RUTGERS

THE STATE UNIVERSITY OF NEW JERSEY

FACIAL SKETCH TO COLORED IMAGES USING GANs

COMPUTER VISION – CS534

PROJECT MEMBERS:

PARTH GOEL

PRASHANT KANTH

RISHIKA BHANUSHALI

UNDER GUIDANCE OF: -

PROF. AHMED ELGAMMAL

TERM PROJECT

Facial Sketch to Colored Images using GANs

Abstract

In this project we implement an image translation model for generating realistic human faces from artist sketches using GANs. This project is an implementation and improvement on the work presented by [1], [2] and [4]. We have implemented [1], [2] and [4] and applied our improvement to get realistic images from thus proposed changes. We utilize the CUHK Sketch Database and FS2K Database for training and evaluation.

1. Introduction

Our main motivation behind this project was that it can be used by the police and security firms to recognize criminals and wanted people. As we know that many times if the CCTV recording isn't present then the police is forced to seek help from local witnesses and sketch artists to sketch the criminals. These sketches however accurate still leave a gap for doubt as they are ambiguous. This is because retrieval of criminal data or their identification using sketches as a basis might yield poor results. Searching of real photos will always be easier and more efficient because real images capture the identifiable features of the face much more accurately as compared to sketches.

In this project, we tackle the problem of generating color photorealistic images of human faces from corresponding hand-drawn sketches. We aggregate and align datasets CUHK and FS2K of facial sketches and corresponding real facial photos for training and evaluation. For our baseline we tried to follow 4 papers sketch2face [Julia Gong et. al.], pix2pix [Philip Isola et. al.], DeepFaceDrawing [Shu-Yu Chen et. al.] and Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks [Jun-Yan Zhu, Alexei A. Efros et. al.] (we use abbreviation CycleGANs for this paper in rest of the report).

All three members of the group have equal contribution in this project development. We divided one paper each among ourselves for exploration, implementation and improvements. Improvements were discussed together and applied to each of our papers. Due to challenges with dataset preparation, unfortunately we had to drop the implementation of DeepFaceDrawing [Shu-Yu Chen et. al.] and instead take up CycleGANs [Jun-Yan Zhu, Alexei A. Efros et. al.] which has a state-of-the-art architecture implementation of Cycle GANs. It is worth mentioning here that although original implementation in CycleGANs uses unpaired data, since we already have paired data, we implement this using paired data for the model to have more contextual knowledge.

Paper	**Member**
Pix2pix [Philip Isola et al.]	Parth Goel
CycleGANs [Jun-Yan Zhu et. al.]	Prashant Kanth
Sketch2face [Julia Gong et al.]	Rishika Bhanushali

Table 1.1 Summary of contribution

2. Related Work

GANs are generative models that learn a mapping from random noise vector z to output image y, $G: z \rightarrow y$. In contrast, conditional GANs learn a mapping from observed image x and random vector z, to y, $G: \{x, z\} \rightarrow y$. Generative Adversarial Networks (GANs) are algorithmic architectures that uses two neural networks, that compete against each other (thus the "adversarial") in order to generate new, synthetic instances of data that can pass for real data. Generator artificially generates fake images in an attempt to fool discriminator for real images. The goal of the discriminator is to identify which outputs it receives have been artificially created. Both models are trained in tandem and follow a cooperative zero-sum game framework to learn.

The **CGAN** is one of the first GAN innovations that made targeted data generation possible. It also adds labels to the discriminator input to distinguish data better.

The **CycleGAN** is an extension of the GAN architecture that involves the simultaneous training of two generator models and two discriminator models. One generator takes

images from the first domain as input and outputs images from the second domain, and the other generator takes images from the second domain as input and generates images for the first domain. Discriminator models are then used to determine how plausible the generated images are and update the generator models accordingly.

