Style Transfer with Convolutional Neural Network(CNN)

ISHAN GUPTA – 19BCE7467

KINSHUK AGARWAL-19BCN7283

CONTENTS

OBJECTIVES:

- Use neural style transfer to randomize texture, contrast, and color
- Allow users to create their own artwork using certain content and style images
- Transferring the style of the source image

PROBLEM STATEMENT

- In recent few years, we have experienced the application of computer vision in almost every nook and corner of our life — thanks to the availability of huge amounts of data and super-powered GPUs, which have made training and deployment of convolutional neural networks(CNN) super easy.
- Transferring the style from one image onto another can be considered a problem of texture transfer. In texture transfer the goal is to synthesise a texture from a source image while constraining the texture synthesis in order to preserve the semantic content of a target image.
- A Neural Algorithm of Artistic Style that can separate and recombine the image content and style of natural images. The algorithm allows us to produce new images of high perceptual quality that combine the content of an arbitrary photograph with the appearance of numerous wellknown artworks
- So the problem statement here is given a content photo X and style photo Y how can we transfer the style of Y to the content X to generate a new photo Z. How can we train CNN to process and optimize the differences(the difference between X and Y)to reach an optimum global(Z)?.

TOOLS USED

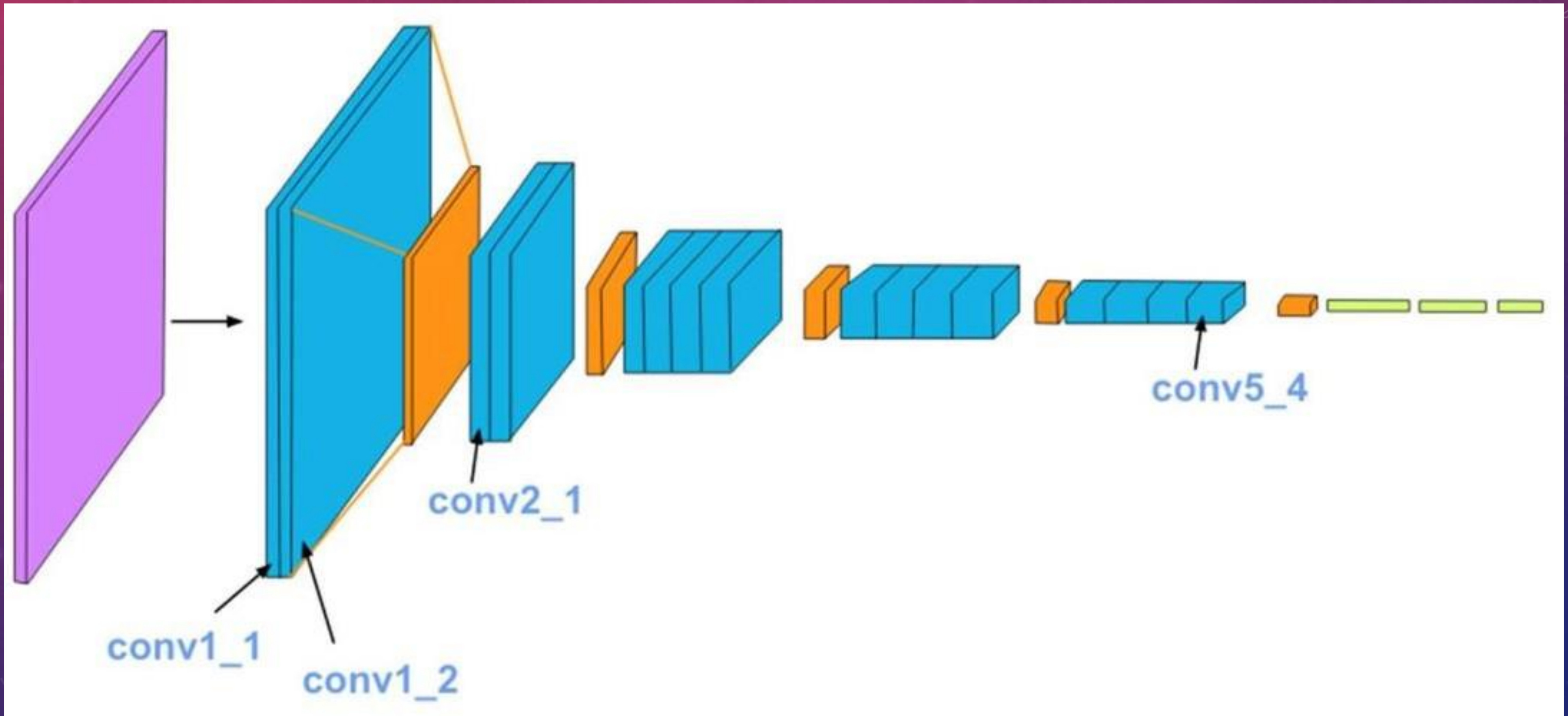
- Pytorch
- Numpy
- matplotlib.pyplot
- Torchvision (models, transforms, utils)
- Optim from torch
- PIL (Image)
- Imageio (imwrite)
- vgg16

