



IJRASET

International Journal For Research in
Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** VI **Month of publication:** June 2022

DOI: <https://doi.org/10.22214/ijraset.2022.43990>

www.ijraset.com

Call:  08813907089

E-mail ID: ijraset@gmail.com

Image Style Transfer Using Machine Learning

Dr. Bagam Laxmaiah¹, Jonnadula Narasimharao², Abhigna Maringanti³, SriTeja Govindula⁴, Suresh Dharamsoth⁵

^{1, 2, 3, 4, 5}Dept. Of CSE, CMR Technical Campus

Abstract: The principle of Image style transfer is to define two distance functions, one that describes the content image and the other that describes the style Image. By using these content and Style Images[5][6] as inputs we will be getting the desired output which has the content image merged with style image. The output will be in the graphical model of the content image.

In summary, we'll take the base input image, a content image that we want to match, the style image that we want to match by undergoing the process of convolutional neural network [7] firstly the content image is undergoing the process of content loss and style image as style loss after content loss and style loss it will undergo the process of gram matrix and the final image will be formed.

Keywords: Content Image, Style Image, Content Loss, Style Loss, Convolutional Neural Networks, Gram Matrix Deep Lab Semantic Segmentation.

I. INTRODUCTION

Dealing With the Image Formation, The Image style transfer is an approach to convert one image form into another image form by using convolutional neural network. In this we have to take two images 1) content- image, 2)style-image. The content-image is nothing but it is a main basic image like a dog or human being etc, which exist in real world. The style image is a graphical image like tree graphical image and etc..which is in 3d format. By using Some machine learning functionalities and modules like we need to make content loss and style loss. After these we will be getting the desired output image.

II. LITERATURE SURVEY

In this project, we have taken a content image and the style image we will save the content image[6] and the style images[5] from any website and we need to use them by their location in the system. Here we use many machine learning modules, Algorithms. Convolutional neural network algorithm[7] is used here for formation of the images. OpenCv[8] ,TensorFlow [9] are the functionalities used to process the images. By using these we can get output image in the merged form of both content image and the style image. We need to calculate content loss and the style loss Then undergoes some modules in machine learning like deep lab semantic segmentation[10],open cv[8],tensor flow[9] making the images forming the output desired image. Content loss[11] is nothing but the difference of content of the content image to the content of the output image is the content loss and the content loss is calculated by Euclidean distance and the content loss can be calculated with the content image only.

The content loss is: content of content image –content of output image.

The Euclidean distance is:

The Style loss[12] is defined as the difference between the output image and the style image. Which in turn gives the style loss. Here

$$d(x, y) = \sqrt{\sum_{i=1}^n (y_i - x_i)^2}$$

we cannot compare intermediate features of style loss and content loss so, that is why we introduce gram matrix

In the gram matrix[13] we have to equal the both pixel sizes. Let us consider a content image pixel size is [3280*4250] and the style image pixel size is [4290*5690] so the both pixel sizes is different by using the gram matrix we have to equal the both pixel sizes of content image and style image, and we have to calculate style image by using gram matrix. In gram matrix we have to calculate each layer style information the formulae is:

$$L_{style}(a, x) = \sum_{l \in L} w_l E_l$$

III. PROPOSED SYSTEM

We have proposed Image style transfer using machine learning in this there are two images 1) content-image 2) style-image so, we have to take these two images as an input image to training model. Here in proposed system we will create a data sets of images for target Image. We give an input as any image and using this data sets the image is converted to our target image. Firstly the content image is given and then it undergoes deeplab semantic segmentation which makes the content image as the mask of the content image then the formed mask of the content image and the style image goes image style transfer which in undergoes machine learning modules like opencv and tensorflow. And finally the desired merged form of the image is formed. The below Image shows the Architecture of the image style transfer using machine learning.

