Extensions to Image Style Transfer

Vishal Batchu, Aurobindo Munagala, Ameya Prabhu

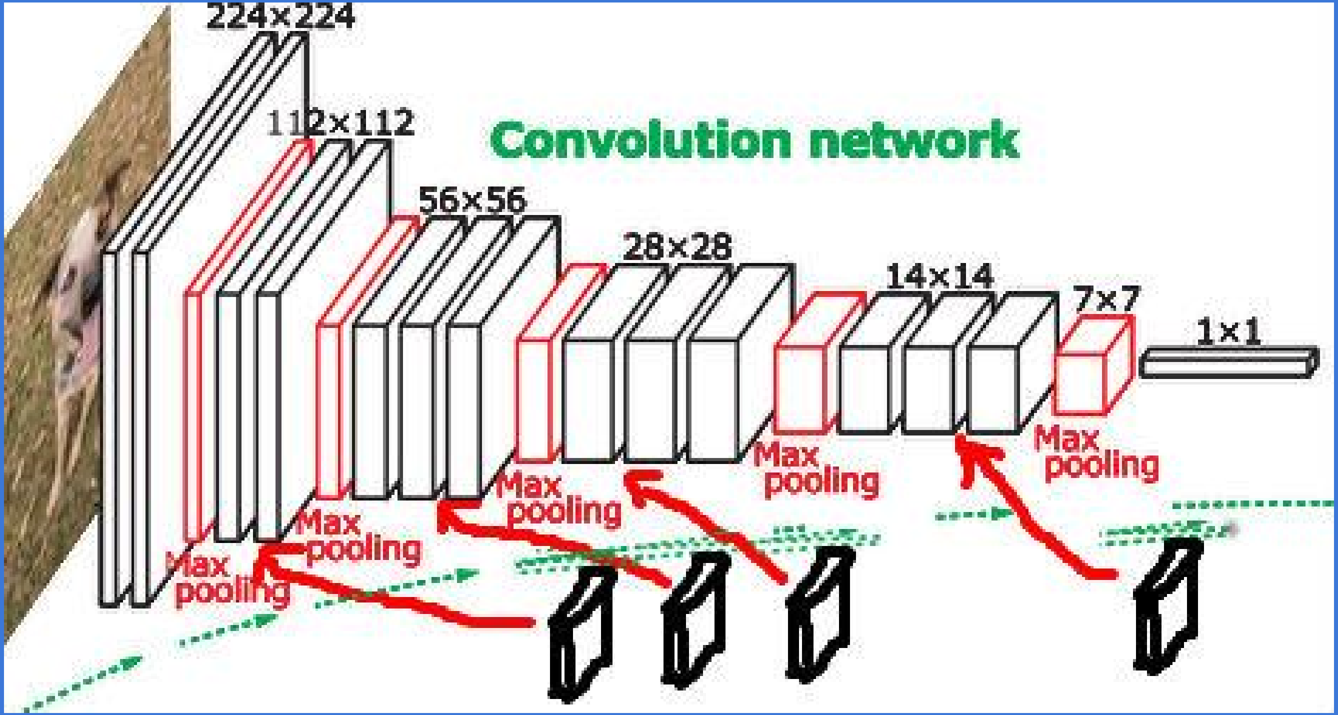
Overview of style transfer

Combine style and content from different images to generate a new image.

Introduction

- Lack of good content representation techniques in the past.
- Use CNNs which are optimized to perform object detection and localization.
- Given a content and style image, generate an image which captures the texture from the style image and the semantic details from the content image.

Module Insertion in Network



VGG

Algorithm Overview

Initialization

- Select layers after which we insert style and content loss modules.
- At each of these layers, store the output of the content image at the specific layer as our targets.
- Similarly for the style we store the gram matrix representations of style images as our targets.
- Randomly initialize an image as our input (white noise)

Learning - Deconvolution to Input Layer

- Propagate the current input image through the network and at the end set the gradient to 0.
- On backpropagation as the image goes through the style and content loss modules, gradients are added and they get accumulated.
- Input image modified according to the gradients to reduce overall loss.
- Loss here is CLoss + SLoss + RLoss where CLoss is the content loss, SLoss is the style loss and RLoss is the regularization loss.
- Repeat this process for around 1000 iterations to get good results.

