Neural Style Transfer

Under Guidance : Prof. Madhav Vaidya Sir.

Name : Ashish Anil Waykar
Reg. No. : 2021BIT507

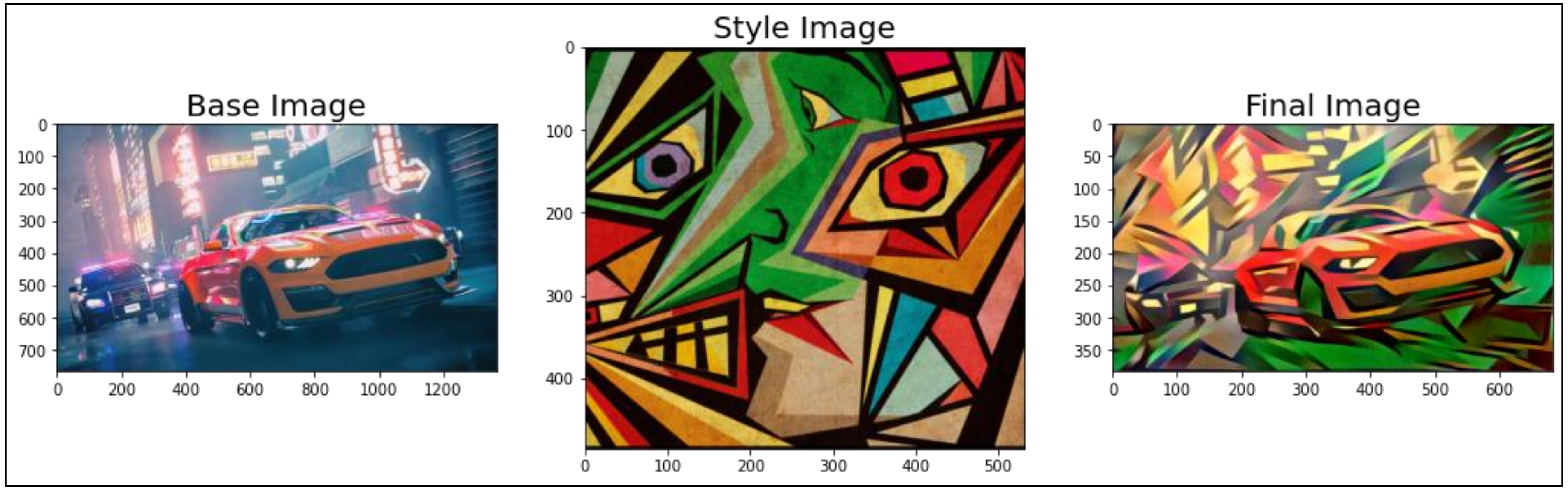
TABLE OF CONTENTS

Neural Style Transfer Introduction

- In the world of fine art, especially in painting, people have become skilled at creating unique visual experiences by blending the content and style of an image in a sophisticated way. Strikingly, the complex rules guiding this artistic process are still not fully understood, and no computer system has matched the abilities of human artists in this regard.

- Yet, in important aspects of visual understanding, such as identifying objects and faces, we've seen impressive progress recently, reaching levels close to human performance. These advancements come from a type of vision models inspired by biology, called Deep Neural Networks.

CNN Neural Style Transfer



We introduce a computer system based on a Deep Neural Network, capable of producing artistic images with exceptionally high visual quality. This system uses neural representations to separate and then combine the content and style elements from any given images. Essentially, it offers a computerized method for creating artistic images.

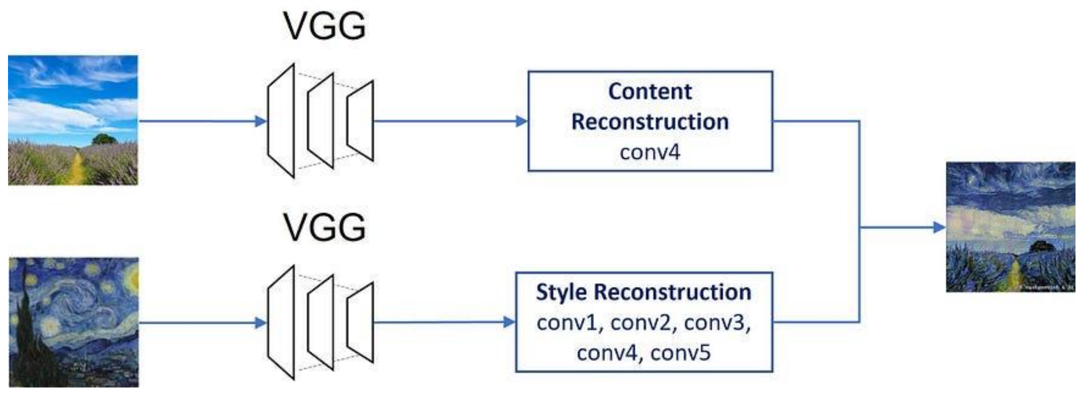
CNN!

The Eyes of Machine !

Process Image & make machines learn from images to process & evolve the technology

