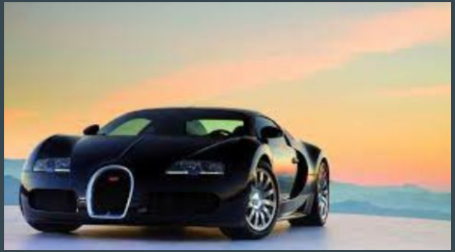
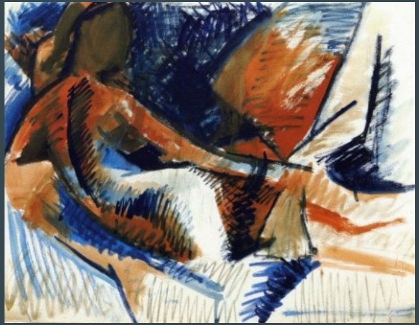
Application of CNN Neural Style Transfer

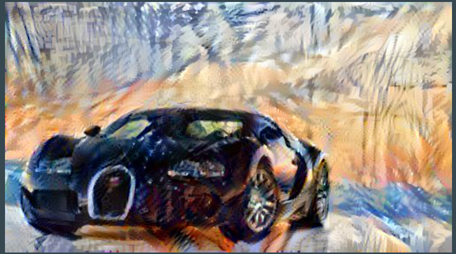
Saksham Gupta
Somesh Verma
Pratham Kailasiya
Archit Gangwal

Overview

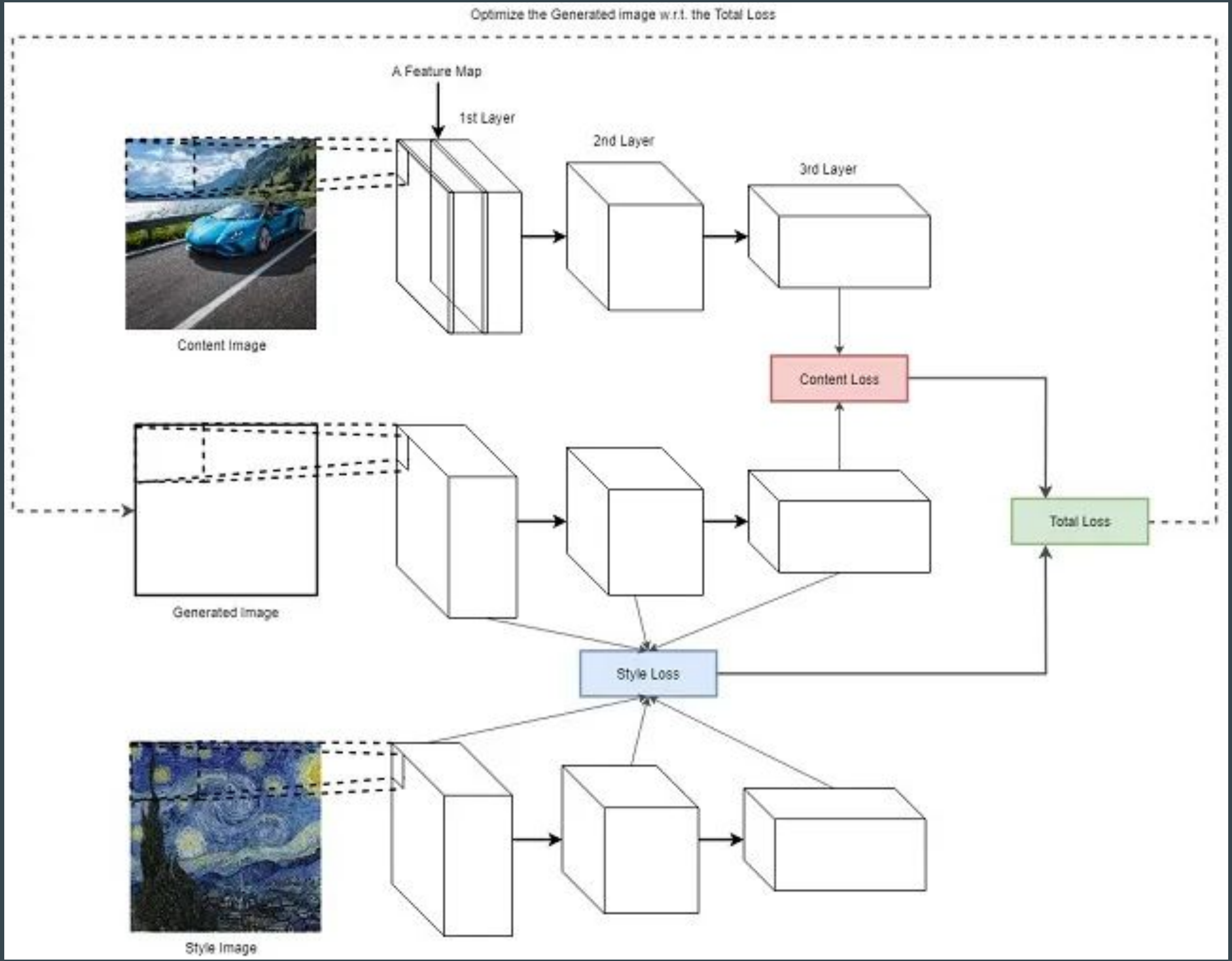
Neural style transfer is a technique that blends two images, a content image and a style reference image, to create a new image with the same content as the original but with the style of the reference image.


Content Image


Style Image


Output Image

Architecture



The Algorithm

STEP 1	● Preprocessing image and making it fit ● Plotting to check if it's right
STEP 2	● Importing vgg model ● Computing total loss(style+content)
STEP 3	● Normalization (AdaIN) ● Running gradient descent
STEP 4	● Improving the Image (Total Variation Loss) ● Running gradient descent

Preprocessing Image

- The content and style images are first taken as input.
- The path is used to preprocess image in form of tensors.
- To verify that it’s been preprocessed correctly, we see it through by plotting it with the help of matplotlib .

VGG19

- Importing the vgg model in tensorflow
- Finding values of hidden layers for loss calculation
- Its feature extraction property will be used to generate the image

Loss Function

STYLE LOSS

Compute gram matrix, then use mean squared error of gram matrices of both images(style and generated)

CONTENT LOSS

Simply finding errors between final layers of content and generated Image

TOTAL LOSS

After multiplying weights to each style and content loss and adding them gives total loss which needs to be minimized

Regularisation

- After training, we saw that there were distortions in image.
- This is due to the noises caused due to high frequency components of image
- Calculated total variation loss through high pass filters
- Running gradient descent with extra added variation loss in total loss

Integration with Application: Dev Part

Techstack

The website takes content and style image as input and displays the output image.

Frontend + Backend :

- HTML, CSS styling and JS is used for frontend using template feature from Django.
- We chose backend framework, Django, because the model being in Python programming language.

The input images as such were used to run the model and generate the output image, which can be downloaded by the user.

But, with image_path we had some difficulty integrating website with the model, so we were unable to implement it.

The site successfully worked with input and feeding to model but gave some TypeErrors at nst.py code deployment.

A Big Hurdle

We initially thought this model requires training data to train model, instead there is no model, it’s just we are exploiting the property of feature extraction of a pre-trained model and minimizing a loss function and updating image pixels.

Thank You