WHAT IS STYLE TRANSFER?

- Style transfer is the technique of recomposing images in the style of other images. It all started when Gatys et al. published an awesomepaper on how it was actually possible to transfer artistic style from one painting to another picture using convolutional neural networks 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.
- Neural style transfer is an optimization technique used to take two images—a *content* image and a *style reference* image (such as an artwork by a famous painter)—and blend them together so the output image looks like the content image, but “painted” in the style of the style reference image.
- This is implemented by optimizing the output image to match the content statistics of the content image and the style statistics of the style reference image. These statistics are extracted from the images using a convolutional network.

In the image below, the convolutional layers are named by stack and their order in the stack.

1. Conv_1_1 is the first convolutional layer that an image is passed through, in the first stack.
2. Conv_2_1 is the first convolutional layer in the *second* stack.
3. The deepest convolutional layer in the network is conv_5_4.



WHAT DOES THIS ALGORITHM HELPS US IN?

- The algorithm modifies the target image, replacing some highfrequency information with the source texture. Although synthesis and transfer operations share many of the same challenges, there are significant differences. First, a clear criterion of success exists in texture synthesis: the result has to look like the input. For texture transfer, the degree of similarity with the original target image is usually adjusted based on user preferences. The case of artistic style transfer is probably the best illustration of this. The definition of artistic style is subjective; success in attaining this style is a matter of personal preference.

