Photo to Sketch convertor

Name: Het Bharatbhai Patel
Enrolment No. - 18012011058
Model : CNN
Architecture: UNet

1. Introduction

Nowadays, people like to capture images through their phones and especially selfie. On that selfie, they usually apply multiple kind of filter, and one of that popular is sketch.

Sketching from an selfie image is a difficult and tedious task. So, for overcome the problem and automatically creating sketch, we have made this project. This project is about converting any colorful or grayscale image into sketch.

2. Environment

Google Colab - Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

I have used this environment during the preparation of whole project.I choose to work on Colab as it allow to work on notebook which is on cloud with around free GPU and 2GB RAM. GPU make image training faster.

3. Libraries

Libraries	Why I Used?
Numpy	For performing different kind of high level mathematical operations on arrays
OpenCV	For converting Images in dataset to array
Matplotlib	For visualising training and predicted data
Keras	For model training and output prediction

4. Dataset

I had gathered the dataset from Kaggle.

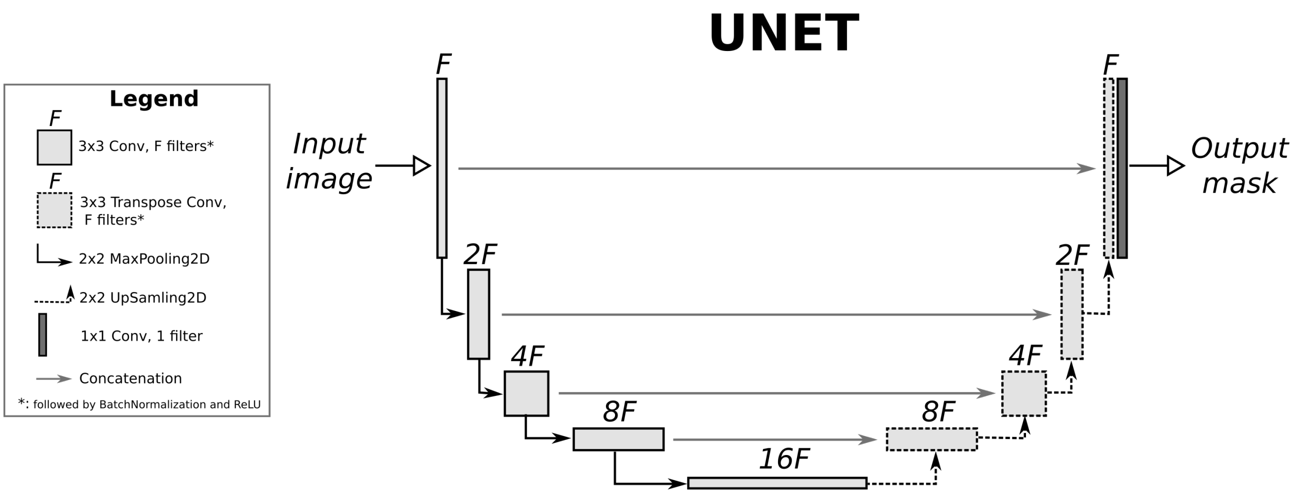
https://www.kaggle.com/arbazkhan971/cuhk-face-sketch-database-cufs

The dataset contains around 188 color as well as sketch images. Size of images are 256x256. Color images are converted to RGB(3 channels) whereas sketches are converted in grayscale(1 channel) and both are scaled to 0-1. All the images includes human faces. Each image is flip and rotated in 4 direction which gives 8 different images from the same. So in total I got 1504 images for training and testing.

Training Images - (1400,256,256,3)
Training Sketches - (1400,256,256,1)
Testing Images - (104,256,256,1)
Training Sketches - (104,256,256,1)

5. Model

I have use CNN model with UNet Architecture.