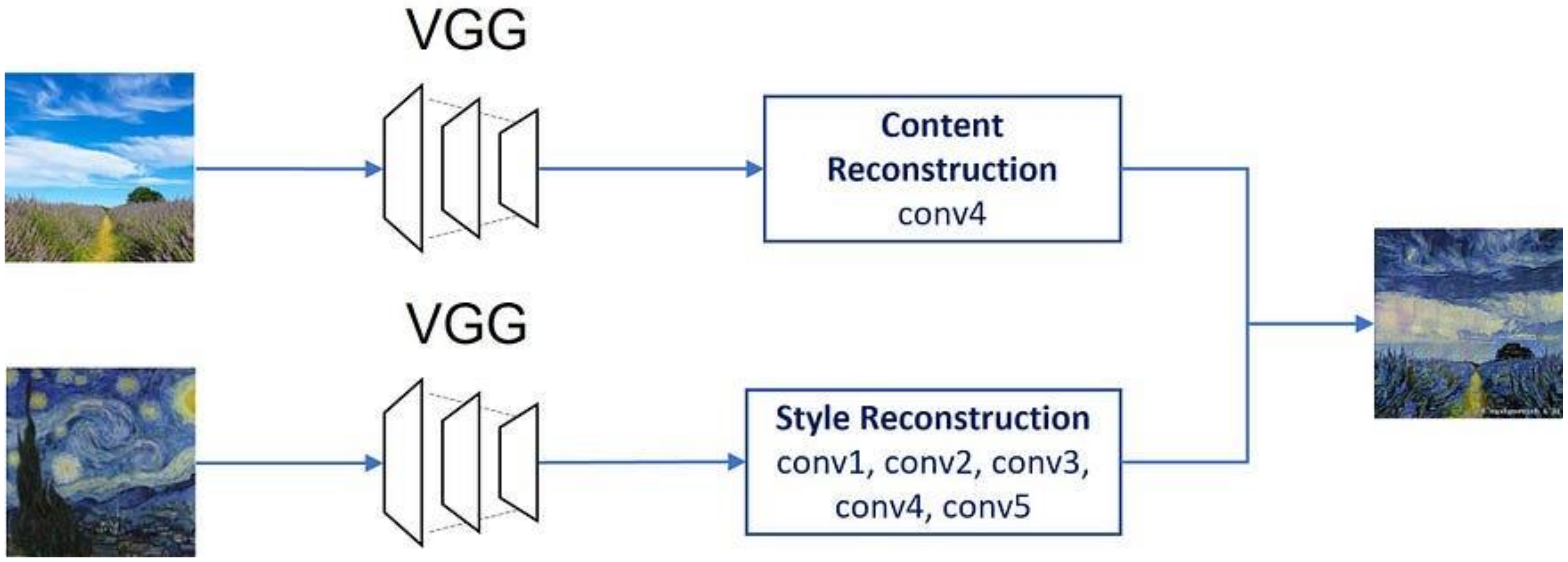
01

Image Reconstruction


VGG
VGG
Content Reconstruction conv4
Style Reconstruction conv1, conv2, conv3, conv4, conv5

We can virtualize the information on different processing stages of layers in CNN with re-building the input image from only knowing the network's responses in several layers.

We rebuild the input image from layers 'conv1 1' (We can virtualize the information on different processing stages of layers in CNN with re-building the input image from only knowing the network's responses in several layer), 'conv2 1' (b), 'conv3 1' (c), 'conv4 1' (d) and 'conv5 1' (e) of the original pretrained VGG19-Network.


VGG
VGG
Content Reconstruction conv4
Style Reconstruction conv1, conv2, conv3, conv4, conv5

02 Style Reconstruction

- On top of the original CNN layers representations we built a new feature space layers that captures the style of an input image. The style representation computes correlations between the different features units in different layers of the CNN.
- Where presentations 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)).
- We reconstruct the style of the input image from style This creates images that match the style of a input image on an highly increasing scale while discarding information of the default arrangement of the scene.

