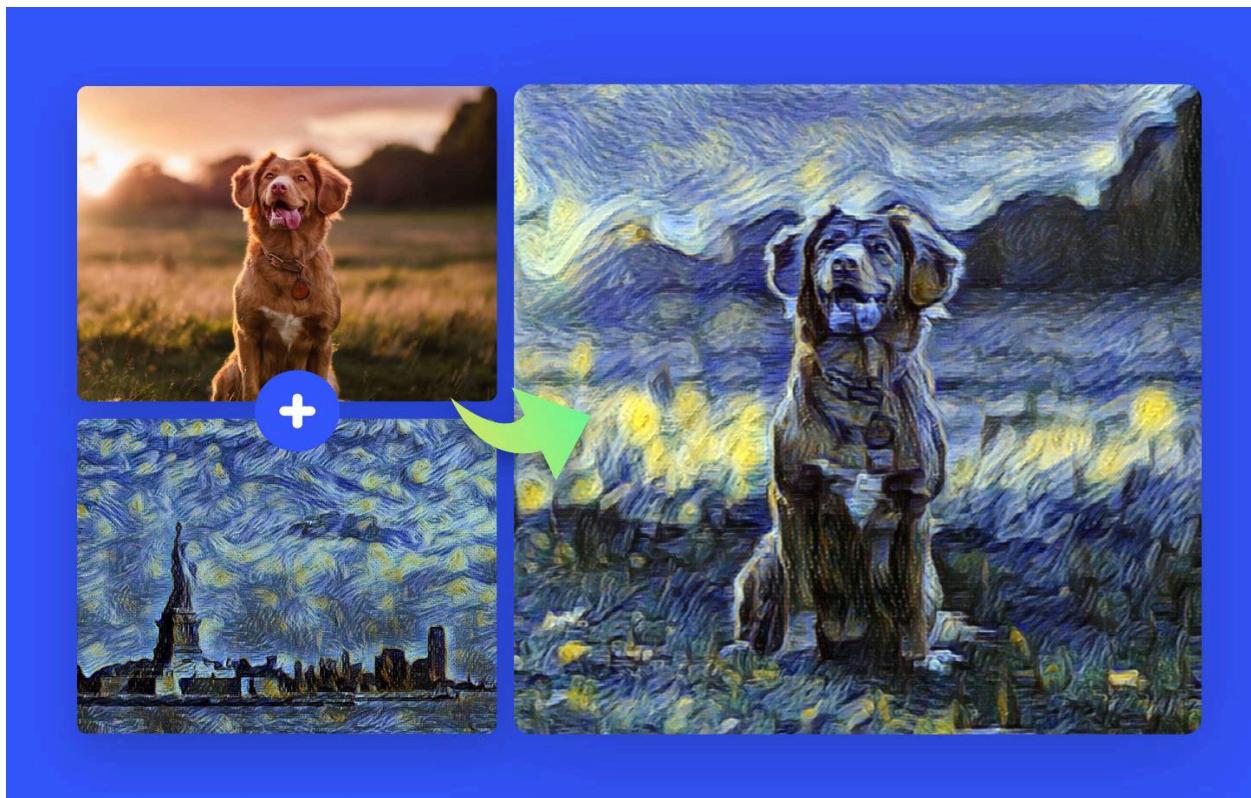
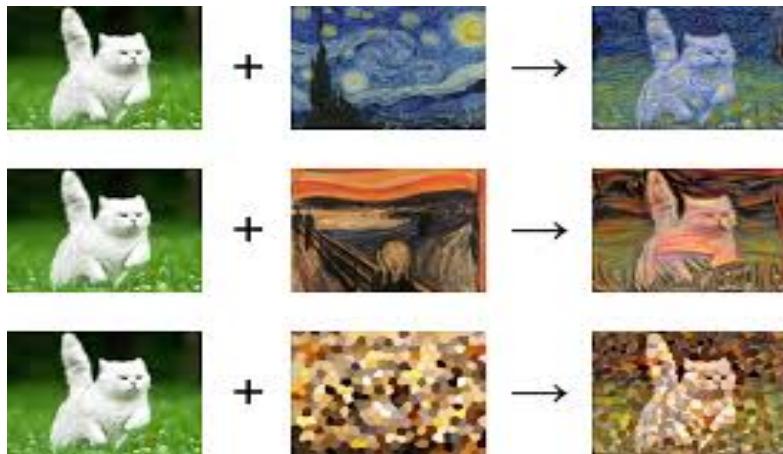


PROJECT REPORT

STYLE TRANSFER:

Style Transfer is a technique in computer vision and graphics that involves generating a new image by combining the content of one image with the style of another image.