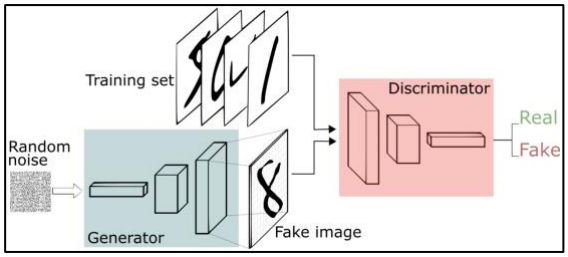


Fig 1.1 GAN Framework (Image Credit: Thalles Silva)

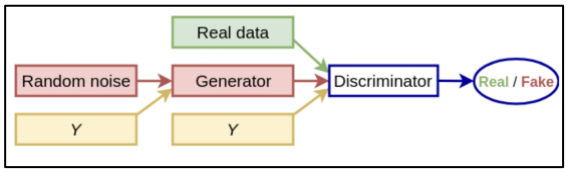


Fig 1.2 Conditional GAN Framework (Image Credit: Manish Nayak)

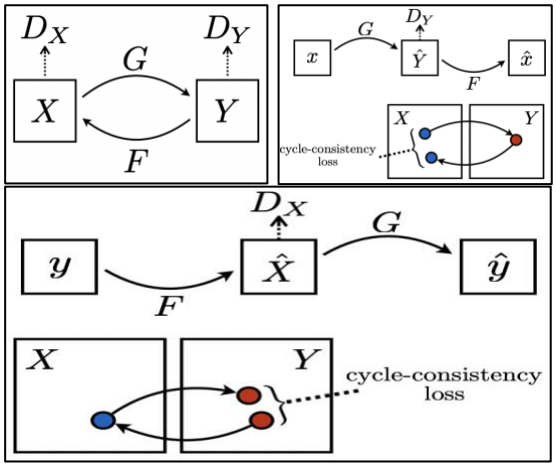


Fig 1.3 Cycle GAN Framework (Image Credit: Jun-Yan Zhu et al.)

3. Approach

Paper I: Image-to-Image Translation with Conditional Adversarial Networks / Pix2pix [Philip Isola et al.]

This paper adopts the U-net Architecture that is a convolutional network architecture for fast and precise segmentation of images. As named, it is a U-shaped architecture consisting of a specific encoder-decoder scheme. A U-Net architecture allows low-level information to shortcut across the network by severing the skip connections in the U-Net. The objective of a conditional GAN can be expressed as:

$$\mathcal{L}_{cGAN}(G,D) = \mathbb{E}_{x,y}[logD(x,y)] + \mathbb{E}_{x,z}[\log(1 - D(x,G(x,z)))]$$

where G tries to minimize this objective against and adversarial D that tries to maximize it. In addition, authors also use $L1$ loss as it encourages less blurring:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[||y - G(x,z)||_1]$$

Final objective thus becomes:

$$G^* = argmin_G max_D \mathcal{L}_{cGAN}(G,D) + \lambda \mathcal{L}_{L1}(G)$$

Paper II: sketch2face: Conditional Generative Adversarial Networks for Transforming Face Sketches into Photorealistic Images [Julia Gong et al.]

In this paper, modifications are done to the existing pix2pix model by introducing four variations of an iterative refinement (IR) model in an attempt to find the best conditional GAN for image-to-image translation. The IR architecture builds off of the conditional GAN framework and uses the same discriminator model as the baseline; however, it instead involves two generator segments that combine to form the generator network. The second generator learns to fine-tune the image obtained from the first generator. To train this model, a transfer learning is performed on top of the pix2pix baseline's weights. Specifically, both $G1$ and $G2$ are initialized to the baseline G weights, and the baseline D weights are used as the initial weights for D.

We only implement the best performing model out of the four IR model, which is IR Model and cGAN-Initial Loss. The cGAN-Initial Loss model has the same architecture and $L1$ loss; however, the first generator receives the cGAN Loss. Its final objective is:

$$G^* = argmin_G max_D \mathcal{L}_{cGAN}(G_1,D) + \lambda \mathcal{L}_{L1}(G_1) + \lambda \mathcal{L}_{L1}(G_2)$$