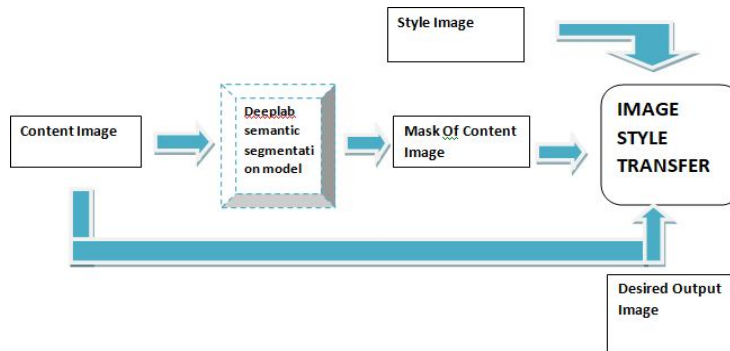
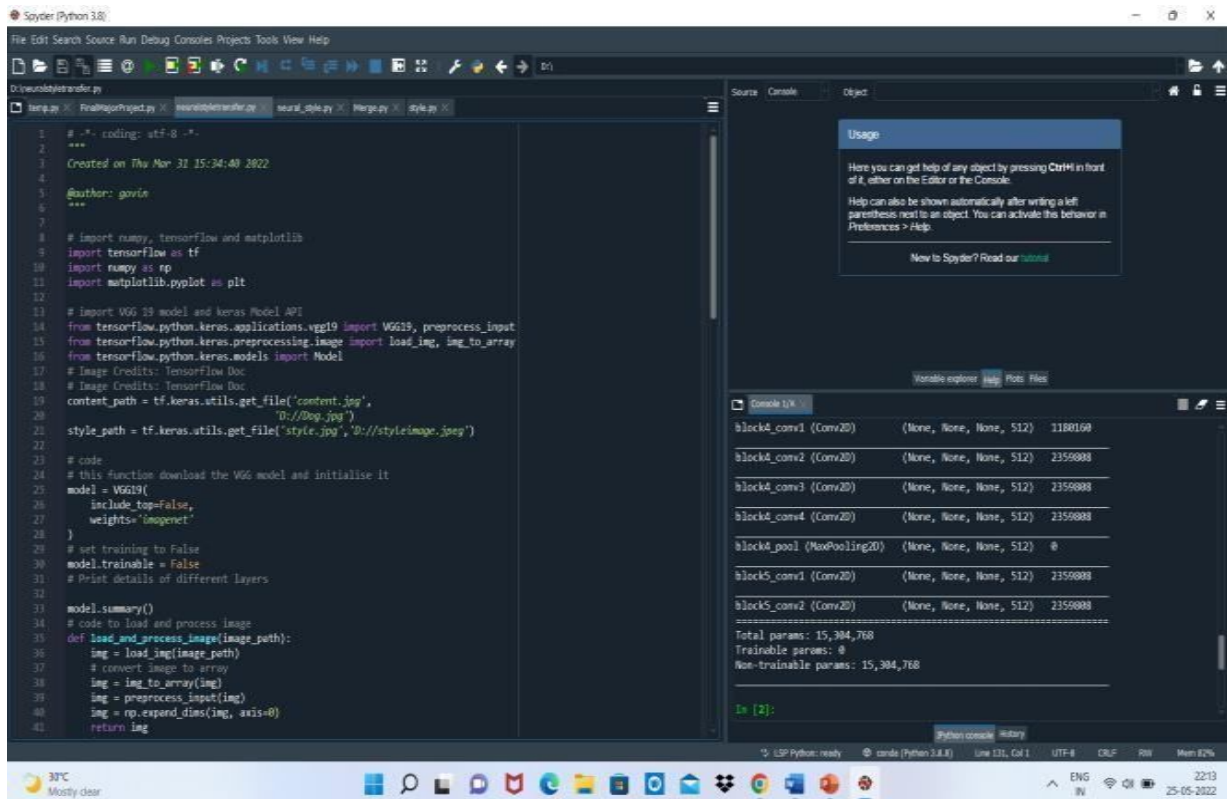


Figure 3.1: Architecture of the Model.