MILESTONES:

S.No	Milestone	status
1	Making a working code for neural style transfer	Achieved
2	Building on the code such that it can take multiple styles	Achieved
3	Creating a user friendly interface	Partially achieved

Milestone 1:

I used a pre-trained convolutional neural network (CNN) vgg-19 is used to transfer styles from one image to another. To do this, a loss function is defined which will attempt to minimize the differences between a content image, a style reference image, and a generated image.

Milestone 2:

Style transfer code takes 1 content image and 1 style image. I improvised the code so that it can take multiple style images and the output image will have segments and each segment will have different style.

Used Images:





Milestone 3:

I used Gradio to make user interface. I feel the user interface could be done better by using another method.

A screenshot of a web-based user interface for image stylization. The interface is divided into four main sections: 'content_image_path' (top-left), 'style_image_path' (bottom-left), 'output' (top-right), and 'Flag' (bottom-right). The 'content_image_path' section contains a file upload area with the placeholder 'Drop File Here - or - Click to Upload'. The 'style_image_path' section also has a similar file upload area. Below these sections are two buttons: 'Clear' (gray) and 'Submit' (orange). The 'output' section is currently empty, showing a placeholder icon. The 'Flag' section is also empty.

How to Run the Project

1. Inputs Required

- **Content Image:** A JPEG file that represents the content you want to stylize.
- **Style Image:** A JPEG file that represents the artistic style you want to apply to the content image.

2. Files and Their Locations

- **Script File:**

1. Upload the Zip to Colab: In Google Colab, go to the Files panel on the left, click the Upload icon, and select the .zip file.
2. Unzip the File: In a code cell, they can use the following command to unzip it:
`!unzip your_file.zip`
3. Open the Notebook: Once unzipped, they'll see the .ipynb file in the Colab file browser. Double-clicking it will open it directly in Colab

- **Image Files:**

- **Content Image:** Place your content image in any location in your system.
 - **Location:** Any location in your system.
 - **Example Filename:** <content_image>.jpg
- **Style Image:** Place your style image in any location in your system.
 - **Location:** Any location in your system.
 - **Example Filename:** <style_image>.jpg

3. Running the Project

Install Dependencies: Ensure you have all required libraries installed. You can use pip to install any missing libraries:

PyTorch: Colab usually has PyTorch installed, but you can ensure you have the latest version

Gradio: For creating the web interface.

Pillow: For image processing.

Matplotlib: For displaying images.

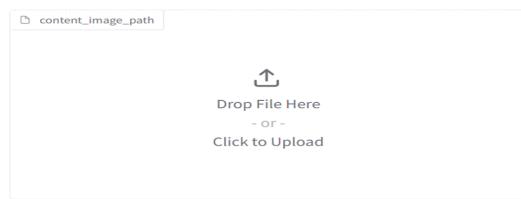
NumPy: For numerical operations

Using GPU: Make sure to enable GPU support in Colab. You can do this by going to Runtime > Change runtime type and selecting GPU.

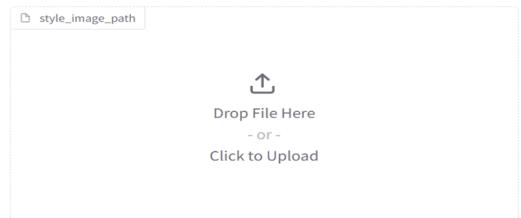
- On CPU it will take a lot of time to apply the style transfer

Run the Script:

- Copy the code attached code in Appendix A and put in Google Colab.
- Before run the script select T4 GPU in Change Run time option in Runtime menu
- Run the script in Google Colab by clicking the run all option in Runtime menu
 - Run the code and choose the number of segments i.e. number of style images
 - link to gradio interface will be provided.
 - Click on that link.
- Provide the JPEG content image by clicking Content window.



