EAST WEST UNIVERSITY

Summer 2022

Course Title: Artificial Intelligence
Course Code: CSE366
Section No: 02

Project Report

Project Name: Neural Style Transfer

Submitted by
Group #7

Rafa Tasnim	2019-1-60-081
Malyha Bintha Mabud	2019-1-60-128
Md. Huzaifa	2019-1-60-079
Abdul Malek Sardar	2019-1-60-087
Mohsenul Kabir Mithun	2019-3-60-046

Submitted to

Nishat Tasnim Niloy
Lecturer
Department of Computer Science and Engineering
Faculty of Science and Engineering

Contents

Abstract

This project explores methods for artistic style transfer based on convolutional neural networks. The core idea proposed by Gatys et. al became very popular and with further research Johnson et. al overcame a major limitation to achieve style transfer in real-time.

We want to use machine learning programs to transfer daily pictures to artistic style images by using a style image for obtaining the transferred style and a content image which we want to transfer the style onto. This project can demonstrate how well machine learning techniques perform on image processing and how artificial intelligence can, to some extent, achieve considerable success in the field of art and creativity.

Introduction

The process of using image pairs to combine the "style" of one image with the "content" of the other to create a synthetic work of art is called style transfer. Style transfer is a popular topic in both academia and industry due to its unique appeal and wide applicability.

In visual arts, especially painting, people have acquired the ability to create unique visual experiences by constructing complex interactions between image content and style. So far, the basis of the algorithm for this process is unknown and no artificial system has similar functionality. However, in other important areas of vision, such as object and face recognition, human-like performance has recently been demonstrated by a class of biologically inspired visual models called deep neural networks. Here, we will introduce an artificial system based on a deep neural network that produces artistic images of high perceptual quality. The system uses neural representations to separate and rejoin the content and style of any image to provide a neural algorithm for creating artistic images. In addition, given the striking similarities between performance-optimized artificial neural networks and biological vision, our job is to algorithmically understand how humans create and perceive artistic images.

Gatys et al. demonstrated a generalized style transfer technique by exploiting feature responses from a pre-trained CNN, opening up the field of neural style transfer, or the process of using neural networks to render a given content image in different artistic styles. Since then, many improvements, alternatives, and extensions to the algorithm of Gatys et al. has been proposed.

In this work, we implement Gatys' algorithm – with some minor modifications – to produce some synthetic artwork.

Convolutional Neural Networks

Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Convolutional neural networks were originally developed for image classification but have recently been used for a variety of other tasks such as image segmentation, neural styling, and other computer vision and natural language processing tasks. CNNs are one of the most interpretable models in Deep Learning because of our ability to visual their representations and understand what they might be learning. A neural network is made up of neurons that are organized in layers. There are three types of layers: an input layer, an output layer, and a hidden layer.

There are two main parts to a CNN architecture

- A convolution tool that separates and identifies the various features of the image for analysis in a process called as Feature Extraction
- A fully connected layer that utilizes the output from the convolution process and predicts the class of the image based on the features extracted in previous stages.