IV. RESULT

Here for a sample we took a dog as a content image and the graphical image as the Input. We need to give input as the location name in the code i.e, where the image is saved in the system. So before the process we need to save the input images which are content image and the style image in the files on the system. While giving input we need to give the location of the image.



```

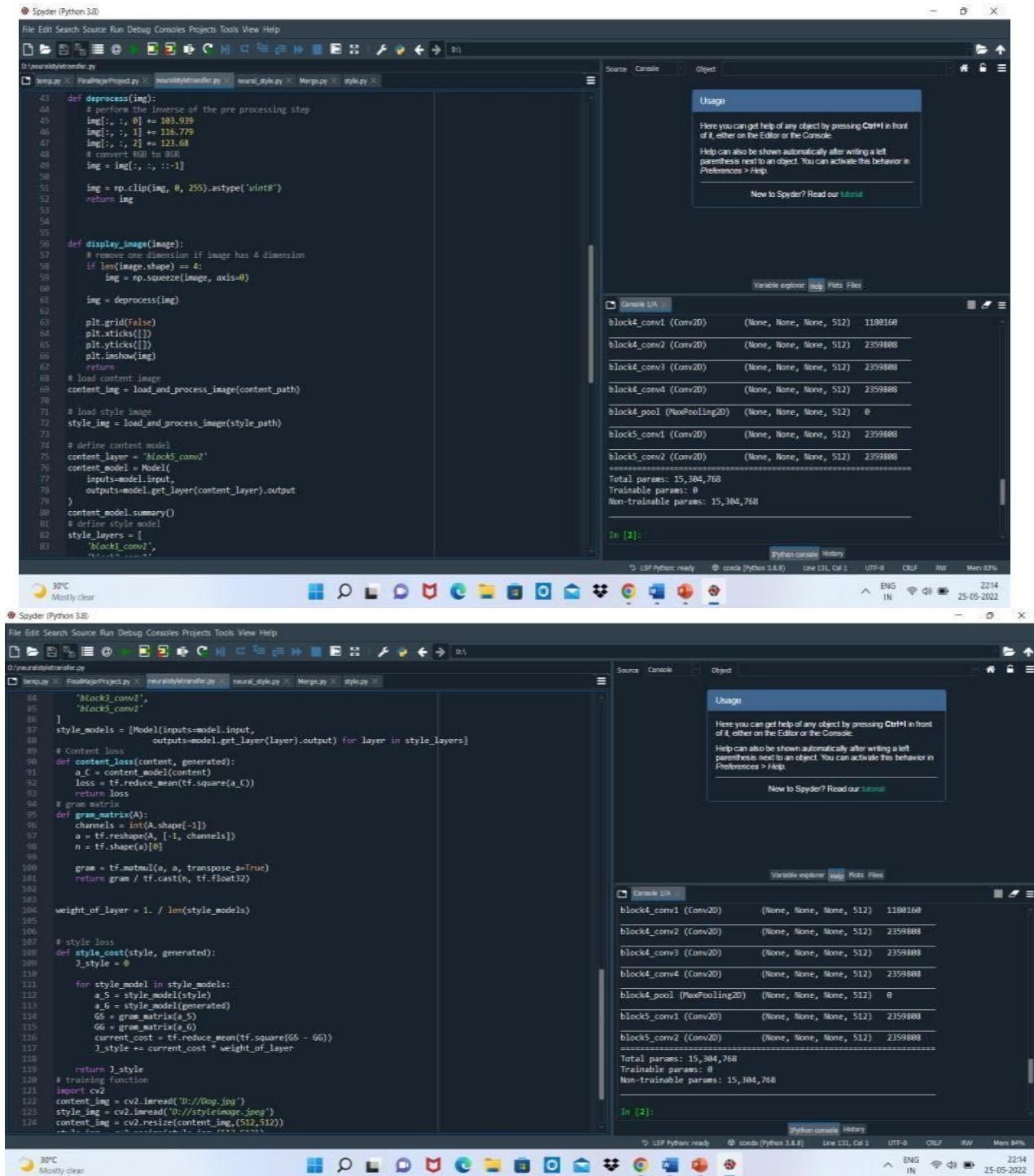
1 # -*- coding: utf-8 -*-
2
3 Created on Thu Mar 31 15:34:40 2022
4
5 @author: gavin
6
7
8 # import numpy, tensorflow and matplotlib
9 import tensorflow as tf
10 import numpy as np
11 import matplotlib.pyplot as plt
12
13 # import VGG 19 model and keras Model API
14 from tensorflow.python.keras.applications.vgg19 import VGG19, preprocess_input
15 from tensorflow.python.keras.preprocessing.image import load_img, img_to_array
16 from tensorflow.python.keras.models import Model
17 # Image Credits: TensorFlow Doc
18 # Image Credits: TensorFlow Doc
19 content_path = tf.keras.utils.get_file('content.jpg',
20                                     'D://dog.jpg')
21 style_path = tf.keras.utils.get_file('style.jpg', 'D://styleimage.jpeg')
22
23 # code
24 # this function download the VGG model and initialise it
25 model = VGG19(
26     include_top=False,
27     weights='imagenet'
28 )
29 # set training to False
30 model.trainable = False
31 # Print details of different layers
32
33 model.summary()
34 # code to load and process image
35 def load_and_process_image(image_path):
36     img = load_img(image_path)
37     # convert image to array
38     img = img_to_array(img)
39     img = preprocess_input(img)
40     img = np.expand_dims(img, axis=0)
41     return img
  
