Universal Style Transfer using Deep Neural Nets

Rachit Jain
Swapnil Gupta
Chanakya Vishal KP
Aashish Kumar

Project Mentor: Sanjoy Chowdhary
Instructor: Dr. Ravi Kiran

What is Universal Style Transfer ?

Universal Style Transfer

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.

Style transfer is an important image editing task which enables the creation of new artistic works.

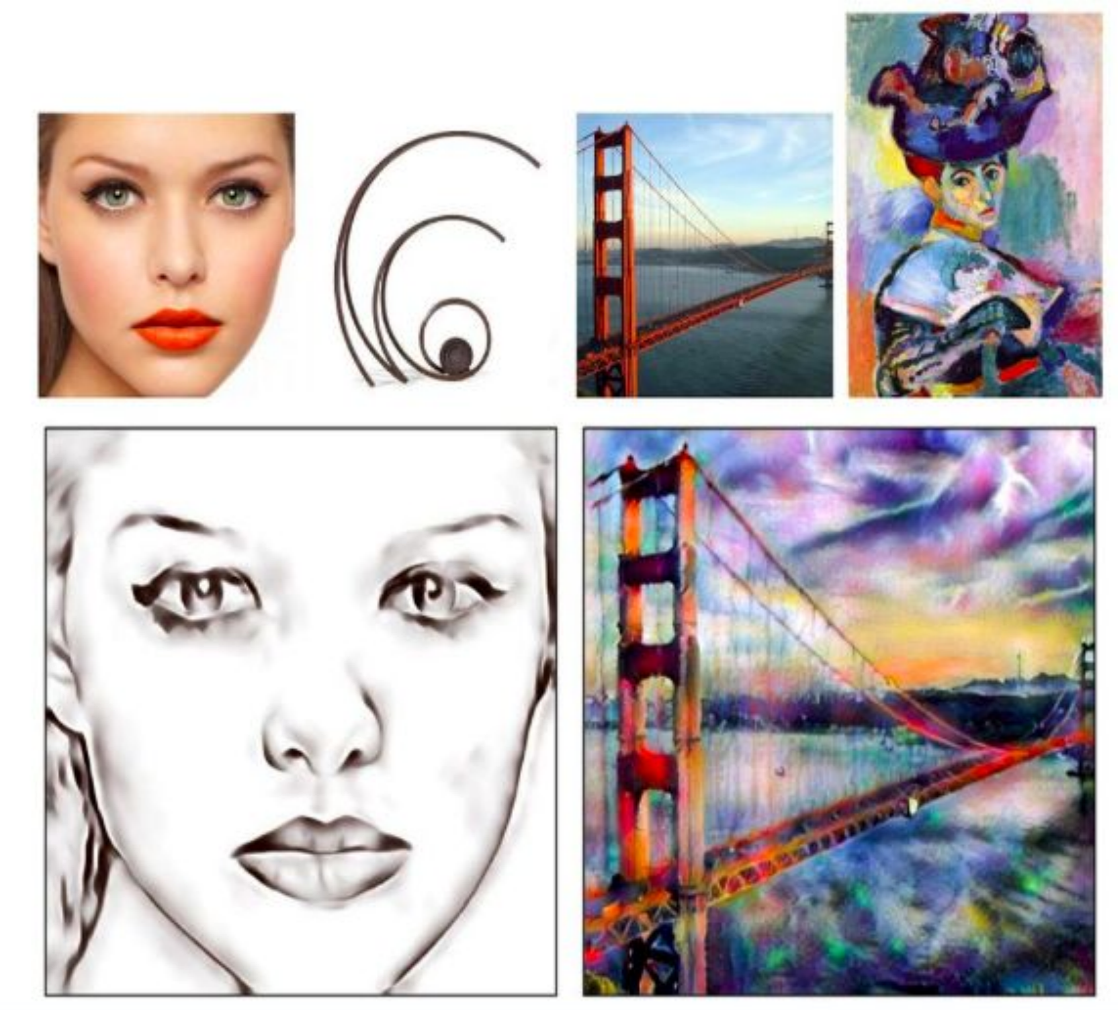
The key challenge lies in how to extract effective representations of the style and then match it in the content image.

