ARTISTIC STYLETRANSFER USING CONVOLUTIONAL NEURAL NETWORKS

A Project Report Submitted as a Part of Major Project - 2

Master of Technology

in

Information Systems

by

Sagar

2K17/ISY/13

Under the Supervision of

Dr. Dinesh K. Vishwakarma
(Associate Professor – Department of Information Technology)

Department of Information Technology
Delhi Technological University
(Formerly Delhi College of Engineering)
Shahbad Daulatpur, Bawana Road, Delhi – 110042

July-2019

DECLARATION

We hereby declare that the Major Project-2 work entitled "**ARTISTIC STYLE TRANSFER USING CONVOLUTIONAL NEURAL NETWORKS"** which is being submitted to Delhi Technological University, in partial fulfillment of requirements for the award of degree of Master of Technology (Information System) is a bonafide report of Major Project-2 carried out by me. The material contained in the report has not been submitted to any university or institution for the award of any degree.

SAGAR

2K17/ISY/13

CERTIFICATE

This is to certify that Major Project report-2 entitled "ARTISTIC STYLE TRANSFER USING CONVOLUTIONAL NEURAL NETWORKS" submitted by **SAGAR (Roll No. 2K17/ISY/13)** for partial fulfillment of the requirement for the award of degree Master of Technology (Information Systems) is a record of the candidate work carried out by him under my supervision.

Dr. Dinesh K. Vishwakarma

Project Guide
Associate Professor, Department of Information Technology
Delhi Technological University

ACKNOWLEDGEMENT

First of all, I would like to express my deep sense of respect and gratitude to my project supervisor Dr. Dinesh K. Vishwakarma for providing me the opportunity of carrying out this project and being the guiding force behind this work. I am deeply indebted to him for the support, advice and encouragement he provided without which the project could not have been a success.

Secondly, I am grateful to Dr. Kapil Sharma, HOD, Department of Information Technology, DTU for his immense support. I would also like to acknowledge Delhi Technological University library and staff for providing the right academic resources and environment for this work to be carried out.

Last but not the least I would like to express sincere gratitude to my parents and friends for constantly encouraging me during the completion of work.

SAGAR

2K17/ISY/13

ABSTRACT

One of the exciting research field has implemented known as Neural Style Transfer, which is a technique to transform images in an artistic way. Two images are taken as input image namely style image and content image to transform another base image with the help of optimization technique. This NST can be done with the help of Convolutional Neural Networks model as many researchers have tried to achieve good results using CNN network architecture. One of the famous and efficient pre-trained architecture is VGG16 and Gatys et al. [2] were able to generated good results based upon the VGG model. [2] Many famous Mobile and Web applications like DeepArt, Prisma and Pikazoapp have used these models to transformed images in an artistic way. [6] [27]

We primarily have discussed different Neural Style Transfer techniques then we have classified the artistic style transfer. We have implementation the model in Keras with the pre-trained CNN model that is VGG19 where we have adjusted the hyperparameters and transformation coefficients. VGG19 model has been trained on ImageNet dataset and we used it for feature extraction where for testing we have used two datasets namely Caltech101 and Caltech256. The fundamentals of NST is also discussed in depth literature survey which can be found in chapter 2.

Keywords: Neural Style Transfer, ConvNet, Deep Learning, Artistic Style Transfer, Convolutional Neural Networks.

TABLE OF CONTENTS

LIST OF FIGURES..X

LIST OF TABLE……………………………………………………………….…II

LIST OF ABBREVIATIONS…………………………………………………XIII

LIST OF EQUATIONS………………………………………………………..VIII

CHAPTER 1: INTRODUCTION……………………………………………..01-10

1.1 ARTISTIC STYLE TRANSFER 01-04

1.2 1.2 TRANSFER LEARNING 04-05

1.3 TRANSFER LEARNING 05-07

1.4 TRAINING A NEURAL NETWORK..07-10

1.2.2 PARAMETER INITIALIZATION 08-09

1.2.2 FORWARD PASS 9

1.2.3 BACKWARD PASS 9

1.5 MOTIVATION AND PROBLEM FORMULATION............................. 10

CHAPTER -2: LITERATURE REVIEW……………………………………11-17

2.1 OVERVIEW 11-12

2.2 LOSS FUNCTIONS ………………………………………………12-15

2.2.1 CONTENT LOSS………………………………………..13-14

2.2.2 STYLE LOSS ………………………………………......14-15

2.2.3 TOTAL LOSS ……...……………………………………….15

2.3 CLASSIFICATION OF ARTISTIC STYLE TRANSFER ………..15-16

2.3.1 CLASSIFICATION OF NEURAL STYLE TRANSFER 16

2.4 NEURAL STYLE TRANSFER CHALLENGES ..………………16-19

CHAPTER 3: THE PROPOSED WORK……………………………………18-23

3.1 OBJECTIVE………………………………………………………….....18
3.2 TRAINING NEURAL NETWORKS IN NST……………………......18-20
3.2.1 NETWORK ARCHITECHTURE …………………………....19-20
3.3 NEURAL STYLE TRANSFER ALGORITHM EVALUATION ……20-21
3.4 VGG MODEL ……………………………………………………..21-23

CHAPTER 4: EXPERIMENTAL WORK AND RESULTS………...………...24-33

4.1 OVERVIEW ………………..……………………………………24
4.2 SETUP AND DATASET …………...…………………………...24-30
4.3 PERFORMANCR EVALUATION ……………………………..30-33

CHAPTER -5: CONCLUSION AND FUTURE WORK…………………….34-35

5.1 CONCLUSION 34
5.2 FUTURE WORK 34-35

REFERENCES……..……………………………………………………………35-39

LIST OF FIGURES

S No	Figure Name	Page no.
1	Illustration of Neural Style Transfer by Gatys.	03
2	Neural Networks having 1 hidden unit with 2 neurons and a biase.	08
3	Art works by some of the famous artists.	12
4	Neural Style Transfer classification.	17
5	Receptive field illustration shows the deeper layer is able to see broad view of the image.	20
6	Block diagram of layers in network architecture.	21
7	Convolutional Neural Network architecture VGG19.	23
8	Images from different classes of Caltech101 dataset.	25
9	Images from different classes of Caltech256 dataset.	26
10	Images taken from different classed of Caltech101 and Caltech256 datasets.	26
11	Style Transfer performed on image “Delhi Technological University” with 5 different artistic paintings.	27
12	Style Transfer performed on image “Caltech101 and Caltech256 datasets” with 5 different artistic paintings.	28
13	Loss value graphical representation in terms of DTU images.	29
14	Loss value graphical representation in terms of Caltech101 dataset.	29
15	Loss value graphical representation in terms of Caltech256.	30

LIST OF EQUATIONS

S No	Equation Name	Page No.
1	Content loss	13
2	Derivation of the loss function	14
3	Gram matrix	14
4	Style representation	14
5	Style loss	15
6	Total loss	16
7	Feed forward network	25
8	Total loss	27

LIST OF TABLES

S No	Table Name	Page no.
1	VGG19 Network architecture.	22
2	Dataset	25
3	Loss value comparison with DTU images	32
4	Loss value comparison with Caltech101	32
5	Loss value comparison with Caltech256	33

LIST OF ABBREVIATIONS

S No	Abbreviated Name	Full Name
1	NST	Neural Style Transfer
2	DL	Deep Learning
3	CNN	Convolutional Neural Networks
4	NN	Neural Networks
6	VGG	Visual Geometry Group
7	PSPM	Per Style Per Model
8	MSPM	Multi Style Per Model
9	ASPM	Arbitrary Style Per Model
10	MRF	Markov Random Field
11	ST	Style Transfer
12	DNN	Deep Neural Network
13	ReLU	Rectified Linear Unit

Chapter 1
Introduction

1.1 Artistic Style Transfer

Artistic style transfer is the problem of transferring the texture of an image onto another image while constraining the semantic content, it can be achieved by several techniques but the most used technique by researchers is Style Transfer via Convolutional Neural Networks called Neural Style Transfer (NST). Artistic Style Transfer is often referred as Texture Synthes, there are divers amount of algorithms to synthesize texture of images. [1][2][5] Non-Photorealistic Rendering (NPR) has been around since 1990's to 2015 which is a way to stylize an image using computer algorithms. Neural Style Transferred using Convolutional Neural Networks is first studied by Gatys *el al.* [2] in which they were able to generate perceptual high quality synthesized image by combining one sample image called "Content Image" with another painting like image called "Style Image". Neural Style Transfer can be consider a sub-field of Artistic Style Transfer, which is nothing but a technique of transferring style using convolutional neural network.

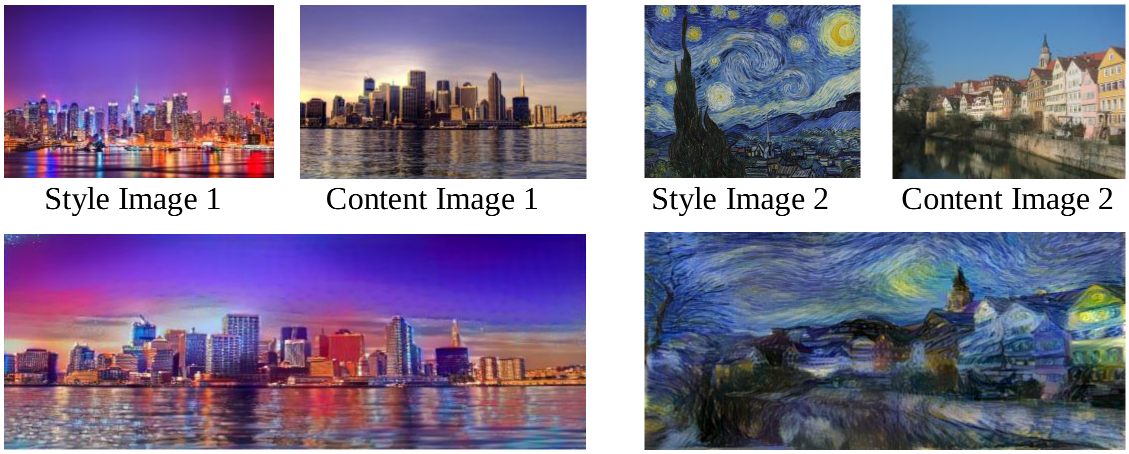
Gatys *et al.* [2] showed how to extract style and content from an image and recombine them to form an artistically pleasant image which can be further used for numerous purposes. Style is known as the texture of an image, in other words, how an image looks can be treated as it's style. Feature response is treated as content of an image from pre-trained model. Hence, authors were able to separate the style and content of images to recombine them to generate an image in the style of any painting.