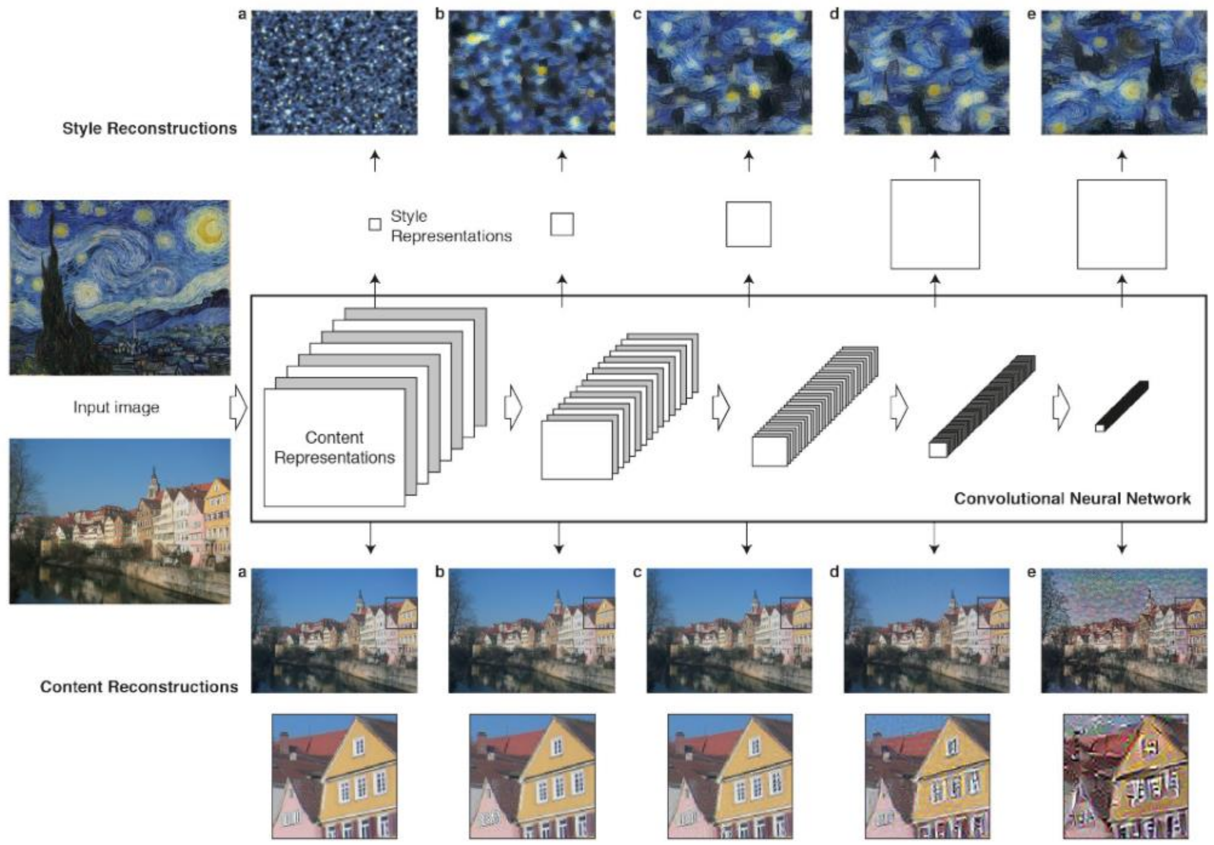


Figure 1

In Figure-1 A given input image is represented as a set of filtered images at each processing stage in the CNN. While the number of different filters increases along the processing hierarchy, the size of the filtered images is reduced by some down sampling mechanism (e.g., max pooling) leading to a decrease in the total number of units per layer of the network. Content Reconstructions. We can visualize the information at different processing stages in the CNN by reconstructing the input image from only knowing the network's responses in a particular layer. We reconstruct the input image from layers 'conv1 1' (a), 'conv2 1' (b), 'conv3 1' (c), 'conv4 1' (d) and 'conv5 1' (e) of the original VGG-Network. We find that reconstruction from lower layers is almost perfect (a, b, c). In higher layers of the network, detailed pixel information is lost while the high-level content of the image is preserved (d, e). Style Reconstructions. On top of the original CNN representations, we built a new feature space that captures the style of an input image. The style representation computes correlations between the different features in different layers of the CNN. We reconstruct the style of the input image from style representations built on different subsets of CNN layers ('conv1 1' (a), 'conv1 1' and 'conv2 1' (b), 'conv1 1', 'conv2 1' and 'conv3 1' (c), 'conv1 1', 'conv2 1', 'conv3 1' and 'conv4 1' (d), 'conv1 1', 'conv2 1', 'conv3 1', conv4 1' and 'conv5 1' (e)). This creates images that match the style of a given image on an increasing scale while discarding information of the global arrangement of the scene.

Image-Based Neural Style Transfer

We start by reimplementing the first NST algorithm proposed by Gatys et al. Given a content image Ic and a style image Is, the algorithm seeks a stylized image I that we minimize the following loss:

$$\arg\min_{I} \mathcal{L}(I_c, I_s, I) = \arg\min_{I} \ \alpha \mathcal{L}_c(I_c, I) + \beta \mathcal{L}_s(I_s, I)$$

where Lc and Ls represent the content loss and style loss respectively, both are computed based the layers of the VGG-19.

VGG-19

There are the four pre-trained networks you can use for computer vision tasks such as ranging from image generation, neural style transfer, image classification, image captioning, anomaly detection, and so on.

- ➔ VGG19
- ➔ Inceptionv3 (GoogLeNet)
- ➔ ResNet50
- ➔ EfficientNet

For this project I would like to describe VGG19 model. VGG19 is a variant of VGG model which in short consists of 19 layers (16 convolution layers, 3 Fully connected layer, 5 MaxPool layers and 1 SoftMax layer). By reconstructing the middle layers of the VGG-19 network, Gatys et al. find that a deep convolutional neural network is capable of extracting image content and some appearance from well-known artwork. Figure 2 shows the architecture of VGG-19. Designed for image classification, we are now more interested in the structure of the network.

