Journal of Harbin Engineering University Vol 44 No. 10
ISSN: 1006-7043 October 2023

Implementing Neural Style Transfer Using Python-Based Keras Library And VGG-19 Convolution Neural Network

Dr. T. Deepa[1*], Ms. S. Srisowmiya[2], Ms. A. Priya[3], Mrs. M. Jeevitha[4], Mrs. S. Sri Sakthi Hamrish[5]

[1*]Associate Professor & Head, Department of Computer Science, KPR College of Arts Science and Research,
Email ID: deepa.t@kprcas.ac.in
[2]Assistant Professor, Department of Computer Technology & Data Science, Sri Krishna Arts & Science College,
Email ID: srisowmikrishna@gmail.com
[3]Lab Technician, Department of Computer Science and Engineering, United Institute of Technology,
Email ID: priyasport14@gmail.com
[4]Assistant Professor, Department of Computer Science and Engineering, United Institute of Technology,
Email ID: jeevithamaruthachalam92@gmail.com
[5]Assistant Professor, Department of Computer Science and Engineering, United Institute of Technology,
Email ID: sakthisrinivasan2198@gmail.com

***Corresponding Author: Dr**.T. Deepa
*Associate Professor & Head, Department of Computer Science, KPR College of Arts Science and Research,
Email ID: deepa.t@kprcas.ac.in

Abstract— A series of software algorithms known as Neural Style Transfer (NST) makes use of a neural network to modify and alter media scenes and environments. Unlike conventional techniques, NST is used in image and video editing software to enable image stylization based on a general model. Because of this, NST has become a hot topic in the entertainment sector. Experienced editors and media producers can produce content more quickly and make it available for public consumption. This work presents a comprehensive analysis of the state-of-the-art developments in Neural Style Transfer, including all relevant features including still photos and videos. The writers examined the various architectures in use and contrasted their benefits and drawbacks. This research makes use of the CNN (Convolution Neural Network) VGG-19 model and the Python-based Keras framework. It aims to give Deep Learning models the ability to distinguish between style representations and content images. The position of the style and content photos creates the illusion that the content image has been "Painted" to resemble the style reference image. It is an application for computer vision that uses image processing techniques along with deep convolution neural networks. The mean squared error between the content's features and the output image is the content loss. Using the features taken from the content and style images, the VGG-19 model feature extractor is utilized to determine the content loss and style loss. The difference in mean squared is the style loss
.

Keywords—Content Image, PyTorch, Style image, Content loss, Style loss.

I. INTRODUCTION

Neural Style Transfer focuses on PyTorch implementation of neural style transfer. An application of deep learning called neural style transfer enables us to blend the style of one image with the content of another image. Understanding the neural style transfer process and how it might be applied to produce beautiful images is the aim of this. The pre-trained VGG-19 network, a convolutional neural network that has been trained on millions of photos for object detection, is loaded first in this process. A pre-trained convolution neural network is utilized in this method, which is based on the neural style transfer algorithm, to extract features from both the content image and the style image. The algorithm operates by minimizing a total loss function, which is composed of the sum of two individual loss functions, namely, content loss and style loss. In order to achieve the required style transfer, the total loss is then minimized using an optimizer to update the target image. The Python programming language and PyTorch deep learning framework are used to carry out the research. To achieve neural style transfer, we make use of a number of PyTorch modules, including torch.nn, torch.optim, and torch vision.transforms. To demonstrate the effectiveness of the neural style transfer technique, a visualisation of the this paper intermediate and final results is also provided.