Content Loss

Content Loss module implemented

- A single content loss module has been added in the network
- The 3 most commonly used images for style transfer were used and the results were recorded. The tubingen image, brad pitt's face and the golden gate image.
- This module just stores an expected content image representation during initialization.
- When we pass the white noise image it compares the L2 norm between the noise image representation at the current layer and the expected output of the content image at that same layer and adds this to the gradient while backpropagating.

Style Loss

Gram Matrices

- Represent the style of an input volume with the help of a 2D matrix
- An input volume of dimensions (C,H,W) would generate a matrix of dimensions (C,C)
- The generation of the gram matrix is done as follows,

(C,H,W) -> (C,HxW) -> ([C,HxW],[C,HxW]) -> ([C,HxW],[HxW,C]) -> (C,C)

- This matrix is symmetric and also satisfies all the other properties of kernels. This is essentially a kernel that captures the style of an image quite well.

Style Loss modules implemented

- Around 4-5 style loss modules have been added in the network at various places
- We ran the network on images of our faces and various other pictures and stored the results.
- This module just stores an expected gram matrix of the style image representation initially
- When we pass the white noise image it compares the L2 norm between the gram matrix of the noise image representation at the current layer and the expected gram matrix at that same layer and adds this to the gradient while backpropagating.

Total Variational Loss

TV Loss module used

- We used an existing TV (Total Variational) loss module and added it to the start of our network.
- This helped reduce the variation between adjacent pixels and helped us achieve a lot of smoothness in the output image based on the weight assigned to the TV Loss module. We ended up using a 1e-3 weight for the TV Loss layer.
- The results seemed fine without this layer as well but it helped reduce the overall random noise in the output image which made it seem unreal and odd at times.

Sample Runs

Examples with Input Normalization

(And TV, Regularization Losses)