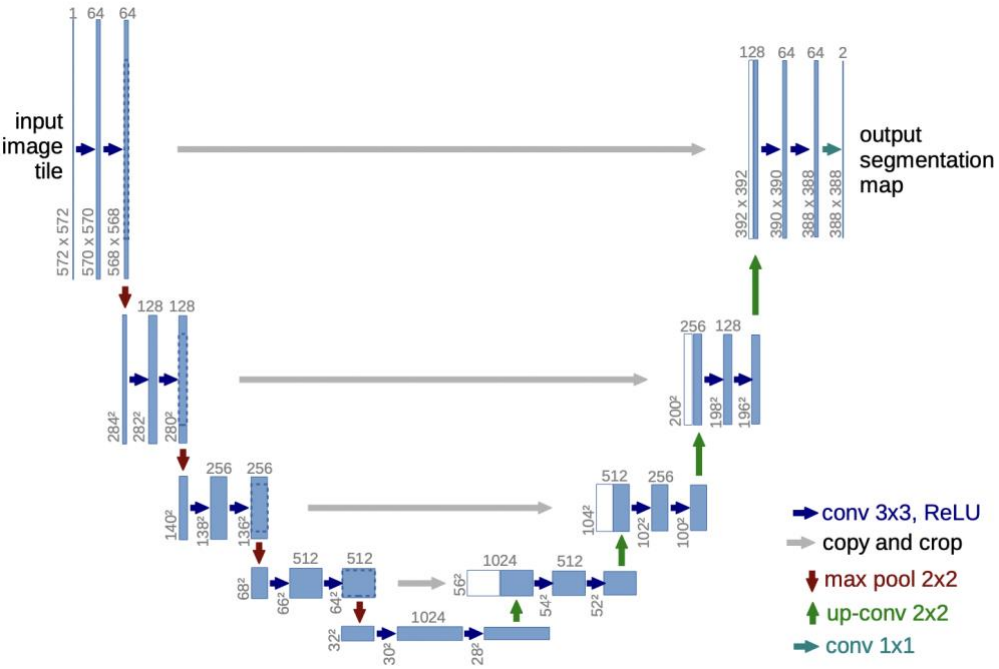


Fig 1.4 U-Net Architecture

Paper III: Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

The problem of this paper can be more broadly described as an image-to-image translation, converting an image from one representation of a given scene, x, to another, y. In this paper, the authors present a method that can learn to capture special characteristics of one image collection and figure out how these characteristics could be translated into the other image collection, all in the absence of any paired training examples. Although the paper implements unpaired image-to-image translation, since we already have paired images dataset, we implement it for paired images. The model includes two mappings $G: X \rightarrow Y$ and $F: Y \rightarrow X$. In addition, two adversarial discriminators D_y and D_x. There are two types of losses applied to each mapping $G: X \rightarrow Y$ and $F: Y \rightarrow X$; adversarial loss ($\mathcal{L}_{GAN}$) and cycle consistency loss ($\mathcal{L}_{cyc}$). Its final objective is:

$$\mathcal{L}(G, F, D_x, D_y) = \mathcal{L}_{GAN}(G, D_y, X, Y) + \mathcal{L}_{GAN}(G, D_x, Y, X) + \lambda \mathcal{L}_{cyc}(G, F)$$

$$G^*, F^* = argmin_{G,F} max_{D_x D_y} \mathcal{L}(G, F, D_x, D_y)$$

For model stabilization, negative log-likelihood is replaced by a least-square loss. To reduce model oscillation, discriminators are updated using a history of generated images from a buffer that stores 50 previously created images rather than the ones produced by the latest generators. Learning rate is kept same for first 100 epochs and then linearly decayed over the next 100 epochs.

4. Improvements

a. The dataset was limited to only one ethnic group, and hence to make the model more robust to diverse sketches we combined two datasets, CUHK and FS2K, and trained our model on that.

b. The original model architecture uses weight initialization of gaussian distribution with $\mu = 0 \; and \; \sigma = 0.02$. We tried different weight initialization $\mu = 1 \; and \; \sigma = 0.02$ in BatchNorm layer to help in training process.