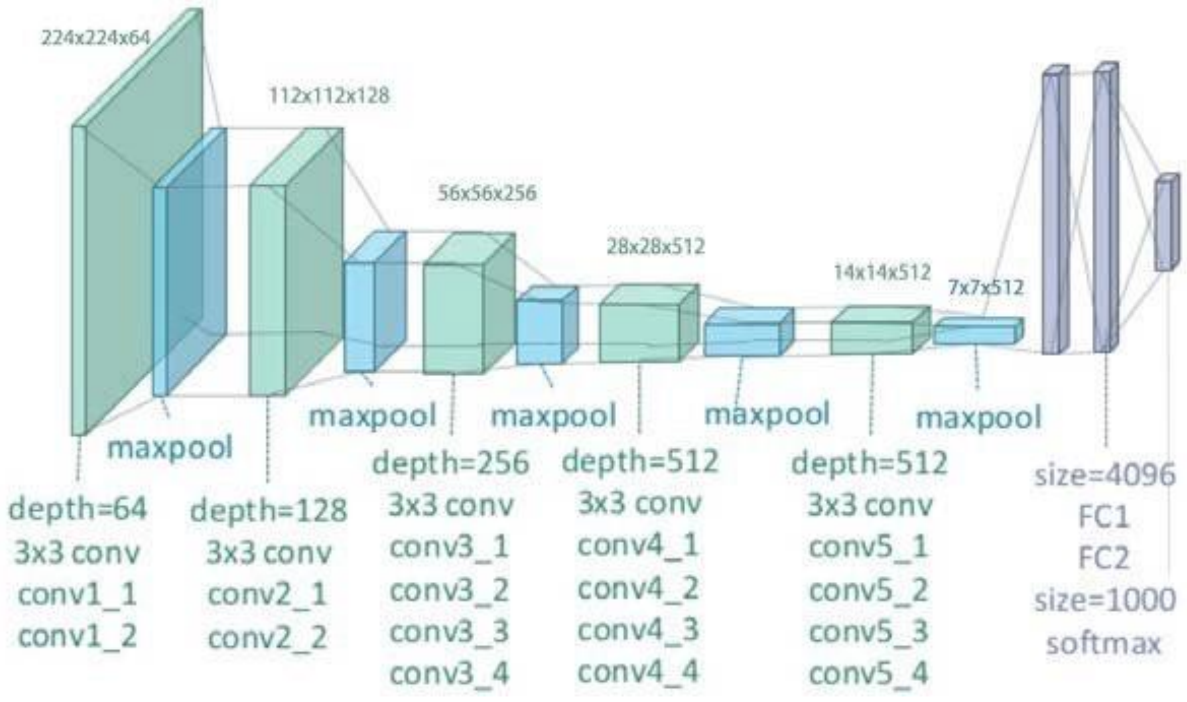


Figure 2

Architecture of the network VGG19 - The network has 16 convolutions with ReLUs between them and five maxpooling layers. The number of filter maps of the convolutions start at 64 and grow until 512. After the convolutions, there is a linear classifier made-up three fully connected layers with dropout between them, the first two have 4096 features while the last one has 1000. The last fc layer is connected to a softmax which maps each value to the probabilities of belonging to each of the 1000 classes of the ImageNet competition.

Method

Artistic style transfer may be defined as creating a stylized image x from a content image c and a style image s. Typically, the content image c is a photograph, and the style image s is a painting. A neural algorithm of artistic style posits the content and style of an image may be defined as follows:

- ➔ If the high-level features extracted by the image recognition system are close to each other at Euclidean distance, the contents of the two images are similar.
- ➔ If the low-level features extracted by the image recognition system have the same spatial statistics, the styles of the two images are similar.

The first definition is motivated by the observation that the higher level of functionality of the pre-trained image classification system matches the semantic information in the image. The second definition is based on the hypothesis that the style of painting can be seen as a visual texture. Extensive literature suggests that repetitive motifs representing visual textures may be characterized by lower-order spatial statistics. Images with the same low-order spatial statistics look perceptually identical and capture the visual texture. The assumption that the visual texture is spatially uniform means that low-order spatial statistics can be represented by a Gram matrix that represents the spatially averaged correlation of the entire filter within the representation of a particular layer.

Data Collection

The data set consists of two types of images: style image and content image. For simplicity we did not collect any data to train our CNN. We used pre-trained CNN model to complete this project. A pre-trained model is a model created and trained by someone else to solve a problem that is similar to ours. In practice, someone is almost always a tech giant or a group of star researchers. They usually choose a very large dataset as their base datasets such as ImageNet or the Wikipedia Corpus. Then, they create a large neural network (e.g., VGG19 has 143,667,240 parameters) to solve a particular problem (e.g., this problem is image classification for VGG19). Of course, this pre-trained model must be made public so that we can take these models and repurpose them.

Deep learning libraries already host many of these pre-trained models, which makes them more accessible and convenient like:

- TensorFlow Hub
- Keras Applications
- PyTorch Hub

In this project we used TensorFlow Hub.

TensorFlow Hub

TensorFlow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications.