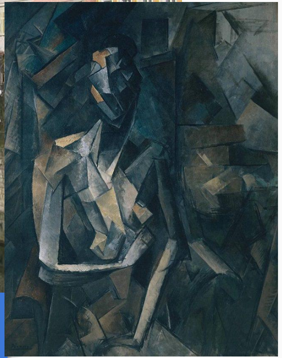
Example - 1

brad_pitt.jpg

blackwhite.jpg

red.jpg

Example - 2

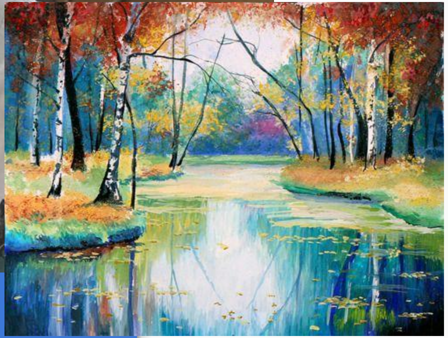


tubingen.jpg

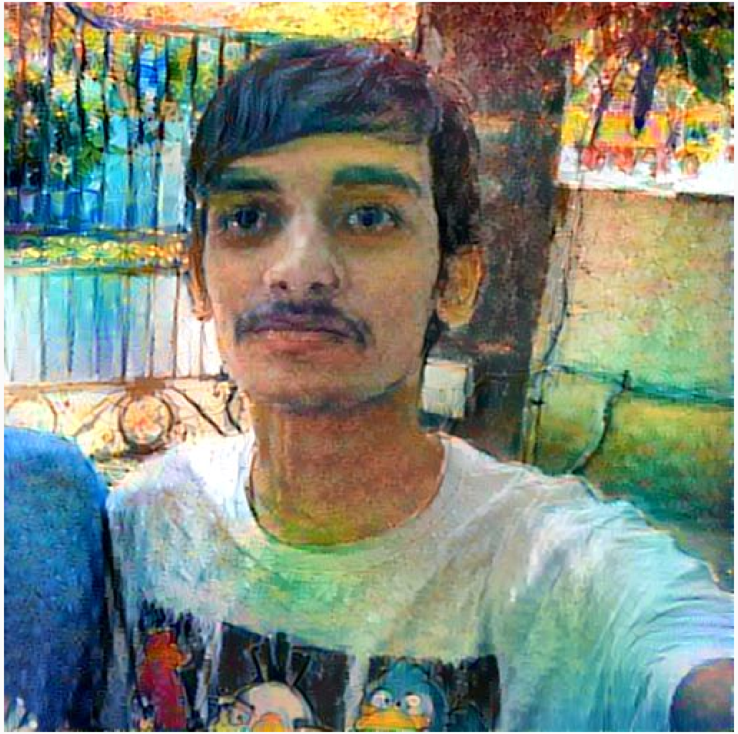
seated-nude.jpg

Example - 3

vishal.jpg



scenery4.jpg



Example - 4

house.jpg

style2.jpg

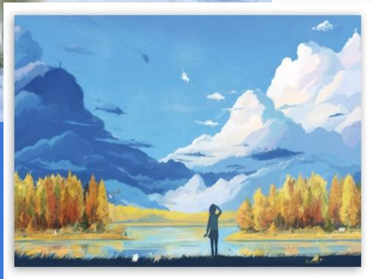
Example - 5

bear.jpg