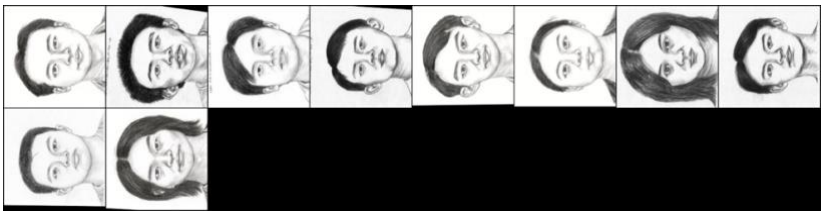
c. Modified BatchNorm Layers and Dropouts to get improved results.

d. We used photoshop to remove background from the all the images, so that model could ignore other details and learn facial images better.

e. We also experimented with different input size and architecture layout of the model.

5. Results

Paper 1 [Pix2pix]:



Input Sketch

Source Truth

Generated Result

Fig 1.5 Paper 1 result [256x256 input dim] weight initialized $\mu = 1$ and $\sigma = 0.02$

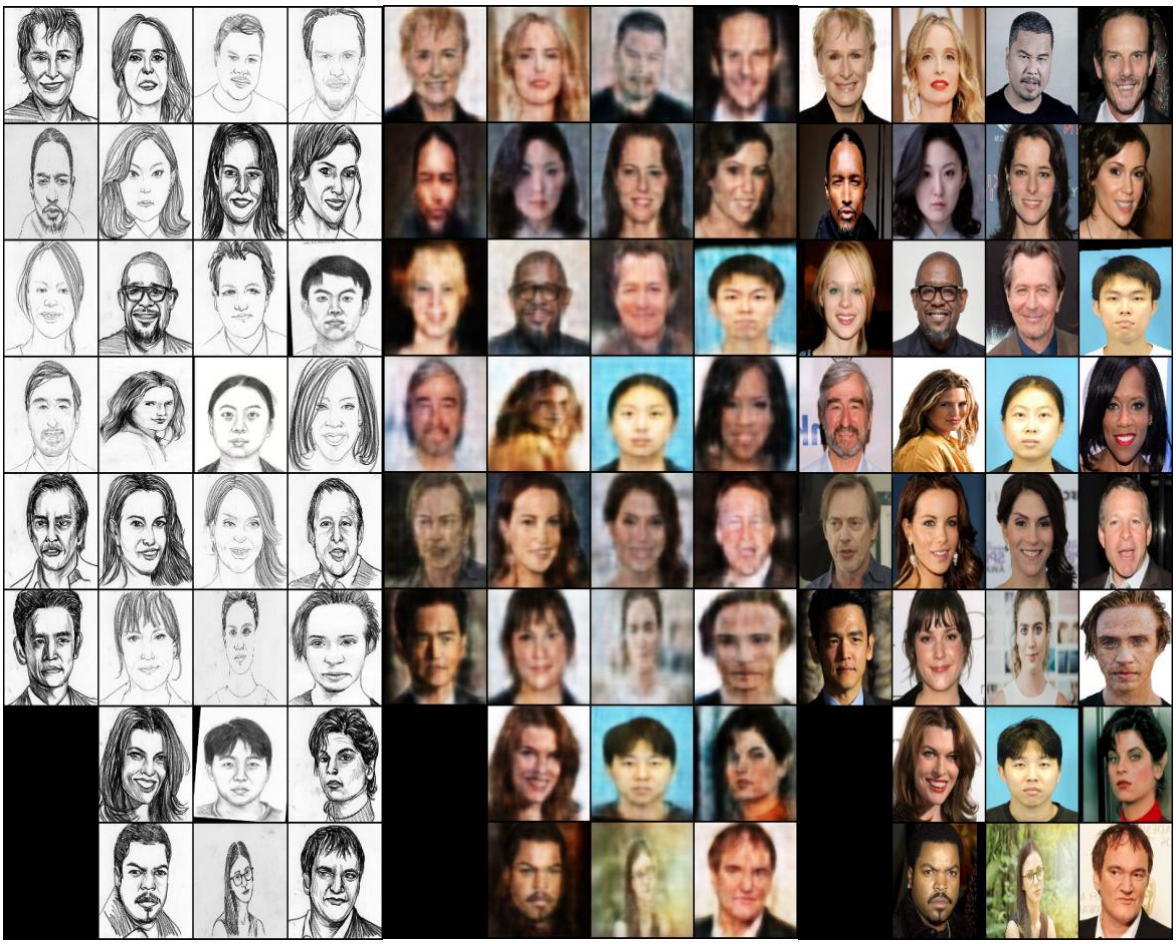
Fig 1.6 Paper 1 result [256x256 input dim] Test Result on sketches from same dataset and outside dataset

Input Sketch | Generated Result | Source Truth

Fig 1.7 Paper 1 result [128x128 input dim] Test Result

Input Sketch Generated Result Source Truth

Fig 1.8 Paper 1 result [128x128 input dim] Test Result



Input Sketch Generated Result Source Truth

Fig 1.9 Paper 1 result [128x128 input dim] merged dataset CUHK and FS2K

Input Sketch Generated Result Source Truth

Fig 1.10 Paper 1 result [256x256 input dim] merged dataset CUHK and FS2K, removed background

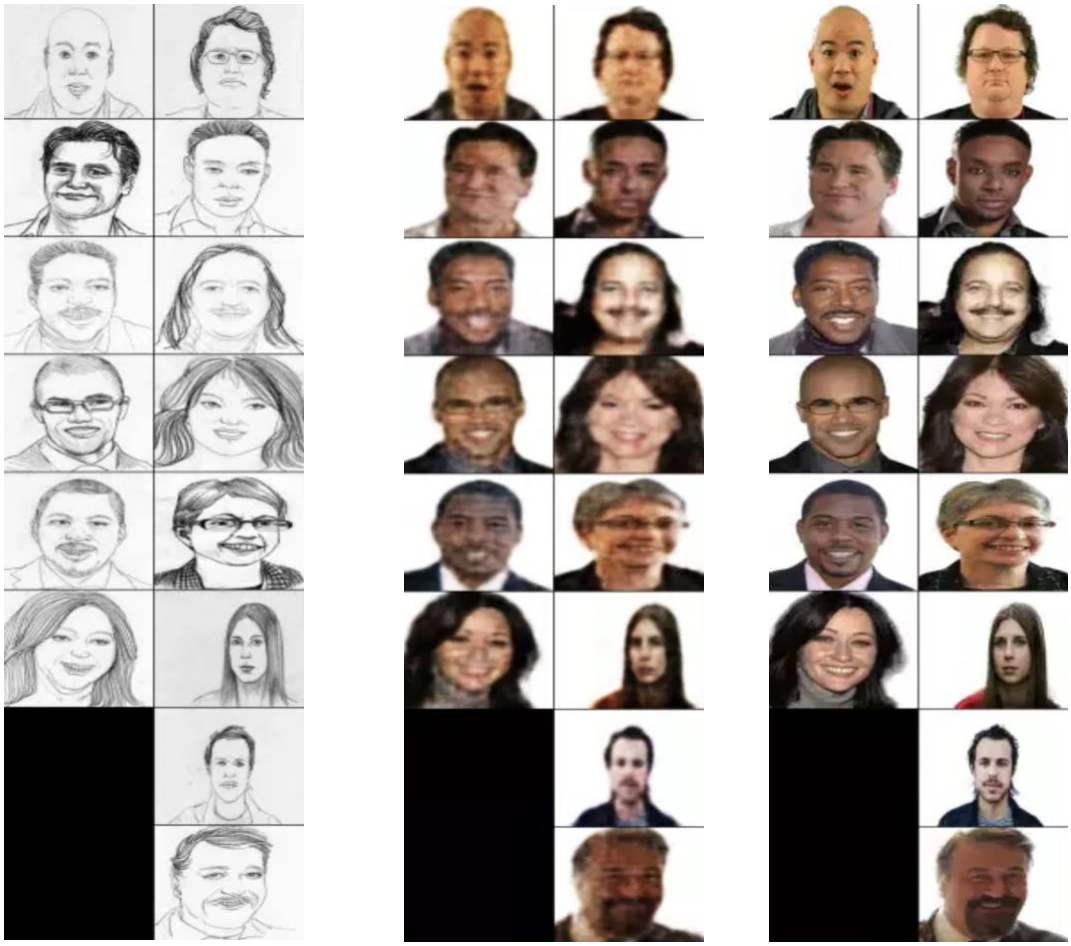


Fig 1.11 Paper 1 result [256x256 input dim] merged dataset CUHK and FS2K, removed background

Paper 2 [sketch2face]:

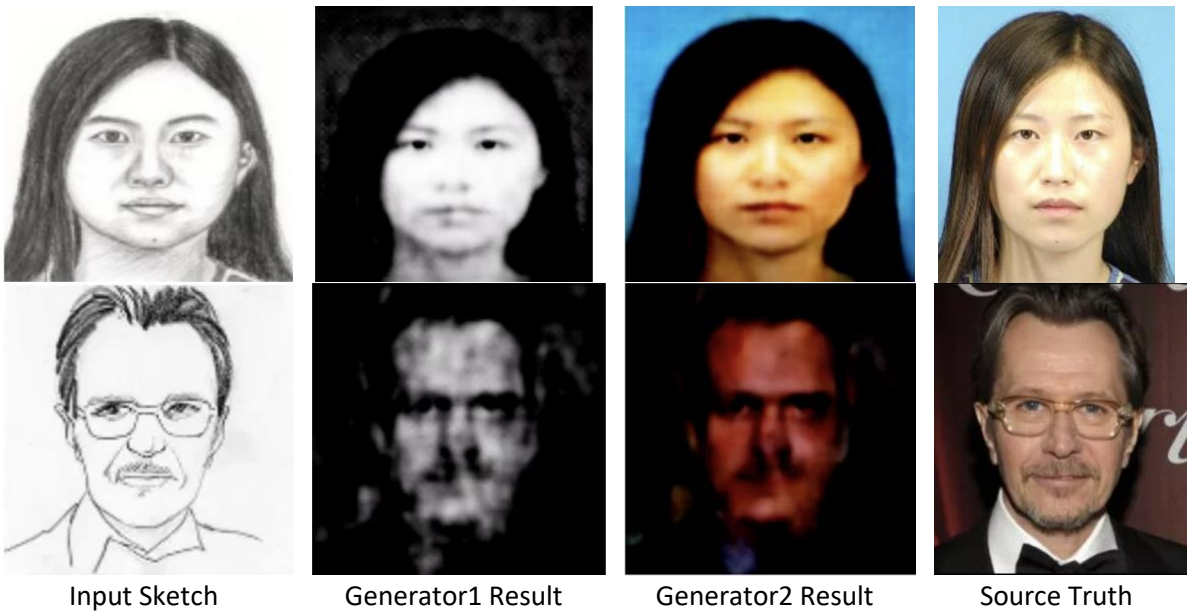
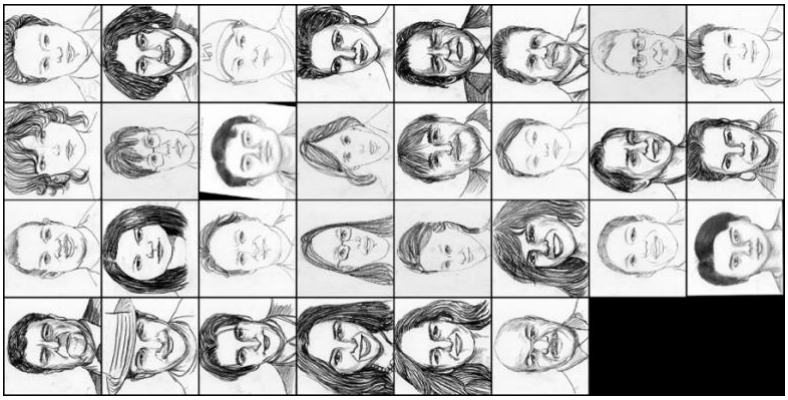