Aim of the project

The main task

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


Content
Final Image
Style



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

Research Paper and Dataset used

Li Y, Fang C, Yang J, et al. Universal Style Transfer via Feature Transforms

- It is learning free.
- Existing feed-forward base techniques would need to be trained on predefined styles and then fine-tuned for new styles.
- The method in paper, is completely independent of the style during training phase.

About the Dataset

Microsoft COCO dataset has 330K images out of which around 200K are labelled. This is for training the decoder. Currently trained on 15K images.

The MNIST database (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems.

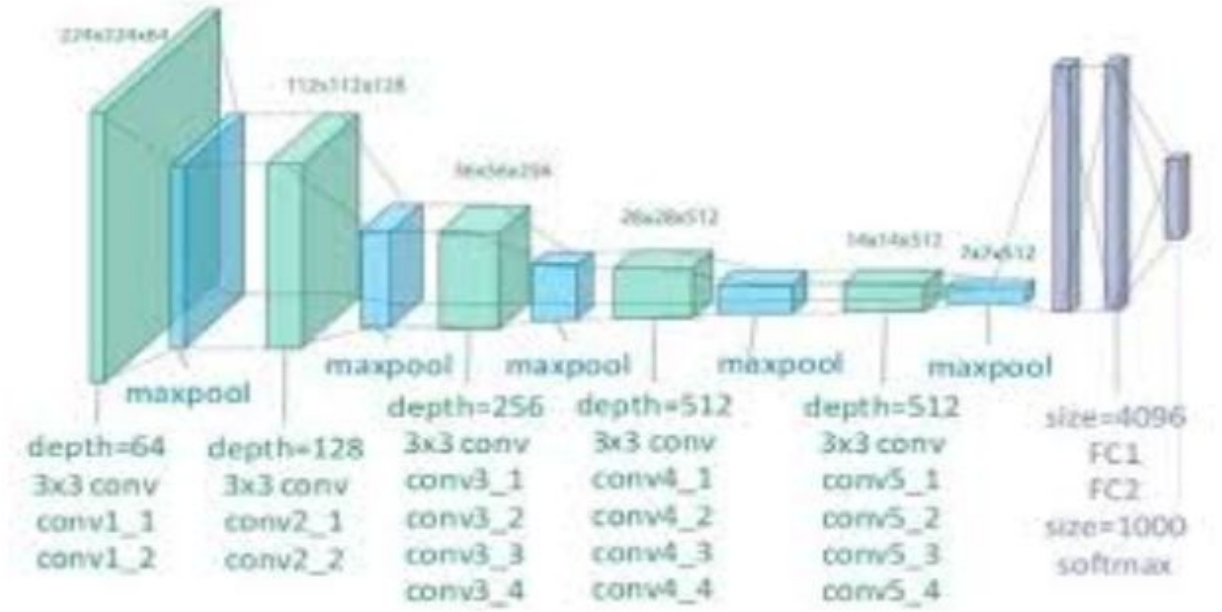
Describable Textures Dataset has around 5640 images organized according to a list of 47 terms which are inspired from human perception.