Why TensorFlow

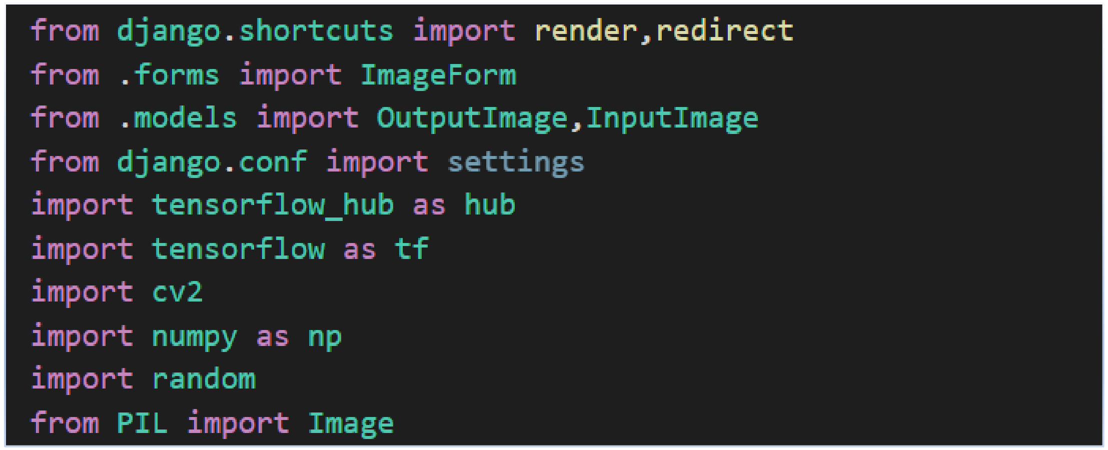
- ➔ Open-Source library
- ➔ Easy model building
- ➔ Robust ML production anywhere
- ➔ Powerful experimentation for research

Implementation

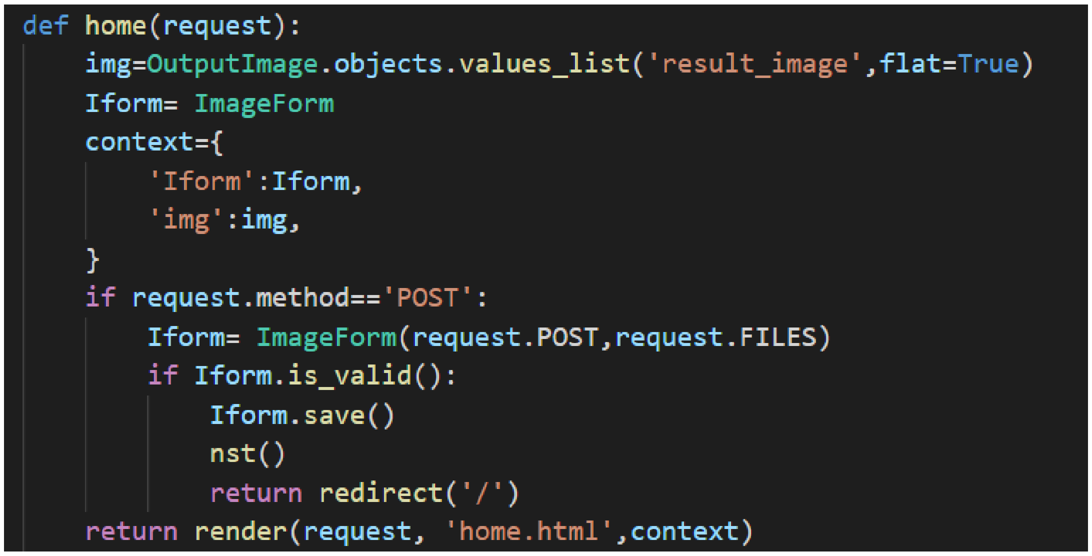
The original work for artistic style transfer with neural networks proposed a slow optimization algorithm that works on any arbitrary painting. Subsequent work developed a method for fast artistic style transfer that may operate in real time but was limited to one or a limited set of styles.

Like we said we will be using pre-trained convolutional neural networks. A way to cut short this process is the concept of transfer learning where libraries like **TensorFlow** have provided us with these giants and let us experiment with them on our own problem statements. We have used several other libraries. Cv2 library is very useful for computer vision applications such as image analysis. We used PIL for its image editing capabilities.

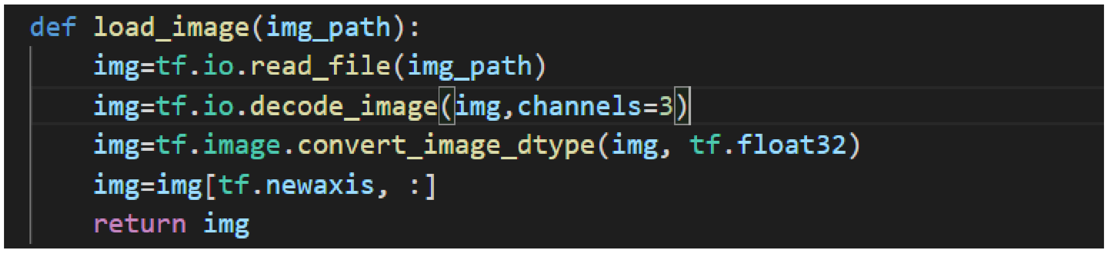
Let's start importing all relevant dependencies.


```python
from django.shortcuts import render,redirect
from .forms import ImageForm
from .models import OutputImage,InputImage
from django.conf import settings
import tensorflow_hub as hub
import tensorflow as tf
import cv2
import numpy as np
import random
from PIL import Image
```