walk.jpg

Example - 6

bear.jpg



scenery.jp
g

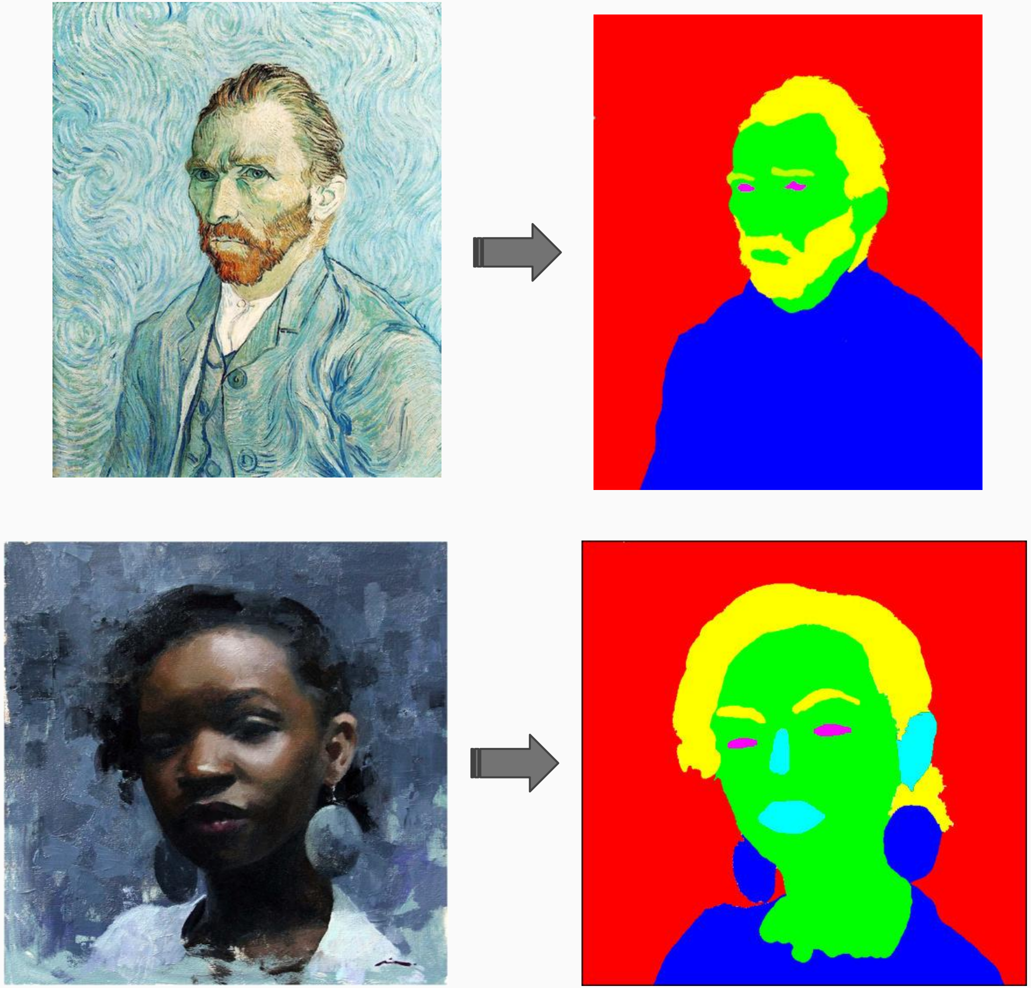
Example - 7

bear.jpg

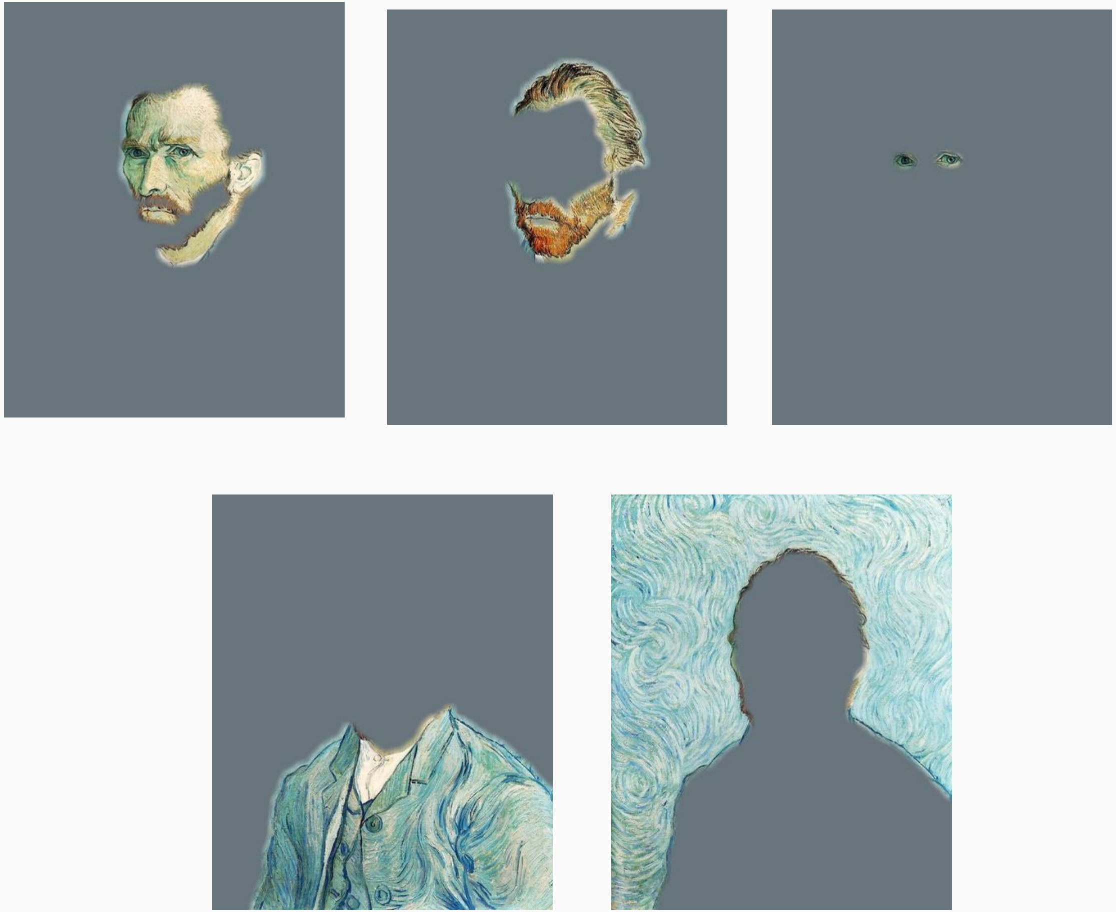
walk2.jpg

Segmented Style Transfer

Segmented Map Representations



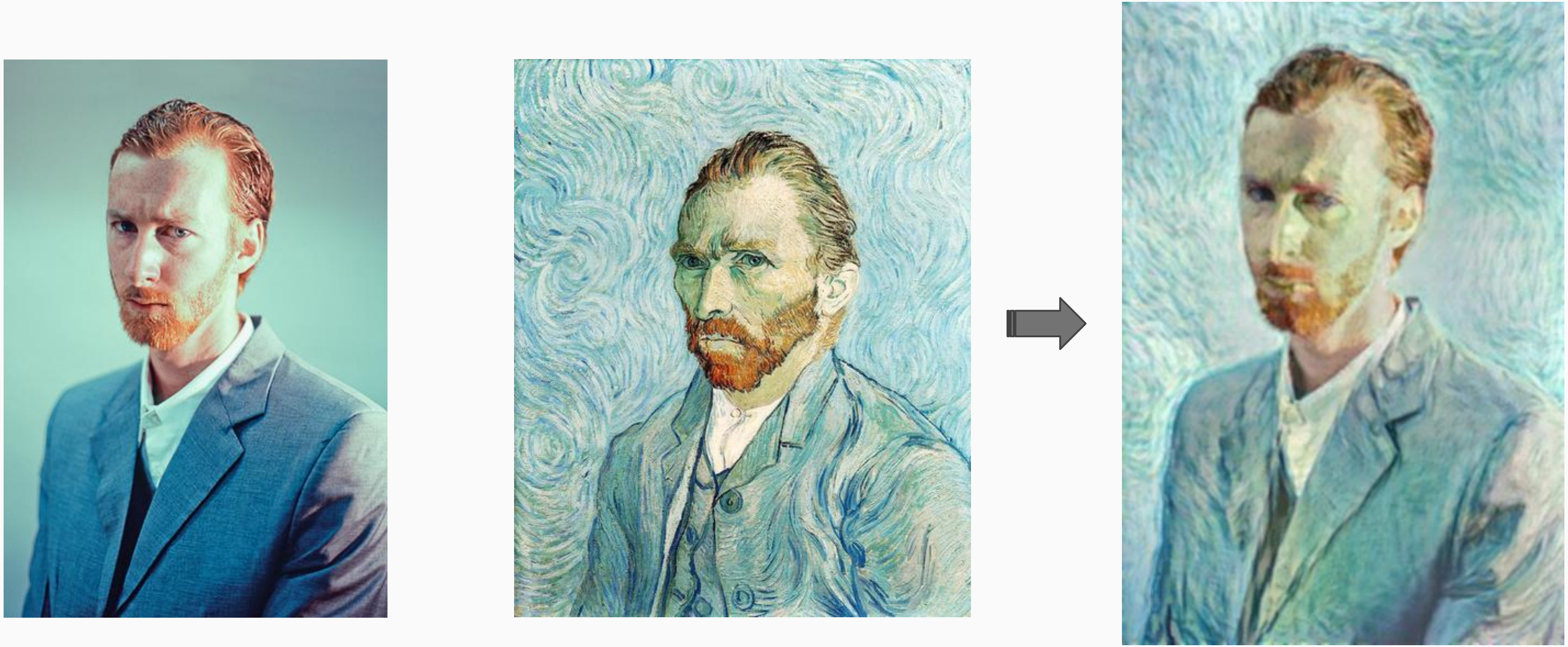
Split images based on segment maps



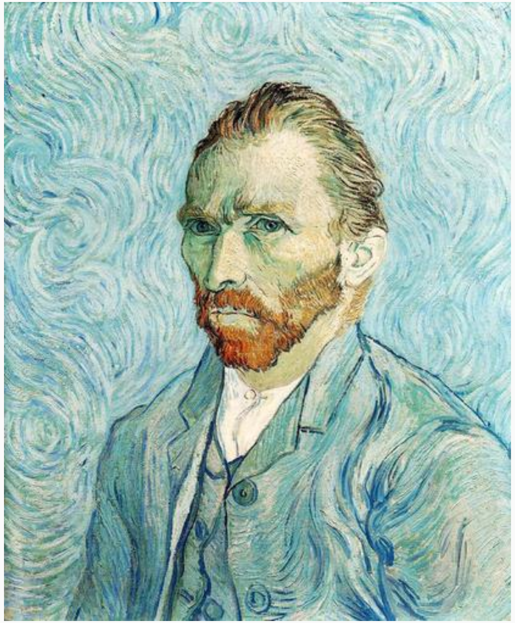
Segmented Style Transfer

