Computer Vision Project

Photorealistic Style Transfer

Project Information

Github Repository: [Link]

Reference Paper: Deep Photo Style Transfer ([link here])

Team Members:

Shahbaz Shaik : 2018111025

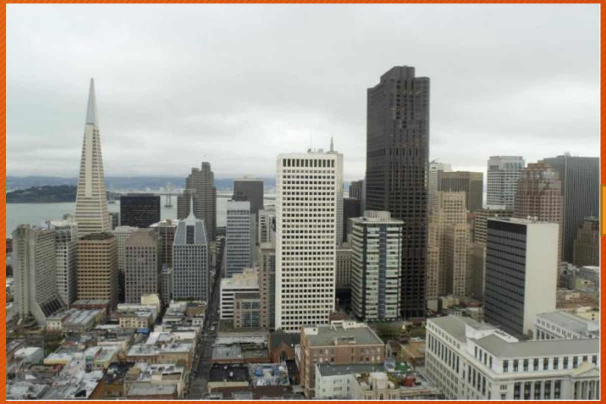
Jayant Duneja : 2018102003

Tanvi Kamble : 2018114004

Nikhil E : 2018114019

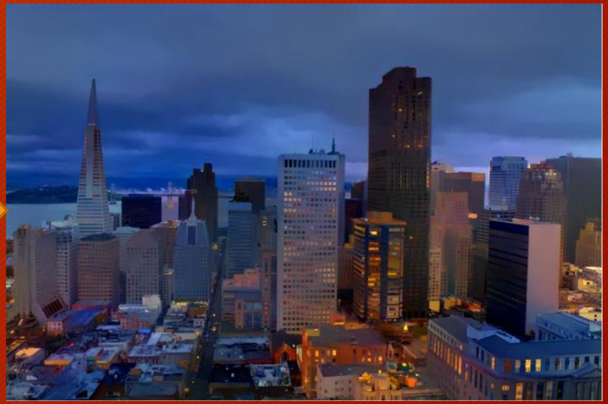
Problem Statement

Style Image



Content Image

Photo Style Transfer



Transformed Image

Introduction

We want to change the illumination, weather, hue etc. of the input photo.

We want to prevent painting-like distortions and ensuring that the photorealistic properties of the input images.

Maintain the semantics of the image and not match incompatible parts of the reference and the input image.

Proposed Model Features

Segmentation

The process of partitioning a digital image into multiple segments i.e. sets of pixels, also known as image objects.

Used to locate objects and boundaries like lines, curves, etc. in images.

We create the segmentations of both content and style images

We use the segmented images in the dataset.

VGG-19 Base Model

- We use this to get the base weights from content and style image.
- It is a pre-trained model.
- VGG-19 is a Convolutional Neural Network which has 19 layers.
- It has been trained using more than 1 million images from the ImageNet Database

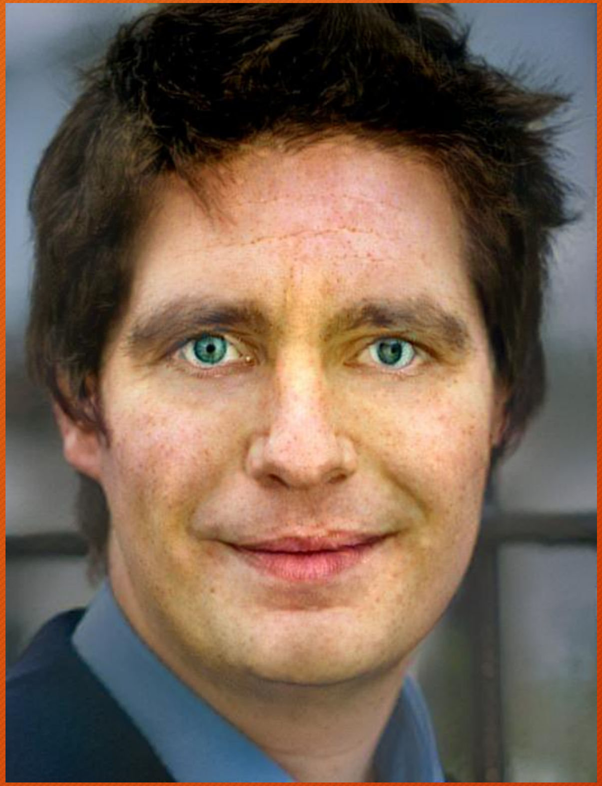
Matting Laplacian

- A natural Image Matting method
- Extracting FG from BG so that the colour of one object does not go into another.
- We perform matting on the content image and return Matrix
- Matrix is used for affine loss

$$\sum_{k|(i,j)\in w_k}\left(\delta_{ij}-\frac{1}{|w_k|}\left(1+(I_i-\mu_k)\left(\Sigma_k+\frac{\epsilon}{|w_k|}I_3\right)^{-1}(I_j-\mu_k)\right)\right)$$

Comparison with Matting

Without Matting



With Matting

Affine Loss

- Make the image smoother
- More Photo realistic images
- Uses the Matrix returned from Matting Laplacian
- Conserves the locally affine nature of the image.
- Leads to sharper images to prevent distortions.