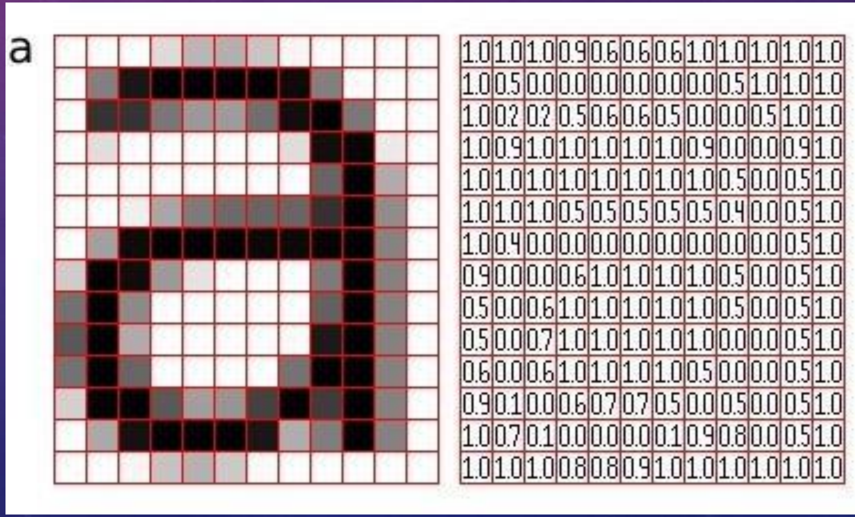
SOME BASIC CONCEPTS USED

LAYERS OF NEURAL NETWORK

- In a regular Neural Network there are three types of layers:
- 1.Input Layers: It's the layer in which we give input to our model.The number of neurons in this layer is equal to total number offeatures in our data (number of pixels in case of an image).
- 2.Hidden Layer: The input from Input layer is then feed into the hidden layer. There can be many hidden layers depending upon ourmodel and data size. Each hidden layer can have different numbers of neurons which are generally greater than the number of features. The output from each layer is computed by matrix multiplication of output of the previous layer with learnable weights of that layer and then by addition of learnable biases followed by activation function which makes the network nonlinear.
- 3.Output Layer: The output from the hidden layer is then fed into alogistic function like sigmoid or SoftMax which converts the output of each class into probability score of each class.

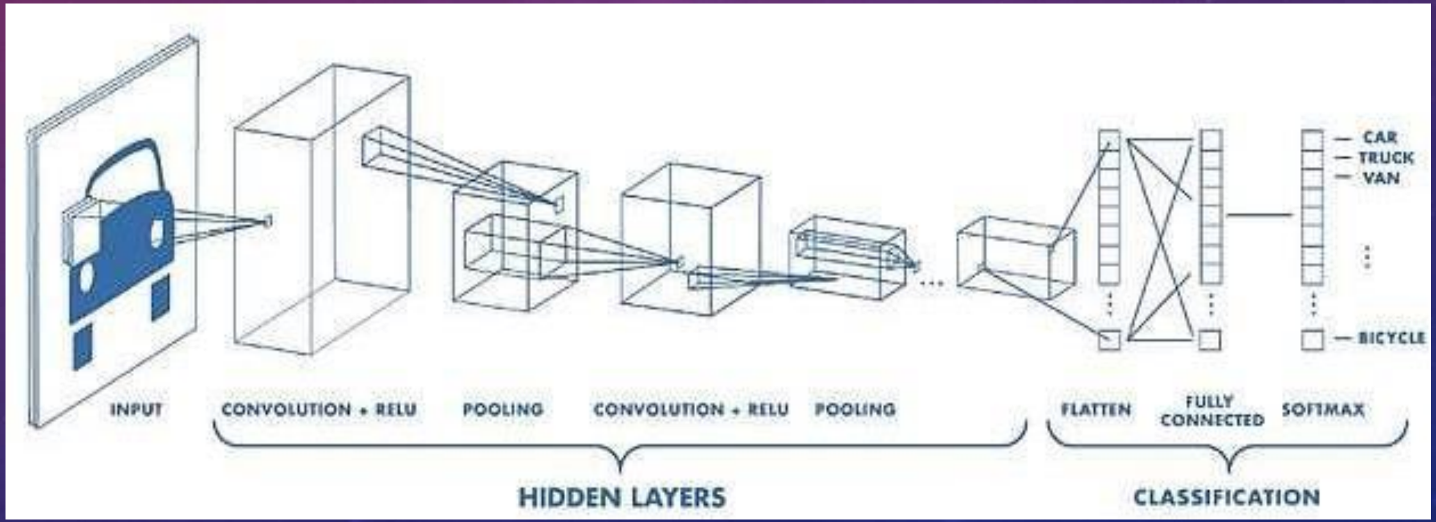
WHAT IS CNN?

- A Convolutional Neural Network, also known as CNN or ConvNet, is a class of neural networks that specializes in processing data that has a grid-like topology, such as an image. A digital image is a binary
- representation of visual data. It contains a series of pixels arranged in a grid-like fashion that contains pixel values to denote how bright and what color each pixel should be.



CONVOLUTIONAL NEURAL NETWORK ARCHITECTURE

- A CNN typically has three layers:
- 1.a convolutional layer,
- 2. pooling layer,
- 3.and fully connected layer.