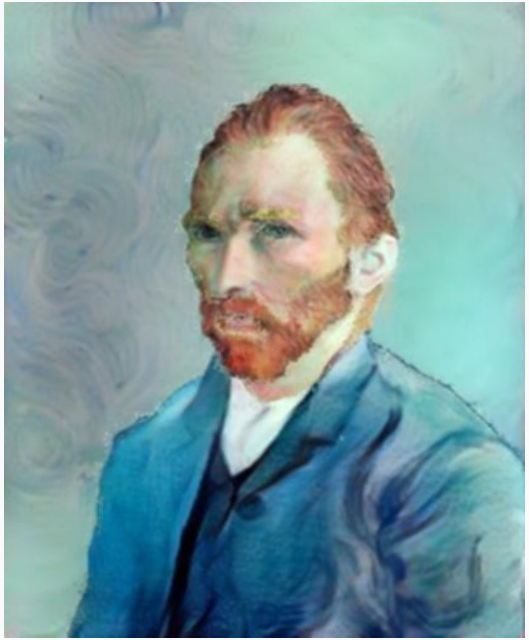
- Extrapolate masks to ensure there are no empty places left.
- Extract segments from the content image.
- Extract segments from the style image.
- Considering each pair at a time, perform style transfer individually.
- Combine generated images using the same extrapolated masks that were used in the initial segmentation.

Results - 1



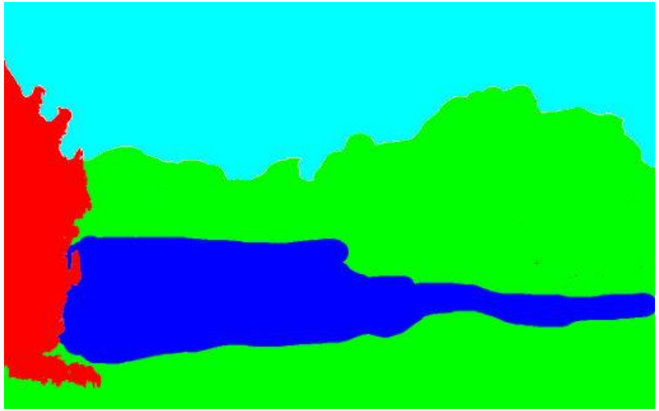
Results - 2

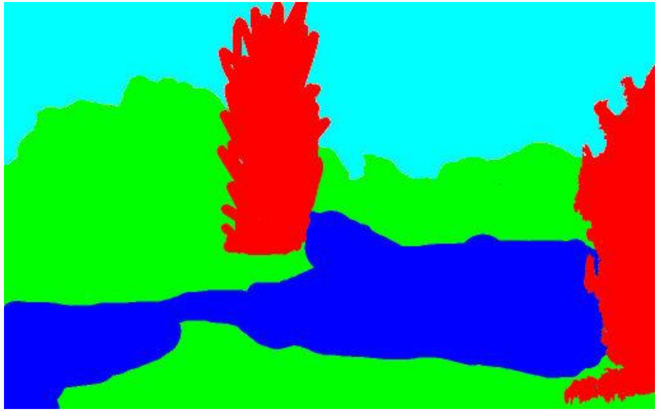




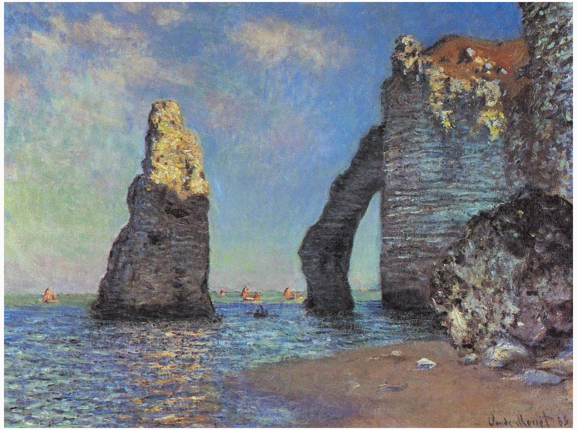
Results - 3

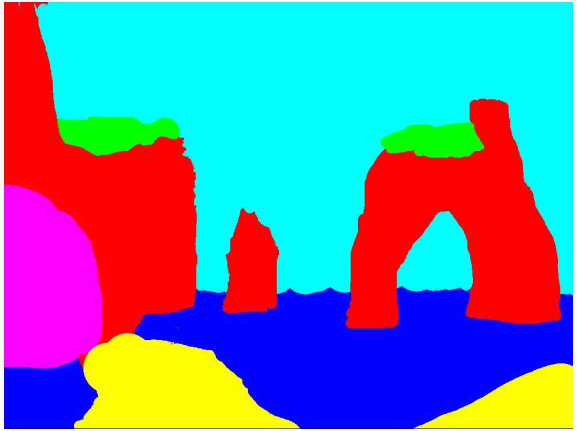
Results - 1

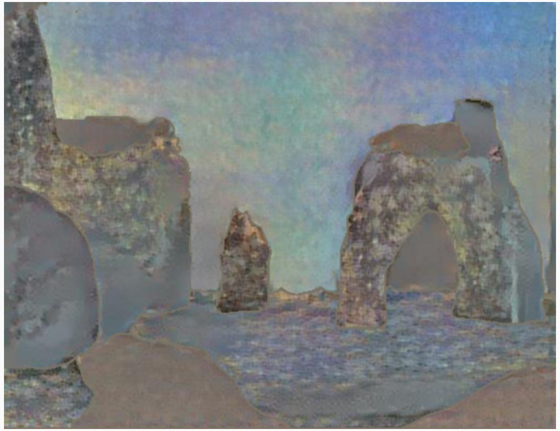




Results - 2







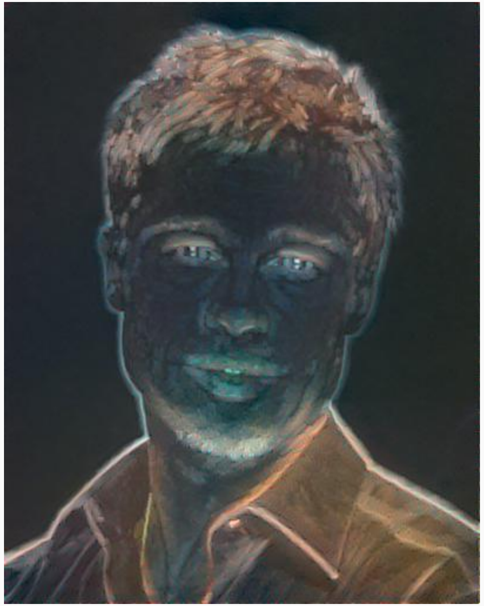
Pyramid based style transfer

Generate Laplacian Pyramid of the Content Image

Pyramid based style transfer

- Instead of using the same image at various places in the network for the content loss, use the images of the laplacian stack which provide richer detail.
