AIP COURSE PROJECT

K.Kalyan(21361), Ramesh Babu(20950)

Paper 1 : **Convolutional Sketch Inversion (https://arxiv.org/pdf/1606.03073.pdf)**

Paper 2 **: Controlling Deep Image Synthesis with Sketch and Color (https://arxiv.org/pdf/1612.00835.pdf)**

Paper 1 : Convolutional Sketch Inversion

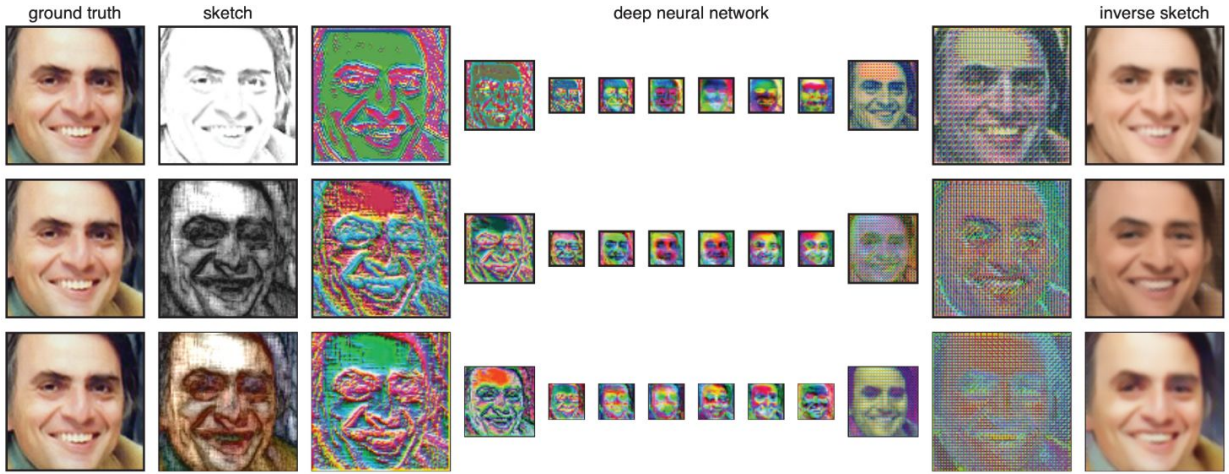
- Goal : Develop a deep learning model that can invert a hand-drawn sketch into a realistic image.

- To do this, we used a convolutional neural network (CNN) that is trained on a dataset of sketches and their corresponding images.
- The code for this paper has written from scratch(using inbuilt sklearn and pytorch libraries)
- Original paper had trained over 2 lakh images, but due to colab constraints we trained over 3600 images for 50 epochs.

Dataset used: CUHK Face Sketch (CUFS) database

Training Images : 3600

Testing Images : 236



CNN Architecture

Layer	Type	in_channels	out_channels	ksize	stride	pad	normalization	activation
1	con.	1 or 3	32	9	1	4	BN	ReLU
2	con.	32	64	3	2	1	BN	ReLU
3	con.	64	128	3	2	1	BN	ReLU
4	res.	128/128	128/128	3/3	1/1	1/1	BN/BN	ReLU
5	res.	128/128	128/128	3/3	1/1	1/1	BN/BN	ReLU/+x
6	res.	128/128	128/128	3/3	1/1	1/1	BN/BN	ReLU/+x
7	res.	128/128	128/128	3/3	1/1	1/1	BN/BN	ReLU/+x
8	res.	128/128	128/128	3/3	1/1	1/1	BN/BN	ReLU/+x
9	dec.	128	64	3	2	1	BN	ReLU
10	dec.	64	32	3	2	1	BN	ReLU
11	con.	32	3	9	1	4	BN	tanh

Loss Function

The loss consists of 3 components:

The first component is the standard Euclidean loss for the targets and the predictions, also called the pixel loss.(lp)
The second component is the Euclidean loss for the feature-transformed targets and the feature-transformed predictions, also known as the feature loss.

$$\ell_f = \frac{1}{n} \sum_{i,j,k} \left(\phi(t)_{i,j,k} - \phi(y)_{i,j,k} \right)^2$$

The third component is the total variation loss for the predictions, which measures the smoothness of the output image.

$$\ell_{tv} = \sum_{i,j} \left((y_{i+1,j} - y_{i,j})^2 + (y_{i,j+1} - y_{i,j})^2 \right)^{0.5}$$

The total loss is then given by , weighted combination of these components

$$\ell = \lambda_p \ell_p + \lambda_f \ell_f + \lambda_{tv} \ell_{tv}$$

Results:

These results were obtained we trained on 3600 images for 50 epochs(3hrs).

Results

Average PSNR(db) of all test images is : 42.7

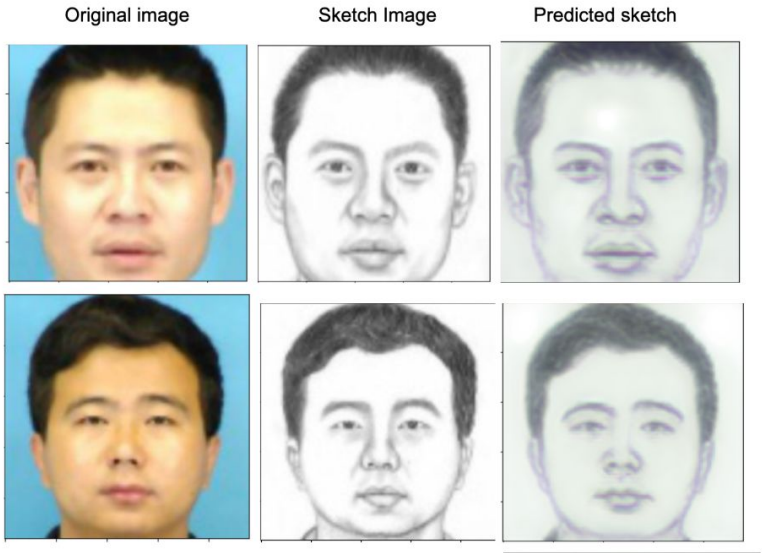
Average SSIM of all test images is : 0.64

The reported average PSNR of 42.7 dB indicates that, on average, the difference between the original and the reconstructed images is relatively small. A higher PSNR value generally implies better reconstruction quality, but it is worth noting that PSNR is known to be less sensitive to perceptual differences than other metrics.