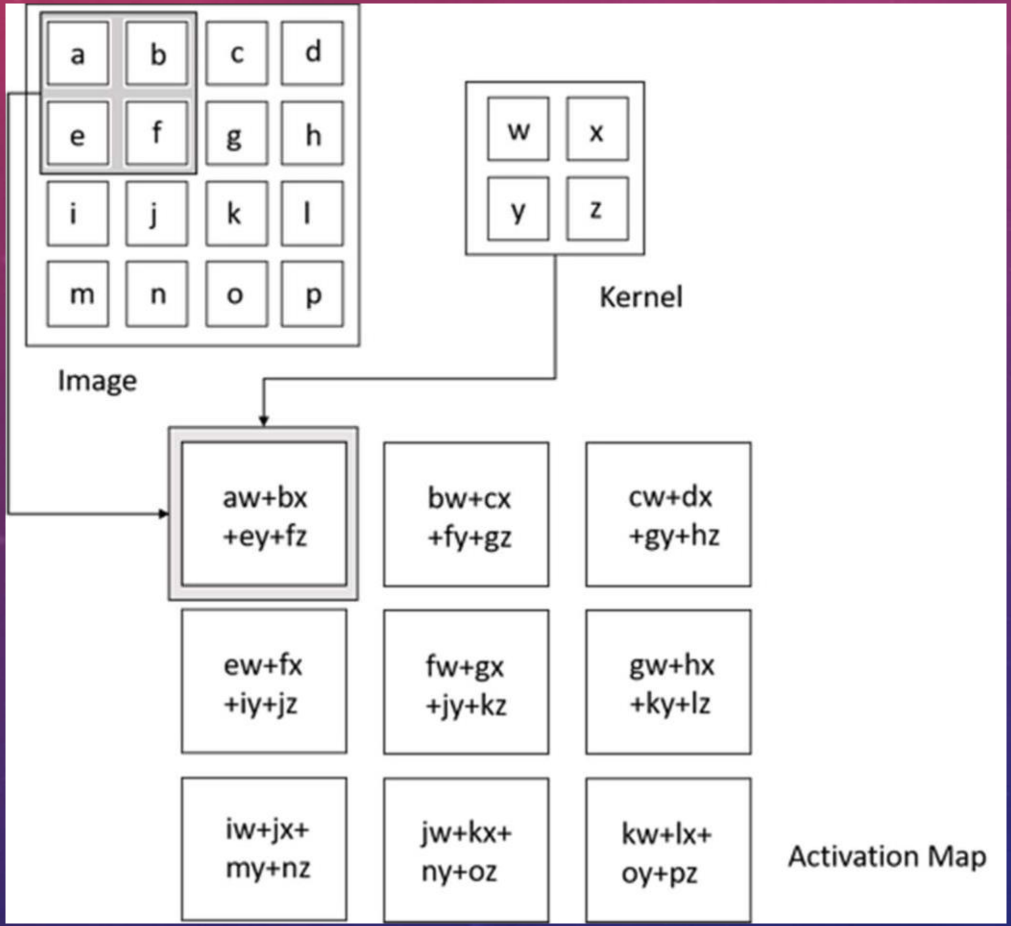
CONVOLUTION LAYER

- The convolution layer is the core building block of the CNN. It carries the main portion of the network's computational load.
- This layer performs a dot product between two matrices, where one matrix is the set of learnable parameters otherwise known as a kernel, and the other matrix is the restricted portion of the receptive field.

- The convolution layer is the core building block of the CNN. It carries the main portion of the network's computational load.
- This layer performs a dot product between two matrices, where one matrix is the set of learnable parameters otherwise known as a kernel, and the other matrix is the restricted portion of the receptive field.

IF WE HAVE AN INPUT OF SIZE W X W X D AND DOUT NUMBER OF KERNELS WITH A SPATIAL SIZE OF F WITH STRIDE S AND AMOUNT OF PADDING P, THEN THE SIZE OF OUTPUT VOLUME CAN BE DETERMINED BY THE FOLLOWING FORMULA:

$$W_{out} = \frac{W - F + 2P}{S} + 1$$

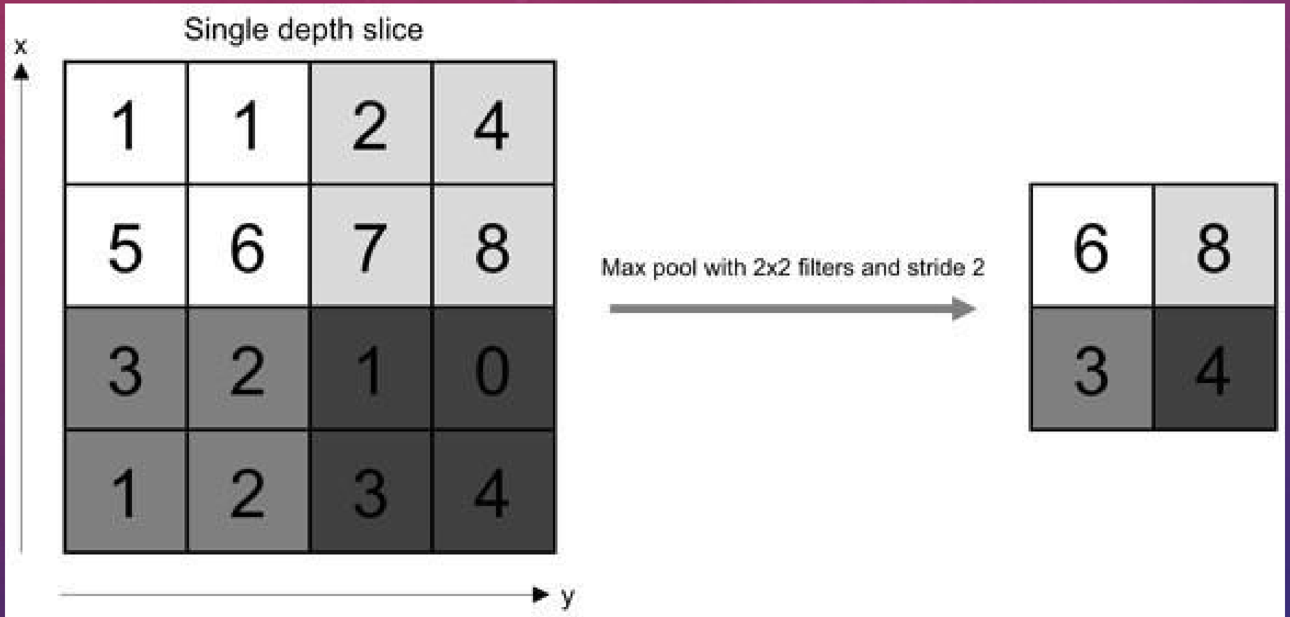


POOLING LAYER

- The pooling layer replaces the output of the network at certain locations by deriving a summary statistic of the nearby outputs. This helps in reducing the spatial size of the representation, which decreases the required amount of computation and weights. The pooling operation is processed on every slice of the representation individually.
- There are several pooling functions such as the average of the rectangular neighborhood, L2 norm of the rectangular neighborhood, and a weighted average based on the distance from the central pixel. However, the most popular process is max pooling, which reports the maximum output from the neighborhood.