Approach

Method Outline

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

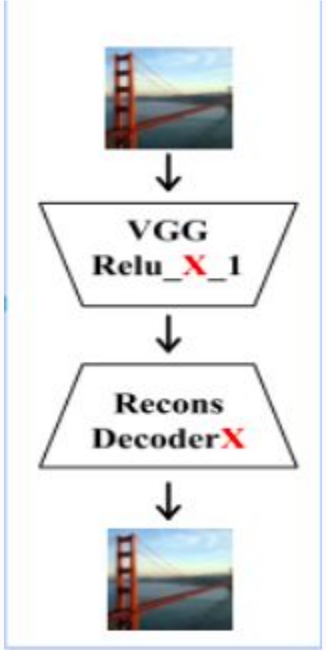
Method : VGG19 architecture



Method: Image Reconstruction

- As a first step, the existing pre-trained VGG-19 network as the Encoder.
- The decoder is then trained to reconstruct the Image. The decoder is designed to be symmetrical to that of VGG-19 network with the nearest neighbour upsampling layer used for enlarging feature maps.
- Important to note that 5 decoders are trained for reconstruction.
- The pixel reconstruction loss and feature loss are employed for reconstructing an input image. The following is the function : $L = \|I_o - I_i\|_2^2 + \lambda \|\Phi(I_o) - \Phi(I_i)\|_2^2$
- Here, Ii and Io 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.
- Lastly, after training the decoder, it is fixed and will not be fine-tuned later on. This will be used as a feature inverter.

Image Reconstruction



Architecture for image reconstruction. ‘X’ represents the layer number.

Method: Whitening and Coloring Transform

1. WCT does some cool math which plays a central role in transferring the style characteristics from style image while still preserving the content.
2. 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.
3. 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

Transformation Math

1. Whitening Transform

$$\hat{f}_c = E_c \, D_c^{-\frac{1}{2}} \, E_c^\top \, f_c$$

2. Coloring Transform

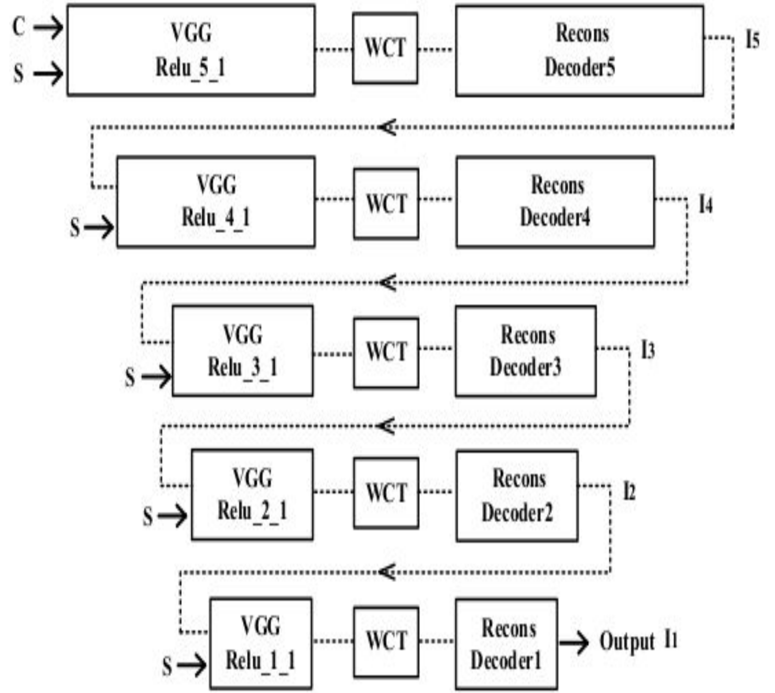
$$\hat{f}_{cs} = E_s \, D_s^{\frac{1}{2}} \, E_s^\top \, \hat{f}_c$$

Original

Whitened Image

Multi-level coarse-to-fine stylization

1. High layer features capture more complicated local structures while lower layer features carry more low-level information eg: colors.
2. 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 image present in the next slide) and I_1 is the output of Decoder1(the final output).



(c) Multi-level stylization

Multi Level Stylization using 5 Decoders and VGG Layers

Expected Output by end

Style Image

Content Image

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

Our Work

Pipeline of the project

- 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

Challenges Faced

1. Loss function depends on feature of input image, input image, feature of output image and output image

$$L = \|I_{output} - I_{input}\|_2^2 + \lambda\|\Phi(I_{output}) - \Phi(I_{input})\|_2^2$$