Style transfer using Convolutional Neural network is a step-by-step process in which we first resize content image, style image and a randomly generated image to equal shapes. A pre-trained CNN like VGG19 is loaded, we can also separate the CNN layers independently. It can also be considered as an Optimization Problem because we calculate the loss functions and try to reduce it to an extent. We define Content loss as Squared error loss between feature representations and take derivation with represent to the activation on a particular layer. Style is often refer to the texture of an image which represent the brush strokes, patterns, angular geometric shapes and alteration between colors. [3] At last we calculate the total loss which is the linear combination of style loss and the content loss, we take the derivative of this loss with respect to the pixel value with the help of error back-propagation.

Finally, we take gradients to iteratively update the picture until the input style features and content features coming out to be similar to style image and content image. Gatys *et al* [2] showed how style of an image can be captured via Gram Metrix feature response which can be defined as correlation between feature maps from different layers of a network like VGG19 which is a pre-trained Convolutional Neural Network trained on a dataset called ImageNet used for object classification. The formulation and mathematical representation of various loss functions with gram matrix is discussed in chapter 2 in brief.

As stated above Artistic style transfer is an achievable task but majority of the algorithm used non-parametric approach for texture synthesis using different methods to preserver the semantics of the output image, these methods involves high-frequency texture manipulations, edge orientation information, coarse scale preservation, matching of image intensity and image analogy. For instance, Correspondence map then introduced to include objects such as image intensity for constraining the texture synthesis procedure by Efros and freeman. [5] Image analogy is used for transferring the style by Hertzman et al. [6] These non-parametric approaches worked well in pre-neural era but style transfer parametric approach seemed to work better but despite of the excellent performance it turned out to be a slow iterative optimization process.

Yongcheng Jing et al. [16] reviewed the broad overview on Neural style transfer in which they classified style transfer techniques according to their drawbacks and advantages which we will discuss in chapter 2. Before jumping into details one should have a clear understanding about Style Transfer and as we are limiting our research to Neural Style Transfer on images, we shall first define it. Neural Style transfer is a technique to synthesize multiple images to form an artistically pleasant image using Convolutional Neural Networks. In the fig. 1 two examples of neural style transfer are shown in which, example 1 contains Style Image 1: “New York by Night”, Content Image 1: “London by day” and Transformed image 1: London by day painted in the style of New York by night painting whereas example 2 contains Style Image 2: “The Starry Night Painting”, Content Image 2: “Tubingen image” and Transformed image 2: Tubingen image painted in the style of The Starry Night Painting, These illustration are on the based on Gatys et al. [2]



Example 1: Transformed Image 1　　Example 2: Transformed image 2

Fig. 1 Neural Style Transfer Illustration by Gatys et al. [2] First example is the combination of Style Image 1 (New York by night) and Content Image 1 (London by day) resulting a Transformed Image 1. Second example is the combination of Style Image 2 (Starry Night painting) and Content Image 2 (Tubingen image) resulting a Transformed Image 2.

Neural Style Transfer is gaining a lot of attention in not only research field but also in the commercial field, we all have used the famous apps like Prisma, DeepDream, Deepart, Pikazoapp and so on these applications provides us a nice and easy interface with which one can apply painting filters on their pictures. [6][27] The idea behind this magical transformation of images is Convolutional Neural Networks. Applications like Prisma has 250 plus modern art filters and they have more than 110 Million registered users now imagine the amount of images that are being transformed and computational cost required to synthesize. [6] Hence we can conclude that Convolutional Neural Networks are the powerful and efficient Networks in Neural Style Transfer, not only that we all have seen the power of CNN's in fields like Image Classification, Image Recognition, Robotic Vision, Self-Driving Cars and so on.

Manuel Ruder et al. [7] used CNN VGG-19 Network for Video Style Transfer inspired by the advancement in still images Style Transferred proposed by [2]. Authors were able to achieve the satisfactory results even in the fast motion videos or in strong occlusion, they produced visually appealing stylized videos using Neural Networks. Now that we know the power of Neural Networks does not only limit to still images but also it can exhibit great results in Videos as well, we can say that there is a lot to uncover in the field of Neural Style Transfer. [8][15][16]

1.2 Activation Function

1980's were the initial years when researchers first started to study Neural Networks, any single hidden layer having connections with some other neurons can be called universal approximator according to Kurt Hornik. Further the theory stretched stated that to approximate any continuous function having sufficient amount of hidden neurons, multilayer perceptron may be used. [7] As we all know, Neural Networks work as our brain does so before jumping into details of activation function let us look at some of the basics of neurons. Millions of neurons are presented in human brain connecting each other they form a hierarchy. Neurons are connected to each other by Axons which passes signals from one entity to other. Hearing, feeling, watching or even thinking are treated as signals and passed by one neuron to another neuron in the form of electric signal also called Synapse, this is the process of information storing and learning from experience.

Referring human brain, Researchers were able to claim that the real cause of universal approximation capability was the structure of the neural network but still they failed to mark the implementation side effects of overlook the choice of activation function. This theory of Hornik encouraged researchers to dig deep and discover more sophisticated activations to reduce the complexity of the network. Now that we know the origin on activation function, let us understand what an activation function is and how many types of activation functions are there.

The output of a Neural network depends upon the type of activation function we are using not only that it is also responsible of model's accuracy, computational efficiency while training our model. The Converge and the Convergence speed are also controlled by activation functions, even in some cases it might prevent from converging the network on the initial stage. [9] The primary task of an activation function is to convert an input signal to output signal of a node in an Artificial Neural Network where this output signal then used as a feedback to the next layer input. Neural Networks works on non-linear activation functions that assists the network to learn more complex date, also helps in solving and leaning complex mathematical representations and provides correct predictions. Majorly we take sum of the products of all the inputs to a model with the corresponding weights and eventually applying the activation function in order to get output of the working layer which then turns into input of the next layer.

In this section we will discuss some of the common non-linear activation functions which are widely used in Neural Networks.

1. *Sigmoid function or Logistic activation*: This function is majorly used due to its well defined range i.e. (0 to 1) hence using Sigmoid function in the case of probability prediction is a practice best practice.
2. *TANH*: This function is similar to Sigmoid activation function but the precision level is wide i.e. (-1 to 1) hence it is better to use in the situation of strong values like negative, positive and neutral.
3. *ReLU*: ReLU stands for Rectified Linear Unit activation function, it is the most used activation function due to its wide range in non-negative axis i.e. (0 to infinite) and its quick convergence property.
4. *Leaky ReLU*: The gradient of the function turns zero in the case of negative or zero inputs this is called The Dying ReLU problem and Leaky ReLU is came into picture to overcome this disadvantage of ReLU activation. The range of Leaky ReLU is between negative infinity to positive infinity.
5. *Softmax activation function*: This function exhibit the property of cumulative distribution function and widely used for binary classification. The range of Softmax function is (0 to 1)

1.3 Transfer Learning

Conventional algorithms related to Deep learning have been used through isolation learning in order to tackle particular problems but due to some drawbacks, like model rebuilding in case of change in feature space distribution, Transfer learning had to inaugurate.

Feature maps are the outputs of image convolution with a filter which are significant and transferable. Previously when transfer learning was not in practice, traditional learning was used to train isolated models while it works isolatedly on particular jobs and specific datasets. There were no knowledge stored to transfer from one network to forward network.

Sharif Razavian et al. [8] trained SVM classifier for multiple recognition exercise like attribute detection, scene recognition, image classification and so on. Yosinski et al. [9] specified that low level features depict better performance even in case of distant tasks and deep learning networks seems to learn low level feature maps. We can say, one can teach to see a Convolutional Neural Network by training large datasets. Now that we have some intuition about learning procedurals it is right time to define Transfer learning with a clear understanding.

Transfer learning aims to solve the similar tasks in the given source domain that is then used to enhance the performance of a model and speed-up the training. In other words it may reuse the weights from pre-trained model for multiple layers in a complete new model. [10][12] Even in the case of few amount of data available, Transfer learning can take advantages from knowledge like features, weights and so on with the help of previously trained network.

The advantages of Transfer learning is listed down below:

- Model that is trained can be reused in case of predictive modeling.
- Transfer learning helps in improvement of overall performance of the model in many cases like multiclass classification.
- Through feature extraction or weight initializing strategy, Transfer learning can accelerate the training process.

According to Yosinski et al. [8] a network first trained on a dataset under the base conduction then the learned features gets remodeled or passed to the another target network for training purpose onto the dataset. [9] Transfer learning has been used in Neural Style Transfer as well, by the virtue of feature representation transfer sometimes also called feature representations. The classification network filters assists feature maps that are then utilized to form styling images.

Not only that simultaneously one can take advantage of feature representations for a fresh new job on comparatively large networks onto tasks like Supervise learning. Overfitting is one of the drawbacks of Transfer learning in case of small dataset and large amount of parameters but on the other hand, overfitting is not a problem in large dataset and small amount of parameters that in-fact helps in fine-tuning and improves the overall performance.

