IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 8 Issue 8, August 2021
ISSN (Online) 2348 – 7968 | Impact Factor (2020) – 6.72
www.ijiset.com

A Faster Neural Algorithm for Artistic Style Transfer

Aaliya M. Basheer[1], Jobin Jose[2] and Dr. Sajith K. [2]

[1,2] Department of Electronics and Communication Engineering, Govt. Engineering college, Mananthavady, Wayanad, Kerala-670644, India

Abstract

Neural Style transfer is an artificial system meant for creating artistic images using a Deep Convolutional Neural Network (DCNN). The visual appearance or style of an image or video can be artistically modified with the help of neural style transfer. All previous approaches of style transfers are implemented using VGG architectures, where residual/skip connections are absent. But a large number of parameters present in the VGG architecture increases the execution time of the algorithm. So in this paper, we are performing style transfer using ResNet50, a lightweight CNN model to improve the speed of the algorithm.

Keywords: *CNN, ResNet50, Style Transfer, VGG19, Artistic Style.*

1. Introduction

The main problem with texture transfer is in transferring style from one image to another. The goal behind texture transfer is to combine textures from input images while maintaining the texture production for preserving the contents of the target image. Photorealistic textures are produced using existing non-parametric algorithms by resampling the pixels for a given texture of input. Most of the previous approaches of style transfer algorithms are focused on these nonparametric techniques of texture synthesis which uses different methods for preserving the textures of the target image.

However, recent advances in Convolutional Neural Networks (CNN) has created powerful systems which are trained for extracting high-level linguistics info from pictures. It was shown that CNNs that are trained on object recognition are learned for extracting high-level feature representations that generalize datasets and even different visual scientific discipline functions, as well as texture identification and artistic style transfer classification [1].

This work shows how the general feature renditions trained by these Convolutional Neural Networks are used in manipulating the content and also the style of images. Here “A Faster Neural Algorithm of Style Transfer”, an algorithmic program is introduced to perform quicker image style transfer. Simply, it is a style or texture transfer algorithmic program that employs a texture synthesis method by feature representations from progressive CNNs. Since this model supports deep image representations, the style transfer technique minimizes to an optimization problem inside a network. New images that match feature representations of example pictures are generated by achieving pre-image search. In fact, our style transfer algorithm combines a faster texture model by using neural networks.

2. Literature Survey

The paper “Neural Style Transfer: A Review” demonstrates the power of Convolutional Neural Networks (CNNs) in creating artistic images by the separation and recombination of style and content images [1]. This process of using neural networks to obtain a content image in distinct styles is known as Neural Style Transfer (NST). NST has always been an attractive topic in various applications. In this paper, the current progress towards NST is mentioned. There are 2 types of evaluation methods for style transfer - Qualitative and Quantitative evaluations. The paper concludes by discussing different applications of NST and problems in the future.

The paper "Very Deep Convolutional Networks for Large-Scale Image Recognition" introduces the depth effect of convolutional networks based on its precision in the large-scale image identification setting [3]. The main improvement of this work is by a detailed study of networks of expanding depth incorporating an architecture that uses very small (3 × 3) convolutional filters, which showed an interesting improvement about present configurations of that point which may be obtained by extending the depth to 16-19 weight layers. Rectified Linear filter (ReLu) is used for obtaining a better classification by introducing non-linearity. These findings were on the basis of ImageNet Challenge 2014. It was also noted that these representations generalize well to other datasets, where current results are achieved.

The paper "Feature Guided Texture Synthesis (FGTS) for Artistic Style Transfer" discusses about how the artistic images (paintings) are regarded as a combination of style and content. The aim of artistic style transfer is to produce a well stylized image taking content of input source image, but that is style from style image. Based on texture generation, an algorithm for artistic style transfer known as Feature Guided Texture Synthesis (FGTS) is proposed in this paper [4]. When we compare this method with the existing methods, the content of the source image is found to be better defined in FGTS since the source image generates the feature field. Even though the style modelling is with low-level features, the feature field which combines both style and content during the process. Also, a L_2 neighborhood distance metric is provided for better perceptual similarity. Comparisons and results of this work indicate FGTS as an efficient technique for artistic style transfer.