II .LITERATURE REVIEW

S.NO	AUTHOR & YEAR	TITLE OF THE PAPER	DATASET USED	ALGORITHM USED	VISUAL ACCURACY
01	Risser et al. 2017	Preserving Color in Neural Artistic Style Transfer	COCO dataset	VGG-19	High
02	P. Karthikeyan 2020	Neural Style Transfer for Indian Classical Dance Poses	Self-created dataset of 12 Indian classical dance forms	VGG-16	Low
03	Dong et al. 2017	Multi-Style Generative Network for Real-time Transfer	Microsoft COCO dataset	Multi-Style Generative Network	High
04	Chen et al 2017	Style Bank: An Explicit Representation for Neural Image Style Transfer	COCO dataset	StyleBank	Superior
05	K. J. AnandhaKumar 2020	Fuzzy Based Neural Style Transfer for Improved Image Quality	COCO dataset	VGG-19	Low
06	M. S. Jithin et al. 2021	Neural Style Transfer for Indian Art Forms: A Comparative Study	Self-created dataset of 7 Indian art forms	VGG-16	High
07	S. K. Padhy et al. 2020	Image Style Transfer Using Multi-Objective Genetic Algorithm	COCO dataset	VGG-19	N/A
08	P. V. Laxmi et al. 2020	Cultural Neural Style Transfer Using Multi Style Images	COCO dataset	VGG-19	High
09	N. Anand et al. 2020	Effective Neural Style Transfer Techniques for Indian Classical Dance Images	Self-created dataset of 12 Indian classical dance forms	VGG-19	High
10	R. R. Reddy et al. 2019	A Study on Neural Style Transfer of Traditional Indian Paintings.	Self-created dataset of traditional Indian paintings	VGG-19	N/A
11	S. Das et al. 2020	Indian Traditional Art Style Transfer using Deep Neural Networks	Self-created dataset of Indian traditional art	VGG-19	High
12	S. S. Sahoo et al. 2019	Neural Style Transfer with Wavelet Transforms for Cultural Artifacts	Self-created dataset of cultural artifacts	VGG-19	High
13	N. Anand et al. 2021	Performance Evaluation of Neural Style Transfer Algorithms on Indian Classical Dance Images	Classical Indian Dance Dataset	VGG-19, MobileNetV2, InceptionResNetV2	N/A
14	A. G. Kulkarni et al. (2020)	Neural Style Transfer for Indian Sign Language Gesture Images	Indian Sign Language (ISL) Gesture Dataset	VGG-19	High
15	N. R. Shetty et al. 2020	Neural Style Transfer for Indian Art: A Comparative Study	Indian Art Dataset	VGG-16, Inception-V3, ResNet-50.	High

II. METHODOLOGY

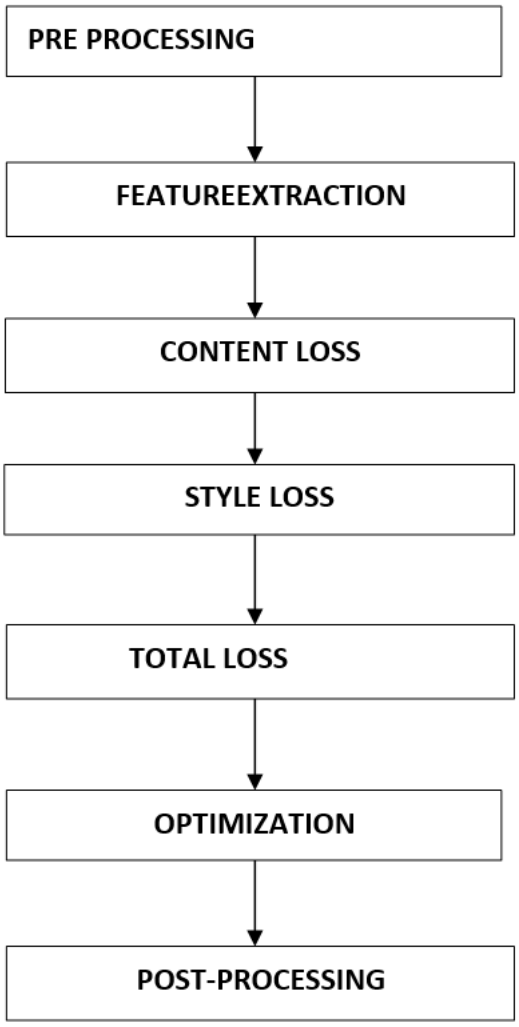

Fig 1 Methodology Overview

A. Pre processing

In Neural Style Transfer (NST), pre-processing refers to the step where the input images, i.e., the content image and the style image, are pre-processed before they are fed into the neural network. This step involves several operations such as resizing, normalization, and conversion to tensor format. Resizing: The first step in pre-processing is to resize the input images to a predefined size. The reason behind resizing the images is to ensure that both the content and style images have the same dimensions. In the above code snippet, both the content and style images are resized to 400 pixels. Normalization: The next step is normalization, which involves converting the pixel values of the input images to a standardized range. Normalization is important because it ensures that the pixel values are not too large or too small, which can cause the network to behave erratically. In the above code, the mean and standard deviation of the ImageNet dataset are used for normalization.Conversion to Tensor format: The final step in pre-processing is to convert the input images to tensors. Tensors are a fundamental data structure in deep learning, and they are used to represent multi-dimensional arrays of numerical data. In the above code, the input images are converted to tensors using the torchvision. transforms. ToTensor()` method.