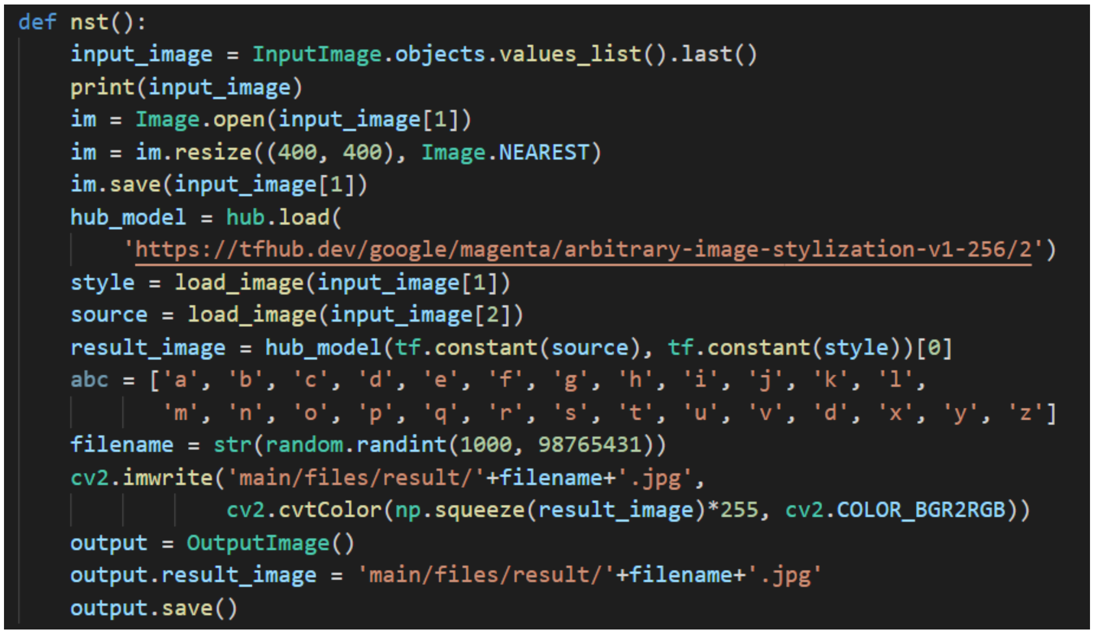
Define visualization functions


```python
def home(request):
    img=OutputImage.objects.values_list('result_image',flat=True)
    Iform= ImageForm
    context={
        'Iform':Iform,
        'img':img,
    }
    if request.method=='POST':
        Iform= ImageForm(request.POST,request.FILES)
        if Iform.is_valid():
            Iform.save()
            nst()
            return redirect('/')
    return render(request, 'home.html',context)
```

Process image and loading function


```python
def load_image(img_path):
    img=tf.io.read_file(img_path)
    img=tf.io.decode_image(img,channels=3)
    img=tf.image.convert_image_dtype(img, tf.float32)
    img=img[tf.newaxis, :]
    return img
```

Import TF Hub module and Demonstrate image stylization.



```
def nst():
    input_image = InputImage.objects.values_list().last()
    print(input_image)
    im = Image.open(input_image[1])
    im = im.resize((400, 400), Image.NEAREST)
    im.save(input_image[1])
    hub_model = hub.load(
        'https://tfhub.dev/google/magenta/arbitrary-image-stylization-v1-256/2')
    style = load_image(input_image[1])
    source = load_image(input_image[2])
    result_image = hub_model(tf.constant(source), tf.constant(style))[0]
    abc = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l',
           'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'd', 'x', 'y', 'z']
    filename = str(random.randint(1000, 98765431))
    cv2.imwrite('main/files/result/'+filename+'.jpg',
                cv2.cvtColor(np.squeeze(result_image)*255, cv2.COLOR_BGR2RGB))
    output = OutputImage()
    output.result_image = 'main/files/result/'+filename+'.jpg'
    output.save()
```

