# DA526 : Image Processing with Machine Learning Project report


| | |
|---|---|
| Ashish kumar Pal | 224101004 |
| Kaja Gnana Prakash | 224101027 |
| Karthik Maddala | 224101029 |
| Tejas Chandra Karredula | 224101052 |
| Vudata Venkata Narendra | 224101060 |


**Submitted to**

Dr. Debanga Raj Neog
Assistant Professor
Mehta Family School of Data Science and Artificial Intelligence


INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI
GUWAHATI-781039, ASSAM

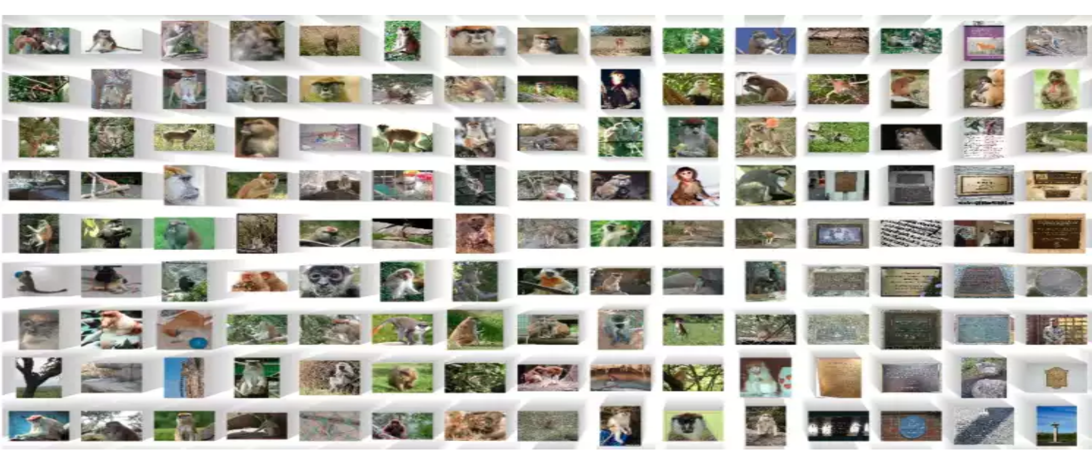
# Table of Contents

## Problem Statement

Imagine that you can create a true artwork by yourself, turning your own photo or a familiar landscape into a painting done like Picasso would do it. Seams impossible right? With the help of deep neural networks, this dream has now become reality. Neural style transfer has become a trending topic both in academic literature and industrial applications.From the deep learning perspective, ideas for style transfer stem from attempts to interpret the features that a deep neural network learns and understand how exactly it works. Neural style transfer is a technique that involves combining the content of one image with the style of another image using deep neural networks. This process involves training a deep learning model to extract the content and style features from the input images and then synthesizing a new image that preserves the content of the first image while applying the style of the second image. This technique has become increasingly popular due to its ability to create visually stunning and unique artworks. The process involves using convolutional neural networks (CNNs) to extract high-level features from both the content and style images. These features are then used to construct a new image that combines the content features with the style features. The neural style transfer technique has been used in a variety of applications, including artistic style transfer, video processing, and even fashion design. It has also been applied in the field of medical imaging to help identify and diagnose various diseases, such as cancer. Overall, neural style transfer is an exciting area of research that has the potential to transform the way we think about image processing and computer vision. With further advancements in deep learning techniques, we can expect to see more sophisticated and effective methods for style transfer that can be applied to a wider range of applications. The main problem statement is to get a better aesthetic look by our improvements in the model.

## Dataset

The ImageNet dataset contains 14,197,122 annotated images according to the WordNet hierarchy. Since 2010 the dataset has been used in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), a benchmark in image classification and object detection.



## Related Work

Jing Y, et al [1] outlines the key components involved in the process of neural style transfer, including the style representation, content representation, and the loss function used to optimize the generated image. It discusses various neural network architectures employed for style transfer, such as the VGG network, and provides insights into the feature extraction process.

Cheng J, et al [2] proposed the Style-Aware Normalized Loss (SANLoss) for improving arbitrary style transfer. By incorporating style information and utilizing Gram matrices, SANLoss enhances the accuracy of style transfer. The multi-level style-aware normalization module further improves the fidelity of style transfer at different levels. The experimental results demonstrate the effectiveness of the proposed approach, offering promising potential for advancing the field of arbitrary style transfer.

GATYS L A, et al [3] presents a neural algorithm for artistic style transfer using pre-trained CNNs. By defining content and style losses and utilizing Gram matrices, the algorithm can generate images that combine the content of one image with the style of another. The experimental results illustrate the effectiveness of the method and its potential for creating visually pleasing artistic images.