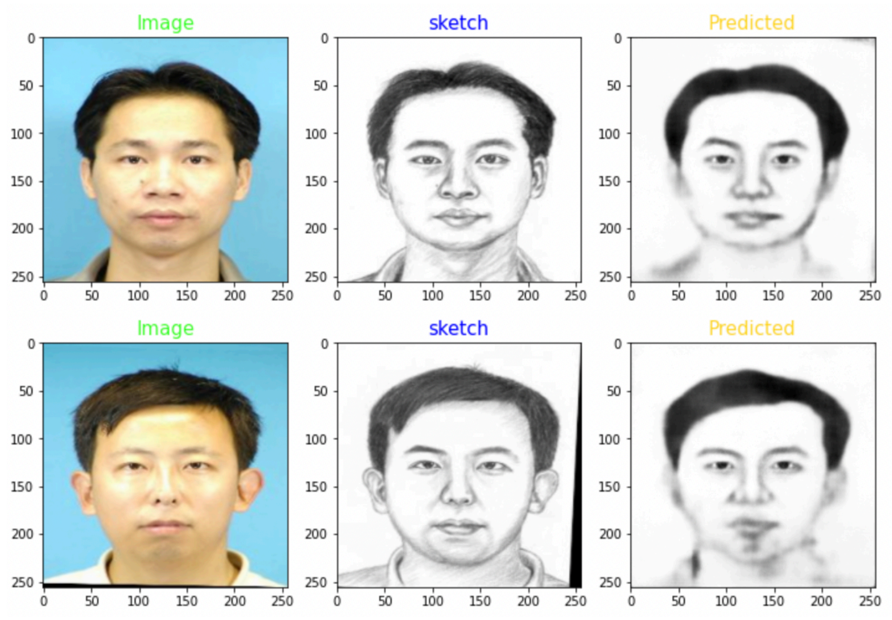


Basically, UNet contains UNet Blocks a Conv2d layer -> ReLU -> Batch normalisation layer -> Conv2d layer. The train images with shape (256,256,3) is passed as input to first UNet Block. The block output is stored and maxpooling of 2x2 is applied.this thing repeats 4 times. The previously stored outputs are concatenated with recent outputs in reverse order and upsampling2D is applied. Then comes UNet Block.On the other side I putted training sketches with shape (256,256,1).

I choose to work with UNet after comparing it with the previous Kaggle notebook works where auto encoder with random layers are used. which provides low accuracy.

6. Results

Autoencoder:

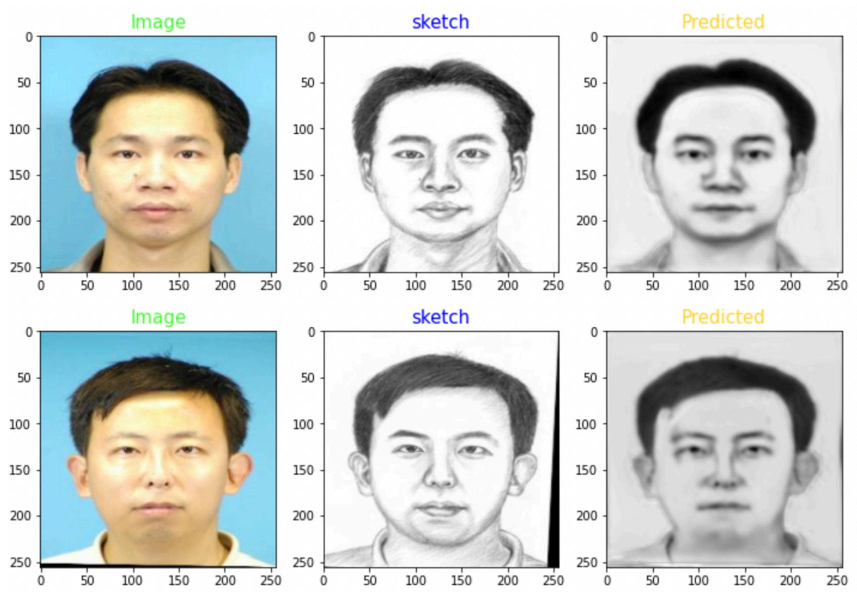


Epoch-100

loss - 0.0809

Accuracy - 0.2817

UNet:



Epoch-61 (EarlyStopping(patience=20))

loss - 0.0322

Accuracy - 0.2836

Reference:

https://www.kaggle.com/theblackmamba31/photo-to-sketch-using-autoencoder/notebook