VGG19 Architecture

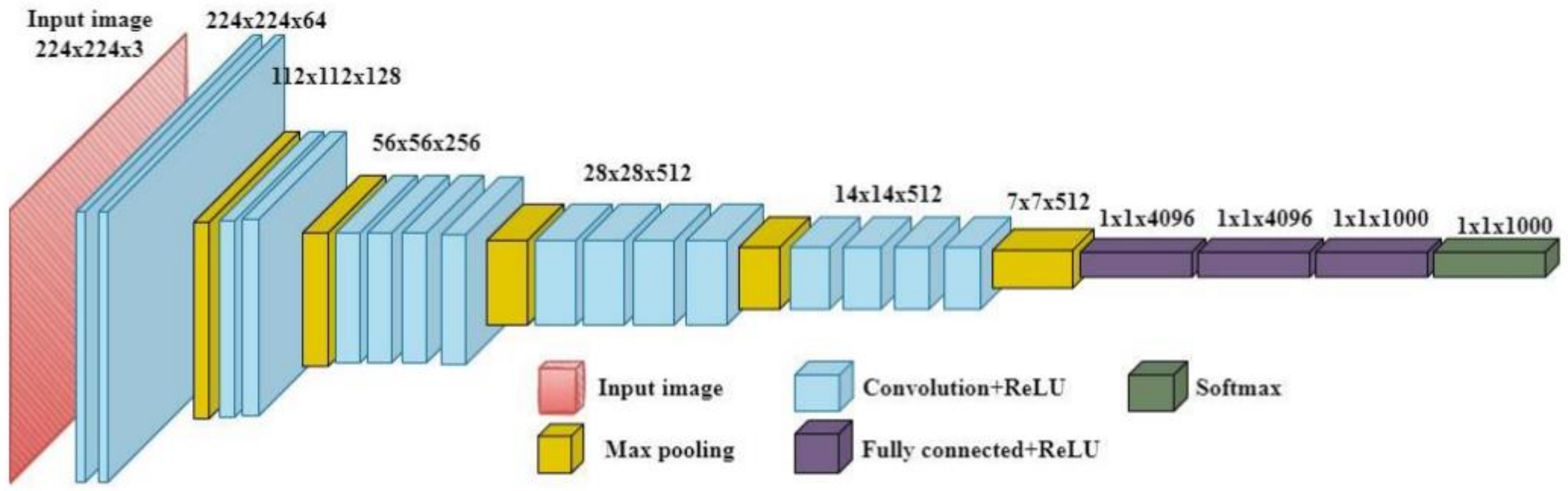
03

VGG19

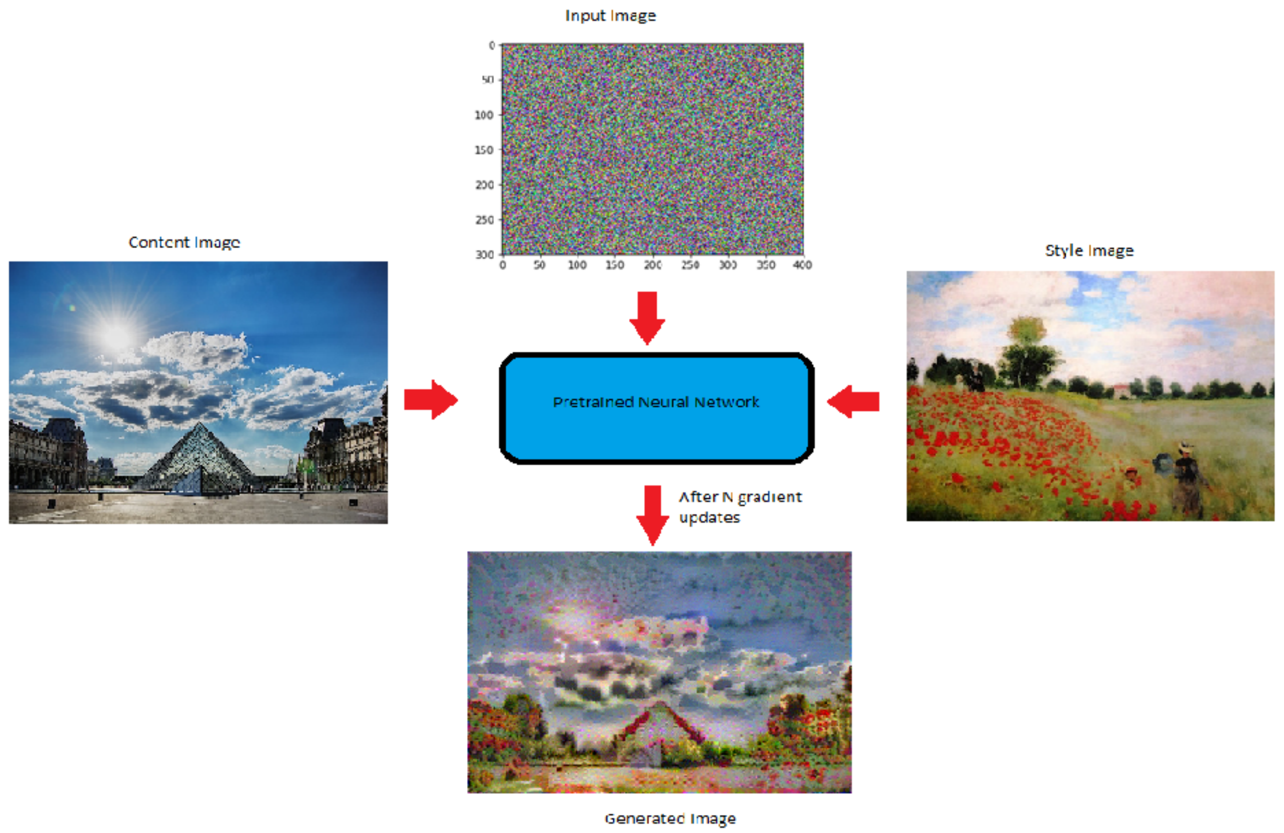
VGG19 network is used for Neural Style transfer. VGG-19 is a convolutional neural network that is **trained on more than a 14 million images** from the **ImageNet database**. The network is **19 layers deep** and trained on millions of images. Because of which it is able to detect high-level features in an image.

VGG19

Architecture as per layer to layer



VGG19



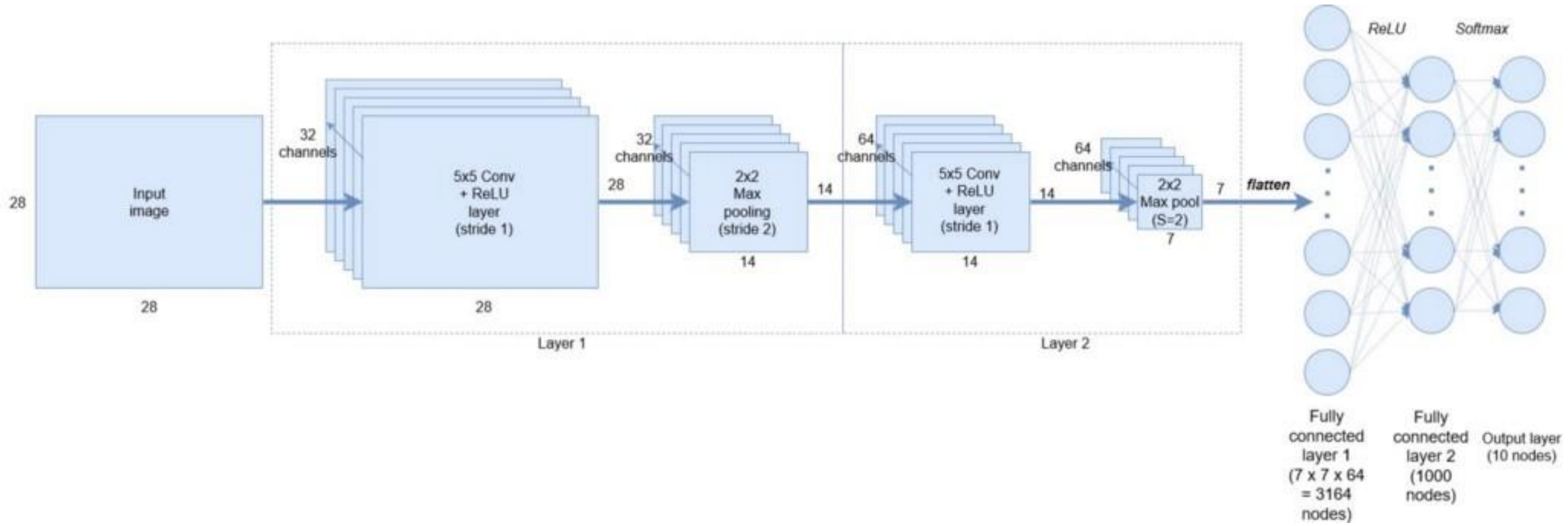
- Now, this ‘encoding nature’ of CNN’s is the key in Neural Style Transfer. Firstly, we initialize a noisy image, which is going to be our output image(G). We then calculate how similar is this image to the content and style image at a particular layer in the network(VGG network). Since we want that our output image(G) should have the content of the content image(C) and style of style image(S) we calculate the loss of generated image(G) with respective to the respective content(C) and style(S) image.