Qu Shicao, et al [4] presents a research study on image style transfer methods based on deep learning. It explores different deep learning architectures, loss functions, and optimization techniques used in style transfer. The author proposes novel methods to improve style transfer quality and efficiency and evaluates their effectiveness through experiments. The findings contribute to the advancement of image style transfer techniques using deep learning approaches.

Leon A.Gatys,et al [5] presents a neural algorithm for artistic style transfer using pre-trained CNNs. By defining content and style losses and utilizing Gram matrices, the algorithm can generate images that combine the content of one image with the artistic style of another. The experimental results highlight the effectiveness of the method and its potential for creating visually pleasing images with artistic styles.

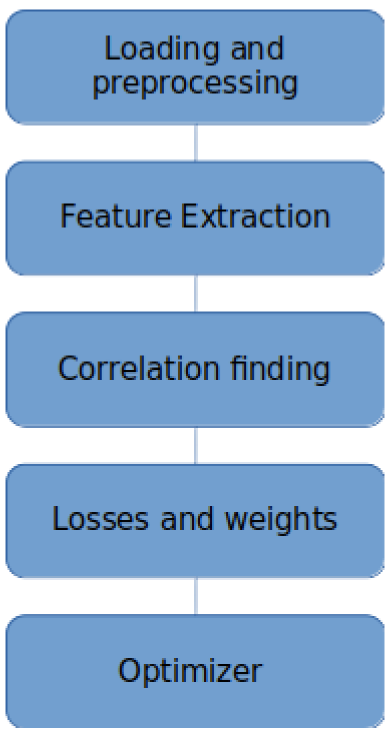
Yanghao Li, et al [6] provides a detailed explanation of the neural style transfer technique, which combines the content of one image with the style of another to generate artistic images. The authors discuss the role of convolutional neural networks (CNNs) in extracting features from the images and the use of loss functions to measure content and style similarity. They explain the optimization process and various techniques to enhance style transfer. The paper also covers implementation details and practical considerations. Overall, it aims to demystify the neural style transfer process for researchers and practitioners.

Fujun Luan et al [7] presents a deep learning-based approach for photo style transfer that addresses the limitations of existing methods. By leveraging intrinsic image decomposition and a neural network architecture, the proposed method achieves high-quality style transfer results by preserving local geometric structures and improving color consistency. The experiments validate the effectiveness of the approach and highlight its advantages over previous techniques.

Prabhumoye et al [8] Style Transfer through Back-Translation" presents a novel approach for text style transfer that utilizes a back-translation framework. By leveraging an autoencoder-based model for style-specific sentence generation and employing back-translation, the proposed method enables style transfer without the need for parallel datasets. The experimental results confirm the effectiveness of the approach in achieving high-quality style transfer in various tasks.

## Methodology

The proposed work has various steps involved; each step has its own importance in the context of style transfer.

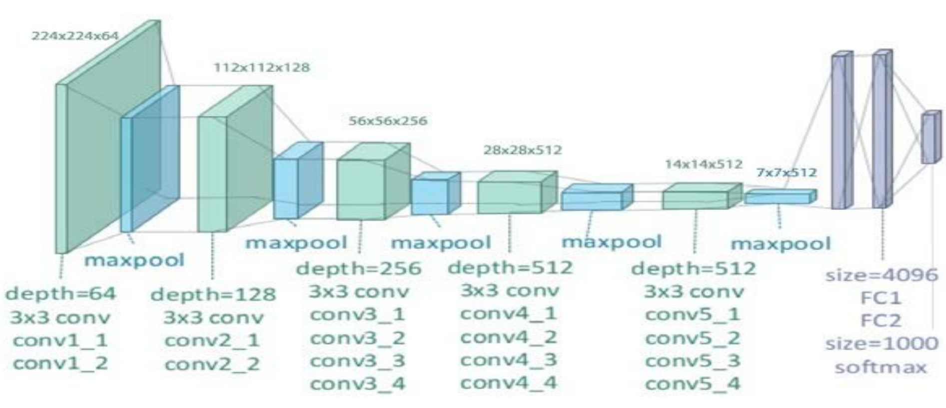


**1) Loading and preprocessing**

Inorder to generalize, we can take input of image in any format, we will convert it in to RGB format. Once the image is converted to RGB format, in order to have a tradeoff between the number of computations and quality of the image we are resizing any given image to 256x256. If an image with size less than 256 is given we will work with the original size of image. If an image with size more than 256 we will resize its image to 256. Once done we will convert the image to tensors to make it ease to apply many image processing techniques. And then we will continue with the normalization process with (0.485, 0.456, 0.406) as mean values of the RGB channels (red, green, and blue) and (0.229, 0.224, 0.225) as the standard deviation of the RGB channels. Once all the transformation on image tensors, in order to display an image we un-normalize the image tensor using numpy. To match the dimension with numpy we transpose the second dimension of the tensor to the first position, the third dimension to the second position, and the first dimension to the last position then we un-normalise the image and clip the values between o and 1.