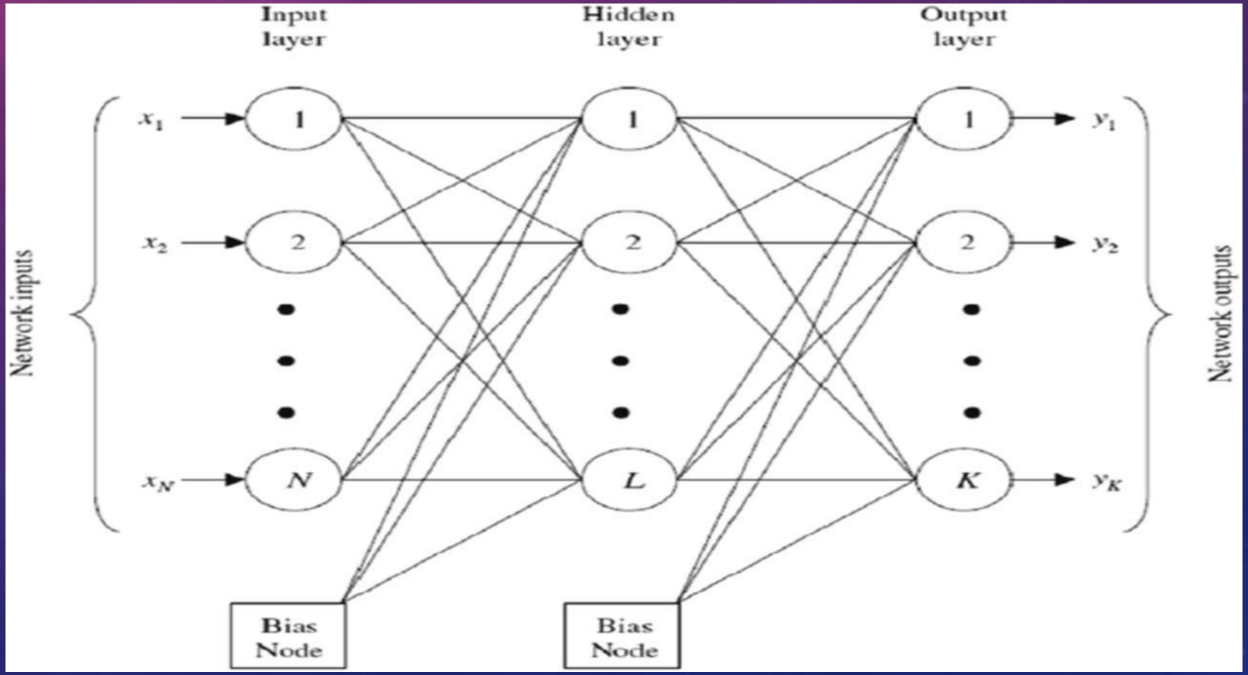
IF WE HAVE AN ACTIVATION MAP OF SIZE W X W X D, A POOLING KERNEL OF SPATIAL SIZE F, AND STRIDE S, THEN THE SIZE OF OUTPUT VOLUME CAN BE DETERMINED BY THE FOLLOWING FORMULA:

$$W_{out} = \frac{W - F}{S} + 1$$



FULLY CONNECTED LAYER

- Neurons in this layer have full connectivity with all neurons in the preceding and succeeding layer as seen in regular FCNN. This is why it can be computed as usual by a matrix multiplication followed by a bias effect.
- The FC layer helps map the representation between the input and the output.



NON-LINEARITY LAYERS

- Since convolution is a linear operation and images are far from linear, non-linearity layers are often placed directly after the convolutional layer to introduce non-linearity to the activation map.
- There are several types of non-linear operations, the popular ones being:
- 1. Sigmoid
- The sigmoid non-linearity has the mathematical form $\sigma(\kappa) = 1/(1+e^{-\kappa})$. It takes a real-valued number and "squashes" it into a range between 0 and 1.
- 2. Tanh
- Tanh squashes a real-valued number to the range [-1, 1]. Like sigmoid,the activation saturates, but—unlike the sigmoid neurons—its output is zero centered.
- 3.ReLU
- The Rectified Linear Unit (ReLU) has become very popular in the last few years. It computes the function $f(\kappa)=\max(0,\kappa)$. In other words, the activation is simply threshold at zero.