The paper "State of the Art: A Taxonomy of Artistic Stylization Techniques for Images and Video" paper examines non-photorealistic rendering (NPR), a technique that focuses on conversion of 2D inputs (images and video) into artistic renditions [5]. A classification of the 2D NPR algorithms that was created over the years is presented according to the behavior and characteristics of every method. Then chronological development of semi-automatic paint systems in the starting of nineties are described, going to the automated rendering systems in the late nineties inspired by picture gradient analysis. Two corresponding methods in the NPR literature were mentioned in this work. First is the blending of NPR algorithm with upper level computer vision which illustrates the trends toward scene analysis in order to drive diversity of favor and artistic abstraction. Second, the advancement of local processing methods by edge-aware filtering stylization of images and video in real-time. The survey comes to an end by discussing the challenges of 2D NPR including topics like aesthetic user evaluation.

The paper "Transfer learning for image classification" mentioned that Convolutional neural network has achieved great recognition for its information mining and powerful extraction of features. CNN are used extensively for wide range of applications such as object identification, semantic segmentation, image super-resolution etc. due to its well-planned learning mechanism and feature extraction. Keeping the standard learning layout constant, different CNN architectures were introduced to enhance the existing performance of the system [6]. CNN architectures like AlexNet, VGG16 and VGG19 are mentioned in this work for object recognition. A pre-trained network (VGG19) variable in image categorization task is used for transfer learning. VGG19 architecture performance is compared with that of the networks VGG16 and AlexNet. In addition, with the CNN architectures, a comparison of another hybrid approach containing a powerful extraction of features from CNN architecture is made which is accompanied by a classifier - support vector machine (SVM). Two databases are used here: GHIM10K and CalTech256 for studying CNN architecture's effect for functional feature extraction. Average recall, F-score and precision are carried out for evaluating the performance evaluation. It was observed that VGG19 architecture outclasses other CNNs in their mixed learning approach for image categorization task.

3. The Proposed System

3.1 Methodology

The purpose of this project is to implement a system using the network ResNet50 for executing a faster style transfer with minimized loss for the creation of artistic images. The steps executed are given below:
Step 1: Choose 2 images – content image and style image as inputs and initialize.
Step 2: Initialize target image with either style or content image.
Step 3: Use ResNet50 network architecture for extracting certain features from input image.
Step 4: Use NAdam optimization algorithm to minimize loss.

Step 5: Suppose we initialize content image as target image. Find loss between input content image and target content image (i.e. content loss= 0).
Step 6: Then find loss between input style image and target content image (style loss not zero).
Step 7: Minimize the style loss.
Step 8: A style will be formed in content image.
Step 9: Use style and content weight ($\mathbb{L}_s$ and $\mathbb{L}_c$) to find total loss ($\mathbb{L}_t$).
Step 10: Update the target image.
Step 11: Run a few train steps and if it works then perform longer optimization with 100 train steps.
Step 12: Save the result.
The block diagram of style transfer is given in Fig. 1.

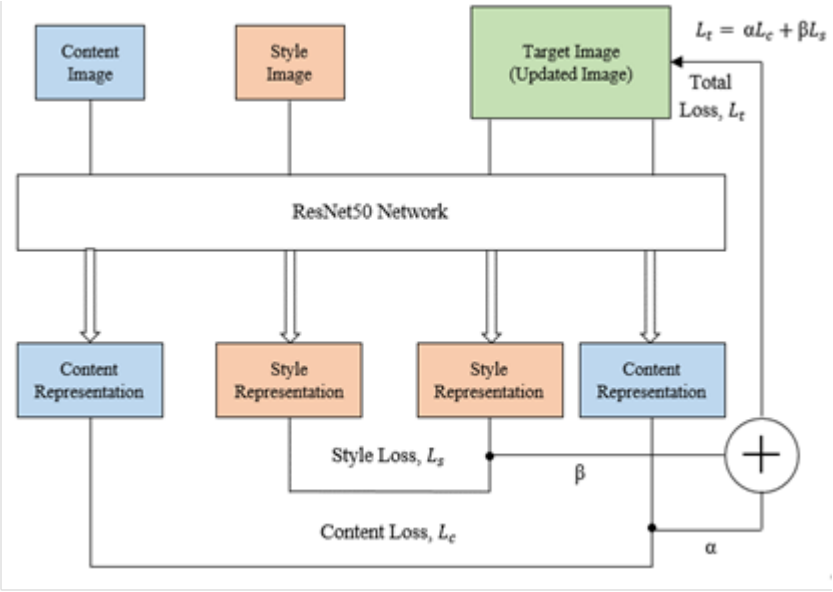

Fig. 1. System structure diagram.