B. Feature Extraction

In NST, feature extraction refers to the process of extracting the style and content features from the input images. This is typically done by passing the input image through a pre-trained convolutional neural network (CNN), such as VGG-19, which has been trained on a large dataset of images for image classification. The convolutional layers of the CNN can extract high-level features, such as edges, textures, and patterns, from the input image. These features are then used to compute the style and content representations of the image. For style representation, the Gram matrix is computed from the feature maps of one or more convolutional layers. The Gram matrix captures the correlations between the feature maps and represents the style of the image. For content representation, the feature maps of one or more convolutional layers are used to capture the content of the image. These feature maps are typically from the higher layers of the CNN, which have a more abstract representation of the image. Once the style and content features have been extracted, they are used to compute the loss

function for optimization, which is used to generate the stylized image that combines the style of one image with the content of another.

C. Content Loss

Content loss is one of the two main components of the loss function used in neural style transfer. It measures the difference between the content of the content image and the stylized image. The content loss is calculated using a pre-trained convolutional neural network (CNN) and a specific layer's feature maps. The VGG19 model is used as the pre-trained CNN. The content image is passed through the VGG19 model, and the feature maps of the chosen layer (in this case, layer 4) are extracted. Similarly, the stylized image is also passed through the VGG19 model, and the feature maps of the same layer are extracted. The content loss is then calculated as the mean squared error between the feature maps of the content image and the stylized image. This measures how different the content of the stylized image is from the content image. The content loss is used in the total loss function along with the style loss to optimize the stylized image. The goal is to minimize the content loss while preserving the style of the style image.

D. Style loss

In neural style transfer (NST), style loss is used to measure the difference between the style of the style image and the stylized target image. Style loss is calculated by comparing the feature maps of the style image and the target image in one or more layers of the pre-trained convolutional neural network (CNN). The style loss is calculated by computing the mean squared error between the Gram matrices of the style and target images. The Gram matrix is a matrix of dot products between the feature maps of a given layer. The dot product is computed between the vectorized feature maps of each pair of positions in the feature map, resulting in a matrix where each element is the dot product of two feature maps. The Gram matrix measures the correlation between the feature maps of a given layer, which can be seen as a measure of the style of the image. The style loss is computed by taking the difference between the Gram matrix of the style image and the Gram matrix of the target image and then squaring and summing the differences. The style loss is then weighted by a factor to balance it with the content loss in the total loss function. The choice of the layer or layers used for style loss calculation and the weight of the style loss are hyperparameters that can be adjusted to control the degree of stylization and level of detail in the final output. The style loss is calculated using the Gram matrix of the feature maps in three selected layers: 'conv1_1', 'conv2_1', and 'conv3_1'. The weight of each layer is defined in the `style_weights` dictionary. The overall style loss is the weighted sum of the individual style losses in each layer, which is multiplied by the `style_weight` hyperparameter. The final stylized target image is obtained by minimizing the total loss function, which is the sum of the content loss and style loss, using gradient descent.

E. Total Loss

The total loss in Neural Style Transfer (NST) is the combination of the content loss and style loss. The total loss is computed as a weighted sum of the content loss and style loss, where the weights are determined by hyperparameters alpha and beta, respectively. The equation for total loss is:

*total_loss = alpha * content_loss + beta * style_loss*

where alpha and beta are hyperparameters that control the relative importance of the content and style in the final stylized image. In the code above, the total loss is computed in the `run_style_transfer` function. The content loss and style loss are computed as described earlier, and the total loss is then computed using the alpha and beta hyperparameters. The optimizer is then used to minimize this total loss, which results in a final stylized image that combines the content of the content image with the style of the style image.