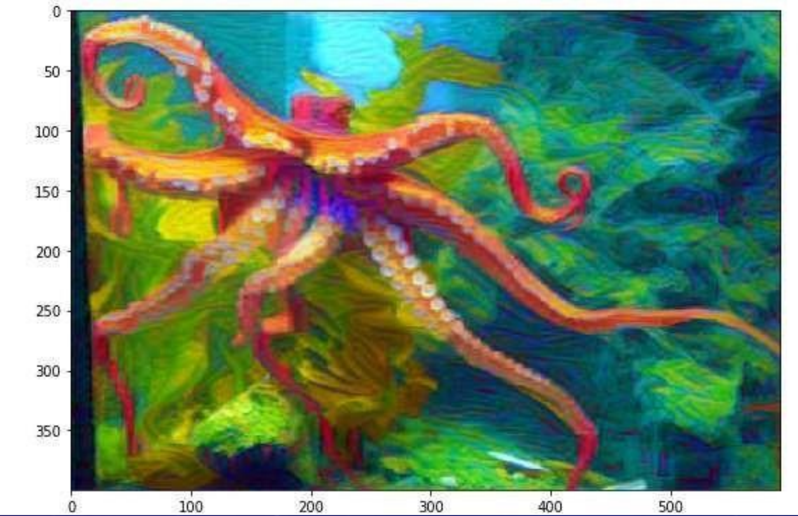
WHY DID WE USE PYTORCH

- PyTorch is an open source machine learning library based on the Torch library,[1][2][3] used for applications such as computer vision and natural language processing.[4] It is primarily developed by Facebook's artificial intelligence research group.[5][6][7] It is free andopen-source software released under the Modified BSD license.Although the Python interface is more polished and the primary focusof development, PyTorch also has a C++ frontend.[8] Furthermore, Uber's Pyro probabilistic programming language software uses PyTorch as a backend.[9]
- PyTorch provides two high-level features:[10]
- Tensor computing (like NumPy) with strong acceleration via graphics processing units (GPU)
- Deep neural networks built on a tape-based autodiff system