- Similar datasets **Resnet18** , **AlexNet** actually **VGG19 pretrained** will use the **ImageNet dataset** where it has been **trained over 14 million images** from **ImageNet**.

Methods

04

- **Content Loss**
- **Style Loss**
- **Total Loss**



28
Input image
28
32 channels
5x5 Conv + ReLU layer (stride 1)
28
28
32 channels
2x2 Max pooling (stride 2)
14
14
Layer 1
64 channels
5x5 Conv + ReLU layer (stride 1)
14
14
64 channels
2x2 Max pool (S=2)
7
7
flatten
Layer 2
ReLU
Softmax
Fully connected layer 1 (7 x 7 x 64 = 3164 nodes)
Fully connected layer 2 (1000 nodes)
Output layer (10 nodes)

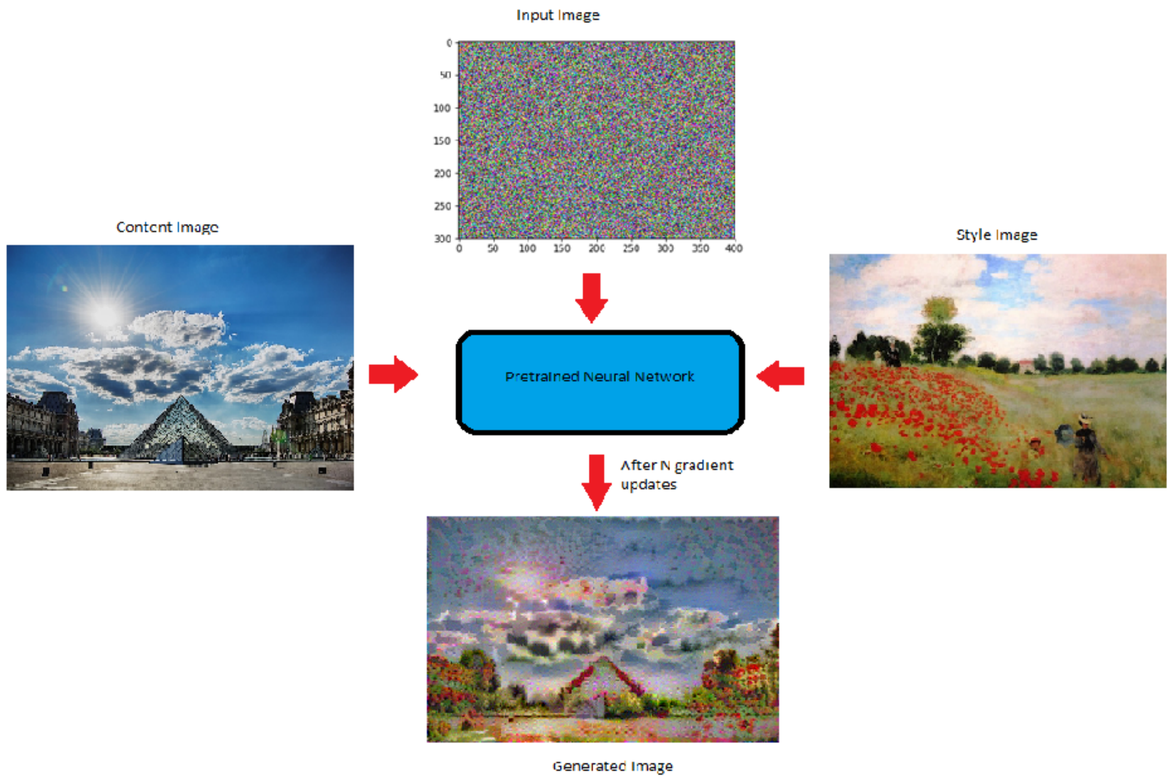
Layered CNN Architecture for pretrained VGG19 with respect to networking in Neural Networks

Now, at Layer 1 using 32 filters the network may capture simple patterns, say a straight line or a horizontal line which may not make sense to us but is of immense importance to the network, and slowly as we move down to Layer 2 which has 64 filters, the network starts to capture more and more complex features it might be a face of a dog or wheel of a car. This capturing of different simple and complex features is called feature representation.

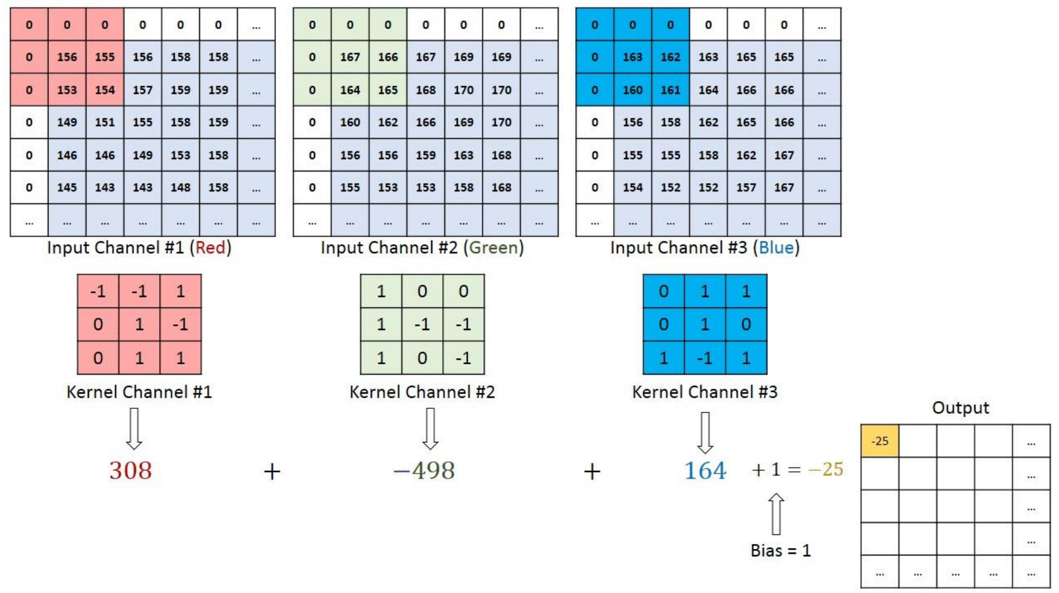
Content Loss:

- Calculating content loss means how similar is the randomly generated noisy image(G) to the content image(C).In order to calculate content loss :
- Assume that we choose a hidden layer (L) in a pre-trained network(VGG network) to compute the loss. Therefore, let P and F be the original image and the image that is generated. And, F[l] and P[l] be feature representation of the respective images in layer L. Now, the content loss is defined as follows:

$$\mathcal{L}_{content}(\vec{p}, \vec{x}, l) = \frac{1}{2} \sum_{i,j} \left(F_{ij}^{l} - P_{ij}^{l} \right)^2$$



How we capture style of an image ?