**2)Feature extraction**

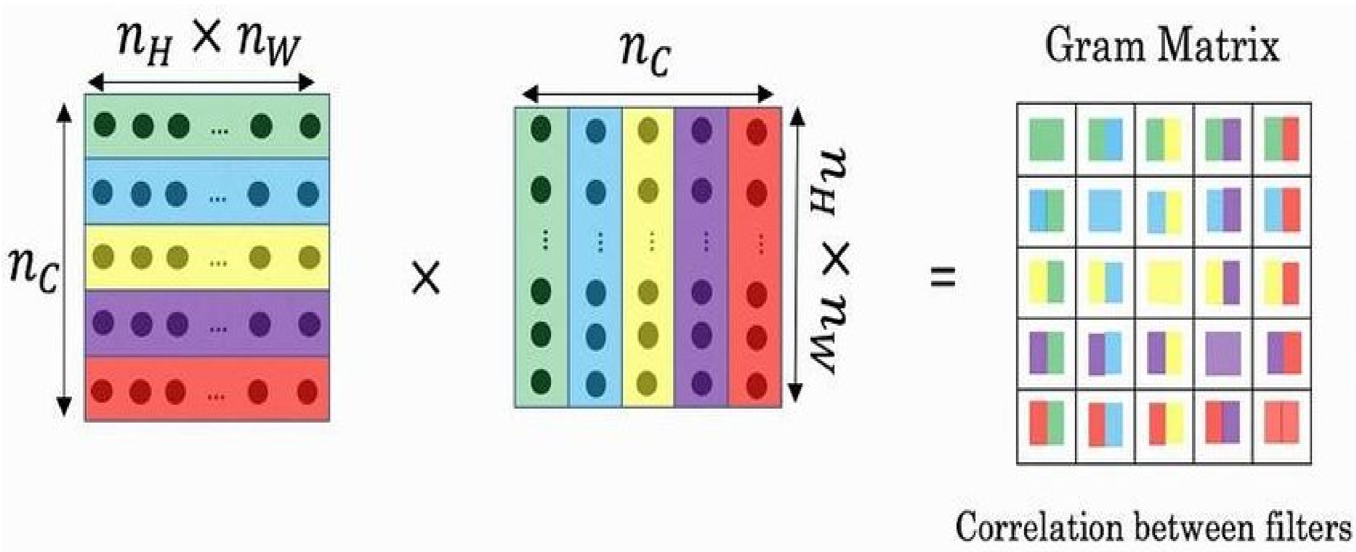
We have used VGG19 to extract features from both the content and style image, the more accurate the feature prediction makes our model more accurate. VGG19 network consists of 19 layers, including 16 convolutional layers and 3 fully-connected layers. The convolutional layers are organized into five stacks, with increasing depth as we go deeper into the network. The layers within each stack are numbered, with the first number indicating the stack and the second number indicating the order of the layer within that stack. The convolutional layers in the VGG Network use small filters (3x3) with a stride of 1 pixel, and are followed by a max pooling layer with a stride of 2 pixels. "conv1_1", "conv2_1", "conv3_1", " conv4_1", "conv5_1" convolutional layers are designated as style feature extraction layers. This design allows the network to learn hierarchical features of increasing complexity, starting with low-level features such as edges and corners, and progressing to higher-level features such as object parts and textures. The VGG Network has been trained on large datasets such as ImageNet, which contains millions of labeled images from thousands of categories. This pre-training model makes our model more accurate in finding features.




**VGG NET**

**3)Calculate the gram matrix**

Gram matrix can be thought of as a measure of the correlation between the features represented by each feature map. When two feature maps have a high correlation, they will have a large value in the corresponding entry of the Gram matrix. In our model to measure the similarity between feature maps in a convolutional layer gram matrix is being used.Gram matrix is calculated for each feature map in a convolutional layer, which is a tensor and reshaped it so that the spatial dimensions are flattened into a single dimension[batch_size, depth, height * width. The difference in Gram matrices between the input image and a style image, neural style transfer can generate a new image that preserves the content of the input image while adopting the style of the style image.




**Gram matrix**

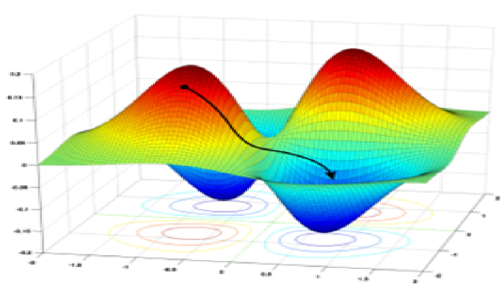
**4)Defining losses and weights**

In style transfer, the loss is calculated by comparing the feature representations of the content and style images with that of the generated image. Content loss is used to preserve the content of the input image in the output image, while style loss is used to match the style of the input image with the output image.Content loss is calculated by comparing the feature maps of the generated image and the content image, obtained from a specific layer of the neural network. The feature maps in the deeper layers of the network represent high-level features, such as the shape and content of the image. Therefore, we typically choose a deeper layer, such as 'conv4_2', to calculate the content loss. The weightage for content loss is typically set to a small value, such as (1)alpha, to ensure that the generated image retains the content of the input image. Style loss is calculated by comparing the Gram matrices of the feature maps obtained from multiple layers of the neural network. Gram matrix is a way of representing the style information in an image, which captures the correlation between different feature maps in the network. The style loss is typically calculated from multiple layers, such as 'conv1_1', 'conv2_1', 'conv3_1', 'conv4_1', and 'conv5_1', with different weightages as 1, 0.75, 0.2, 0.2, 0.2 respectively.

The weightages decrease as we go deeper into the network, as the deeper layers capture more abstract features of the image, which may not be necessary for matching the style. The weightage for style loss is typically set to a large value, such as (1e3)beta, to ensure that the generated image matches the style of the input image.

**5)Optimizer to reduce loss**

The loss function in this model is a weighted combination of style loss and content loss, where the weights are determined by hyperparameters $\alpha$ and $\beta$. Style loss measures the similarity between the style of the style image and the generated image, using the gram matrix of feature maps. Content loss measures the similarity between the content of the content image and the generated image, using the output of a specific layer in a pre-trained
CNN. The loss is calculated as, Loss-total = $\alpha \times$ Loss-content + $\beta \times$ Loss-style.The optimizer used in this model is Adam optimizer, which is a popular optimization algorithm in deep learning. Adam optimizer combines the advantages of Adagrad and RMSprop optimizers, and can effectively deal with noisy gradients and achieve faster convergence. The learning rate used in this model is 0.003, which controls the step size of the optimization process.The model is trained for 500 iterations, with the goal of minimizing the total loss function. In each iteration, the total loss is calculated based on the current generated image, and the optimizer is used to update the generated image by backpropagating the gradients of the loss function. This process continues until the total loss is sufficiently minimized and the generated image converges to the desired combination of style and content.



**Adam Optimizer**

## Experiments and results

The style image is imposed on the content image to get final style transferred image

**Result 1:**

**Result 2:**

**Result 3:**

**Result 4**

**Result 5**

**Result 6**

## Conclusion

In some deep learning applications, it can be challenging to evaluate model performance based on a single metric such as accuracy. Image style transfer is one of them, the goal is to create an image that combines the content of a source image with the style of a reference image. However, the optimal balance between content and style may vary depending on the specific images being used. In some cases, using more style information can lead to a better result, while in other cases using less style information may be preferable.Furthermore, each image has its own unique aesthetic qualities, and what looks good for one image may not necessarily look good for another. Therefore, it can be difficult to compare the quality of output images directly using a single metric. So we are training our model using an iterative optimization process over multiple images and styles can be a useful approach. By minimizing the total loss over many iterations, the model can learn to capture a range of styles and content combinations, and can produce images that are aesthetically pleasing for a variety of inputs. This can lead to a more robust and versatile model that can handle a wider range of input images and styles.Style transfer is an active area of research and development in computer vision and deep learning, and there are several novel approaches and techniques that could be introduced to improve its performance and functionality. Some of them are Multi-style transfer, Interactive style transfer, Adaptive style transfer, Fine-grained style transfer and Real-time style transfer.

## References


Jing, Y., Yang, Y., Feng, Z., Ye, J., Yu, Y., & Song, M. (2019). Neural style transfer: A review. IEEE transactions on visualization and computer graphics, 26(11), 3365-3385.

Cheng, J., Jaiswal, A., Wu, Y., Natarajan, P., & Natarajan, P. (2021). Style-aware normalized loss for improving arbitrary style transfer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 134-143).

GATYS L A, ECKER A S, BETHGE M.A neural algorithm of artistic style[ J]. ArXiv preprint arXiv, 2015(8):13-18.

Qu Shicao. Research on Image Style Transfer Method Based on Deep Learning [D]. Shanghai Normal University, 2021. DOI: 10.27312/d.cnki.gshsu.2021.001768.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge.“A Neural Algorithm of Artistic Style.”Preprint, submitted September 2, 2105.

Li, Y., Wang, N., Liu, J., & Hou, X. (2017). Demystifying neural style transfer. arXiv preprint arXiv:1701.01036.

Fujun Luan, Sylvain Paris, Eli Shechtman, Kavita Bala; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4990-4998

Prabhumoye, S., Tsvetkov, Y., Salakhutdinov, R., & Black, A. W. (2018). Style transfer through back-translation. arXiv preprint arXiv:1804.09000.