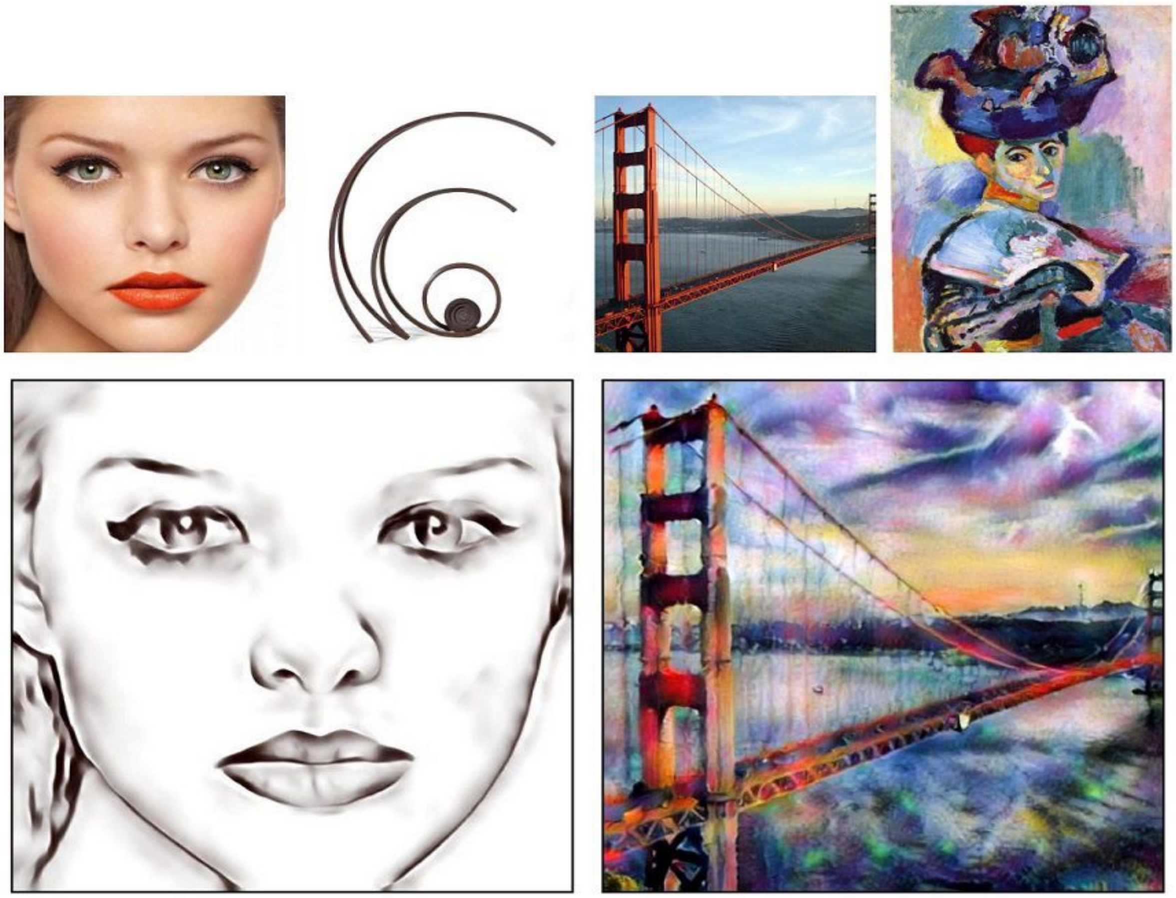
Image Style Transfer using Deep Neural Nets



Team Name: The Last Minute Team

Project Mentor : Sanjoy Chowdhary

Team Members:

Rachit Jain

Aashish Kumar

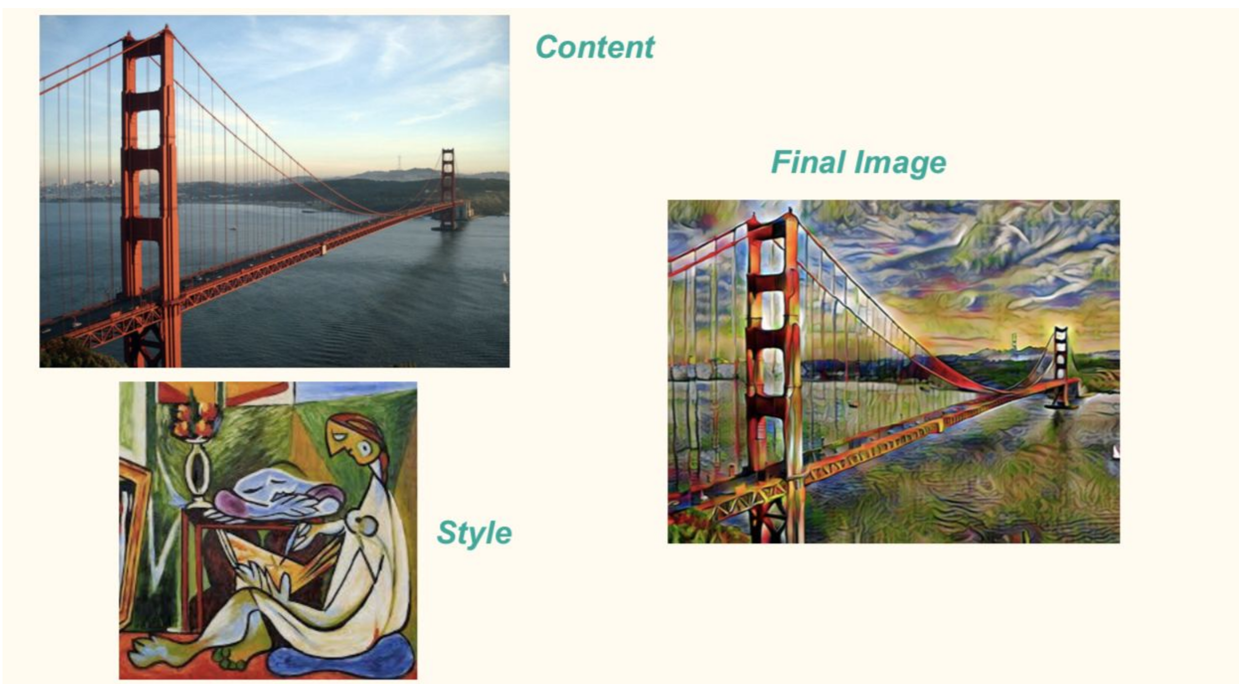
Chanakya Vishal KP

Swapnil Gupta

Instructor : Dr. Ravi Kiran

What is Universal Style Transfer ?

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



Problem Description

The main task of this problem is to extract effective representations of the style and then match it to the content image.

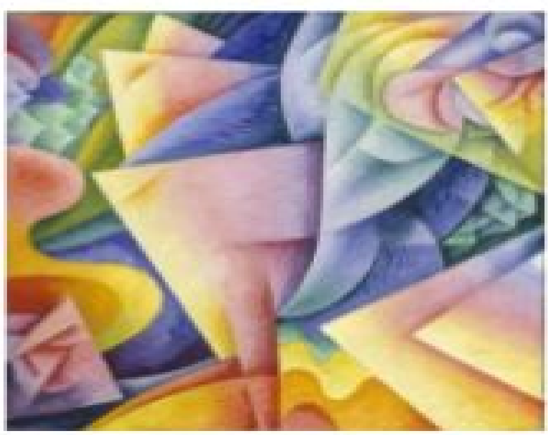
Major Problems with existing techniques

1. Optimization based methods can handle arbitrary styles with pleasing visual quality but it takes many iterations to generate good results hence requires high computational costs.
2. Feed-forward approaches can be executed efficiently but are limited to a fixed number of styles or compromised visual quality

Expected Result

Extracted effective representation of style and matched with the content of image.

Style Image



Content Image

Expected Result from the Style image and content image

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

About the research paper and Problem Design

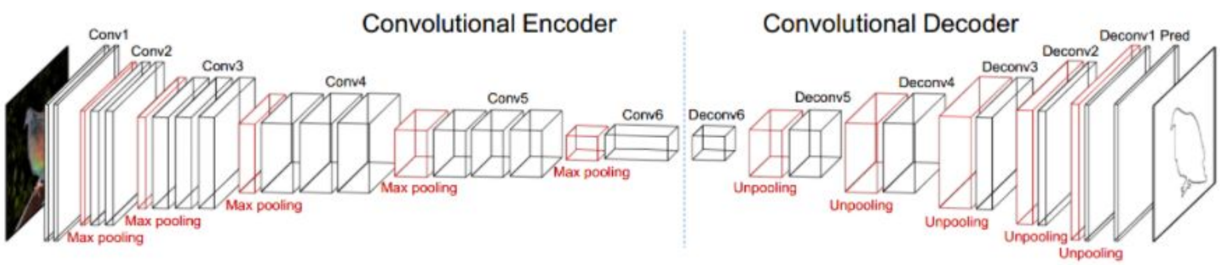
The research paper used id Li Y, Fang C, Yang J, et al. Universal Style Transfer via Feature Transforms. It is learning free, existing feed-forward base need to be trained on predefined styles and then fine tuned for new styles. The method in the paper is completely independent of the style during training phase

Problem Design

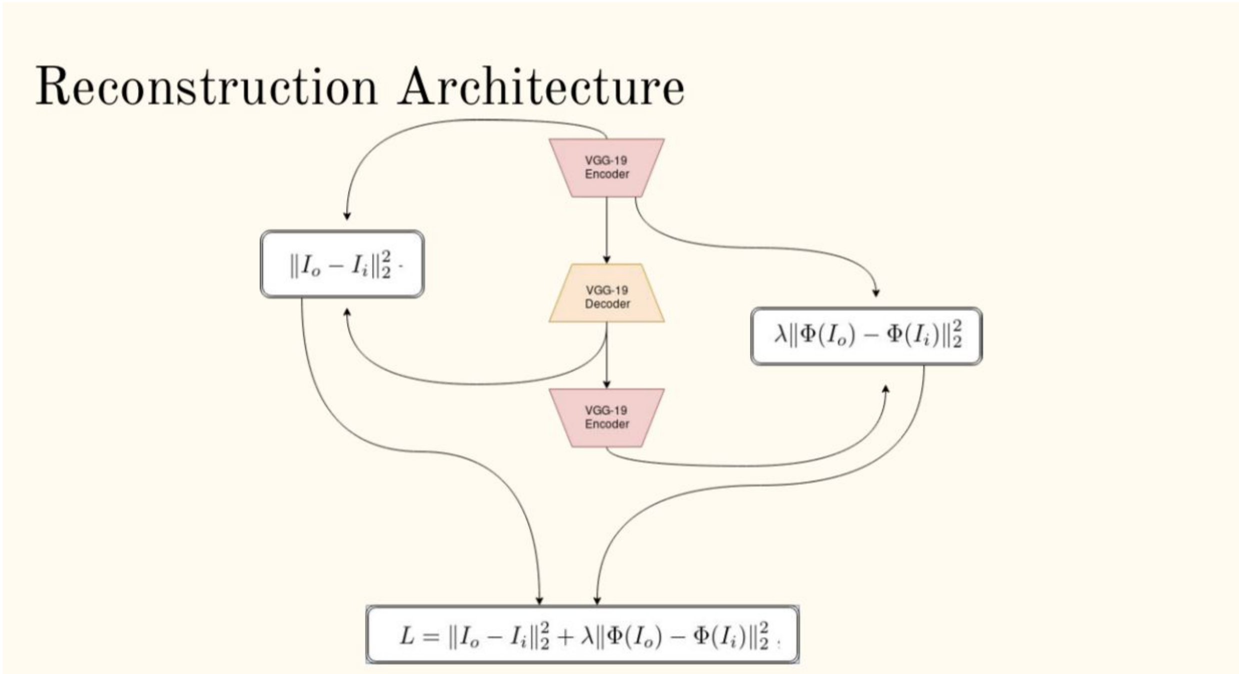
The style transfer problem is formulated as a combination of two processes, viz. Image Reconstruction and Feature transform using Whitening and Color Transform. The reconstruction part is responsible for inverting features back to the RGB space and the feature transformation matches the statistics of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

Image Reconstruction

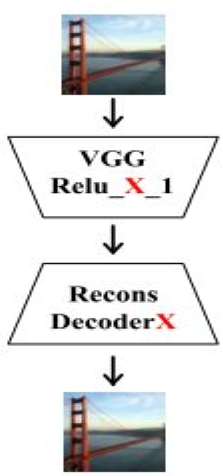
A classic Encoder-Decoder mechanism is the one where a image is fed into an Encoder network which encodes the image forming a representation and passed on to a decoder which tries to reconstruct the original input image.



Reconstruction Architecture



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.

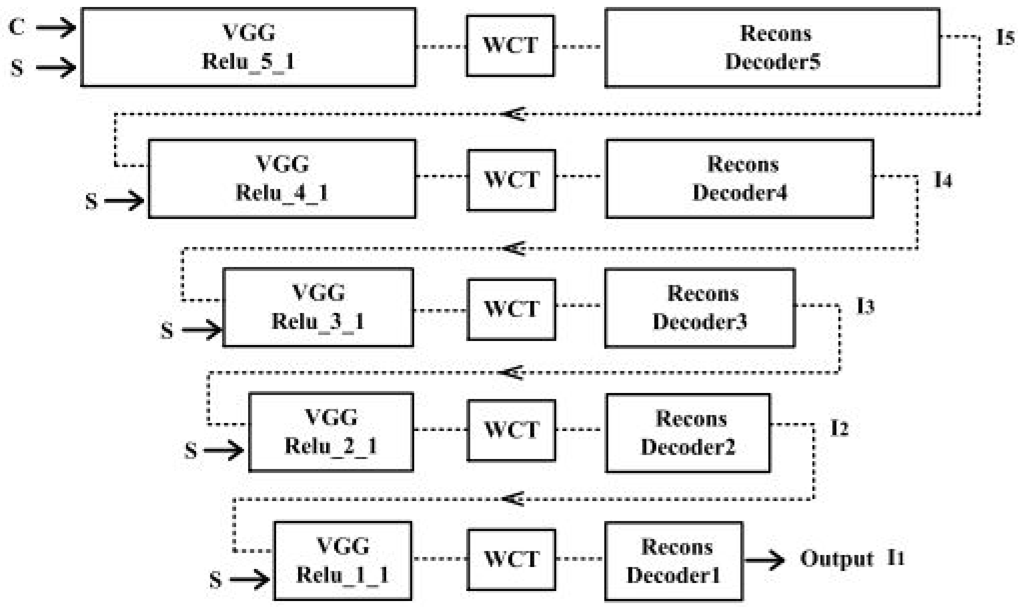
Our goal is to find a transform of f_ct, let us call it f_cst such that the covariance matrix of f_cst is equal to the covariance matrix of f_s.

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

$f_{cst} = E_s{D_s}^{1/2}{E_s}^T f_{ct}$ 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.

Feature Engineering:

We did it using Pytorch from Scratch

- Modules
    - Data Loader
    - Reconstruction
    - Separate Encoder and Decoder
    - Make feature from encoder for style image and content image
    - WCT (Whitening and Coloring Transform)
    - Decode the feature to get stylized image

We have implemented all the proposed in the research paper.

Work Distribution:

- Rachit Jain

  Data Loaders and decoder for MNIST database and presentation part

- Chanakya Vishal KP

  WCT and Encoder for mnist

- Aashish Kumar

    VGG Model

- Swapnil Gupta

    Integration, running of all the code and VGG model