MaxPooling with Respect to Frames & RGB Values

In-order to calculate a correlation between different filters or channels we calculate the **dot-product between the vectors of the activations of the two filters**. The matrix thus obtained is called **Gram Matrix**.

Style Loss:

Before calculating style loss, let’s see what is the meaning of “style of a image” or how we capture style of an image.

Gram Matrix of Style Image(S):

$$Jstyle = \sum_i^H \sum_j^W \left(Aijk^{[l][S]} - Aijk'^{[l][S]} \right)$$

Here k and k’ represents different filters or channels of the layer L. Let’s call this Gkk’[l][S].

Gram Matrix for Generated Image(G):
Here k and k’ represents different filters or channels of the layer L.
Let’s call this Gkk’[l][G].

$$Jstyle.generated = \sum_i^H \sum_j^W \left(Aijk^{[l][G]} - Aijk'^{[l][G]} \right)$$

Now, we are in the position to define Style loss:
Cost function between Style and Generated Image is the square of difference between the Gram Matrix of the style Image
with the Gram Matrix of generated Image.

$$J(S,G) = \frac{1}{\left(2\, H^l\, W^l\, C^l\right)^2} \sum_k \sum_{k'} \left(Gkk'^{[l][S]} - Gkk'^{[l][G]} \right)$$

Total Loss Function :

The total loss function is the sum of the cost of the content and the style image.

Mathematically, it can be expressed as :

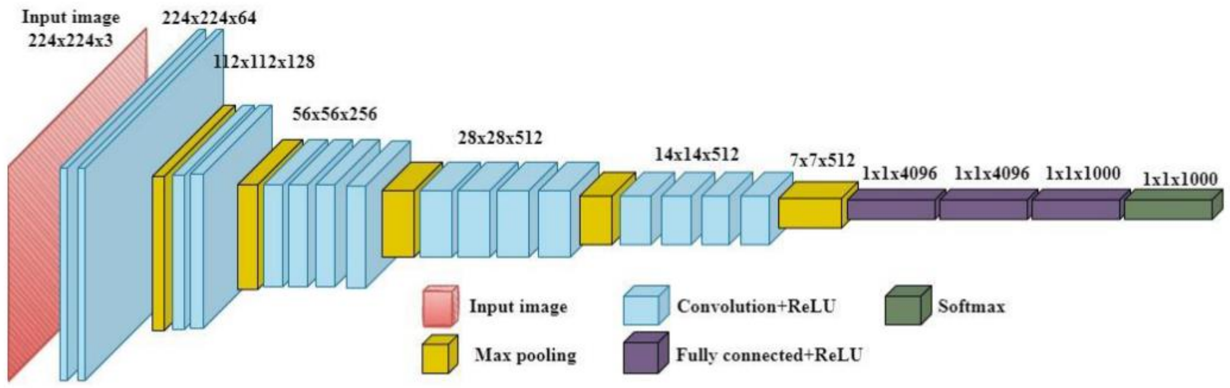
$$\boxed{\mathcal{L}_{total}(\vec{p},\vec{a},\vec{x}) = \alpha\mathcal{L}_{content}(\vec{p},\vec{x}) + \beta\mathcal{L}_{style}(\vec{a},\vec{x})}$$

You may have noticed Alpha and beta in the above equation. They are used for weighing Content and Style cost respectively. In general, they define the weightage of each cost in the Generated output image.

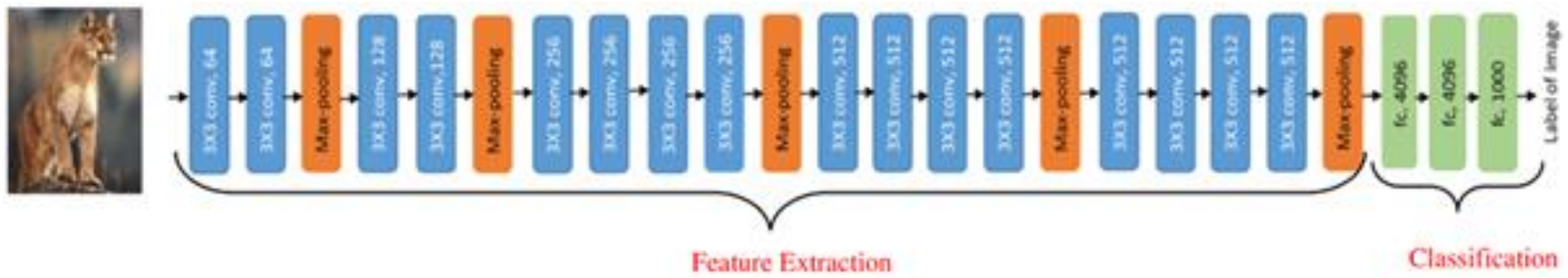
Once the loss is calculated, then this loss can be minimized using backpropagation which in turn will optimize our randomly generated image into a meaningful piece of art.