- Provide the JPEG style image by clicking Style window(s).



- Click on Submit button.

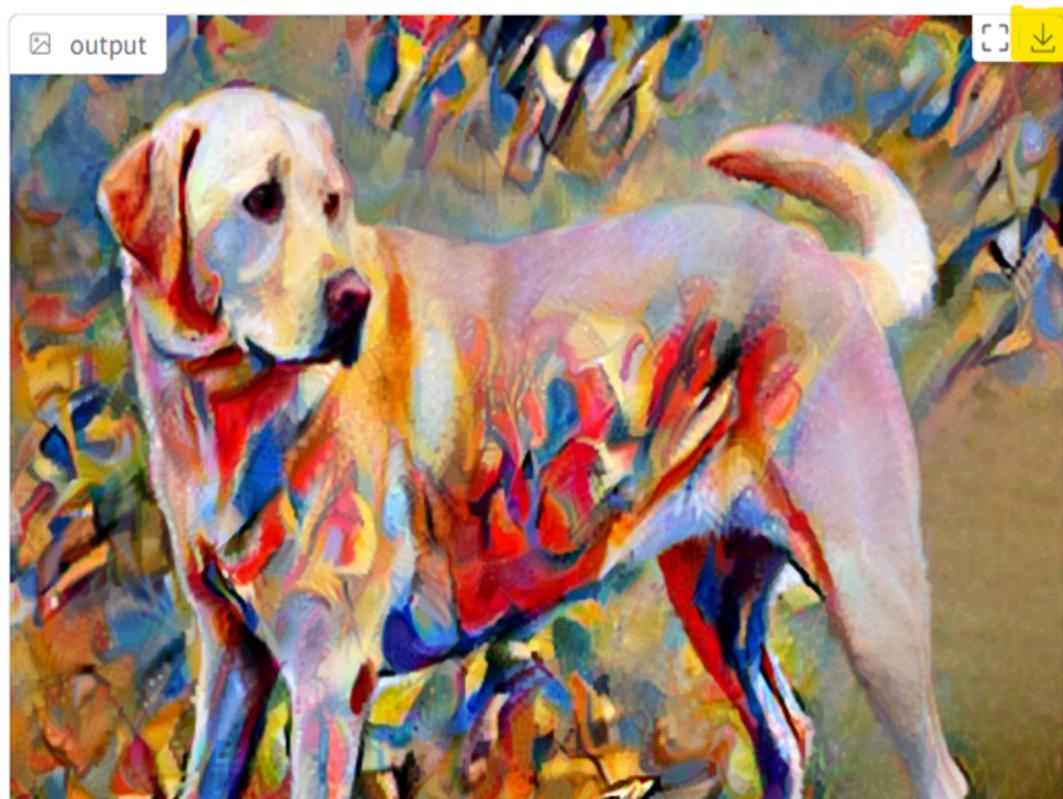
Submit

- To Clear the uploaded Content image or Style image you can click on clear button