Here content loss $\mathbb{L}_c$ and style loss $\mathbb{L}_s$ are multiplied by weighting factors α and β which are content weight and style weight respectively. Losses are added to get total loss and target image is updated every time.

3.2 Deep Image Representations

The results that are obtained were produced using ResNet50 network, which is instructed to execute faster object identification and localization. The feature space given by the normalized genre of the 48 Convolution layers together with 1 Average Pool and 1 MaxPool layer is used. It is a widely used ResNet50 model and its architecture is also explored extensively. The network is normalized by performing weight scaling so that the total activation over images and positions of every convolutional filter equals one. This type of re-scaling can be done using ResNet50 network without affecting the output. The model is executed in a framework known as Tensor Flow [7].

3.2.1 Content Representation

In order to visualize the image information which is encoded at various layers of the hierarchy, a gradient descent needs to be performed on a White noise image for finding another image which complements to the feature responses of the actual image (Fig. 2, content reconstructions) [2]. Let $\vec{p}$ and $\vec{x}$ be the actual image and the image that is to be produced respectively, and P^l and F^l be their representation of features in the layer l respectively. The loss between the two feature representations known as the content representation is given in Eq. (1).

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

Higher layers of the network are capable of capturing higher level contents in terms of their positioning of objects in the input image but does not require the actual pixel values of the reconstructed image (Fig. 2, content reconstructions d, e). Whereas, reconstructions taken from the lower layers just simply indicate the perfect pixel values of the actual image (Fig. 2, content reconstructions, a-c). Consequently, the feature responses taken from upper layers are known as the content representation of the network.

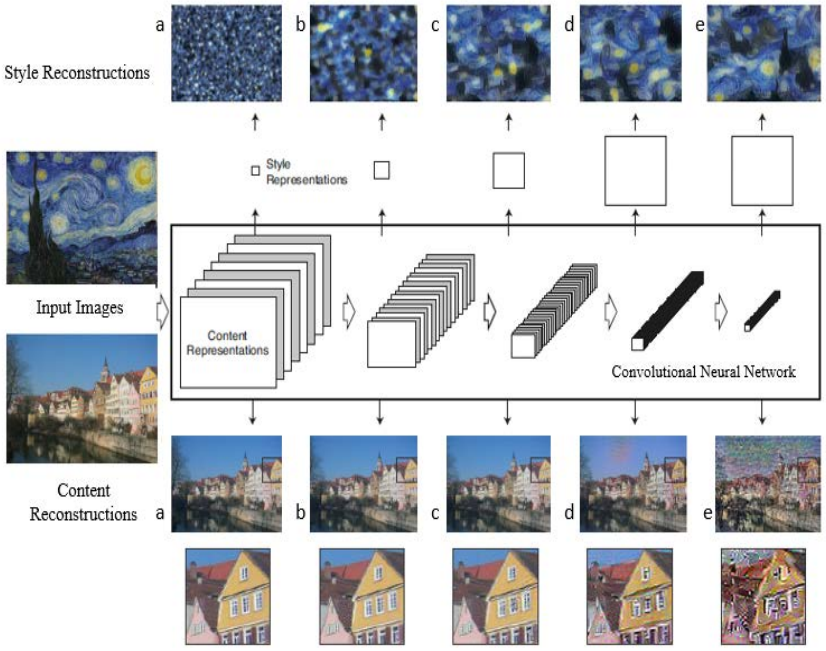

Fig. 2. Image representations in a Convolutional Neural Network [2].

