**Style-Transfer**

Style Transfer is a machine learning project which works on image data and video data. It is an

optimization technique that generally takes in two images – a content image and a style

reference image and blends them together so the output image looks like the content image, but “painted” in the style of the style reference image. This is implemented by optimizing the output image to match the content statistics of the content image and the style statistics of the style reference image. These statistics are extracted from the images using a Convolutional Neural Network. So basically, Style transfer is the task of changing the style of an image in one domain to the style of an image in another domain.

**Web scrape**

1. pip install bs4
2. Open python in cmd

* import bs4
* from urllib.request import urlopen as uReq
* from bs4 import BeautifulSoup as soup

1. pip install selenium
2. Put your chrome web driver in web driver folder
3. Run - python styleimages.py to web scrape style images
4. Run - python contentimages.py to web scrape content images

**Dataset Information**

Dataset can be found here: <https://drive.google.com/drive/folders/1TJEsXmPXGcrixQ-YZHa0hl9sE3BIDabA?usp=sharing>

It has 2 categories of images and videos and further 2 categories - style and content.

*Style Images :*

Currently, I have taken 50 painters out of the famous 101 painters. While web scraping few of the directories were having less than 5 paintings so I deleted it and traversed for a few more out of 101 to make it a total of 50 directories. Every painter has approximately 50 paintings except a few.

1. You can either download the dataset from here and create a folder called dataset and store it in the directory where styleimages.py is

Like in my case :

D:\Style-Transfer\Webscrape\dataset\images\style\_images

Here your styleimages.py file will be in the same directory as that of the dataset.

1. Or you can web scrape yourself based on different parameters(number of painters, number of paintings, etc). Run: python styleimages.py

*Content Images :*

Currently, I have taken 3 broad categories - Cars(Lamborghini, Ferrari, Redbull, Bugatti), players(LeBron James, Virat Kohli, Ronaldo, Hamilton), and museum(Rome, India, Australia, Africa) as content categories. Each category contains approximately 200 images. They can either be downloaded or scraped.

1. To web scrape, Run: python contentimages.py
2. Download the data

**Data Visualization**

Content Images and Style Images :

* Read the images from the respective directories (Content -> 3 directories, Style -> 50 directories)

#### Displaying the total number of images in each category in various visual formats.

* Display sample images from each category(few from style as there are many directories) using OpenCV modules.

Feature Extraction :

* Reading a sample image.
* Demonstrating 3 different methods for feature extraction:

#### Grayscale Pixel Values as Features

#### Mean Pixel Value of Channels

#### Extracting Edge Features

**Data PreProcessing**

#### Importing Libraries

* Reading Image
* Resizing Image
* Denoising Image
* Saving the Image for Style Transfer Process

**Style Transfer process:**

Models: VGG16 and VGG19

Optimizers: Adam, RMSprop, Adagrad, Adadelta

The Style transfer process was run for all the combinations: VGG16 model with optimizers such as Adam, RMSprop, Adagrad, Adadelta, and then for VGG19 model with optimizers such as Adam, RMSprop, Adagrad, Adadelta. The losses were then compared to check the efficiency.

A content image and a style image(constant for all variations) were taken, data processing was done from the Data Preprocessing directory, and passed to the models and used various optimizers to reduce the loss and ran for 2000 epochs.

* Importing Libraries
* Loading Images
* Applying Image Processing techniques
* Model Definition
* Hyperparameter Tuning
* Defining Content and Style Loss Functions
* Applying Style transfer process
* Result

The notebooks are present in the Style Transfer Images directory.

**Results:**

* Compare loss of Optimizers in both the models used
* Adam was the best optimizer out of all the optimizers
* Compare both the models after 2000 epochs
* VGG19 model produced the best result
* VGG19 model with Adam optimizer produced the best result out of all the combinations.
* 10 content images and 10 style images were taken from the image dataset and a style transfer process was applied to get the resulting image.

The results are present in the Results directory.

**Video Dataset:**

Currently, 5 videos are included in the video dataset.