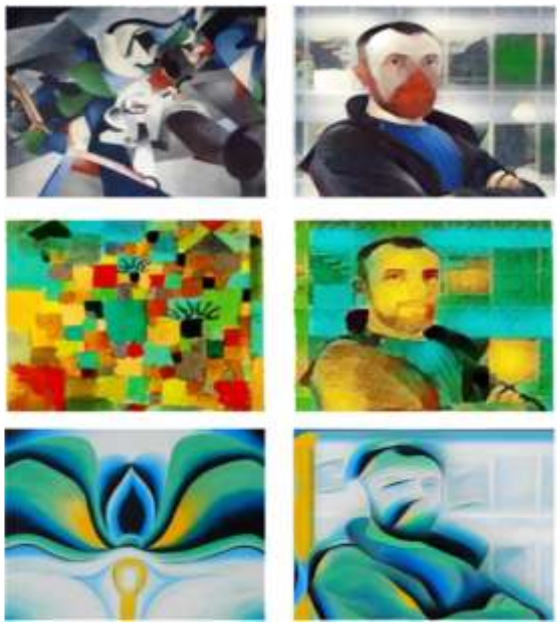


Figure 3. Neural Network Powered Photo to Painting (AI PAINTER)

5.3 GAMING

Google made the announcement of Stadia at a press conference held during the 2019 Game Developers Conference. The business is well-known for providing a cloud-based streaming service for video games. One of the main features of the demo was an in-game style transfer function that recreated the virtual world's textures and colour scheme automatically from a potentially infinite variety of art styles[10].

This, according to Google, is an attempt to "motivate the artist inside of every developer." People who we might not typically think of as artists will have more access to artistic creation thanks to gaming.

5.4 VIRTUAL REALITY

Similar to video games, where users are immersed in captivating virtual environments, virtual reality is also interested in investigating the potential of style transfer.

Although the possibilities are intriguing and bright, style transfer applications in virtual reality are still mostly in the research stage. Facebook claims that style transfer has the power to completely transform the way virtual reality developers tell visual stories in their apps, games, movies, and other media.

5.5 SOCIAL MEDIA

NST is a terrific idea for social media firms that are always looking for new and inventive methods to keep their customers interested on their platform. Because there are so many different content and style images to choose from, as well as the possibility for users to share their artwork with their friends, NST provides businesses with a fresh and simple approach to keep their users engaged.

6. CONCLUSION

Neural Style Transfer is the process of employing CNNs to generate a content image in different styles (NST). Since then, NST has been a popular issue in academic circles as well as in industry. It is gaining popularity, and other techniques to improving or extending the original NST algorithm have been presented. It is demonstrated that a neural system can learn image representations that permit some separation of the content and style of an image after being trained to handle one of the basic computational tasks of biological vision. This is justified by the fact that in order to maintain object identity during object recognition training, the network must become invariant to all visual fluctuations.

REFERENCES

[1] A Neural Algorithm of Artistic Style [*PDF] Gatys, L.A., Ecker, A.S. and Bethge, M., 2015. CoRR, Vol abs/1508.0657*

[2] Neural Style Transfer using ResNet50 with tf.keras by Leo Barbosa. [Link] (https://www.kaggle.com/lbarbosa/neural-style-transfer-using-resnet50-with-tf-keras)

[3] Cai, Xiuxia, and Bin Song. "Combining inconsistent textures using convolutional neural networks." Journal of Visual Communication and Image Representation 40 (2016): 366-375 (9)

[4] Gatys, L. A., Ecker, A. S., & Bethge, M. (2016). Image style transfer using convolutional neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2414-2423). (10)

[5] Balas, V. E., Roy, S. S., Sharma, D., & Samui, P. (Eds.). (2019). Handbook of deep learning applications (Vol. 136). Springer.

[6] Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576.

[7] Tensorflow Documentation [Link] (https://www.tensorflow.org/)

[8] Neural Information Processing Systems 28, 2015. 3, 4

[9] D. J. Heeger and J. R. Bergen. Pyramid-based Texture Analysis/Synthesis. In Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques,

[10] www.fritz.ai