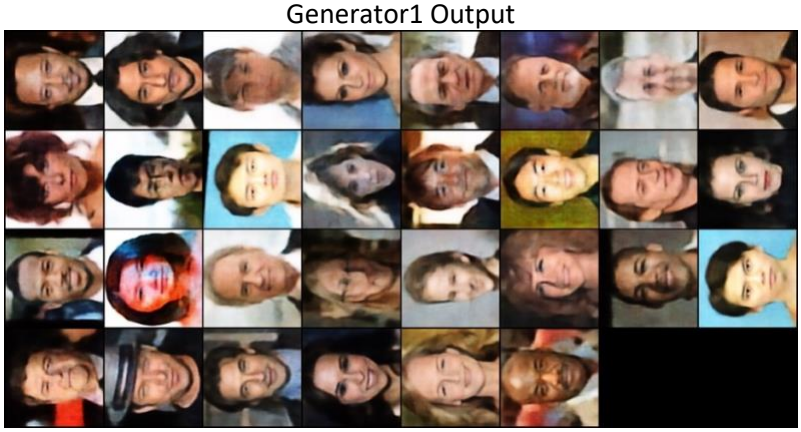
Fig 1.12 Paper 2 Test result [128x128 input dim] merged dataset CUHK and FS2K, with background



Input Sketch

Source Truth

Generator1 Output



Generator2 Output

Fig 1.12 Paper 2 result [128x128 input dim] merged dataset CUHK and FS2K, with background

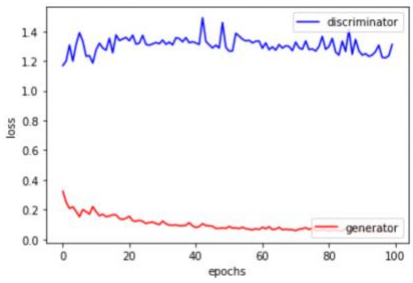
Fig 1.13 Paper 2 result [256x256 input dim] merged dataset CUHK and FS2K, without background and colored generator 1

Paper 3 [CycleGAN]:

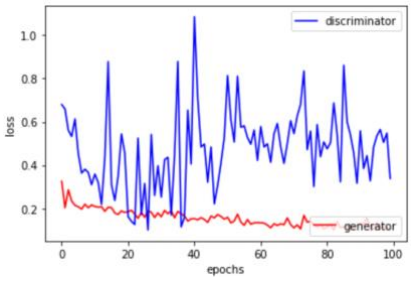
Fig 1.14 Paper 3 result [256x256 input dim] merged dataset CUHK and FS2K, without background, also showing bad result due to dark shading and partial removal of background

Input Sketch | A2B (Sketch2Color) Result | B2A (Color2Sketch) Result | Source Truth

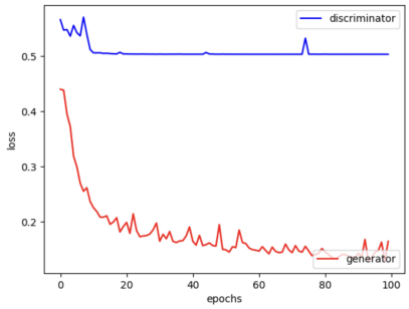
Fig 1.15 Paper 3 result [256x256 input dim] merged dataset CUHK and FS2K, without background