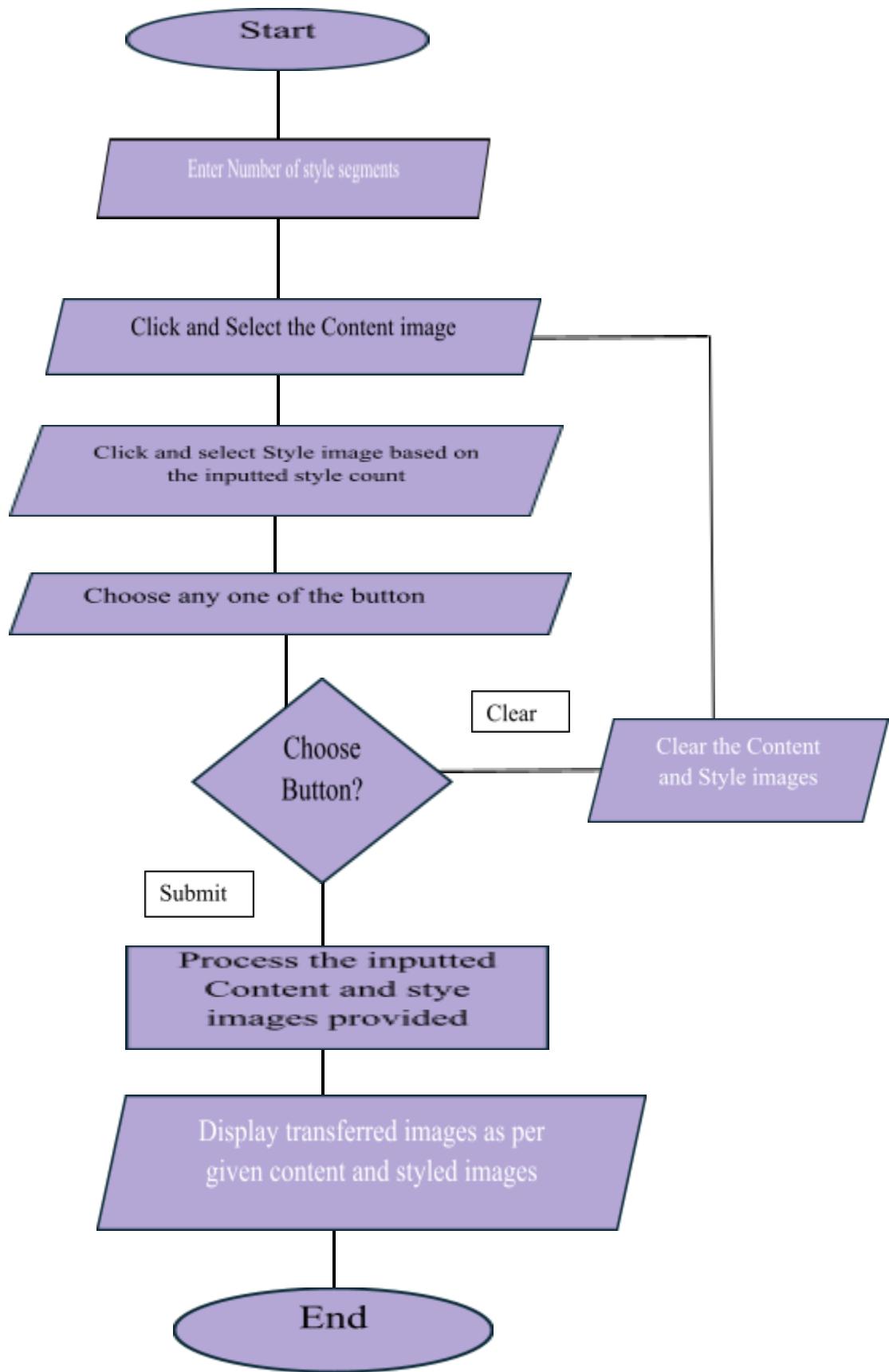
Clear

ü Expected Output

- The output stylized image will be display as output on the screen. We can save into local system by clicking the download button



FLOWCHART:



ALGORITHM:

Import Libraries:

- Import necessary libraries like gradio, PIL, torch, matplotlib, etc.

Set Device:

- Determine the device to use (GPU if available).

ImageLoader Class:

- Initialize with image size and transformations (resize and tensor conversion).
- Method to read and preprocess an image.
- Method to display an image (optionally save it).

MyModel Class:

- Initialize with layers for content and style extraction, normalization values.
- Load VGG19 model and replace pooling layers.
- Forward method to extract features from the input tensor.

NeuralStyleTransfer Class:

- Initialize with content and style images, set loss weights, and target features.
- Static methods to compute variation loss, content loss, style loss, and total loss.
- Method to optimize the image over a specified number of epochs and iterations.

Style Transfer Function:

- Load content and style images using ImageLoader.
- For each style image, instantiate NeuralStyleTransfer and fit the model to get stylized images.
- Save individual stylized images and create a combined image from all stylized outputs.

Launch Gradio App:

- Prompt user for the number of style images.
- Create and launch a Gradio interface for user input and output display.

Run Application:

- Call the function to start the Gradio app.

IMAGES:

Output images:



