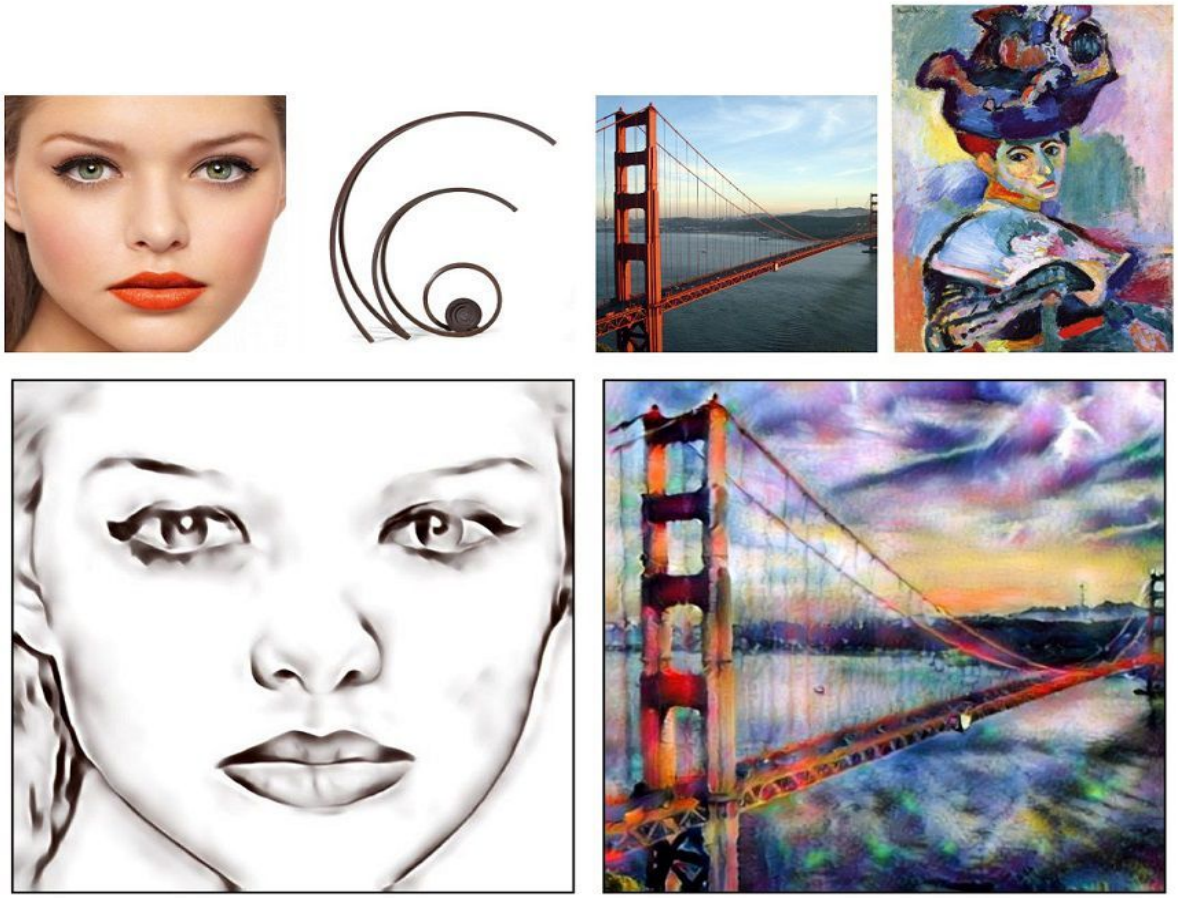
Computer Vision

Universal Style Transfer Via Feature Transform



Team Name: Amateurs
Team Members:
Gangula Rama Rohit Reddy
Suma Reddy
Vallabhaneni Jaswanth

Project Mentor : Chaitanya Patel
Instructors : Dr. Avinash Sharma,
Dr.Anoop Namboodiri

Introduction

Style transfer is an important image editing task which enables the creation of new artistic works. Given a pair of images, i.e content image and style image, it aims to synthesize an image that preserves some notion of the content but carries characteristics of the style. The key challenge is how to extract effective representations of the style and then match it in the content image.

Despite the rapid progress in deep neural networks, the existing works often trade off between generalization, quality and efficiency, which means that optimization based methods can handle arbitrary styles with pleasing visual quality but at the expense of high computational costs, while feed-forward approaches can be executed efficiently but are limited to a fixed number of styles or compromised visual quality.

In this work, the authors propose a simple yet effective method for universal style transfer, which enjoys the style-agnostic generalization ability with marginally compromised visual quality and execution efficiency. The selling point of this paper is that it is “learning free”. Existing feed-forward base techniques would need to be trained on predefined styles and then fine-tuned for new styles. Whereas, this paper presents a method which is completely independent of the style during training phase making it a “learning-free” approach.

Dataset:

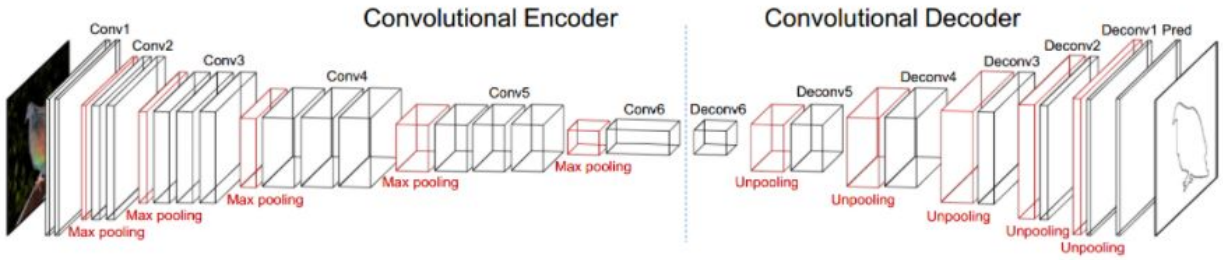
For training the decoder, we use COCO dataset. COCO is a large-scale object detection, segmentation, and captioning dataset. Microsoft COCO dataset has 83K images. Describable Textures Dataset has around 5640 images organized according to a list of 47 terms which are inspired from human perception.

Method

The style transfer problem is formulated as a combination of two processes, viz. Image Reconstruction and Feature transform using Whitening and Color Transform. The reconstruction part is responsible for inverting features back to the RGB space and the feature transformation matches the statistics of a content image to a style image.

Image Reconstruction

A classic Encoder-Decoder mechanism is the one where a image is fed into an Encoder network which encodes the image forming a representation and passed on to a decoder which tries to reconstruct the original input image.



The paper uses a slight modification of this for image reconstruction. As a first step, they use existing pre-trained VGG-19 as the Encoder. The decoder is trained to reconstruct the Image. The decoder is designed as being symmetrical to that of VGG-19 network with the nearest neighbour upsampling layer used for enlarging feature maps.

More than one decoder for reconstruction is trained. 5 decoders are trained for reconstruction. X in the above image refers to the layer number in VGG network.

The pixel reconstruction loss and feature loss are employed for reconstructing an input image.

$$L = \|I_o - I_i\|_2^2 + \lambda\|\Phi(I_o) - \Phi(I_i)\|_2^2$$

where I_i and I_o are the input image and reconstruction output, and Φ is the VGG encoder that extracts the Relu_X_1 features. In addition, λ is the weight to balance the two losses. After training, the decoder is fixed (i.e., will not be fine-tuned) and used as a feature inverter. We never use any style image in the whole training process.

Whitening and Coloring Transforms(WCT):

WCT does some cool math which plays a central role in transferring the style characteristics from style image while still preserving the content. WCT is the process of disassociating the current style of the input image and associating the style of the style image with the input image. It involves two steps, first step is whitening.

We know that input to the WCT block is the output of the Encoder block (Relu_X_1). Relu_X_1 has a shape of C x H x W, where C is the number of channels, H is the height and W is the width of a feature map. We vectorize these feature maps such that we have C vectors of length H x W. Let f_c be the vectorized feature map of shape [C, (H_c x W_c)], where H_c and W_c are respectively the height and width of the feature maps at certain Relu_X_1 due to the content image. Similarly, let f_s be the vectorized feature map of shape [C, (H_s x W_s)], where H_s and W_s are respectively the height and width of the feature maps at certain Relu_X_1 due to style image.

Whitening Transform:

Our goal is to find a transformation of f_c, let us call it f_ct such that the covariance matrix of f_ct is an Identity matrix. This ensures that the feature maps have no correlation.