2. Training Heavy

3. Tried training VGG 19 decoder (around 100M parameters) on MS Coco

4. Training small architecture on MNIST dataset

5. Trained encoder and decoder together for reconstruction module

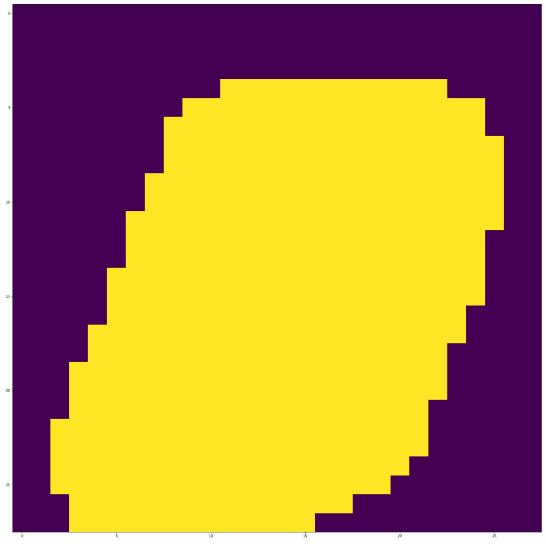
6. Need to separate encoder and decoder after training for WCT

Experiments

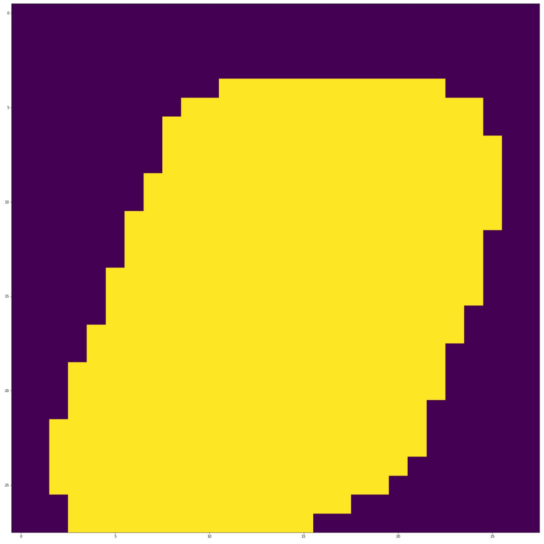
- Decoder Training
  - VGG 19 RELU Layers
  - Microsoft Coco Dataset
  - MNIST Dataset
- Style Transfer
  - Describable Texture Dataset (DTD)
  - User Control

Results

MNIST Results



Content : 0
Style : 1



Content: 3
Style: 2

MNIST Results

Content : 4
Style : 7

Content: 9
Style: 2

Results from pretrained model

Content

Style

Final Image

Results from pretrained model

Content

Style

Final Image

Results from pretrained model

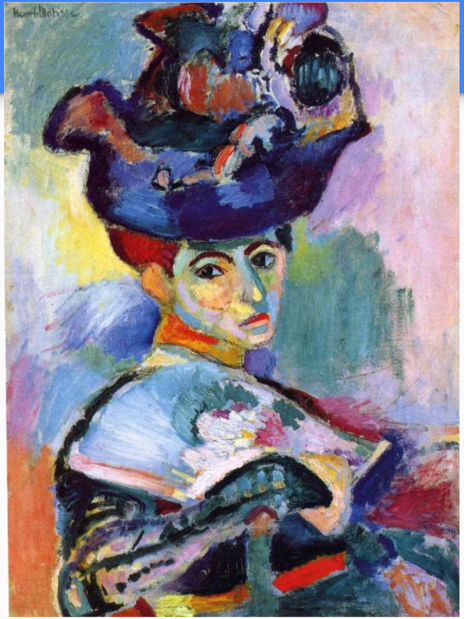
Content

Style

Final Image

Results from pretrained model

Content



Style

Final Image

Applications

Some applications

- Style Transfer
- Texture Synthesis
- Neural style transfer to design clothes
- Displaying images as if it was drawn by an artist.

Few ideas

Few ideas worth trying

- Style Transfer in Videos
- Interpolation between styles
- Spatial control
- Comparison with other architecture (ResNet, GoogleNet, ...)

Work Division

- 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

Thank You