3.2.2 Style Representation

To get the style representation of a given input image, we need a feature space that is designed in a way to express the details about the texture which is constructed above the filter responses of the network. It contain feature correlations which is represented by the Gram matrix $G^l \in \mathcal{R}^{N_l \times N_l}$, where G_{ij}^l is the inner product between the feature maps i and j of layer l as given in Eq. (2).

$$G_{ij}^l = \sum_k F_{ik}^l F_{jk}^l. \qquad (2)$$

Then, the information that is represented by this style feature spaces built on various layers of the network is represented by constructing an image that matches the style representation of given input image (Fig. 2, style reconstructions). Let $\vec{a}$ and $\vec{x}$ be the original image and the image that is to be generated, and let A^l and G^l be the respective style representation in layer l. The total loss is then expressed as Eq. (3)

$$E_l = \frac{1}{4N_l^2 M_l^2}\sum_{i,j}(G_{ij}^l - A_{ij}^l)^2 \qquad (3)$$

and the total style loss is given in Eq. (4)

$$\mathcal{L}_{style}(\vec{a},\vec{x}) = \sum_{l=L0} \omega_l E_l \qquad (4)$$

where ω_l are the weighting factors that contribute to entire loss in each layer.

3.3 Style Transfer

In order to transfer the style of an artistic work $\vec{a}$ over a photograph $\vec{p}$ we have to create a new image that can match both the style representation of $\vec{a}$ and the content representation of $\vec{p}$ (Fig. 3).

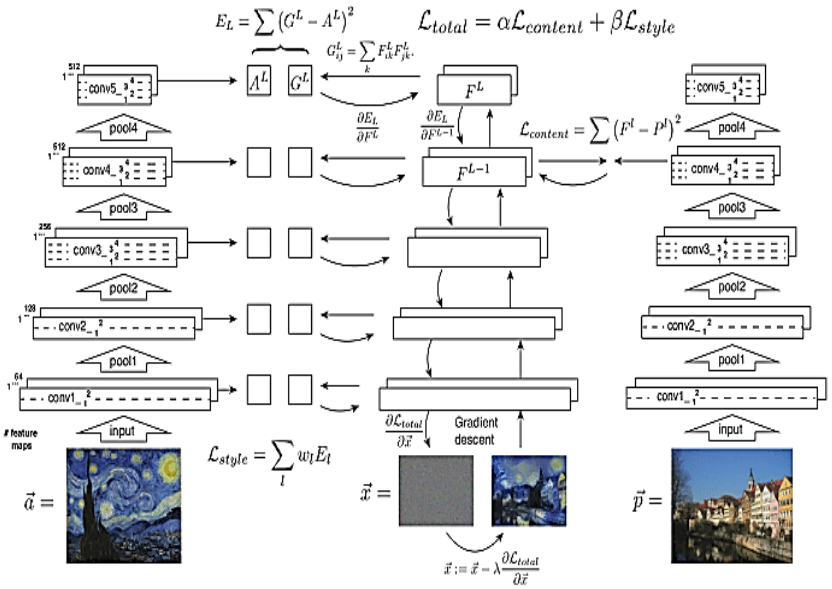

Fig. 3. Style transfer algorithm [2].

The minimized loss function is given by Eq. (5).

$$\mathcal{L}_{total}(\vec{p},\vec{a},\vec{x}) = \alpha\mathcal{L}_{content}(\vec{p},\vec{x}) + \beta\mathcal{L}_{style}(\vec{a},\vec{x}) \qquad (5)$$

Here α and β represents the weighting factors for content representation and style representation respectively.

4. Results and Discussions

The key finding of this paper is that the content and style representations in the Convolutional Neural Network can be done faster using ResNet50. That is, both representations can be utilized independently for producing new and visually meaningful images. NAdam optimization is used which is found to perform well for faster style transfer. Some differences in the image synthesis are quite normal because of the difference in network architecture and optimization algorithm that is used.

4.1 Comparison Table

Previous artistic style transfer was performed using the CNN VGG19. After training, it is observed that by using the network ResNet50 style transfer is executed at a faster rate than by using VGG19 (Table 1.).

Table 1: Comparison between VGG19 and Resnet50.