$f\_ct = W x f\_c$, where W is a transformation matrix. A very common choice of W is the inverse square root of Y, where Y is the covariance matrix. To have Y

= f_c x (f_c.transpose) we will need that the mean value m_c (per channel mean) be subtracted from f_c.

$$f_c = f_c - m_c$$

$$Y = f_c f_c^T$$

$$W = Y^{-1/2}$$

$$f_{ct} = Y^{-1/2} f_c$$

$Y = E_c D_c E_c^T$, where E_c is an orthogonal matrix with its columns being the Eigen vectors of Y. D_c is a diagonal matrix with the Eigen values of Y.

$$Y^{-1/2} = (E_c D_c E_c^T)^{-1/2}$$

Let, $C = (E_c D_c E_c^T)^{-1/2}$

$$C^2 = (E_c D_c E_c^T)^{-1}$$

$$C^2 = E_c D_c^{-1} E_c^T$$

$C = E_c D_c^{-1/2} E_c^T$ satisfies $C^2 = E_c D_c^{-1} E_c^T$

So, $Y^{-1/2} = E_c D_c^{-1/2} E_c^T$

and finally, $f_{ct} = E_c D_c^{-1/2} E_c^T f_c$ ——————————(1)

Reconstruction from the features which are subjected to whitening transformation would preserve the content but removes any information related to style. For example:

Coloring Transform:

By whitening transformation, we effectively disassociated the features of their style. Now by coloring transform, we will associate to these the style of style image.

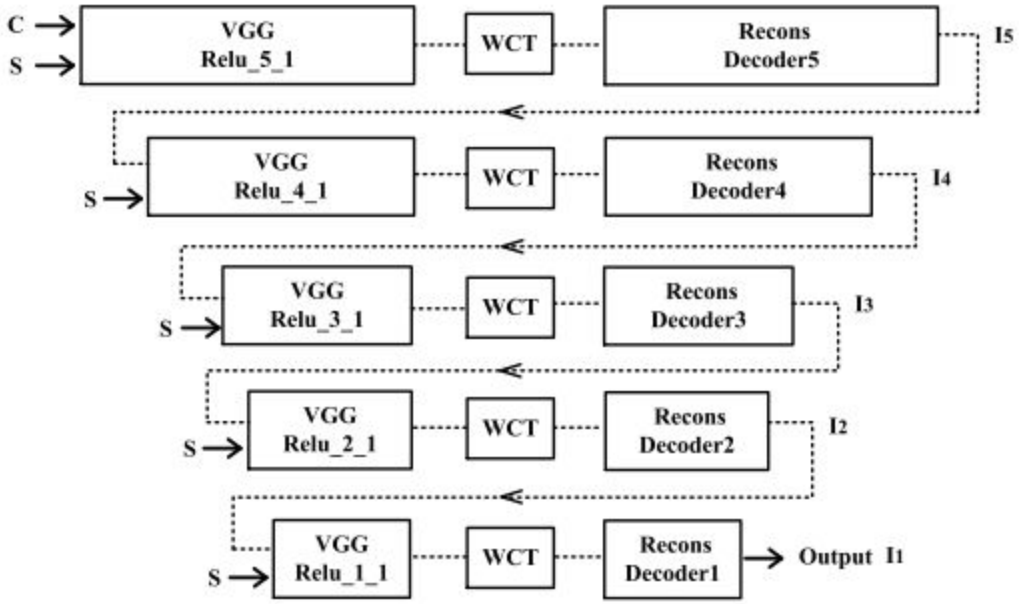
Our goal is to find a transform of f_ct, let us call it f_cst such that the covariance matrix of f_cst is equal to the covariance matrix of f_s.

$$f_{cst} {f_{cst}}^T = Z = f_s {f_s}^T \quad \text{-----------(2)}$$

$f_{cst} = E_s {D_s}^{1/2} {E_s}^T f_{ct}$ where E_s is an orthogonal matrix with its columns being the Eigen vectors of co-variance matrix Z. D_c is a diagonal matrix

Multi-level coarse-to-fine stylization

High layer features capture more complicated local structures while lower layer features carry more low-level information eg: colors. So we proceeded to use the feature from all layers instead of sticking to just one.



We start off with Content and Style Images feeding them to VGG and Relu_5_1 feature is extracted and sent into WCT and then Decoder5. The output of Decoder5 is fed into VGG along with the style image and Relu_4_1 is extracted and the process continues until we get output from Decoder1. The image below shows results from such a multilevel inference. I_5 is effectively the output of first level (in the above image) and I_1 is the output of Decoder1(the final output).

(a) I_5 (b) I_4 (c) I_1

Thus this algorithm is efficient as it is learning free and also efficient as it has no loops of optimization which takes many iterations to generate good results. It is not a style specific network as it does not include style factor while training.

User Interface:

An user interface was made for the model developed. It was created using wxPython. It takes any content image and style image as an input and generates the styled content image as the output. It also takes the user input for the amount of style has to be applied for the content image on a scale of 0-1 according to the user's choice.

This finds many uses because of it being a real-time application which can actually be used for editing images and pictures.

