

## NEURAL STYLE TRANSFER



## OVERVIEW

Neural style Transfer is a technique in AI and computer vision that enables you combine the artistic styles of any two images, applying the style of one image(called the style image) on another(called the content image).

## **AIM OF THE PROJECT**

It has to ensure that the generated image maintains the overall structure and content of the content image. It includes shapes, objects and spatial arrangement. The artistic style of the style image has to be transferred to the target image. It involves capturing the textures, colors, patterns and other characteristics of the image.

## **MY APPROACH**

I made the NST code in Python and developed the solution using VSCode. The first step was installing the necessary libraries-

1. Torch
2. Torchvision
3. Streamlit
4. Pillow

Since this was my first project, with minimal experience is working on Python, I chose to do the project on Pytorch, which required installing Torch and Torchvision. I created a very simple web application using streamlit. I installed PIL(Python Imaging Library) for image handling and processing.

Then, the VGG19 model was loaded, which is pretrained on the ImageNet dataset. It helps extract the features from the images.

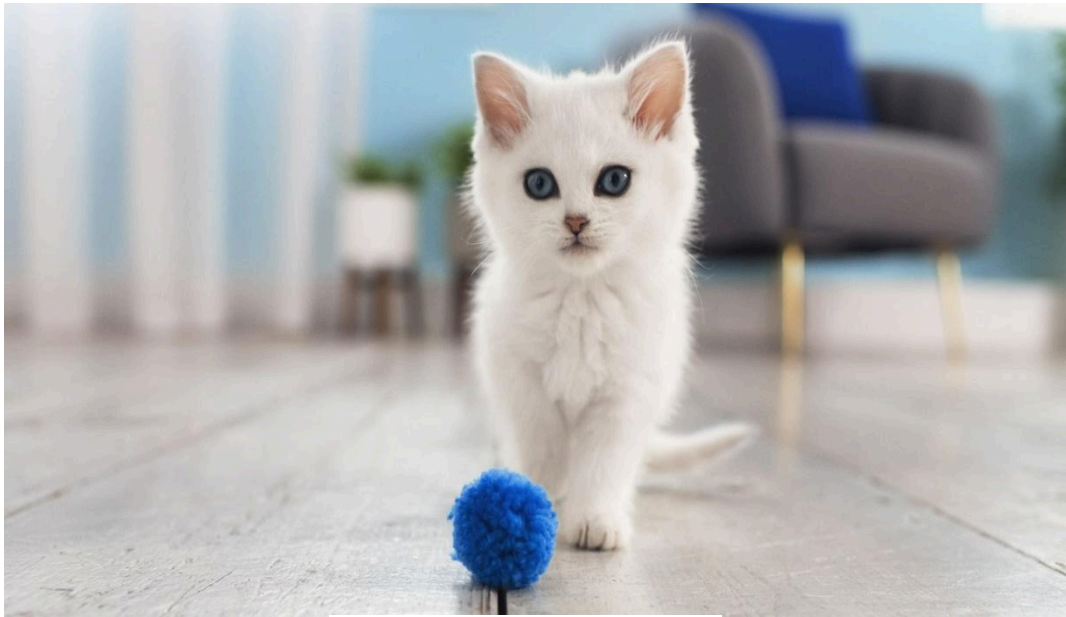
After importing all the libraries, I made the `load_image` function. After the user uploads the image, it'll be converted to RGB mode(if not already), resized, converted to Tensor and Normalized with ImageNet mean and standard deviation. I have created the `content_loss` function to calculate the content loss., which is the mean square error between content and generated image tensors. Then I made the `gram_matrix` function to calculate the Gram matrix which will be further used to calculate the `Style_loss`. Then, the `style_loss` function has been created, which computes the mean square error between the gram matrices of the style and the generated images.

The total loss is  $\alpha * \text{content\_loss} + \beta * \text{style\_loss}$ . Alpha and Beta are pre-defined weights(Hyperparameters), which I have taken by my own intuition. I took the number of steps to be 500. This function iterates through the features extracted from the content, style and generated images.

Then a Streamlit web application is setup. It provides file uploaders for uploading the file and uses the `load_image` function to load them.

The next step involves the running the adam optimizer loop to iteratively update the generated image while minimizing the losses. The loss will be displayed after each 50 steps, and after the total number of steps have been accomplished, the final generated image will be displayed.

## **RESULTS**



**CONTENT IMAGE**



**STYLE IMAGE**



**GENERATED IMAGE**

## **DISCUSSION**

More the number of steps, better will be quality because each step in the optimization process adjusts the pixels of the generated image to adjust the loss. But as the number of steps are increased, the computational time increases. So, I chose the optimal value for the number of steps. Initially, I kept the number of steps to be 1000, then to reduce the time taken, I halved it to 500. I also experimented with 300 and 200 steps, but the quality was too poor.

I initially used the learning rate of 0.001, and further increased it to 0.003, to increase the computational speed.

Coming to the part where I created the web application, I had the option to use Flask or streamlit, where I chose streamlit because of it's user-friendly interface.

The other hyperparameters, alpha and beta, were decided using other resources, and by my own intuition.

## **CHALLENGES**

It was fun to tackle these challenges , as I had to learn Pytorch and Streamlit from Scratch. I studied research papers and utilized Youtube videos to understand this fascinating technique and learn the required tools. Some other challenges include-

- Understanding VGG19
- Optimizing quality
- Deciding upon the hyperparameters
- My PC doesn't have GPU, running without GPU takes a lot of time

## **FUTURE IMPROVEMENTS**

- Using Google colab as GPU is not present in my PC
- Better choice of Hyperparamaters

## **CONCLUSIONS**

Being my first project, it was really interesting to work on. It was an enriching experience as my first foray into deep learning and interactive web applications. I gained hands-on experience with image processing, neural networks and creating a front-end. It has been a valuable learning journey, equipping me with practical knowledge and a deeper appreciation for the potential of AI in creative applications.