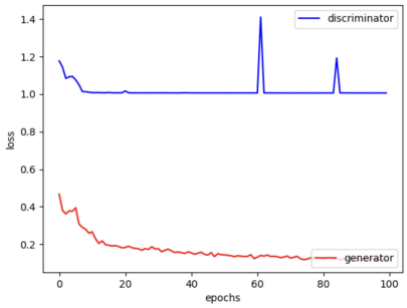
Pix2pix 128x128 CUHK dataset



Pix2pix 256x256 CUHK dataset

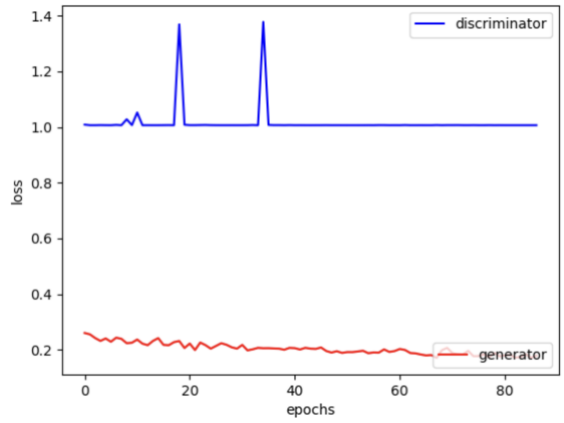


Pix2pix 128x128 Merged Dataset

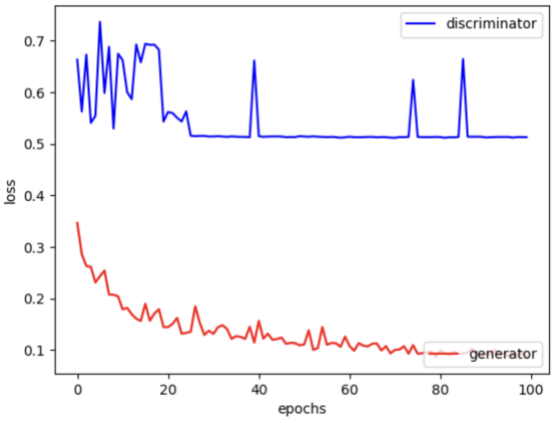


Pix2pix discriminator loss increased by factor of 0.5, less dropout and batch normalization

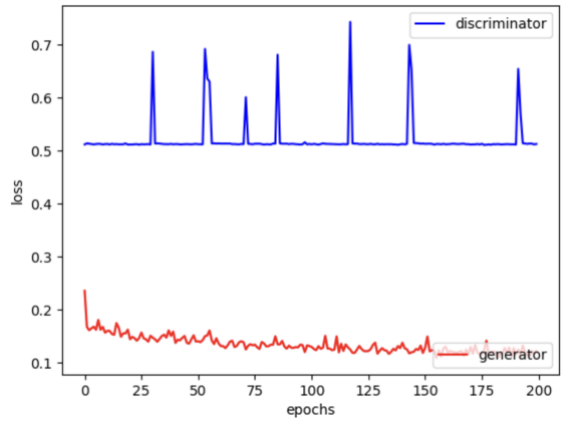
Plot 1.1 Pix2pix Loss for discriminator and generator



Sketch2face 128x128



Sketch2face 256x256 white background



CycleGAN 256x256 white background

Plot 1.2 Sketch2face and CycleGAN (Generator Loss = G1+G2), Discriminator Loss: For Sketch2Face single discriminator, For CycleGAN two discriminators (Discriminator Loss = D1+D2)

6. Discussion and conclusions

From all the papers that we implemented and applied modifications, we found that sketch2face was a better implementation considering large dataset. CycleGAN performed poor with a lingering gap between the generated result and true labels. It does seem to overfit on the training data. Pix2pix performs good on the CUHK dataset.

While implementing all the algorithms we came across a lot of hardware-based challenges which we had to overcome because we had limited GPU access in the Rutgers ilab. We had to reduce the batch size and the number of residual blocks from 9 to 6. We also had to convert our model from double data type to float for faster calculations.

From all the experiments carried out by us through implementing different state of the art algorithms in this field, one thing for certain is that no algorithm can be used as a general approach to convert any facial sketch into the real image, since artistic styles of the sketches like shading regions and ink density etc., in the database matter a lot for the model's performance. Since every artist has his own sketching style, we cannot generalize this by any dataset.

However, we got significantly good results using Sketch2Face algorithm and propose a solution to carry out the main application of this project that is to help the police recognize and find missing as well as wanted people easily, for whom only the facial sketches are available in the police databases. To carry this out we can train our model on each individual sketch artists that the police have and then apply those models to convert facial sketches drawn by them to their real image counterparts.

References

[1] Philip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros: Image-to-Image Translation with Conditional Adversarial Networks.
[2] Julia Gong, Matthew Mistele: sketch2face: Conditional Generative Adversarial Networks for Transforming Face Sketches into Photorealistic Images.
[3] Shu-Yu Chen, Wanchao Su, Lin Gao, Shihong Xia, Hongbo Fu: DeepFaceDrawing: Deep Generation of Face Images from Sketches.
[4] Jun-Yan Zhu, Taesung Park, Philip Isola, Alexei A. Efros: Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks.
[5] StackOverflow
[6] Udemy: Computer Vision A-Z, Practical Deep Learning with Pytorch