VGG19	***RESNET50***
Visual Geometry Group	Residual Network
19 layers deep	50 layers deep
144 million parameters	23 million parameters
Takes more time with reduced accuracy	Takes less time compared to VGG (2 times speed)

4.2 Layers Used

The layers used in ResNet50 are 2 content layers and 8 style layers since style transferred is less in ResNet50 (Fig. 4). The CNN ResNet50 has many convolutional and pooling layers. Convolutional layers are used to extract highly complex features. Pooling layers are used for reducing the dimension of the feature maps and hence reduces the amount of our computation [2]. When layers in CNN increase, the model will have the ability of fitting more complex functions. Intermediate layers are taken here.

```
content_layers = ['conv4_block5_1_relu',
                  'conv5_block3_2_relu']

style_layers = ['conv2_block1_2_relu',
                'conv2_block2_1_relu',
                'conv3_block2_1_relu',
                'conv3_block4_3_conv',
                'conv4_block4_2_relu',
                'conv4_block6_2_relu',
                'conv5_block2_2_relu',
                'conv5_block3_2_relu']
```

Fig. 4. Content layers and Style layers used in the model.

4.3 Hyperparameters used

Parameters that has to be set manually and can be tuned are known as hyperparameters. The determine the network structure and are set before the training. Overall the learning process is controlled by these parameters (Table 2).

Table 2: Hyperparameters used in VGG19 and ResNet50.

Type of CNN	***VGG19***	***RESNET50***
Optimizer	Adam	NAdam
Learning Rate	0.02	0.09
Style Weight	1e-2	0.5e-2
Content Weight	1e4	0.3e4
Processor	Tesla K80 GPU	Tesla K80 GPU
Training Time (10 Train Steps)	49s	40s
Training Time (100 Train Steps)	7m 59s	3m 49s

Some examples of hyperparameters include learning rate, batch size, weights etc. While using NAdam optimizer, β_1 and β_2 values are also given which are the initial decay rates used for estimating first and second moments of gradient [8]. These β_1 and β_2 are multiplied at the end of each training step. It can be noted that while using ResNet50 training time is actually 2 times faster than VGG19 used in the base paper.

4.4 NAdam Optimizer

It is similar to Adam optimization but is RMSprop with momentum (NAdam is Adam with Nesterov momentum). It is abbreviation for Nesterov accelerated Adaptive Moment Estimation. The equation for NAdam optimizer is given in Eq. (6).

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \left(\beta_1 \hat{m}_t + \frac{(1-\beta_1) g_t}{1-\beta_1^{\,t}}\right) \quad (6)$$

where, θ_{t+1}, θ_t - Weight updates
η - Step size/learning rate
$\hat{v}_t$ - Squared gradient
g_t - Current gradient
ϵ - Smoothing term
β_1 - set as default to 0.9
$\hat{m}_t$ – Momentum

4.5 Inputs and Outputs

These are the content image and style image which are given as inputs (Fig. 5). Outputs are generated using the style transfer algorithm mentioned earlier (Fig. 6).

Fig. 5. Input images

Fig. 6. Output images obtained

Images generated by mixing the style and content representations from two different sources are given here. We match the style representations of a well-known artwork with several content representation of photographs (Fig 7.).

Fig. 7. Images that combine the style of an artistic work with the content of several photographs.

4.6 Graphical Representation

During every epoch, we have to calculate the loss function across each data item and the quantitative loss measure is obtained at each epoch. Table 3. gives the values of losses for 3 types of losses which are content loss, style loss and total loss.

Table 3: Table of values for all 3 losses.

Epoch	***Content Loss***	***Style Loss***	***Total Loss***
1	507.839691	0.00124816853	507.84021
2	544.037109	0.000533309882	544.037354
3	553.192	0.000244335795	553.192261
4	485.072144	0.000237490938	485.072418
5	447.96051	0.000260793022	448.990815
6	437.54126	0.000290565222	437.541595
7	411.235748	0.000325598026	411.236115
8	403.667572	0.000365857763	403.667969
9	390.299133	0.000401652273	390.299133

As the epoch increases, loss is minimized and the model goes from underfitting to overfitting. By plotting the curve, it is seen that loss function consistently decreases for all the three losses – content, style and total losses (Fig. 8). Since the model is initialized using style image, style loss is approximately equal to zero. Content loss and Style loss together gives Total loss.