Results :

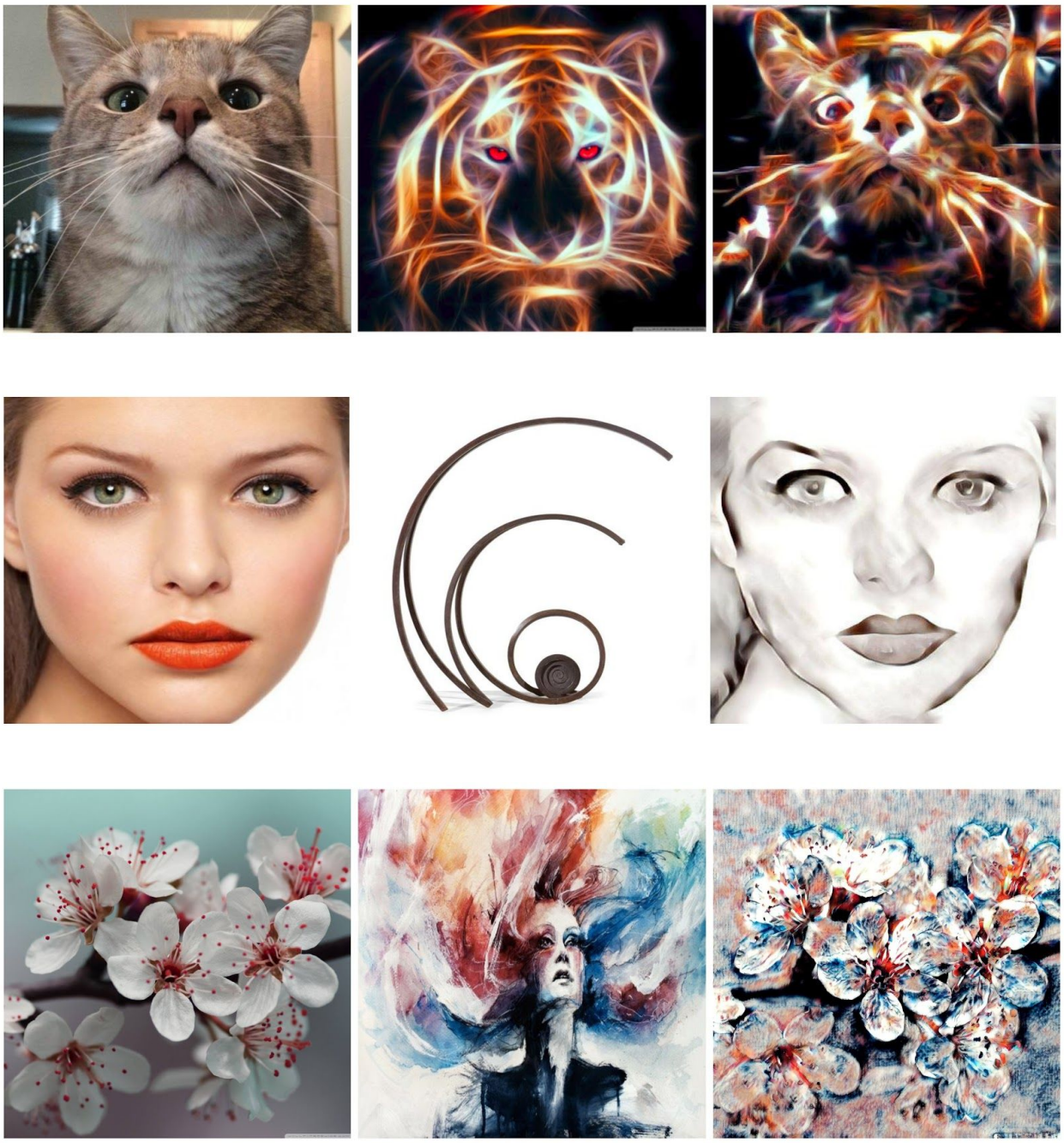
The following are the results generated by the model developed by us :-

CONTENT IMAGE

STYLE IMAGE

RESULT GENERATED

Some more generated results are the following :







AIRFRANCE




Challenges Faced :

The following are some of the challenges faced during the development of the project :-

- Training decoder models were time taking.
- Development of UI was also challenging.

Conclusion :

In conclusion, an universal style transfer model has been developed which is learning free. It also doesn't require learning for each individual style. The architecture is also multi-level stylization pipeline which ensures that all level of information of style is taken into consideration which results in better outputs.

Github Link:

https://github.com/Rama-007/Universal_Style_Transfer