```

Console Output:

```

Usage
Here you can get help of any object by pressing Ctrl+H in front of it, either on the Editor or the Console.
Help can also be shown automatically after writing a left parenthesis next to an object. You can activate this behavior in Preferences > Help.
New to Spyder? Read our tutorial.

Variable explorer
block4_conv1 (Conv2D) (None, None, None, 512) 1188160
block4_conv2 (Conv2D) (None, None, None, 512) 2359808
block4_conv3 (Conv2D) (None, None, None, 512) 2359808
block4_conv4 (Conv2D) (None, None, None, 512) 2359808
block4_pool (MaxPooling2D) (None, None, None, 512) 0
block5_conv1 (Conv2D) (None, None, None, 512) 2359808
block5_conv2 (Conv2D) (None, None, None, 512) 2359808
Total params: 15,384,768
Trainable params: 0
Non-trainable params: 15,384,768
  
```

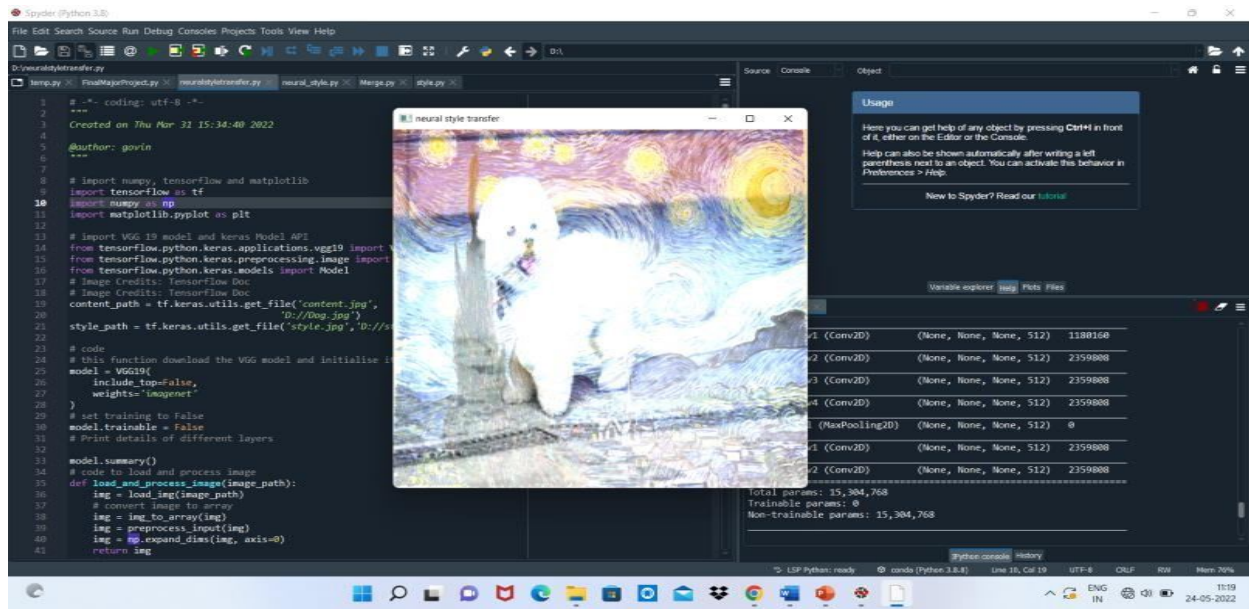


Figure 4.4-Result of the final image

VI. CONCLUSION

By using this principle of Image Style Transfer using Machine Learning we can get our desired image without recreation by using an algorithm called CNN, We use some of the modules of machine learning like opencv and tensorflow and we will get the image with high Accuracy and it takes less time because there is no recreation of the images, only conversion of images directly.

REFERENCES

- [1] N.Ashikhmin.Fasttexturetransfer.IEEECComputerGraph- ics and Applications, 23(4):38–43, July2003.
- [2] M. Berning, K. M. Boergens, and M. Helmstaedter.SegEM: Efficient Image Analysis for High- Resolution Connec-tomics. Neuron, 87(6):1193–1206, Sept. 2015.