Layer By Layer Approach

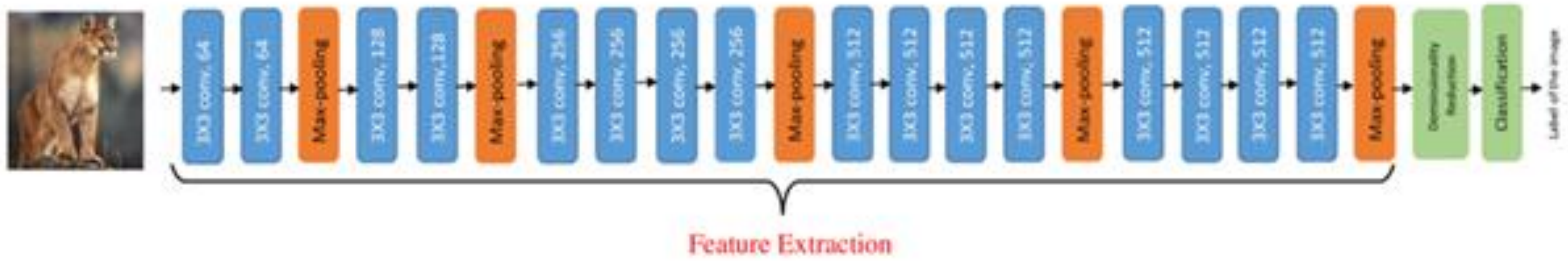
05



Convolutional Blocks and Pooling Layers

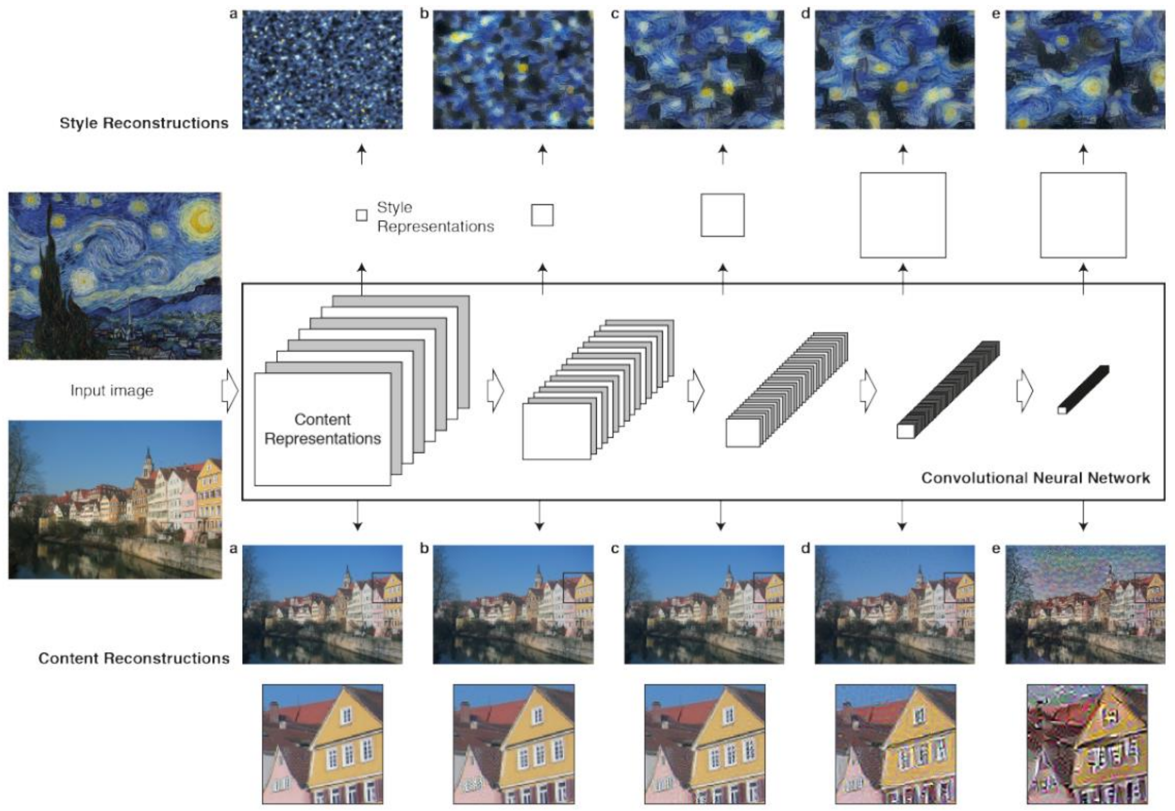


(a) Architecture of VGG19 model



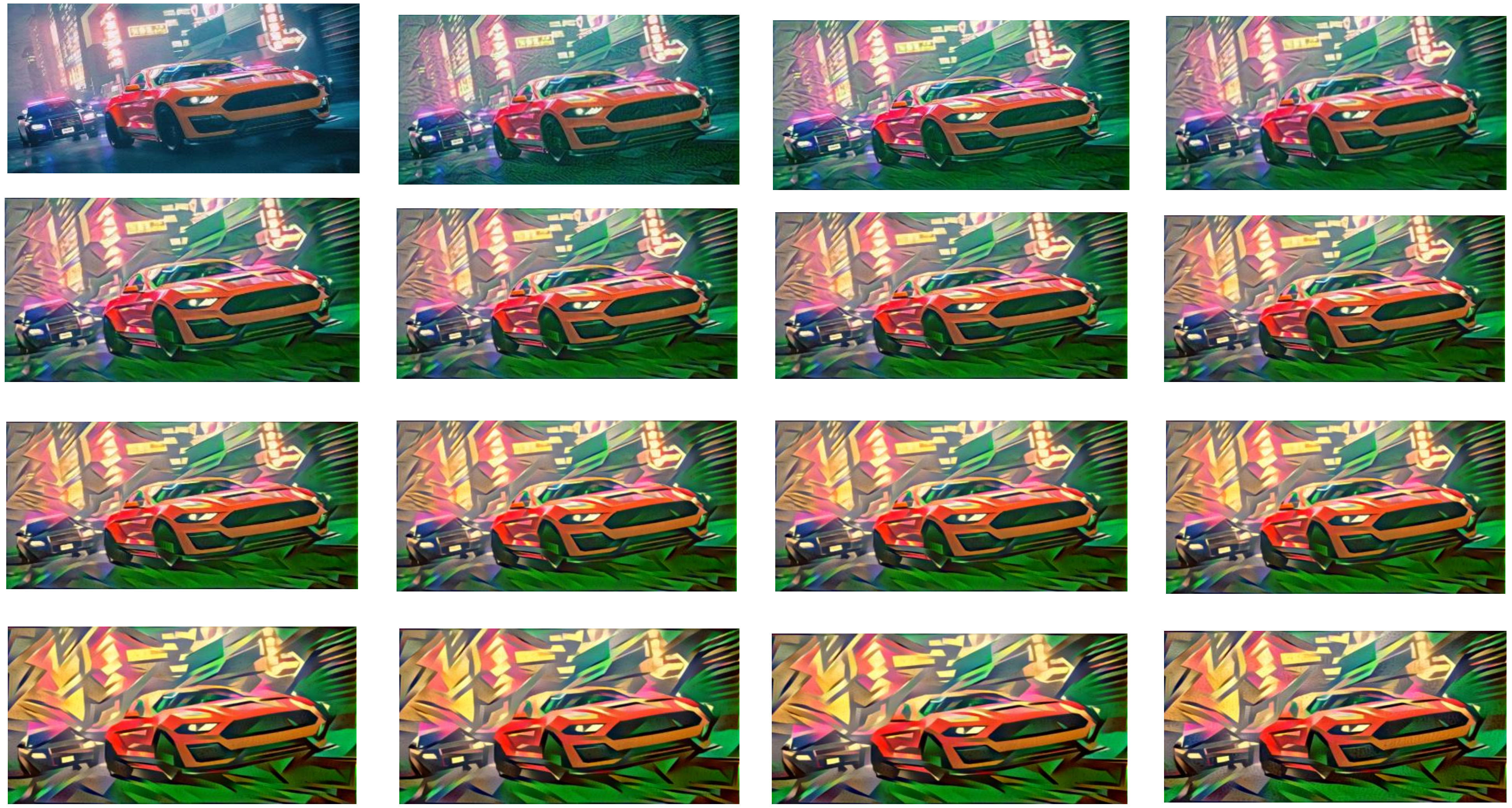
(b) Ensemble of deep feature extraction using VGG19 model and machine learning classification