F. Optimization

In the context of Neural Style Transfer (NST), optimization refers to the process of finding the image that minimizes the total loss function which is the sum of content loss and style loss. The optimization algorithm used in NST is called gradient descent, which is an iterative method that updates the pixel values of the image in small steps to minimize the loss function. The specific variant of gradient descent used in NST is called L-BFGS (Limited-memory Broyden-Fletcher-Goldfarb-Shanno) algorithm, which is a quasi-Newton optimization method that uses an approximation of the Hessian matrix to estimate the direction of steepest descent. L-BFGS is preferred over other optimization algorithms because it has been shown to be efficient for high-dimensional optimization problems and can handle non-convex functions with many local minima. During each iteration of the optimization algorithm, the gradients of the total loss function with respect to the pixel values of the generated image are computed using backpropagation. Then, the pixel values are updated in the direction of the negative gradient, scaled by a learning rate, to minimize the loss function. The learning rate controls the step size of the optimization and is usually set to a small value to ensure that the updates are small enough to avoid overshooting the optimal image. The optimization process in NST can take several minutes or even hours, depending on the complexity of the style and content images and the computational resources available. It is important to note that the quality of the final stylized image heavily depends on the

choice of hyperparameters such as the number of iterations, the learning rate, and the weights assigned to the content and style losses. Proper tuning of these hyperparameters can lead to better results and faster convergence of the optimization algorithm.

G. Post Processing

In the post-processing step of Neural Style Transfer (NST), the stylized image obtained from optimization is converted back into its original format and displayed. This step is crucial in order to visualize the final result and evaluate the effectiveness of the stylization process. The main task in post-processing is to convert the optimized image from a PyTorch tensor to a numpy array, and then rescale the pixel values back to the original range of [0, 255]. This can be done using the `detach()` method to remove the tensor from the computational graph, followed by the `numpy()` method to convert it to a numpyarray.Next, the image is further processed by denormalizing it with the mean and standard deviation values used during the pre-processing step. This is important because the normalization step in pre-processing may have altered the color distribution of the image, so denormalizing it will restore the original color distribution. Finally, the pixel values of the image are clipped to the range [0, 255] to ensure that they are within the valid range of an 8-bit image. The resulting numpy array is then converted back into an image and displayed using the `matplotlib` library. NST is a relatively simple step, but it is important in order to visualize the final result of the stylization process.

II. Experimental result and analysis

A. Model description

Neural style transfer uses the features found in the 19-layer VGG Network, which is comprised of a series of convolutional and pooling layers, and a few fully-connected layers. In the image below, the convolutional layers are named by stack and their order in the stack. Conv_1_1 is the first convolutional layer that an image is passed through, in the first stack. Conv_2_1 is the first convolutional layer in the *second* stack. The deepest convolutional layer in the network is conv_5_4.

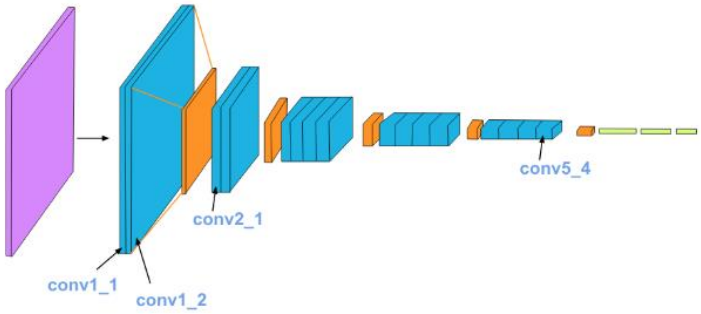

Fig 2 Convolutional layers

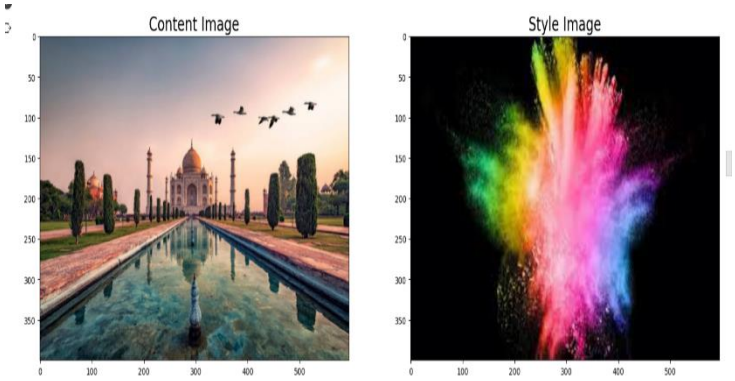

Fig 3 Images of Content and Style

B. Loading vgg19 model

To get the content and style representations of an image, we have to pass an image forward throug the VGG19 network until we get to the desired layer(s) and then get the output from that layer.