- [3] C.F.Cadiou,H.Hong,D.L.K. Yamins,N.Pinto,D.Ardila, E. A. Solomon, N. J. Majaj, and J. J. DiCarlo. Deep Neu- [ralNetworksRivaltheRepresentationofPrimateITCor- tex for Core Visual Object Recognition.PLoSComputBiol, 10(12):e1003963, Dec. 2014.
- [4] L.-C. Chen, G. Papandreou, I.Kokkinos, K.Murphy, and A. L. Yuille. Semantic Image Segmentationwith Deep Convolutional Nets and Fully Connected CRFs. arXiv:1412.7062 [cs], Dec. 2014. arXiv: 1412.7062.
- [5] Different Style images from the web.
- [6] Different content images fromtheweb
- [7] CNN-Convolutional neural Network Algorithm <https://machinelearningmastery.com/convolutional-layers-for-deep-learning- neural-networks/>
- [8] Opencv Module in Machine Learning <https://pypi.org/project/opencv-python/>
- [9] Tensorflowmodule in machinelearning https://www.tensorflow.org/hub/api_docs/python/hub/Module
- [10] Deep Lab Sematic Segmentation forcontent Image <https://towardsdatascience.com/the-evolution-of-deeplab-for-semantic- segmentation-95082b025571>
- [11] Content loss and the style loss for theimages. <https://nnart.org/what-is-content-loss/> <https://www.coursera.org/lecture/generative- deep-learning-with-tensorflow/style-loss-19bTa>
- [12] Image processing Process <https://towardsdatascience.com/convolution- neural- network-for-image-processing-using-keras-dc3429056306>
- [13] Calculation of gram matrix <https://towardsdatascience.com/neural-networks-intuitions-2-dot-product-gram- matrix-and-neural-style-transfer-5d39653e7916>
- [14] Keras in python <https://www.javatpoint.com/keras>



10.22214/IJRASET



45.98



IMPACT FACTOR:
7.129



IMPACT FACTOR:
7.429



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089  (24*7 Support on Whatsapp)