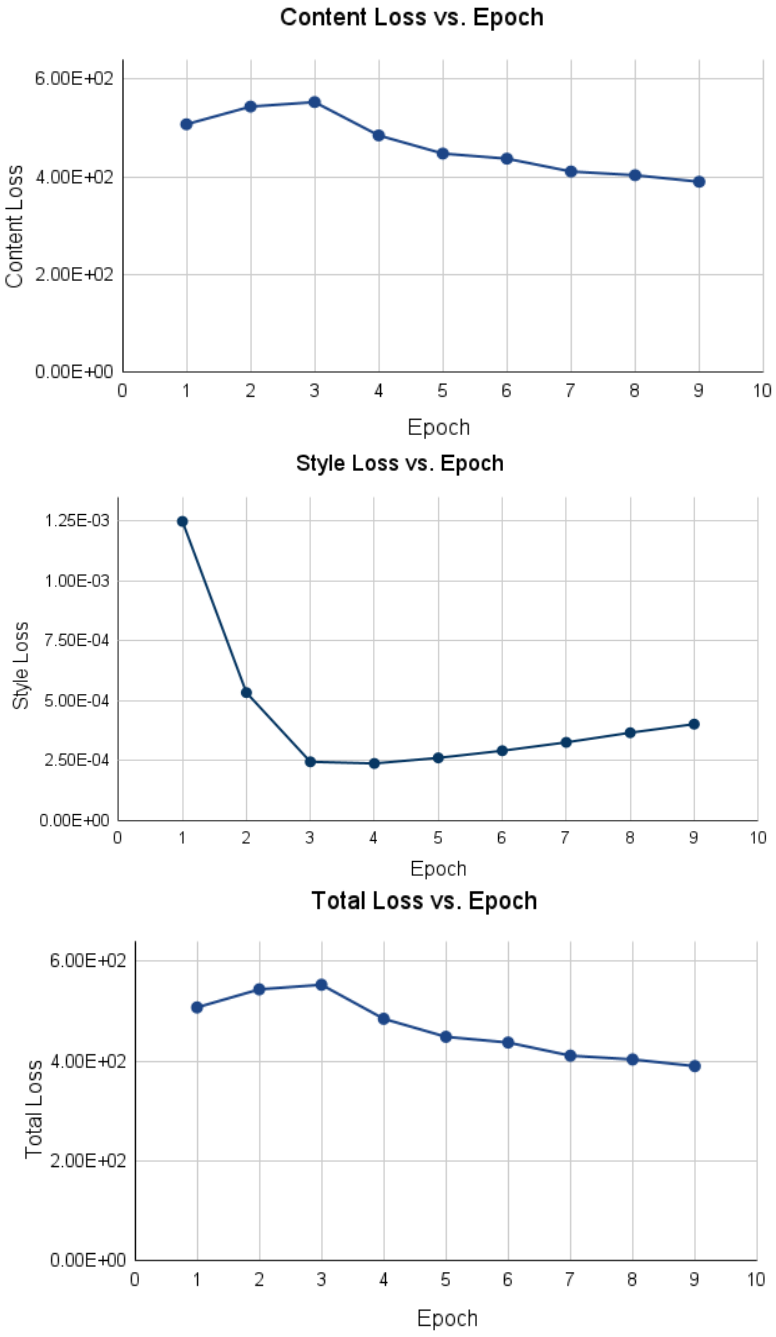

Fig. 8. Plots of Loss Vs Epoch for Content loss, Style loss and Total loss.

4.7 Advantages of ResNet50

- The fundamental breakthrough with ResNet50 was that it can successfully train exceptionally deep neural networks with 150+ layers.
- It is an advancement of ResNet50 which does not have the problem of vanishing gradients and hence training is very effortless with this deep neural network.
- ResNet50 is a powerful model that is used very commonly in many computer vision tasks.
- Model can be trained quickly since the number of parameters are lesser.

4.8 Limitations

- It is observed that due to the complexity of ResNet50, the feature maps do not work as well as those of VGG19.
- Style is not transferred as in VGG19 while using ResNet50.
- There is a reduction in validation accuracy.

5. Conclusions

All previous approaches of style transfer require high execution times, and it varied with the underlying CNN architecture. In this paper, we implemented style transfer using the ResNet50 CNN architecture which helped to produce a stylized image with sufficient quality. As expected, due to the smaller size of the ResNet50 architecture, the algorithm execution time was smaller in comparison with the VGG19 architecture. But due to the skip/residuals present in the ResNet50, it is found that the generated image is not as good as in previous approaches. Hyperparameters were modified for optimizing the visual quality of the final image. For improving the visual quality of stylized images, here we used 8 intermediate layers in the ResNet50 architecture. NAdam optimization techniques were used in this algorithm, which helped to improve the execution time and visual quality of the final image.