The reported average SSIM of 0.64 indicates that the reconstructed images preserve some of the structural similarities and luminance of the original images. SSIM is often considered a more perceptually meaningful metric than PSNR, as it takes into account the human visual system's sensitivity to changes in structure, contrast, and brightness.

Experiments

We tried predicting sketches from images, which surprisingly did pretty well on the same network and same dataset.



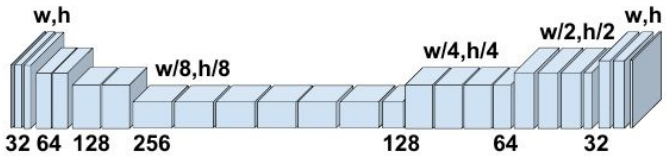
Scribbler: Controlling Deep Image Synthesis with Sketch and Color

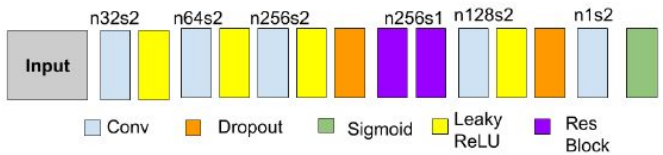
Goal : Develop a deep adversarial image synthesis architecture that is conditioned on sketched boundaries and sparse color strokes to generate realistic cars, bedrooms, or faces.

Dataset:

Dataset used: CUHK Face Sketch (CUFS) database
Training Images : 3600
Testing Images : 236

Network Architecture





- Generator - Downsampling -> Residual blocks -> Upsampling
- Discriminator - Fully Convolutional Network

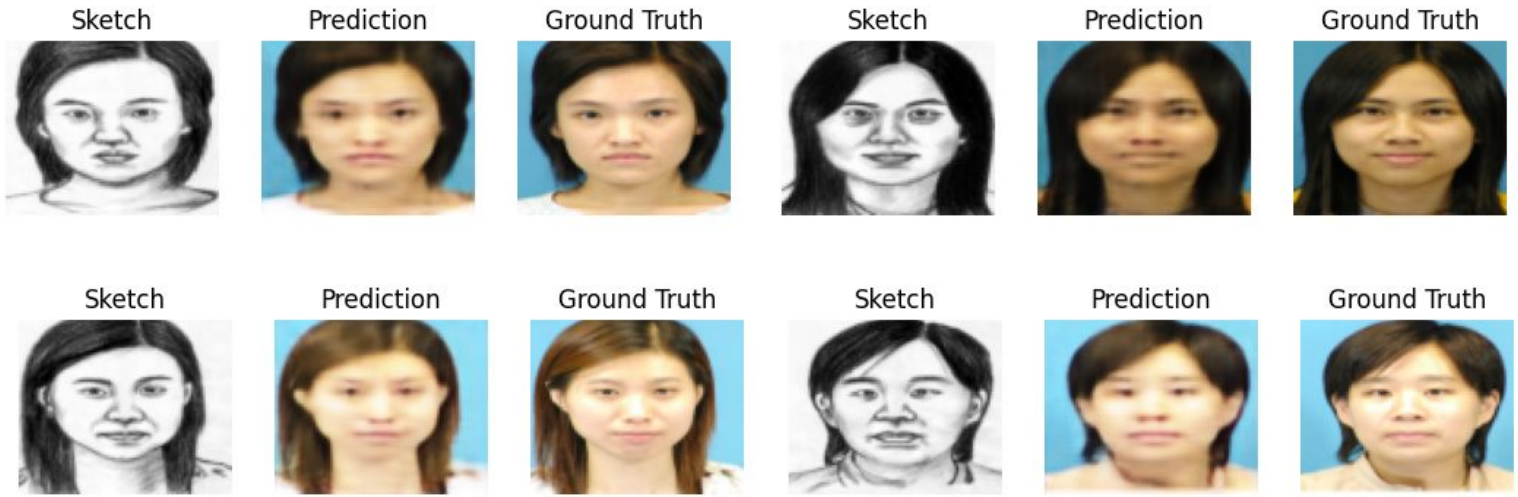
Objective Function

Weighted combination of all these 4 losses

1. Pixel loss
2. Feature loss
3. Variational loss
4. Adversarial loss -

$$\mathbf{L_{adv}} = -\sum log\mathbf{D}_{\phi}(\mathbf{G}_{\theta}(x_i))$$

Observations



Observations

Average PSNR(db) of all test images is : 53.58

Average SSIM of all test images is : 0.90

We can see that the images generated are of better quality when compared to that of using CNN model

Future Work

- Network is to be trained on a larger dataset containing various kinds of images
- The proposed paper claims that the network is a feed forward neural network which means the changes in the sketches result in a different Prediction in real time. This needs to be implemented
- This network is to be trained on input data containing Sketches combined with color strokes which results in predictions generated with those colors as claimed by the paper

THANK YOU