Fig 4 Calculating content loss and style loss

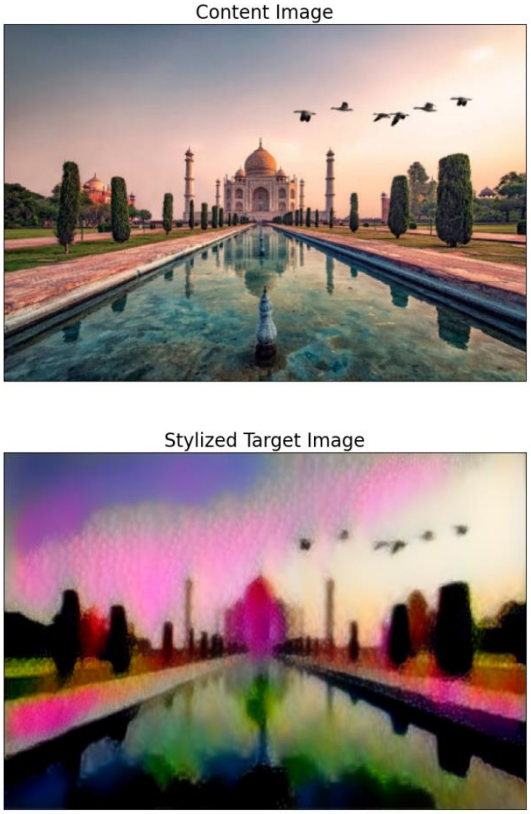

Fig 5 Content image and stylized image

IV. CONCLUSION AND FUTURE ENHANCEMENT

Neural Style Transfer is a technique to convert image into painted or artistic format. It aims to convert a image to artistic image by combining the style with the content of the image. Neural Style transfer is the way to create painted images. the enhanced VGG-19 model described in this research helps us to successfully create artistic image.

CNNs play a crucial role in neural style transfer by enabling the algorithm to extract and manipulate image features to create visually appealing stylized images. The proposed system provides good efficient method for creating painted/artistic image in a safe and powerful manner.There is a wide scope for future development for this technique. The world of computer fields is not static it is always subject to change. The technology which is famous today will become outdated very next day.It is essential to change the technique when a new technique arrives with more advanced features. So it is necessary for future development. Future enhancements can be done in an efficient manner.In this paper, there is a wide scope for the future development of the technique. The study of the paper has only concentrated on transferring the style of one specific image or set of images. Future research could explore ways to transfer styles from more diverse sources, such as video or 3D models

III. ACKNOWLEDGMENT

This work is supported by Dr.F.Paulin during 2022-2023 by Centre for machine learning and intelligence, Avinashilingam Institute for Home Science and Higher Education for Women in Coimbatore, Tamil Nadu, India.

REFERENCES

1.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).
2.Johnson, J., Alahi, A., &Fei-Fei, L. (2016). Perceptual Losses for Real-Time Style Transfer and Super-Resolution.In European Conference on Computer Vision (pp. 694-711).Springer, Cham.
3.Huang, X., Belongie, S., & Adam, W. (2017). Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1501-1510).
4.Ulyanov, D., Vedaldi, A., &Lempitsky, V. S. (2016). Instance Normalization: The Missing Ingredient for Fast Stylization. arXiv preprint arXiv:1607.08022.
5.Luan, F., Paris, S., Shechtman, E., &Bala, K. (2017). Deep Photo Style Transfer.In Proceedings of the

IEEE Conference on Computer Vision and Pattern Recognition (pp. 4990-4998).
6. Chen, D., Yuan, L., Liao, J., Yu, N., &Hua, G. (2018).StyleBank: An Explicit Representation for Neural Image Style Transfer. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 498-514).
7. Li, C., Wand, M., &Fei-Fei, L. (2016). Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis.In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2479-2486).
8. Dumoulin, V., Shlens, J., &Kudlur, M. (2016). A Learned Representation for Artistic Style.In International Conference on Learning Representations (ICLR).
9. Liao, J., Yao, Y., Yuan, L., &Hua, G. (2017). Visual Attribute Transfer through Deep Image Analogy.ACM Transactions on Graphics (TOG), 36(4), 120.
10. Chen, T., Li, M., Li, Y., & Zhu, M. (2017). Style Net: Generating Attractive Visual Captions with Styles. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3137-3146).