Output

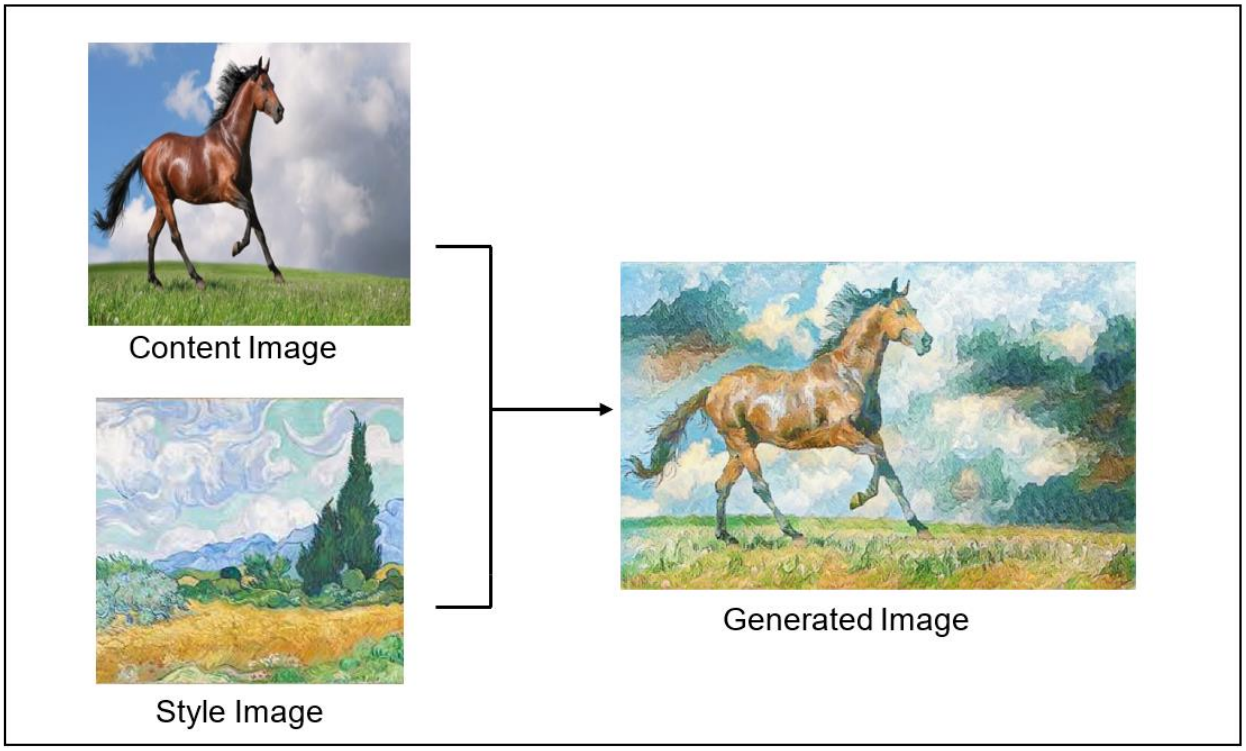


Figure 3

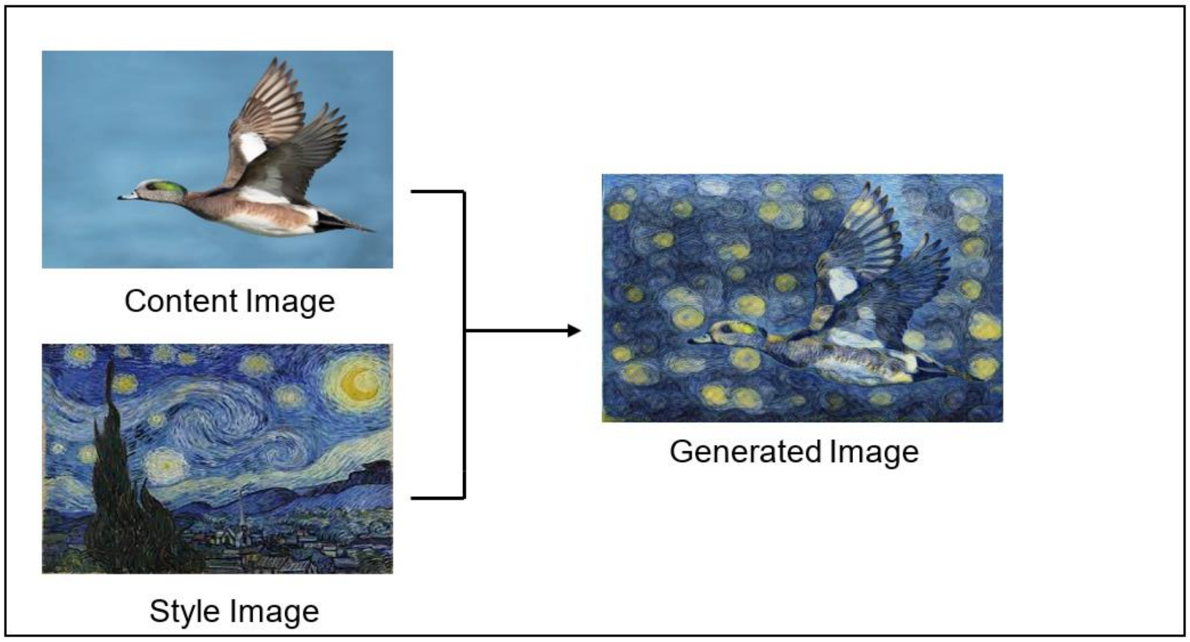


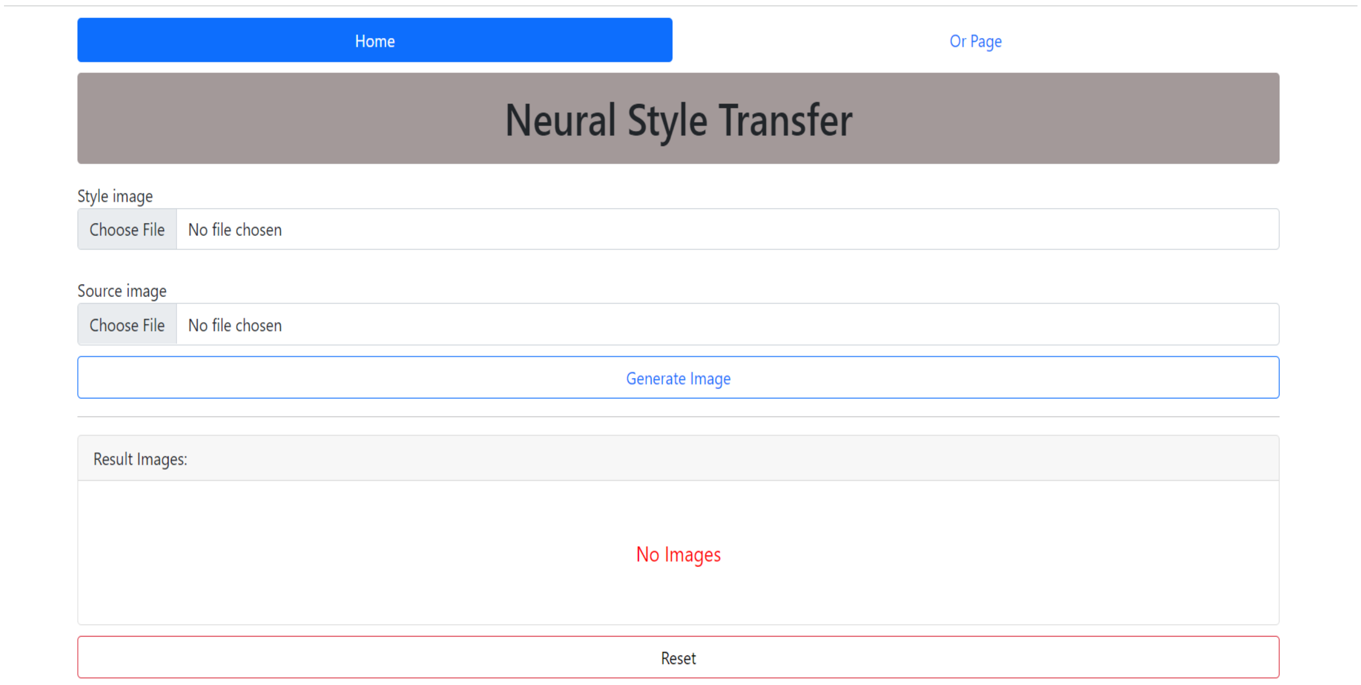
Figure 4

As you can see from the figure above, there are two input images, a content image and a style image, which are used to generate a new image called a stylized image. One thing to note about this image is that it has the same content as the content image and is similar in style to the style image.

We found that convolutional neural network performed very well on this task due to its outstanding performance on analyzing visual imagery. We have acquired decent synthesized images using pretrained CNN model.

Front-End View:

We have designed a web view for our project. Here ,User can Select picture from their devices or they can also choose pic from the given Or Page to make their style image.