References

[1] Jing, Y., Yang, Y., Feng, Z., Ye, J., Yu, Y., & Song, M. (2020). *"Neural Style Transfer: A Review"*. IEEE Transactions on Visualization and Computer Graphics, 26(11), 3365-3385. [8732370].
[2] Leon A. Gatys, Alexander S. Ecker, Matthias Bethge1, *"A Neural Algorithm of Artistic Style"*, 2016 IEEE Conference on Computer Vision and Pattern Recognition.
[3] K. Simonyan and A. Zisserman. *"Very Deep Convolutional Networks for Large-Scale Image Recognition"*, arXiv:1409.1556 [cs], Sept. 2014. arXiv: 1409.1556.
[4] Xie, X., Tian, F. & Seah, H. S. *"Feature Guided Texture Synthesis (FGTS) for Artistic Style Transfer"*, In Proceedings of the 2Nd International Conference on Digital Interactive Media in Entertainment and Arts, DIMEA '07, 44–49 (ACM, New York, NY, USA, 2007).
[5] J. E. Kyprianidis, J. Collomosse, T. Wang, and T. Isenberg. *"State of the 'Art': A Taxonomy of Artistic Stylization Techniques for Images and Video"*, Visualization and Computer Graphics, IEEE Transactions on, 19(5):866–885, 2013.
[6] M. Shaha, M. Pawar, *"Transfer Learning for Image Classification"*, 2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA).
[7] W. Abdulla. *Mask. r-cnn for object detection and instance segmentation on keras and tensorflow*. https:// github.com/matterport/Mask_RCNN, 2017.
[8] *U Güçlü* and M. A. J. v. Gerven, *"Deep Neural Networks Reveal a Gradient in the Complexity of Neural Representations across the Ventral Stream"*, The Journal of Neuroscience, 35(27):10005–10014, July 2015.

Aaliya M. Basheer was born on June 29th, 1997 in Wayanad district, Kerala, India. She received her bachelor's degree (B.Tech) in Electronics and Communication Engineering in 2019 and completed her master's degree (M. Tech) in Communication Engineering and Signal Processing in 2021 from Government Engineering College Wayanad, Kerala Technical University (KTU), Kerala. Her research interests lie in the area of Machine Learning and Image Processing techniques like feature detection and extraction.

Mr. Jobin Jose is currently working as an Assistant Professor in Government Engineering College Wayanad, affiliated to APJ Abdul Kalam Kerala Technological University. He received a bachelor's degree in Electronics and Communication Engineering from Government Engineering College, Kannur, and a master's degree in Applied Electronics and Instrumentation Engineering from College of Engineering, Trivandrum. His area of specialization including Machine Learning and Image Processing.

ISSN (Online) 2348 – 7968 | Impact Factor (2020) – 6.72

www.ijiset.com

Dr. Sajith K. received the B.Tech degree from College of Engineering Thalassery, Kerala in 2011, and M.Tech degree in Pondicherry Central University in 2013, and Ph.D degree in 2020, Electronics and Communication Engineering, Dept. of Electronics Engineering, Pondicherry Central University, under the guidance of Dr.T. Shanmuganantham (Gold medallist in Antenna research from NIT Trichy), Dept. of Electronics Engineering, Pondicherry central University. He has 3 years of teaching experience in various reputed Engineering Colleges and currently he is working as Asst. Prof. in the Dept. of Electronics and communication Engineering, Govt. Engineering College Wayanad, Kerala Technical University (KTU), Kerala. He is a member of IEEE, IEEE APS society, and IEEE MTTs Kerala chapter. During his research received a senior research fellowship grant award from the Government of Kerala. During his research carrier, he developed many metamaterials loaded CPW fed on-body antennas for healthcare monitoring applications, and also he received “Five Best paper awards” in various IEEE conferences. He has authored 15 international journals, 3 chapters in books, and 25 International conference papers. His research interest in the area of planar monopole antennas, FSS for electromagnetic shielding, Wearable and Implantable medical antennas, Microwave and Millimeter-wave antennas, Fractal Antennas, Metamaterial inspired antennas, and RF MEMS reconfigurable antennas, MEMS Phase shifter.