Representation of each layer with respect to style transfer

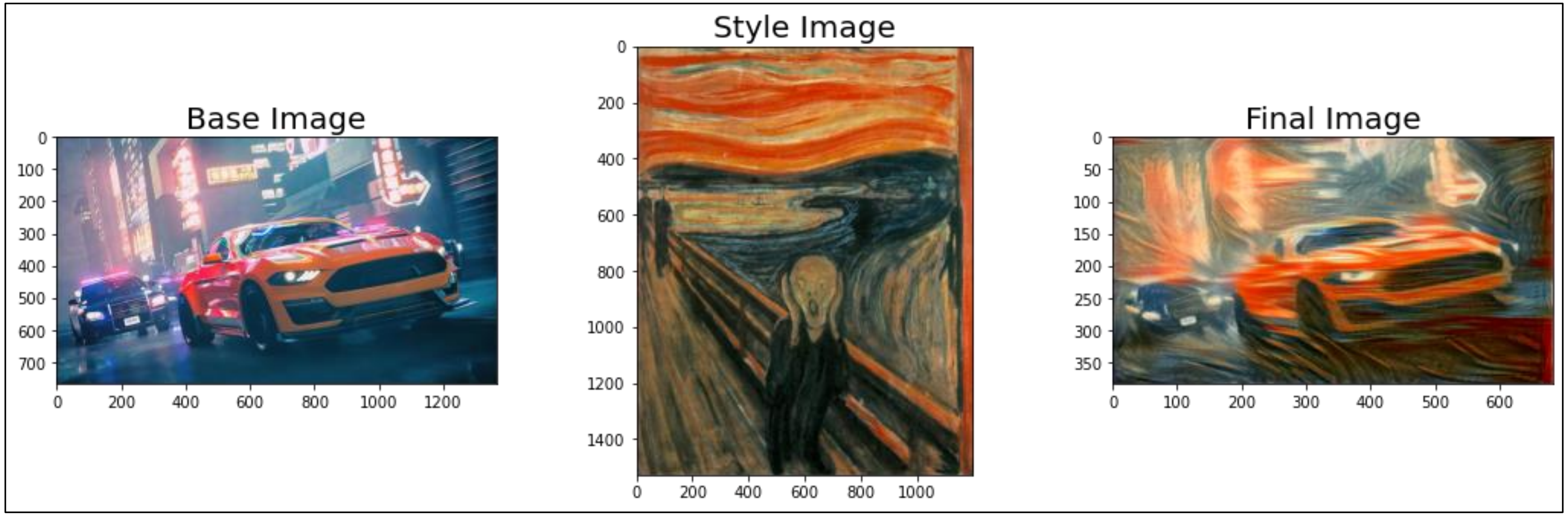


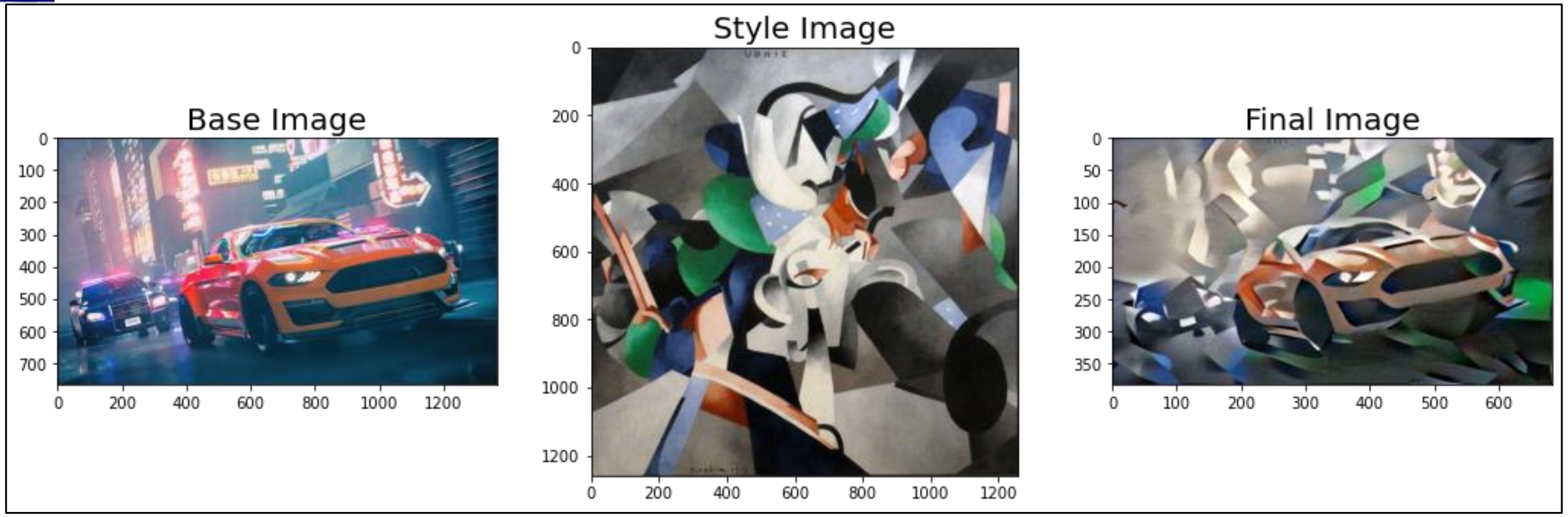
The image is represented by collection of filtered images with each processing stage in CNN. While number of multiple filters are increasing with processing. The size of the above filtered images are reduced by some down sampling mechanisms **MAX POOLING** leading the total number of units per layer network

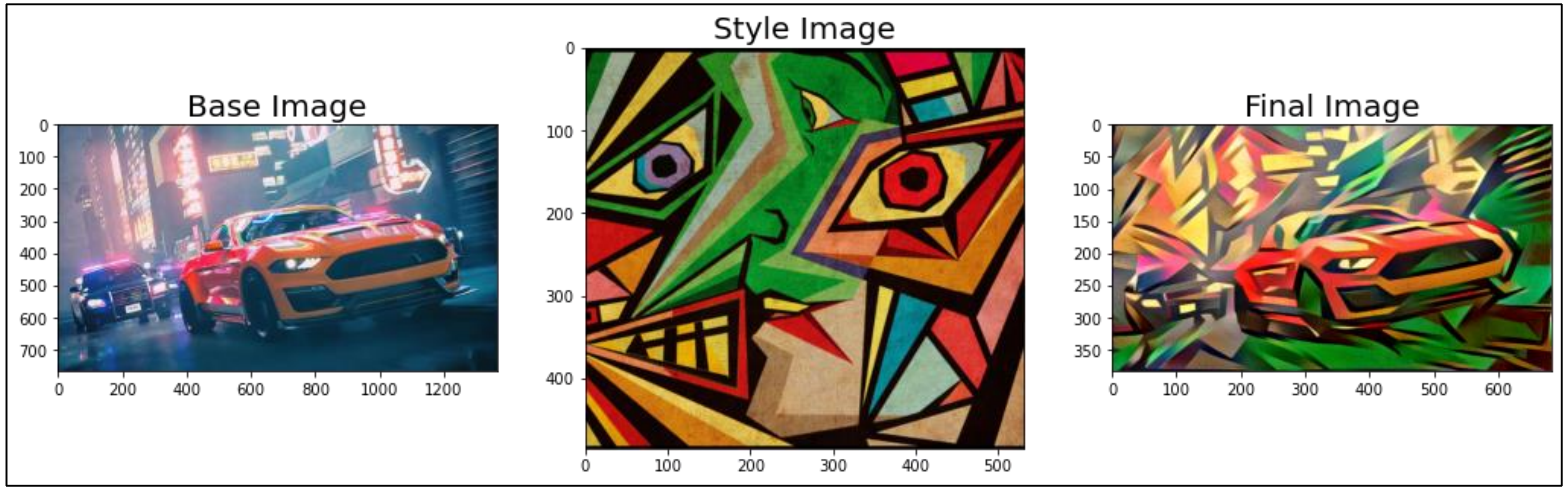
Layer By Layer Epoch Process



The Scream


Base Image
0
100
200
300
400
500
600
700
0
200
400
600
800
1000
1200
Style Image
0
200
400
600
800
1000
1200
1400
0
200
400
600
800
1000
Final Image
0
50
100
150
200
250
300
350
0
100
200
300
400
500
600


Base Image
Style Image
Final Image


Base Image
Style Image
Final Image

References

Title : Understanding Convolutional Neural Network (CNN): A Complete Guide by LearnOpenCV

- https://learnopencv.com/understanding-convolutional-neural-networks-cnn/

Title : Convolutional Neural Networks by **Rijul Vohra**

- https://medium.datadriveninvestor.com/convolutional-neural-networks-3b241a5da51e