CODE

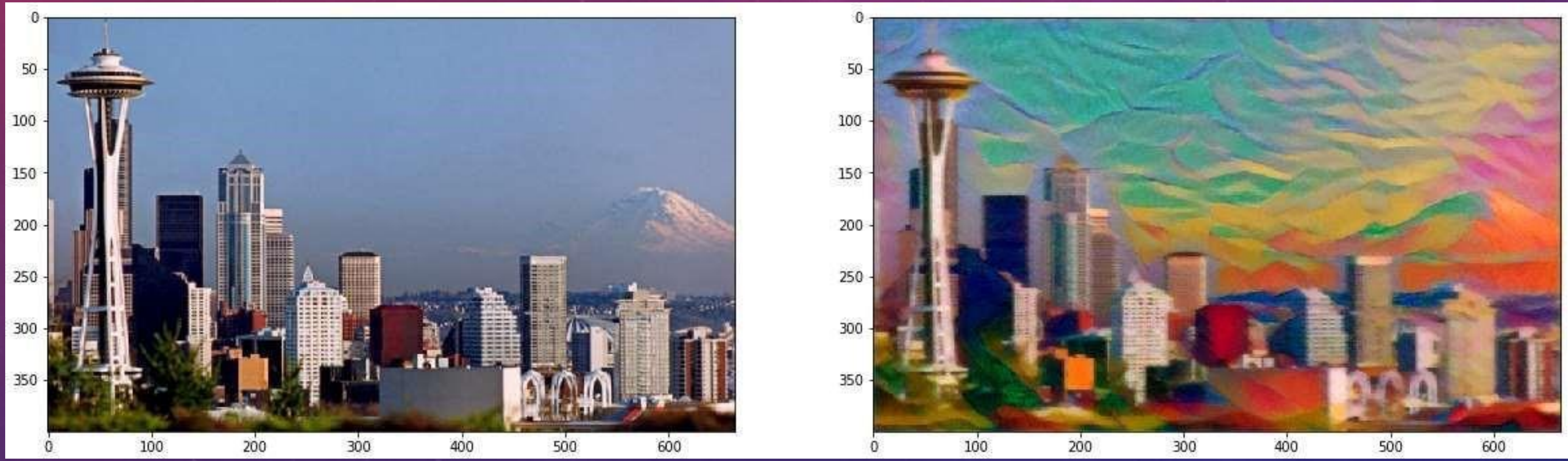
TEST CASE 1

OUTPUT

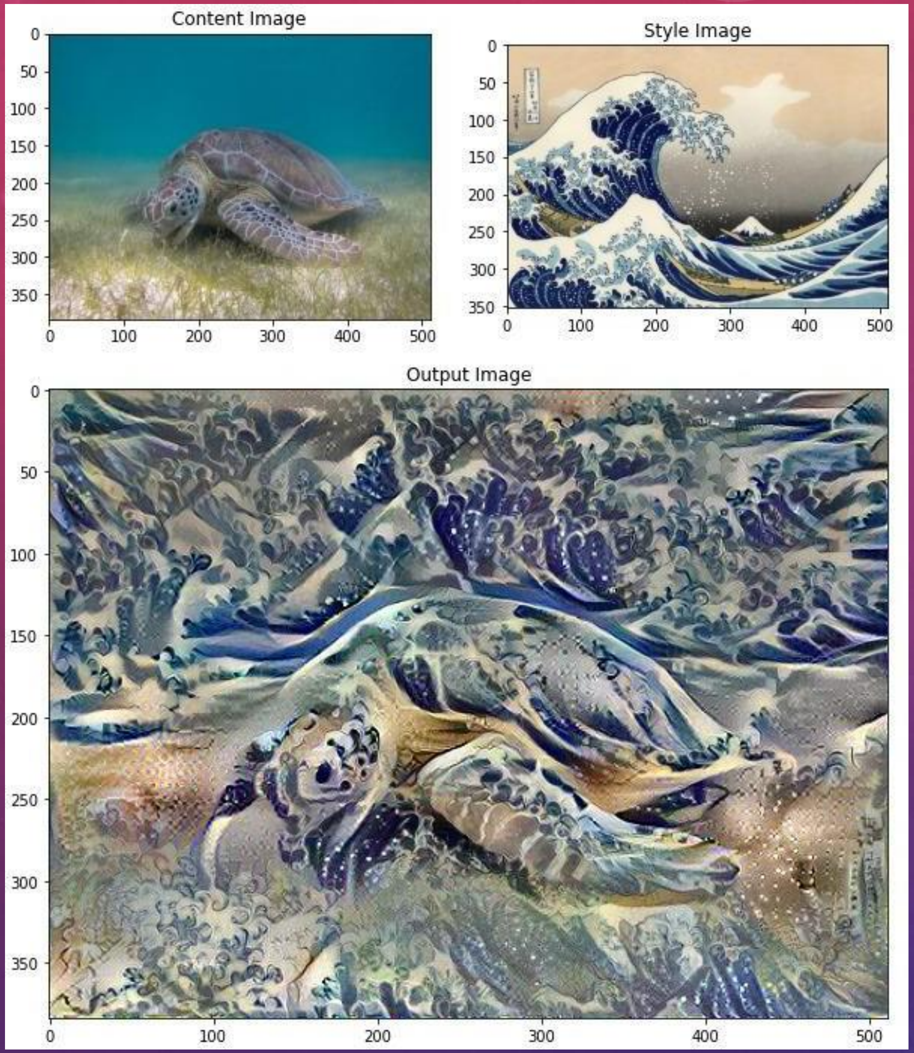


TESTCASE 2

OUTPUT



WHAT IF WE SWITCH THE STYLE IMAGE AND CONTENT IMAGE


Content Image
Style Image
Output Image
0 50 100 150 200 250 300 350
0 100 200 300 400 500


0 50 300
0 100 500
0 50 200
300 400
0 50 300
0 400 500

CONCLUSION

- Thus we conclude the Neural Style Transfer to be an advanced and futuristic technique to render images and generate results from ordinary input images into artistic impressions which previously could only be obtained by an artist. The future holds a lot of improvements in the process but for time being the techniques can be extensively used to ease any problems including customized beautifications, forensics and augmented reality basics.

FUTURE SCOPE

- Neural Style Transfer has also been extended to videos. Most recently, feature transform based Neural Style Transfer methods have been explored for fast stylization that are not coupled to single specific style and enable user- controllable blending of styles, for example the Whitening and Coloring Transform (WCT).
- One of the most intriguing recent ideas is the CycleGAN, which shows how to train image transformation algorithms from collections of before images and after images without requiring correspondences. That is, you can train to convert images of southern France to Monet paintings, without ever needing a photograph to go with each of the Monet training paintings.

THANK YOU!

ISHAN.19BCE7467@VITAP.AC.IN

KINSHUK.19BCN7283@VITAP.AC.IN