1.4 Training Neural Networks

Approximation tasks is the key when it comes to neural networks and it is determined by a metric which then decides how well a neural network is performing. Metrics contain the output values which is determined by functions like mean squared error. Cost function is minimized by changing the weights of neural networks selecting the right cost function or the loss function, also known as backpropagation. Examples are inputted to the neural network, then the total overall loss is computed over all the samples. Finally the overall loss is then derived using derivation w.r.t the weight and then weights are updated. Chain rules are applied while deriving w.r.t biases and weights required by the architecture of neural networks.

Image 3 shows a basic neural networks having one hidden layer containing two neurons (h1 and h2) with weights (w5, w6, w7 and w8) and two input features (x1 and x2) with weights (w1, w2, w3 and w4) while binary output values (y1 and y2). Our neural contains two Biases (b1 and b2) while each layer calculates the linear combination of given inputs pursuing non-linear activation function elementwise.

Now as we have some knowledge about backpropagation, we would see the steps to train neural networks.

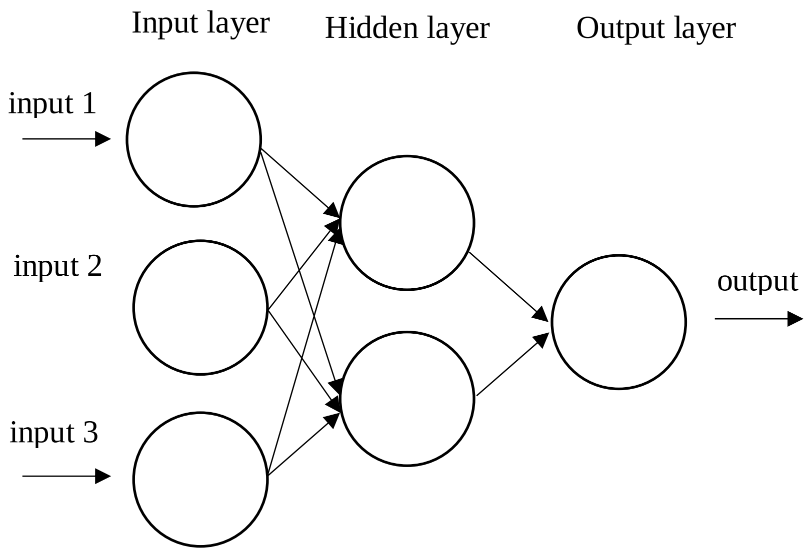


Fig. 2 NN having 1 hidden unit with 3 neurons.

1.4.1 Parameter initialization

Neural network building requires the parameter initialization at the very step in order to achieve accurate results which might turn into a tricky task. We take an examples to understand all the notations, let's take a neural network which is '*L*' layer network where hidden layers would be '*L-1*' with input and output layers attached to it. We may denote weight matrix of dimension '*l*' as '$w^{[l]}$' and bias vector of dimension 'l' as '*b[l]*'.

Neural style transfer is a technique in which initialization matters a lot in fact in general initialization is crucial. [13] In order to get excellent results we need to choose our weights wisely and hence the followings are the best practice initialization parameters.

- *Zero Initialization* : parameters initialized with Zero initialization in a way that weights are with initialized zero and bias with zero too. Weights are initialized with Zero due to derivation as in the further iterations the values for the weights comes out to be same. Generally Zero initializations does not provide satisfactory results.
- *Random Initialization* : parameters initialized in Zero initialization in a way that weights are with initialized randomly and bias with zero. Weights should not be initializes with very high values as it might take significantly long time. Weights should not be initialized with very low values too as it gets mapped to 0.

- *He Initialization* : the assignments are performed randomly with the following formulation and works better than previous initializations.
- *Xavier initialization* : assignments are similar to the He initialization but used for tanh() and also reduce slow convergence problem.

1.4.2 Forward Pass

Accuracy determines how good or bad a model is which is again determined by the forward pass in fact in Neural style transfer it is crucial in finding error of the respective model. [17] Forward pass and Backward pass works collectively to reduce the overall error significantly.

In the process of the forward pass first the values of each neuron is calculated by the linear combination of inputs. Second, we need to apply activation function over these values to attain the non-linearity. Now that the activations at the hidden layers are computed we continue that process for the another pass onto the resulting layers also called output layers. Activation functions are again applied to those output values and iterate the process. Finally the total error get computed by the formulation. The loss functions are given in chapter 2 while only one pass is required to train a feed-forward network in Neural Style Transfer.

1.4.3 Backward Pass

The main purpose behind the Backward Pass is to improve the error that we obtained with the help of Forward Pass by deriving w.r.t. the weights using chain rule. Content loss and the Style loss are added up to compute the total loss as mentioned before. Total loss is then sent to the backward pass to get improved. Linear combination derivatives of activation function are computed and combined all together. [5][18] In the hidden layers the error gets propagate via chain rule with the same technique and weights get adjusted accordingly by computing gradients. Backward pass long with forward pass referred as backpropagation was first used as an optimizer for the network in Landmark Paper 1986. In general researchers avoid only forward pass for the each parameters. Gradients are calculated by keeping, storing the differences and propagating error backwardly.

A technique to calculate gradients is known as gradient descent while taking care of the learning rate. The aim is to get the calculate the gradient of the whole network in order to update weights which then turn the total error down by significant amount.

1.5 Motivation and Problem Formulation

This work has been carried out on the basis of the problem statement which is, given an input image also called content image and another input image called a style image, we are to transfer style of the style image onto the third base input image which is called generated output image while preserving the content of the content image with further optimized loss values and enhancing the perceptual quality.

First study in the field of Neural Style Transfer (NST) has been performed by [2] significant results had been generated where. Neural Style Transfer is the process of first training the Convolutional Neural Network and then optimization of the loss in order to achieve optimum global. We are to calculate activations within the intermediate layer of Convolutional Neural Networks which then recorded into the Gram Matrix. Hence our main goal of the project is the synthesize images in order to minimize the loss to the significant level through a proper activation function with the adjustment in Hyperparameters.

Chapter 2
LITERATURE REVIEW

2.1 Overview

We as humans have always been more attached to the handy works such as crafts, sculptures or paintings resulting many individuals developing their own styled artworks. Further, these artworks then classified accordingly forming a way to express their feelings, self-awareness, social issues and so on. Since the beginning of mankind arts have been used for numerous purpose and played an important role in our culture, not only that art is one of the reasons that we still have some knowledge of prehistoric and ancient world. In the prehistoric era arts had been used primary to preliterate until cultures learned to write and keeping records in-fact these earliest human artifacts left the evidences for us which help uncovering the hidden facts and civilization. Arts have played significant role not only in prehistoric era but also in modern world, paintings turned out to be one of the most appreciated forms of art. Paintings were not just for entertainment purpose but also affected the world in a very revolutionary manner in-fact paintings grew in a periodic way from Medieval to Romanticism and Realism. [10][18] Throughout the development of humans these art-movements affected to world and became a channel to showcase individual's talent while conferring ones perspective.

Many creative artists came into picture throughout evolution of civilization such as Vincent Van Gogh, Pablo Picasso, Leonardo da Vinci, Frida Kahlo and so on where their paintings like "The Starry Night", "Mona Lisa", "The Scream", "Guernica" and so on, also gained a lot of attentions and created many master pieces as shown in the fig. 1. [2]

These artworks were primarily appreciated for their beauty and emotions but time taken to develop such style has always been huge. Every artist spend years to develop their own style and the amount of efforts they incorporate is remarkable to generate a quality painting hence this issue taken into account by many computer researchers.

Researchers and computer scientists gradually started putting their efforts to reduce the complexity faced and the time incorporated by painters during the development of a painting hence new techniques introduced that assisted in various forms.

Later, the concept of Computer art or Digital art introduced which can be defined as any

Fig. 3 Art works by some of the famous artists.

art generated with the help of any digital device. Terms like Computer art and Digital art used interchangeably back then but these terms are being used distinctively now a days. 1960's have been salient years for digital art history because many artists and researches showed interest in Computer art.

Neural style transfer is an optimization in which first we try to calculate the loss of content image as well as style image independently and then we try to optimize it by updating weight through backpropagation. [19] As we are aware of the fact that the set of image on which we aim to train our model is not going to be less instead we train our model on tens of thousands of images with multiple classes hence the amount of complexity is high. Our aim in neural style transfer is to recombine the input images in a way that their style gets blended but not the content in another words the generated image and the content images are same w.r.t. their content but not the style. [3][16] This can be achieved by separating the losses in terms of content and style which we will see in the next section.

2.2 Loss Functions

Loss functions are of three types mentioned down below:

- *Content loss*
- *Style loss*
- *Total loss*

As stated before, we are able to calculate the content loss and style loss or in other words we are able to separate content and style through the content loss and the style loss. Once the loss gets calculated, we then sum up these losses in order to update weights by finding the gradients. [2][3][19]

2.2.1 Content loss

Content loss calculation takes fundamental formulations but before actually calculating the loss function one should have a clear understanding of what content is in Neural Style Transfer. Content is the basic details and patterns of the image that is to be preserved in generated image, by the recent researches it has been noticed that higher layers of Convolutional Neural Networks is able to capture content of the input image while the lower layers of the network mostly captures the pixels hence the higher layers are used in order to formulated loss functions. [1][22] Two images have the same content can be traced by checking whether these images have the same activations on the higher layers or not.

Now that we have some intuitions about content loss now let's formulate the content loss function which is pretty fundamental to find. As we know we input content image, style image and the base output image. Taking the intermediate layers output, we calculate the Euclidian distance between these representation on the intermediate layer of the images or in other words, content loss can be defined as the Euclidian distance of the squared-error loss between the input image *'x'* and the generated image *'p'* (feature representations). Content

loss function can be defined as:

$$L_{content}\left(\vec{P},\vec{x},l\right) = \frac{1}{2}\Sigma_{ij}\left(F_{ij}^{l} - P_{ij}^{l}\right)^2 \qquad (1)$$

Where F_{ij}^{1} and P_{ij}^{1} are the feature activations and p and x are the content image and generated image. In order to minimize the content loss we take backpropagation into account. The gradients gets calculated to update weights and simultaneously generated image tends to achieve the similar response hence by doing this we achieve the generated image having the close feature activations as to the content image. [2] We then derive this loss function with respect to the activation layer *'l'* as:

$$\frac{\partial \mathcal{L}content}{JF_{ij}} = \begin{cases} (F^l - P^l)_{ij} & if \quad F_{ij} > 0 \\ 0 & if \quad F_{ij} < 0 \end{cases} \tag{2}$$

2.2.2 Style loss

Formulation of the style loss is more engaging while it follows the same procedure. The base input images are inputted to our model and accordingly the output image gets generated. The style loss is calculated with the help of Gram matrices in which feature response gets stored. Style can be extracted from image on top of every layer by calculating the correlation between the activations. The correlation then stored into the Gram matrix $G^l \in R^{N_l \times N_l}$ where G^l is referred as the inner product between features i and j on the layer l and every entry of the Gram matrix can be calculated via dot product of activations:

$$G_{ij}^l = \sum_{i=0}^{L} F_{ik}^l F_{jk}^l \tag{3}$$

Lower layers refers to the local structures but on the other hand the higher layers reflects the high styling perception. In simple words we can say that the entries in Gram matrix get to decide the activeness of the activations at the same time. Simultaneous activations or non-simultaneous activations are the leads, in order to create the stylize base image. Actually to decide whether two vectors are similar or not, dot products can be taken into account. If these dot products come out to be large, the results get large hence we may say it is a similarity index. Now that we have Gram matrix, we are to compute the loss between Gram matrix and the style image and total style loss has to be computed at layer l between the input style image and the generated image with the A^l and G^l style representations where the formulating is given below:

$$E_l = \frac{1}{4N_l^2 m_l} \sum_{ij}. \left(G_{ij}^l - G_{ij}^l\right)^2 \tag{4}$$

The above equation gives the total style loss as:

$$\mathcal{L}_{style}\left(\vec{a},\vec{x}\right) = \sum_{j=0}^{L}.w_l E_l$$

(5)

Where w^l and E^l are the weighting factors and standard error back propagation.

2.2.3 Total loss

The addition of the total content loss with total style loss gives the total loss and hence can be expressed as:

$$\mathcal{L}_{total}\left(\vec{P},\vec{a},\vec{x}\right) = \alpha\mathcal{L}_{content}\left(\vec{P},\vec{x}\right) + \beta\mathcal{L}_{style}\left(\vec{a},\vec{x}\right) \tag{6}$$

Where α and β are the hyperparameters which are used for weighing content and style weight. These parameters (i.e. α and β) are sort of a balancing factor which decides the amount of style to be transferred. Varying α and β might reflect significant visual changes and can be seen in the paper. First total loss is computed then we try to minimize this loss by backpropagation in order to optimize the generated image.

2.3 Classification of Artistic Style Transfer

Artistic Style Transfer can be broadly classified in 5 categories, by a deep research Jing et al. [16] were able to classify Artistic Style Transfer into the below given categories.

- Stroke base Rendering
- Image Analogy
- Neural Style Transfer
- Image Filtering
- Texture Synthesis

In this section, we would be focusing onto classify the style transfer techniques and optimizations of the fundamental technique by Convolutional Neural Networks.

2.3.1 Classification in Neural Style Transfer

Style Transfer via Convolutional Neural Networks can be further classified in the below given categories:

A. Slow neural method based on online image optimization: It is based onto the slow image reconstruction technique while it is referred as iterative optimization technique.

   I. Parametric slow neural method with summary statics: It falls in the slow neural method again where style is referred as spatial summary statistics.

   II. Non-Parametric slow neural method with MRF: This technique is slow neural method where style transfer takes place via operating on patches.

B. Fast neural method based on offline model optimization: This category of Neural Style Transfer came into picture due to the less efficiencies and limitations.[16] Taken large set of images I_c having bunch of styles I_s we may optimize the feed-forward network g and hence the formulation can be express as given below:

$$\theta^* = \underset{\theta}{\arg min} L_{total}(I_c, I_s, g_\theta * (I_c)), \quad I^* = g_\theta * (I_c) \tag{7}$$

As g produces individually, it can be further classified by Fast neural method based on offline model optimization as follows:

   I. Per style per model: It is considered as the fast neural method in terms of testing where it is pre-trained feed forward style specific network. It only takes a single forward pass to stylize.

   II. Multiple style per model: It also falls in the fast neural method because of the improve flexibility.

   III. Arbitrary style per model: With only one trainable model it is able to transfer arbitrarily and hence fall in the fast neural method again.

2.4 Neural Style Transfer Challenges

Image transformations with CNN initially quite challenging in which one of the key challenge is not to transfer the geometric features while preserving the semantic content hence capturing the style and content individually needed to carried out. Another challenge faced was, as there are wide variety of artistic styles and content input images hence pairing these styles with the right content in order to tune them together is quite difficult. As the real world contains real complex scenes with so much distortion so in order to get fairly good results, tuning of the input images is necessary.

Loading the pre-trained model and working with it in terms of Style Transfer whereas and retrieving layer activation is quite challenging.

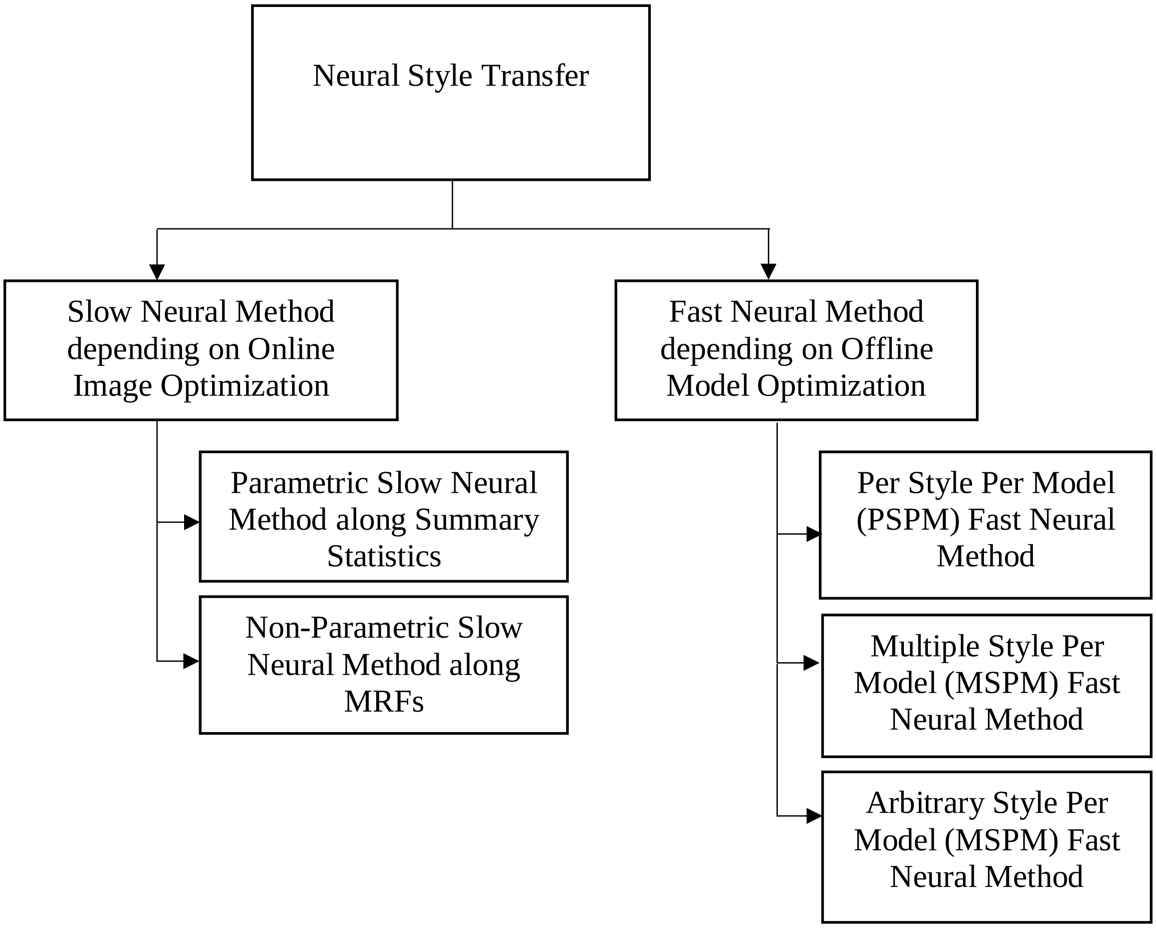


Fig. 4 Neural Style Transfer classification.

Chapter 3

THE PROPOSED WORK

3.1 Objective

The work that has been carried out is inspired by Gatys et al. [2] in which they were able to extract content and style from an image separately therefore style transfer became achievable. From then Neural Style Transfer (NST) got all the attentions and became the trending research field. The biggest challenged faced by authors was the processing time, because the whole point was to reduce the time taken by the artist and reduction of the whole lot of efforts that an artist has to incorporate. Another challenges that were faced by the authors were that the transformed pictures were generated depending upon the low-level noise. Convolutional Neural Networks are primarily appreciated for the classification and feature recognition tasks but in order to achieve style transfer initially the network architecture and configuration was explored while on the other hand setting up the network architecture with the adjustment of hyperparameter would help in achieve more pleasant results. These factors are to decide the perceptual quality of the network architecture and resolution of the results while the output might get affected if chosen wrong parameters. This architecture is used to test and train our model and to achieve better results.

3.2 Training the Neural Network in NST

In order to transform an image into an acceptable image, it requires tens of thousands of iterations with required graphics card. One can train Neural networks in order to Transfer the style onto one image or many images that are to be fed to the network, with universal function approximators (UFA). As we have also seen in the previous section, the total loss can be written as:

$$\mathcal{L}_{total}\left(\vec{P},\vec{a},\vec{x}\right) = \alpha\mathcal{L}_{content}\left(\vec{P},\vec{x}\right) + \beta\mathcal{L}_{style}\left(\vec{a},\vec{x}\right) \tag{8}$$

Where α is β are the hyperparameters for content and style losses respectively. Initially the forward pass takes less computations as compared to the optimization schemes which is again illustrated by Gatys et al. [2] Hence the stylization can be achieved very quickly even on the mobile phones as wells whereas the rate of speed to transfer the style comes at the cost which is the training the network beforehand. Not only that, tens of thousands of input images (content images) are used in order to train the network which again takes 1.5 hours even in Nvidia 1080TI.

3.2.1 *Network Architecture*

As we already aware about the Convolutional Neural Networks where the image classification usually carries out but for image transformation we need something different than this or some fresh network architecture. In almost every image transformation network architecture consist convolution which means they tend to not use the layer like fully-connected-layers. Hence, by doing this we achieve reduction in terms of parameters when it comes to train the neural network and transform images when training is over.

The main advantages of this network is the down-sampling or in another words, image size reduction which then let us exploit even bigger networks having somewhat similar computational cost as we used to do with small networks with non-down-sampling picture. The another advantage of using this network is, it increases receptive field by the size of 2D using down-sampling factor of D. We can have some intuition about the receptive field by referring to the fig. 6. Neurons would have seen the complete image if there was reduction of one pixel on every side.

Transformation of the style needs a better network in which Residual blocks plays an important role which applies the convolutions series wise and then the input fed back to the output which is also called the shortcut mapping. According to He et. al., a network can be leant easily identity function if used shortcut mapping. Also, we also have batch normalization used for normalization techniques which distributes pictures throughout every filter. Due to shortage or limitation of memory the images that are to be fed to the networks are handful and normalization carried out across the batch of images residing in network. The drawbacks like internal covariate shift can be avoided.

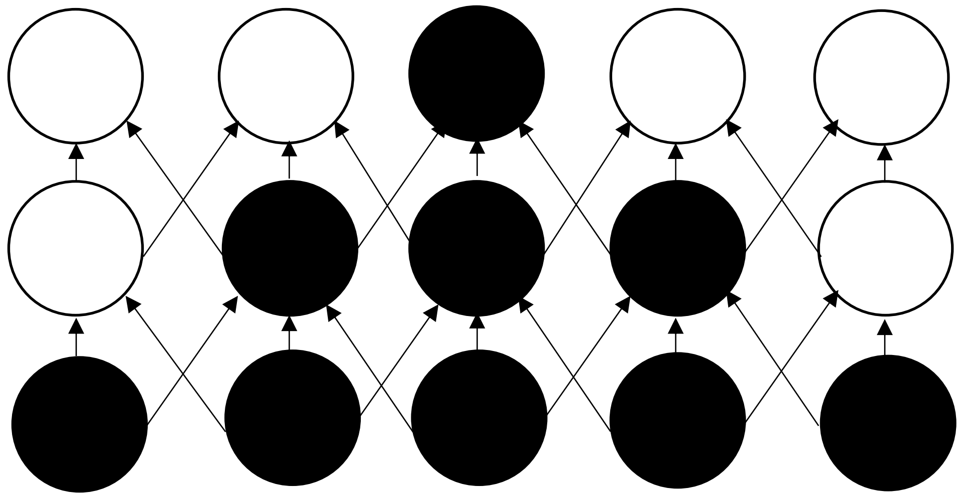


Fig. 5 Receptive field illustration shows the deeper layer is able to see broad view of the image.

3.3 Neural Style Transfer algorithm evaluation

Artificial Neural Network is inspired by neuroscientific research of brain where our brain is way more complex hence it, Deep Neural Networks are analogized to crudely inspired by birds. In place of matrix multiplication, if convolution has used it becomes convolutional neural networks rather a neural networks. These Convolutional Neural Networks has played a significant role in the field of Deep Learning they are the one closest to the biological neural networks and been exploited by many industries as well.

In order to transform images in an artistic way we first need to extract the style and content of the images separately, secondly the style must be transferred onto the generated image preserving the content of another input image.

secondly the style must be transferred onto the generated image preserving the content of another input image. Algorithmically, we have taken $\boldsymbol{x}$ as the white noise image, $\boldsymbol{p}$ is the content image and $\boldsymbol{a}$ is the style image where in layer '*conv4_2*' $\boldsymbol{P}$ represents the content representation. In order to transfer the style and content onto the base image we first need a measure which can calculate the difference of the both the images. The measure has taken into account in the form of loss functions, where for style and content we have taken separate loss functions. These loss functions are summed up and optimized in every iteration while adjusting the learning rate $\boldsymbol{Y}$ and simultaneously updating the weights.

Algorithm: Neural Style Transfer

$x \leftarrow$ white noise of shape (height, width, channels)
$p \leftarrow$ content photograph of size $x.shape$
$P \leftarrow$ content representation of p in layer 'conv4_2'
$a \leftarrow$ style artwork of size $x.shape$
$A \leftarrow$ style representation of style artwork
$\gamma \leftarrow$ learning rate
$\boldsymbol{for}$ $i \leftarrow 1$ to $n\_iteration$ $\boldsymbol{do}$
  $F \leftarrow \mathcal{L}$style representation of x
  $G \leftarrow$ style representation of x
  $\mathcal{L}_{total} \leftarrow \mathcal{L}_{\text{style}}(G, A) + \mathcal{L}_{\text{content}}(F, P)$
  $\boldsymbol{x}_{\text{ij}} \leftarrow \boldsymbol{x}_{\text{ij}} - \gamma \frac{\partial \mathcal{L} total}{\partial \boldsymbol{x}\text{ij}}$
return $\boldsymbol{x}$

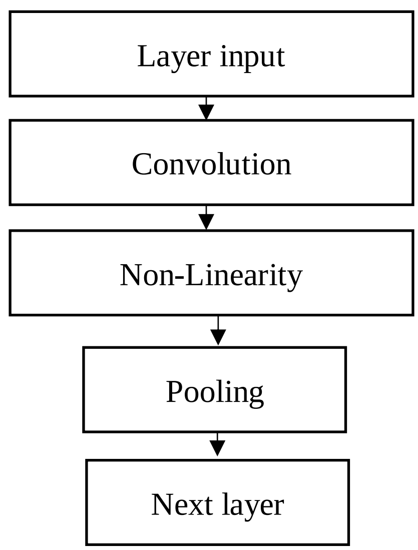


Fig. 6 Block diagram of layers in network architecture.

3.4 VGG Model

Simonyan and Zisserman introduced VGG in the year of 2014 in which there were only 3*3 convolutional layers increasingly. Max-pooling takes care of reduction in volume size, there are two layer that are fully-connected, at every layer there is 4096 nodes following by Softmax layer. Now that we have some intuitions about VGG let's dive deep into VGG network. So basically VGG networks are of two types, VGG16 and VGG19 where these numbers reflect the number of layers.

Neural style transfer using convolutional neural networks (CNN) has gained tremendous

attentions due the outstanding results and that have come true with the help of a pre-trained convolutional neural network that is VGG19. This model has been trained on ImageNet dataset which contains more than million images.

Table 1: Convolutional Network Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	
Input(224*224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv2-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Neural style Transfer is indeed a fun technique and able to achieve high quality extraordinary results but in the whole story VGG, which stands for visual geometry group, played significant role. VGG16 has been used by many authors including Gatys et al. [2] which was able to achieve fine results but with the recent advancement VGG19 seems to outperform the previous network because it is trained on more than one million images of dataset ImageNet. [2][4][7]

ImageNet dataset is pretty famous dataset due to its wide variety of image categorization and labelling. It has 22000 different object segregation that are being used for many purposes primarily researchers. The label "Convolutional neural network" associated with ImageNet generally refers to ImageNet large scale visual recognition challenge. Training in ImageNet carried out with 50000 images while for testing there are 100000 images. Hence ImageNet dataset has become a benchmark in terms of image classification task. Some of the pre-trained well performing convolutional neural networks in the main Keras library is able to outperform other many networks. Convolutional neural networks are always appreciated for the classification tasks but again Vgg19 convolutional neural network is able to classify images into 1000 categories like animals, roads, trains and so on and hence it has learned the majors that is feature representation. It has predefined image input size that is 224*224.

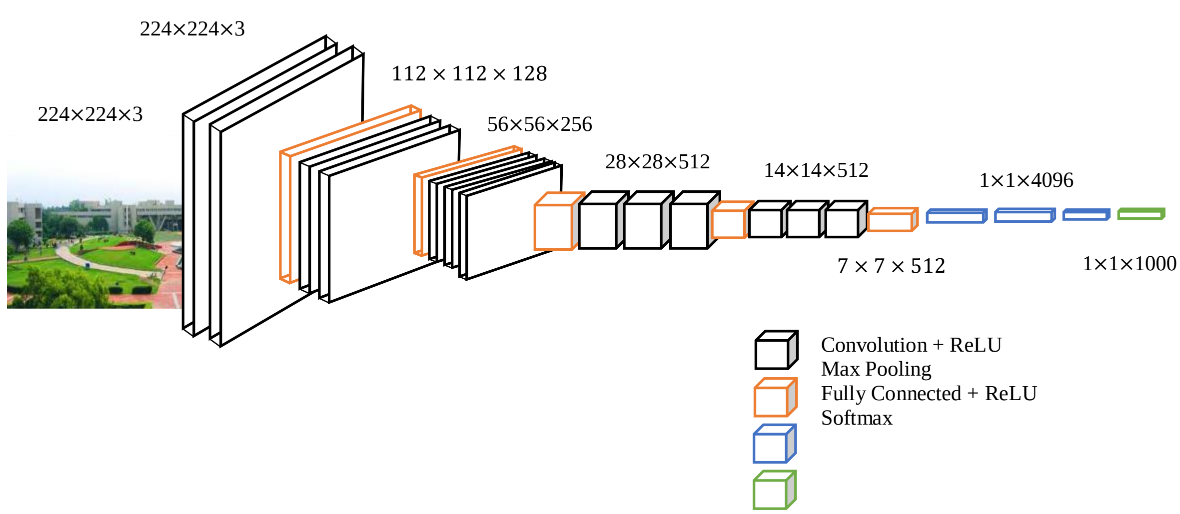


Fig. 7 Convolutional Neural Network architecture VGG19.

CHAPTER 4
EXPERIMENTAL WORK AND RESULTS

4.1 Overview

In Neural Style Transfer using convolutional neural networks we performed many experiments in order to get perceptually pleasant results, we explored the configuration and settings of the network architecture in hope to get satisfactory results to the NST problem. We aim to reduce the loss by adjusting the weights through calculating the gradients and hyperparameters via manual updation. We also tried Neural Style Transfer on different dataset images like Caltech101 and Caltech256 where we found that the images included in each datasets are of wide variety and hence Style Transfer differs in every case. We chose different styles to transform the content image which is the image of "*Delhi Technological University, India*", results are shown below. VGG19 worked fabulously out of many pre-trained convolutional neural networks and hence we have used VGG19 network to train our model. VGG19 from Oxford has been pre-trained on ImageNet dataset which includes 50,000 images where 22000 different object segregation categories are present.

Black and white images were tested with the colored stylization where we found the reduction of the processing time and error as well. Although in terms of time and loss black and white images seems to outperform but it does not give the satisfactory results. The processing time to transform a black and white image came out to be 307 seconds in 5th iteration which is almost half of any colored image. Loss of the black and white image was 9.352594e+31 where in colored image the loss was way more.

4.2 Setup and Datasets

Neural Style Transfer is an optimization technique which emits magical results with the help of Neural Networks. Convolutional Neural Networks (CNN) primarily used for image classification task but in Neural Style Transfer it works as a feature extractor. As we used VGG19 which is a pre-trained CNN on the ImageNet dataset, we shall first discuss the training dataset and secondly the testing datasets in the next section.

Name	Published Date	Total Categories	Images
Caltech101	2003	102	9144
Caltech256	2006	257	30607

Table 2. Average Loss values on Caltech101and Caltech256 datasets.

We have used 2 dataset in the experiments which are caltech101 and caltech256 as given below:

Caltech101 dataset used which contains 101 object categories taken from Google Images, there is another class for background category hence it contains 102 object categories. Caltech101 contains 9144 images where each class has almost 40-800 images where the size of the pixels are 300×200. The sample images can be found below in fig. 8 of Caltech101 where images were taken from different categorical classes.

Caltech256 dataset used which has 30,000 plus images, Caltech101refinements reflect in Caltech256 like the categories are doubled up with minimum 80 images are there for every class. This shows that Caltech256 has more images as compared to Caltech101 hence it takes

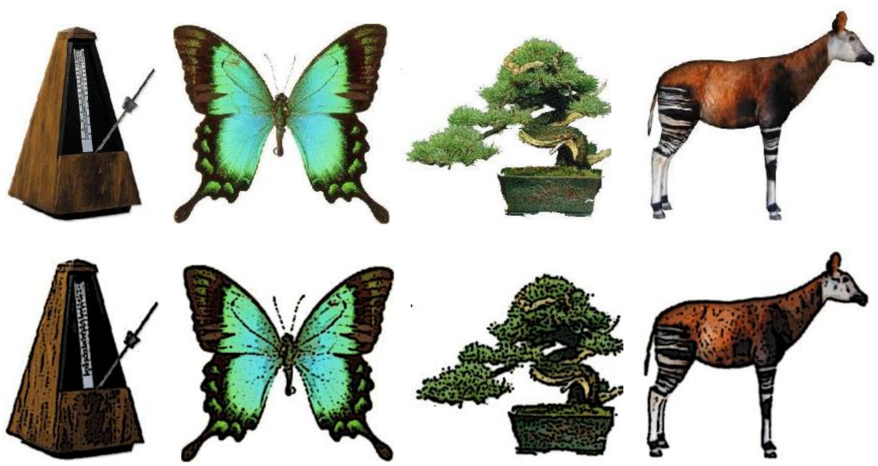


Fig. 8 Images from different classes of Caltech101 dataset.

much time. The sample images can be found below in fig. 8 of Caltech256 where images were taken from different categorical classes.

Fig. 9 Images from different classes of Caltech256 dataset.

(i) Sample image from Caltech101 (ii) **Sample image from Caltech256**

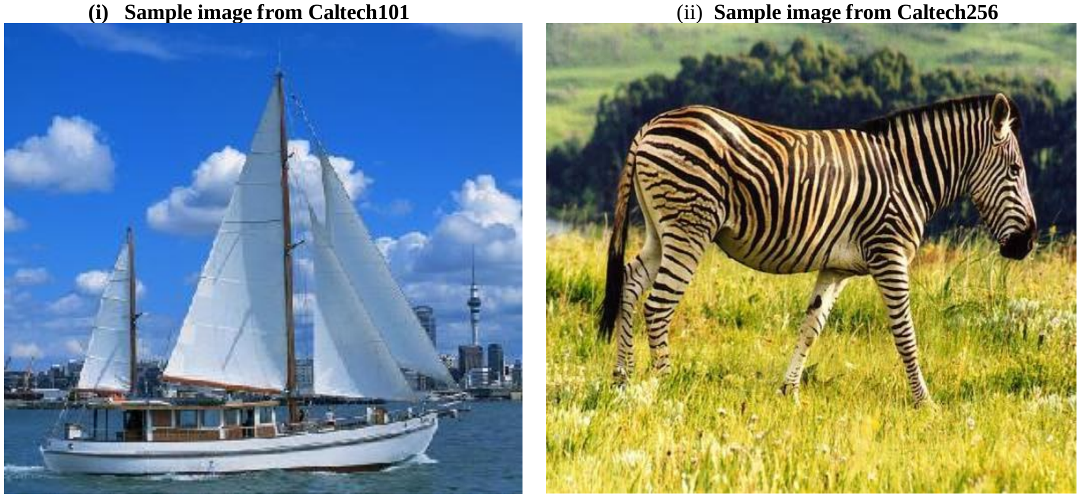
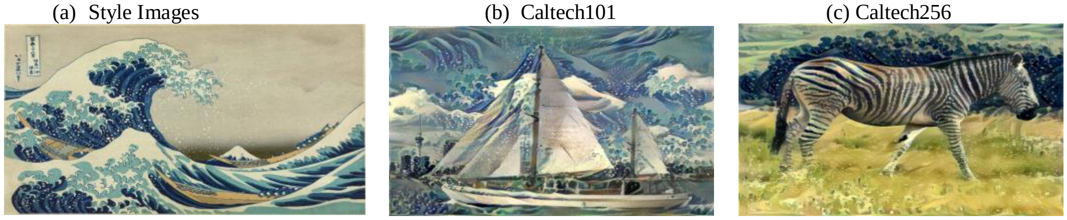


Fig. 10 Images from different classes: (i) left image is taken from Caltech101 dataset ‘ship’ category (ii) right image is taken from Caltech256 dataset ‘zebra category.

(a) Style Images (b) Caltech101 (c) Caltech256



Recombination of wave image with Caltech 101 ship and Caltech256 Jebra.

Recombination of Starry Night with Caltech 101 ship and Caltech256 Jebra.

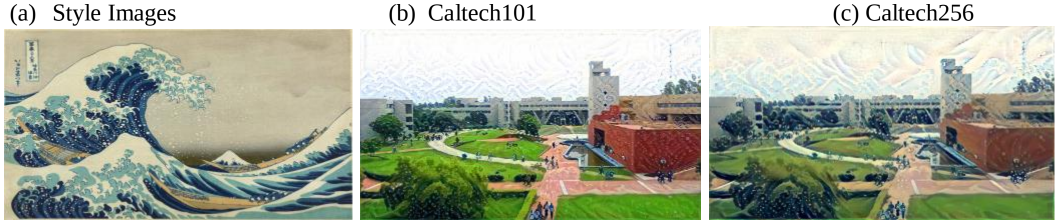
Recombination of Mossaic painting with Caltech 101 ship and Caltech256 Jebra.

Recombination of forest painting with Caltech 101 ship and Caltech256 Jebra.

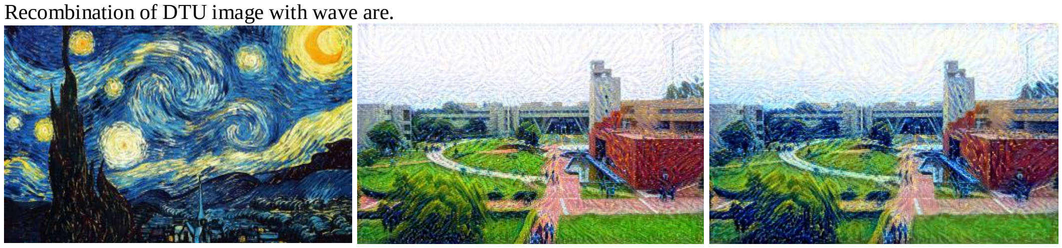
Recombination of geometry painting with Caltech 101 ship and Caltech256 Jebra.

Fig. 11 Style Transfer performed on image “Delhi Technological University” with 5 different artistic paintings from (i) Caltech101 dataset and (ii) Caltech256 dataset

(a) Style Images (b) Caltech101 (c) Caltech256



Recombination of DTU image with wave are.



Recombination of DTU image with Starry Night painting.

Recombination of DTU image with Mossaic painting.

Recombination of DTU image with the forest painting.

Recombination of DTU image with geometry painting.

Fig. 12 Style Transfer performed on image "Delhi Technological University" with 5 different artistic paintings.

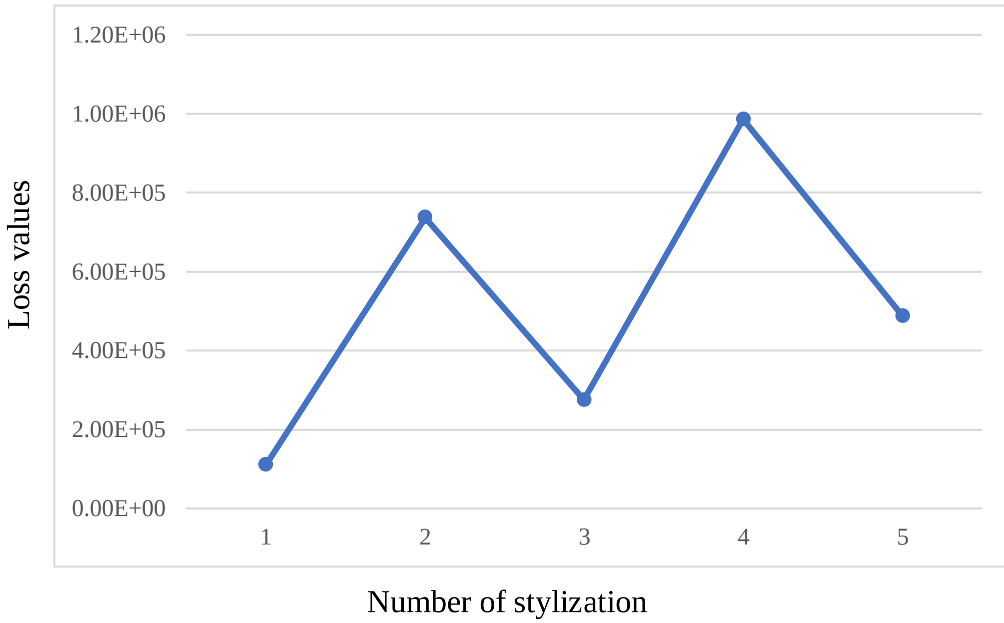


Fig. 13 Loss values of the illustration from Caltech101 dataset.

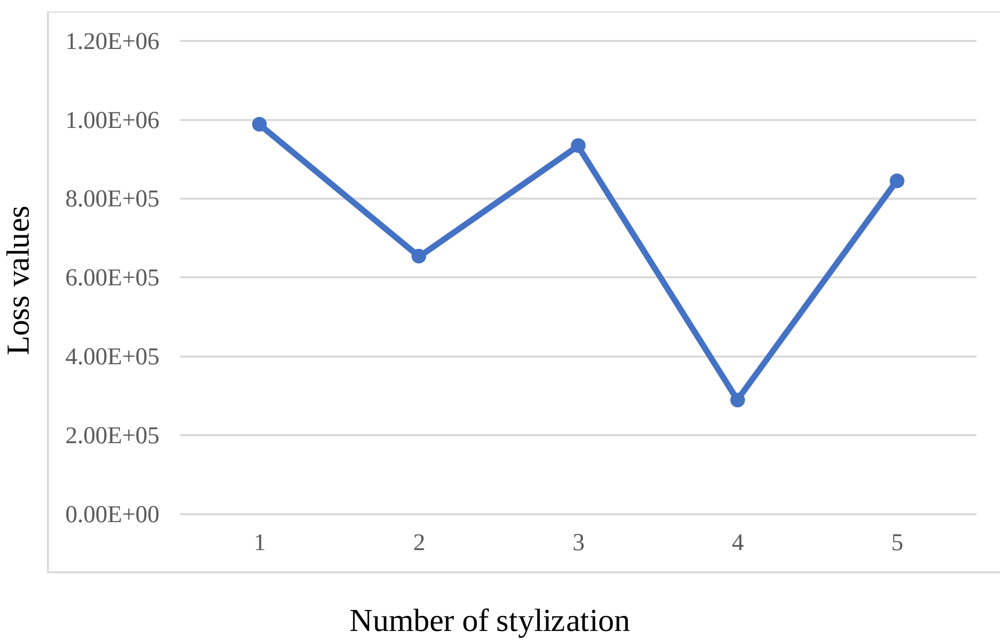


Fig. 14 Loss values of the illustration from Caltech256 dataset.

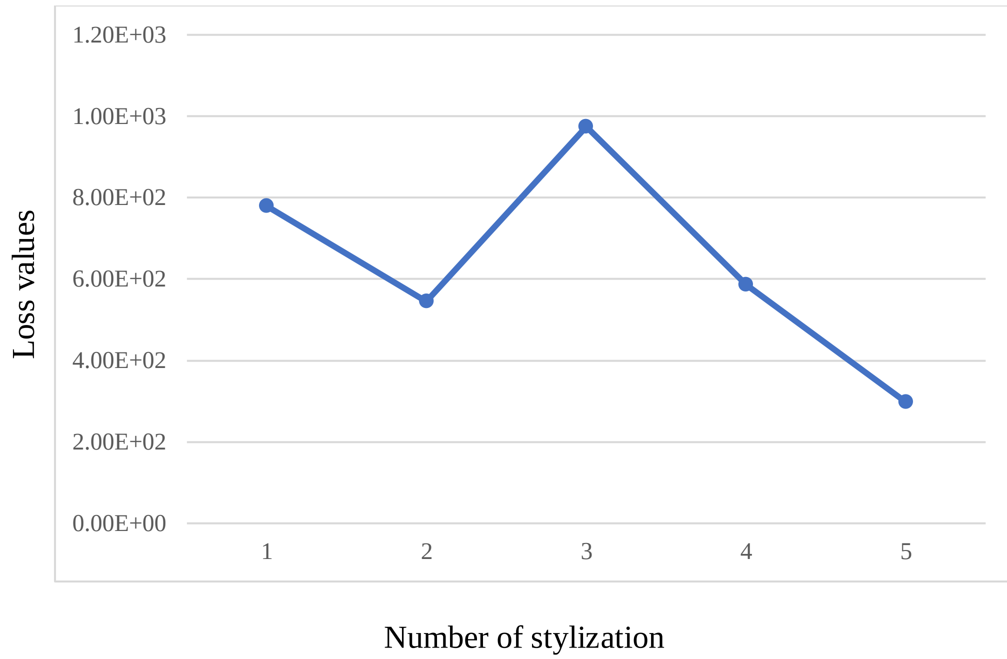


Fig. 15 Loss values of the illustration mentioned in section 4.3.

We have shown 5 examples of the Neural Style Transfer using Convolutional Neural Networks in which we have taken 5 paintings from different artists and 1 single content image. The examples show the difference between the iteration 1 transformed image and iteration 10 transformed image. The content image that we have taken is from "*Delhi Technological University*" to perform various style transformation. We have taken 10 iterations for single image transformation where 1 iteration takes around 600-700 seconds hence for 10 iterations 1 image took almost 1.5-2.0 hours on a CPU. In the given example we have shown 5 transformation of Delhi Technological University's image and time taken to generate all these transformation on CPU was higher compared to while we used NVIDIA TITAN RTX GPU to run these iterations.

The below graphical representation fig. 11 shows the loss value representation of the content image taken from "*Delhi Technological University*" on the above 5 stylization examples shown in the fig. 10 in section 4.3. Hence we may conclude the transformation of an image in an artistic way also depends upon the type of styled image we choose so the loss values in case of third example coming out to be large.

4.3 Performance Evaluation

Neural Style Transfer primarily proposed by [2] which became a popular research topic in 2015, their method performed very well and were able to generated quality results. Authors used Parametric slow Neural Method which had few limitations that were technical in nature. The state-of-the-art approach in order to satisfy the long range consistency was proposed by Berger et al. [22] in which they used Simonyan and Zisserman (2014) the same pre-trained model which was again motivated by Gatys. [2] Non-Parametric using MRFs which is a slow neural method was used by Li and Wand in which the photo realistic styles were generated having few plausibility. They preserved the Mesostructured in the transformed pictures, but the main limitation was the control of Mesostructured. Fast Neural Methods was also implemented depending upon the offline model optimization in which researchers were able to generated exciting results. Per Style Per Model was used by Jonson et al. [19] where they combined the idea of feed-forward image transformation and feed-forward optimization methods having the perceptual loss. [19]

Dumouling et al. [21] proposed new method based upon the Multi Style Per Model in which multiple images were transformed via conditional instance normalization in an efficient and scalable modifications. A different method was proposed by Tian Qi Chen and Mark Schmidt in which their primary goal was the speed and adaptation to the styles that is arbitrary in nature in the form of Arbitrary Style Per Model. They focused the single layer which consists of content and style together including convolutional neural networks layer. Excellent results were generated despite of the one layer constraint. They proposed a method which was simplified in nature but it had to suffer with the quality over efficiency.

Table 3: Comparison of loss values with the earlier state of the art algorithms for DTU images

	Loss		
	1st Iteration	50th Iteration	100th Iteration
Gatys et al. [2]	3.3500846e+33	6.3338757e+31	3.6105e+31
Jonson et al. [19]	1.205e+04	6.532e+03	5.653e+03
Ulyanov et al. [5]	4.823e+02	7.877e+05	9.575e+05
Li and Wand [4]	1.642e+03	8.964e+04	8.276e+05
Ours	**2.329e+02**	**9.667e+05**	**3.015e+05**

The above table 3 shows the loss value comparisons with the earlier state of the arts algorithms with the DTU images we have shown in the section 4.2. We have compared the our results with the above mentioned algorithms and our methods seems to outperform the above algorithms.

Table 4: Comparison of loss values with the earlier state of the art algorithms on Caltech101 dataset

	Loss		
	1st Iteration	50th Iteration	100th Iteration
Gatys et al. [2]	6.284e+33	4.3963e+32	2.472e+32
Jonson et al. [19]	3.763e+5	6.532e+03	5.653e+03
Ulyanov et al. [5]	4.637e+04	7.986e+04	9.561e+03
Li and Wand [4]	7.564e+04	5.874e+04	4.354e+05
Ours	**5.894e+03**	**7.853e+04**	**3.843e+04**

The above table 4 shows the loss value comparisons with the earlier state of the arts algorithms with the Caltech101 dataset. we have shown in the section 4.2. We have compared the our results with the above mentioned algorithms and our methods seems to outperform the above algorithms.

Table 5 Comparison of loss values with the earlier state of the art algorithms on Caltech256 dataset

	Loss		
	1st Iteration	50th Iteration	100th Iteration
Gatys et al. [2]	2.72e+33	7.942e+32	1.093e+32
Jonson et al. [19]	6.229e+4	5.832e+04	3.934e+04
Ulyanov et al. [5]	7.843e+04	9.638e+03	7.587e+03
Li and Wand [4]	8.263e+04	3.093e+04	2.751e+04
Ours	**2.847e+02**	**6.983e+03**	**7.423e+03**

:

The above table 5 shows the loss value comparisons with the earlier state of the arts algorithms with the Caltech256 dataset. we have shown in the section 4.2. We have compared the our results with the above mentioned algorithms and our methods seems to outperform the above algorithms.

CHAPTER 5
CONCLUSION AND FUTURE WORK

In this thesis, the network architecture is manipulated in order to reduce the loss and enhance the perceptual quality of resulting image, Hyperparameters which are the transformation control parameter are also chosen in a way that our model does not overfit. This way we were able to generate the better quality artistic images with the help of Convolutional Neural Networks which was mainly used for classification purposes. Accuracy enhancing results were generated using the VGG16 Network Architecture from 83% to 87%. The other network architecture which is VGG19 also performed well in terms of accuracy. Using VGG19 network architecture the accuracy enhanced from 84.5% to 86%. This method seems to work effective in the outdoor sceneries instead of indoor photographs. The tradition Neural Style Transfer model worked well but the output was distorted, this modification seems to generated even more smooth transformed images.

Fundamentals and the current advancement in the field of Neural Style Transfer are discussed where we found out that the most deployed Networks is VGG by Oxford which seems to work well on ImageNet dataset. Image transformation in the artistic way has become the most growing research field, not only researchers are incorporated in NST but industries are also showing interest towards it. The work has been successfully implemented on Neural Style Transfer but it also includes many limitations which again opens the door for future scope, hence one of the scopes is to take classification loss into account in order to optimize the network. [2][34] Classification loss if considered as a mean to update network one could have the channel to recognize the uncertain hyperparameters this way the weights might be optimized.

Another challenge might be the low-level noise if accrued in both of the images, i.e. content image and style image, which may affect the final generated image hence the denoising techniques may be incorporated once the optimization procedure has been carried out. [2][36][37][38] We may make the algorithm vigorous with the wide range of the input images which can be achieved by fixing the regions for the selections which are richer in nature and by segmentation process.

Our initial concern was the perceptual quality where time taken to transform the image was 10-15 minutes for an iteration while in order to run the complete 20 iteration it costs 200-300 minutes which shows the time to transform the images can be optimized and can be considered as a future implementation.

References

[1] Y. Chen, Y. Lai and Y. Liu, "Transforming photos to comics using convolutional neural networks," *IEEE International Conference on Image Processing (ICIP)*, pp. 2010-2014, 2017.

[2] L. A. Gatys, A. S. Ecker and M. Bethge, "Image Style Transfer Using Convolutional Neural Networks," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2414-2423, 2016.

[3] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann and E. Shechtman, "Controlling Perceptual Factors in Neural Style Transfer," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3730-3738, 2017.

[4] C. Li and M. Wand, "Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2479-2486, 2016.

[5] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky, "Texture Networks: Feed-forward Synthesis of Textures and Stylized Images," In ICML, pp. 1349-1357, 2016.

[6] Prisma Labs, “Prisma: Turn memories into art using artificial intelligence,” [Online], Available: http://prisma-ai.com, 2016.

[7] A. J. Champandard, “Deep forger: Paint photos in the style of famous artists,” [Online], Available: http://deepforger.com, 2015.

[8] M. Elad and P. Milanfar, "Style Transfer Via Texture Synthesis," in IEEE Transactions on Image Processing, vol. 26, no. 5, pp. 2338-2351, 2017.

[9] Y. Li, N. Wang, J. Liu, and X. Hou, "Demystifying neural style transfer," arXiv preprint arXiv:1701.01036, 2017.

[10] E. Risser, P. Wilmot, and C. Barnes, "Stable and controllable neural texture synthesis and style transfer using histogram losses," arXiv preprint arXiv:1701.08893, 2017.

[11] S. Li, X. Xu, L. Nie, and T. Chua, "Laplacian-steered neural style transfer," In Proceedings of the ACM on Multimedia Conference, pp. 1716-1724. ACM, pp. 1716-1724, 2017.

[12] D. Holden, I. Habibie, I. Kusajima and T. Komura, "Fast Neural Style Transfer for Motion Data," in IEEE Computer Graphics and Applications, vol. 37, no. 4, pp. 42-49, 2017.

[13] W. Chen, G. Sun, H. Li, Q. Sun and M. Shi, "A Style Transfer Algorithm of Caricature," International Conference on Multimedia Information Networking and Security, Nanjing, Jiangsu, pp. 49-53, 2010.

[14] X. Huang and S. Belongie, "Arbitrary Style Transfer in Real-Time with Adaptive Instance Normalization," IEEE International Conference on Computer Vision (ICCV), pp. 1510-1519, 2017.

[15] S. Prasad and B. N. Keshavamurthy, "An efficient artistic image-video transfer approach," International Conference on Mechatronics and Machine Vision in Practice (M2VIP), pp. 1-4, 2016.

[16] Y. Jing, Y. Yang, Z. Feng, J. Ye, and M. Song, “Neural Style Transfer: A Review,” CoRR, abs/1705.04058, pp 1-21, 2017.

[17] A. Mordvintsev, C. Olah, and M. Tyka. "Inceptionism: Going deeper into neural networks," pp. 1-17, 2015.

[18] J. E. Kyprianidis, J. Collomosse, T. Wang and T. Isenberg, "State of the "Art”: A Taxonomy of Artistic Stylization Techniques for Images and Video," IEEE Transactions on Visualization and Computer Graphics, vol. 19, no. 5, pp. 866-885, 2013.

[19] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual Losses for Real-Time Style Transfer and Super-Resolution,” ECCV, pp. 619-711, 2016

[20] V. Dumoulin, J. Shlens, and M. Kudlur, “A Learned Representation For Artistic Style,” CoRR, abs/1610.07629, 2016.

[21] T. Q. Chen, and M. Schmidt, "Fast patch-based style transfer of arbitrary style," arXiv preprint arXiv:1612.04337, 2016.

[22] G. Berger, and R. Memisevic, “Incorporating long-range consistency in CNN-based texture generation,” CoRR, abs/1606.01286, pp. 1-17, 2016.

[23] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu and M. Yang, "Diversified Texture Synthesis with Feed-Forward Networks," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-11, 2017.

[24] Y. Li, C. Fang, J. Yang, Z. Wang, X. Lu, and M. Yang, "Universal style transfer via feature transforms." In Advances in Neural Information Processing Systems, pp. 386-396. 2017.

[25] S. Gu, C. Chen, J. Liao, and L. Yuan, "Arbitrary Style Transfer with Deep Feature Reshuffle," CoRR, abs/1805.04103, pp. 1-10, 2018.

[26] R. Sreeraman, "Neural styler: Turn your videos/photos/gif into art," [Online], Available: http://neuralstyler.com/ 16, 2016

[27] "DeepArt," [Online], Available: https://deepart.io/, 2016

[28] K. Zeng, R. Zhang, X. Lan, Y. Pan and L. Lin, "Color style transfer by constraint locally linear embedding," IEEE International Conference on Image Processing, pp. 1121-1124, 2011.

[29] F. Luan, S. Paris, E. Shechtman and K. Bala, "Deep Photo Style Transfer," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6997-7005, 2017.

[30] E. Risser, P. Wilmot, and C. Barnes, "Stable and controllable neural texture synthesis and style transfer using histogram losses," arXiv preprint arXiv:1701.08893, 2017.

[31] B. J. Joshi, K. Stewart, and D. Shapiro, "Bringing impressionism to life with neural style transfer in come swim," In Proceedings of the ACM SIGGRAPH Digital Production Symposium, p. 5, 2017.

[32] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explain- ing and harnessing adversarial examples," in International Conference on Learning Representations, 2015. 17

[33] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," arXiv preprint arXiv:1801.00553, 2018. 17

[34] Y. Deng, C. C. Loy, and X. Tang, "Image aesthetic assess- ment: An experimental survey," IEEE Signal Processing Magazine, vol. 34, no. 4, pp. 80–106, 2017. 17

[35] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018. 18

[36] P. Upchurch, J. Gardner, G. Pleiss, R. Pless, N. Snavely, K. Bala, and K. Weinberger, "Deep feature interpolation for image content changes," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7064–7073. 18

[37] C. Chen, X. Tan, and K.-Y. K. Wong, "Face sketch synthesis with style transfer using pyramid column feature," in IEEE Winter Conference on Applications of Computer Vi-

sion. Lake Tahoe, USA, 2018.

[38] Sagar and D. K. Vishwakarma, "A State-of-the-Arts and Prospective in Neural Style Transfer," *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, Noida, India, 2019, pp. 244-247. doi: 10.1109/SPIN.2019.8711612