- Finer level images from the higher end of the laplacian stack are used at the earlier layers of the network.
- Style losses are the same as earlier.
- Perform style transfer with reduced content weights since there are more of them present now.

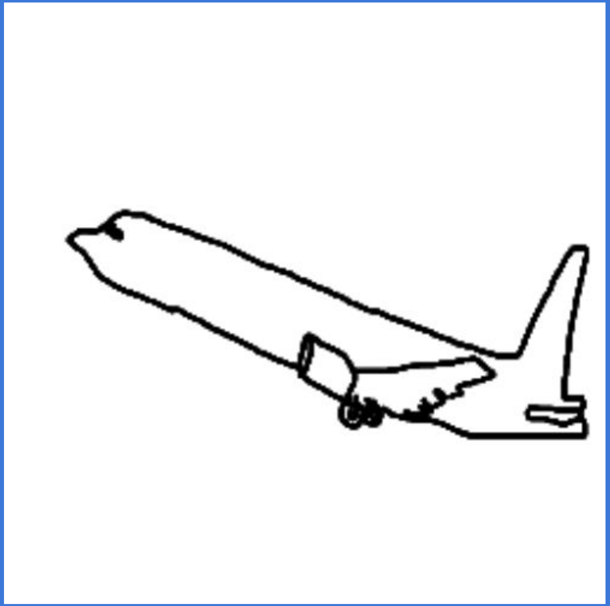
Results - 1



Results - 2

Sketch automatic coloring

Example - 1

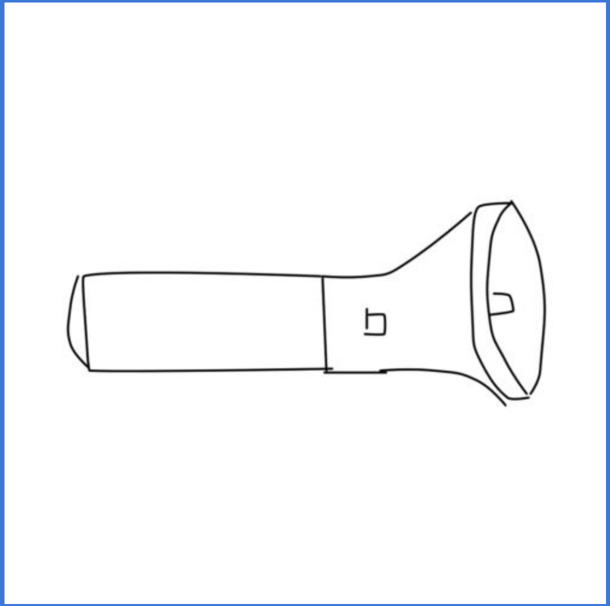


Input Airplane Sketch

SBIR retrieved airplane

Colored Sketch (We will overlay the input sketch on this image to get a good coloring)

Example - 2



Input Torch Sketch

SBIR retrieved torch

The distortion is mainly due to bad segmentation of the images

Thank you!