$$\mathcal{L}_m = \sum_{c=1}^{3} \left(V_c[O]^T M_I V_c[O] \right)$$

Comparison after Smoothing

Base Neural Style Algorithm

Base Content Loss

$$\mathcal{L}_s^l = \frac{1}{2N_l^2} \sum_{ij} (G_l[O] - G_l[S])_{ij}^2$$

Base Style Loss

$$\mathcal{L}_c^l = \frac{1}{2N_l D_l} \sum_{ij} (F_l[O] - F_l[I])_{ij}^2$$

Augmented Functions

Photorealism Loss

$$\mathcal{L}_m = \sum_{c=1}^{3} (V_c[O]^T M_I V_c[O])$$

Augmented Style Loss

$$\mathcal{L}_{s+}^{l} = \sum_{c=1}^{C} \frac{1}{2N_{l,c}^2} \sum_{ij} (G_{l,c}[O] - G_{l,c}[S])_{ij}^2$$

Augmented Objective

$$\mathcal{L}_{total} = \sum_{l=1}^{L} \alpha_l \mathcal{L}_c^l + \Gamma \sum_{l=1}^{L} \beta_l \mathcal{L}_{s+}^l + \lambda \mathcal{L}_m$$

Without augmented style loss

Style Image

Content Image

Photo Style Transfer

Transformed Image

With augmented style loss

Style Image

Photo Style Transfer

Content Image

Transformed Image

Outputs:

Content Image

Style Image

Final Image

Content Image

Style Image

Final Output

Content-Image

Style Image

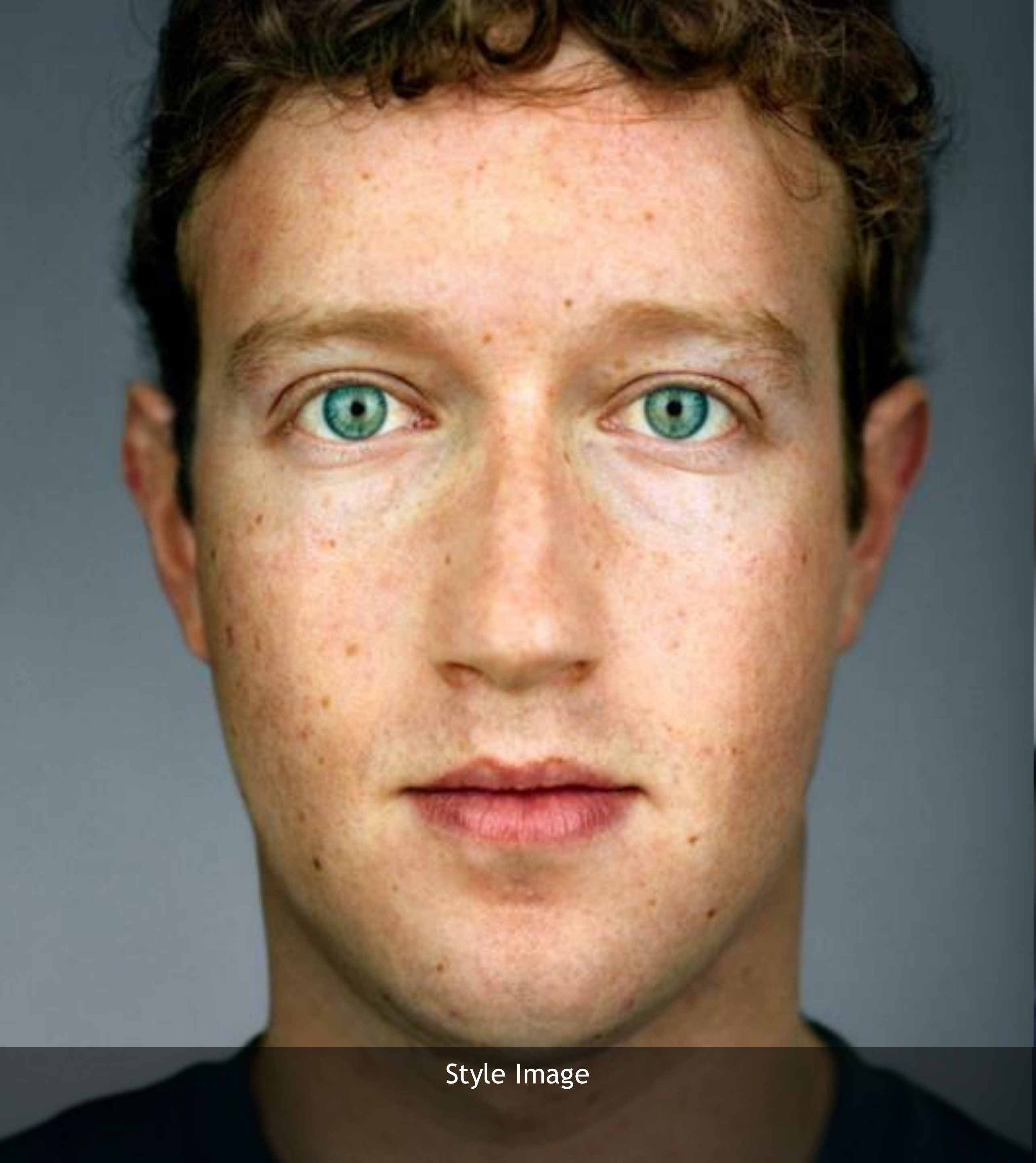
Final Image

Content-Image

Style Image

Final Image

Progress over iterations


Style Image


Content Image

Varying Parameters – Content Weight

Content Weight = 3

Content Weight = 4

Content Weight = 5

Varying Parameters – Style Weight

Style Weight = 100

Style Weight = 1000

Style Weight = 10000

Varying Parameters – Photorealism Weight

Photorealism Weight = 1000

Photorealism Weight = 10000

Photorealism Weight = 100000