Title : Interactive Artistic Multi-style Transfer By **International Journal of Computational Intelligence Systems**

- https://link.springer.com/article/10.1007/s44196-021-00021-0

Title : Recognizing Image Style by **Sergey Karayev**, **Matthew Trentacoste**, **Helen Han**, **Aseem Agarwala**, **Trevor Darrell**, **Aaron Hertzmann** & **Holger Winnemoeller**

- http://arxiv.org/abs/1311.3715

Title : A Parametric Texture Model Based on Joint Statistics of Complex A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients

- https://link.springer.com/article/10.1023/A:1026553619983

Title : Understanding Deep Image Representations by Inverting Them from Aravindh Mahendran, Andrea Vedaldi

- http://arxiv.org/abs/1412.0035

Title : Texture Synthesis Using Convolutional Neural Networks by Leon A. Gatys, Alexander S. Ecker, Matthias Bethge

- http://arxiv.org/abs/1505.07376

Title: Youtube Link to **Andrew Ng** videos

- https://youtu.be/R39tWYYKNcI (Part 1)
- https://youtu.be/ChoV5h7tw5A (Part 2)
- https://youtu.be/xY-DMAJpIP4 (Part 3)
- https://youtu.be/b1I5X3UfEYI (Part 4)

Thank You All !

Your feedback will be appreciated ...