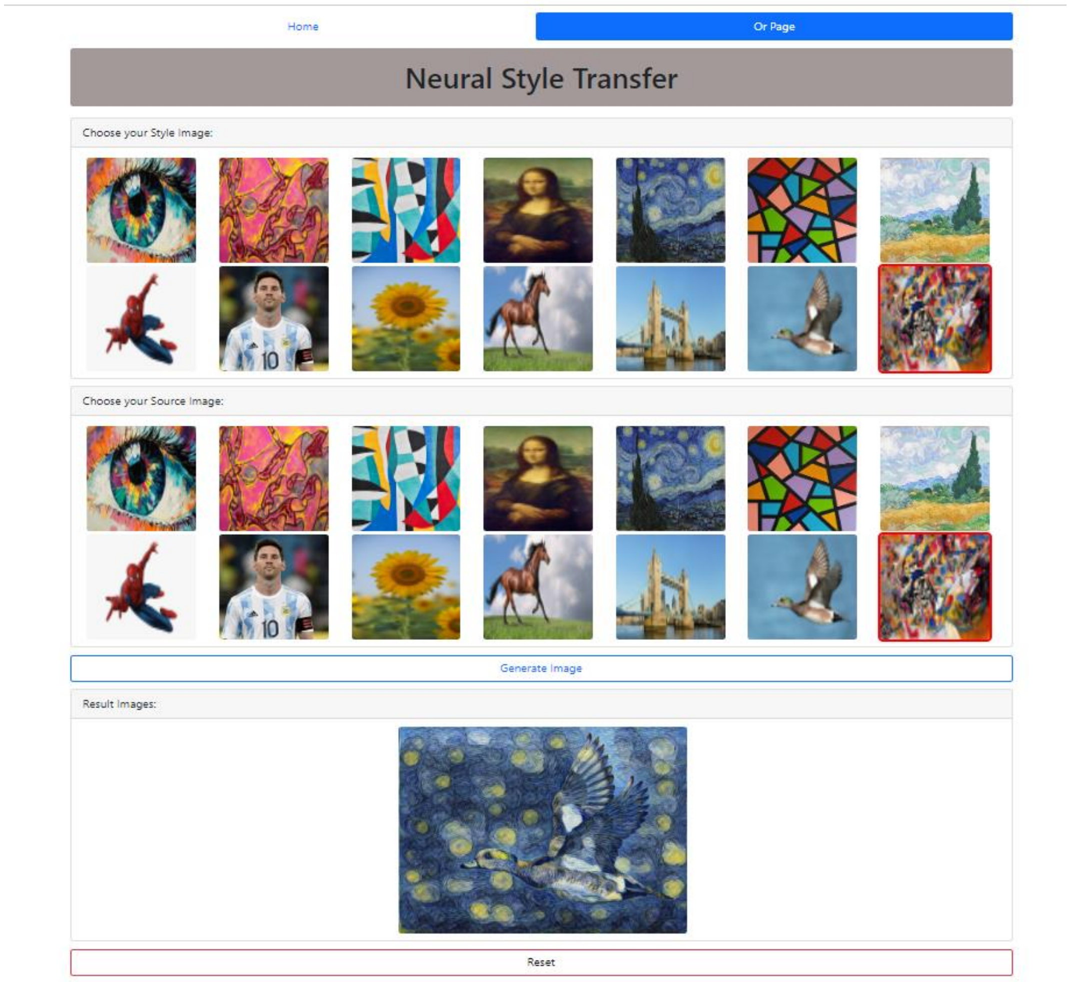

Home
Or Page
Neural Style Transfer
Choose your Style Image:
Choose your Source Image:
Generate Image
Result Images:
Reset

Figure: 5

Contribution

Name	Student ID	Contributions
Rafa Tasnim	2019-1-60-081	Report Writing
Mohsenul Kabir Mithun	2019-3-60-046	Report Writing
Md. Huzaifa	2019-1-60-079	Code
Malyha Bintha Mabud	2019-1-60-128	Code
Abdul Malek Sardar	2019-1-60-087	PPT slide

We complete this project together. Here we implement the neural style transfer algorithm to solve the problem and show the results. We have divided our works. So, we can say that our collaboration is successful.

Learnings

From this project we learn about neural style transfer algorithm, how it takes two image and combine them and create a new artistic image

Conclusion

In this project, we implemented a neural style transfer algorithm to create a new artistic image. Over the past few years, Neural style transfer has spread quickly in the arts, academic research, and commercial applications. There is no other artificial intelligent system with this capability where we combine an image with another style image to create a new style image. This technique is used in Photo and video editing, Commercial art, Gaming, Virtual reality etc.

In future we can achieve arbitrary style transfer. Arbitrary style transfer would also allow us to do a thorough analysis of how artist identification models are affected by stylized images. Another improvement would be to modify the Conditional fast style transfer network to produce the conditional input vector on-the-fly bypassing the style image through a small convolutional neural network.

This system encourages the development of the computer field. It also captures the attention of the other sectors. So, it is important to grow more in neural-style transfer systems.

References

https://www.v7labs.com/blog/neural-style-transfer

https://www.ijert.org/research/generating-artistic-styles-using-neural-style-transfer-IJERTV10IS060440.pdf

https://openaccess.thecvf.com/content_ICCV_2017/papers/Gupta_Characterizing_and_Improving_ICCV_2017_paper.pdf

https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Gatys_Image_Style_Transfer_CVPR_2016_paper.pdf

https://arxiv.org/abs/1508.06576.

https://s3.amazonaws.com/arena-attachments/1382090/b474c4153d87c78ce0729c02df689330.pdf

https://www.semanticscholar.org/paper/A-Learned-Representation-For-Artistic-Style-Dumoulin-Shlens/99542f614d7e4146cad17196e76c997e57a69e4d

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style, 2015.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016.

Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, 1097–1105 (2012)

Taigman, Y., Yang, M., Ranzato, M. & Wolf, L. Deepface: Closing the gap to human-level performance in face verification. In Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, 1701–1708 (IEEE, 2014)

Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. arXiv preprint arXiv:1610.07629, 2016.