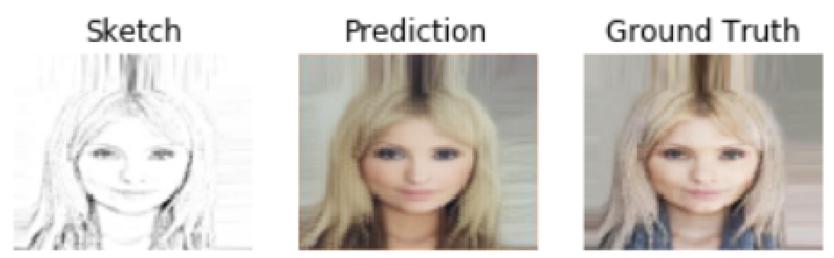
Sketchback

Badr AlKhamissi and Yousef Nassar
Computer Science, The American University in Cairo



Abstract

In this research we are using deep convolution neural networks to synthesis photo-realistic images from pencil sketches. This research focuses on sketch inversion of human faces and architectural drawings of buildings. We trained the model using a dataset generated from a large database of face images, and then fine-tuned the network to fit the purpose of architectural sketches.

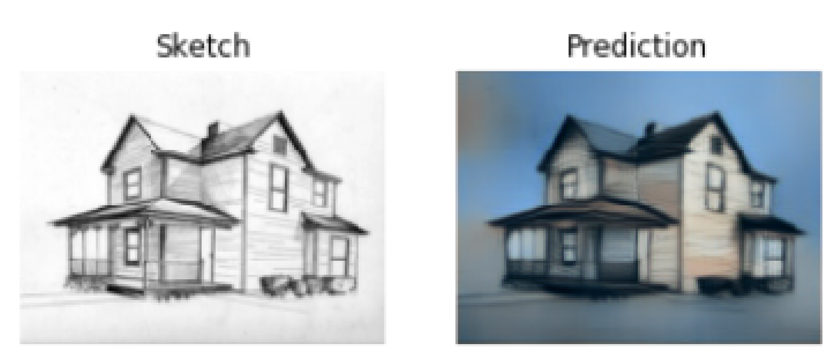
Introduction

Generating photo-realistic images from hand-drawn sketches has been an exciting area of research in the past couple of months. Sketches play a large role in architecture; as architects use their sketches to visualize, and capture how their work will look in reality. In the past year, a lot of research has been conducted on sketch inversion; naming a few: [1] has applied sketch inversion techniques to sketches of faces, for use in the fine arts and forensics. [2] expanded this technique to sketches of cars, and bedroom interiors, and added user-inputted color controls. This research introduces sketch inversion techniques to architectural sketches, in an attempt to empower architects to convey and visualize their work.

Data

The following datasets were used to train, test, and validate our model:

- *Large-scale CelebFaces Attributes (CelebA) dataset.* CelebA dataset contains 202,599 celebrity face images of 10,177 identities. It covers large pose variation and background clutter. We used this dataset to train the network.
- *ZuBuD Image Database.* The ZuBuD dataset is provided by the computer vision lab of ETH Zurich. It contains 1005 images of 201 buildings in Zurich; 5 images per building from different angles.
- *CUHK Face Sketch (CUFS) database.* This dataset contains 188 hand-drawn face sketches and their corresponding photographs. We used the CUHK student database for testing our model.
- We finally used various building sketches from Google Images for testing

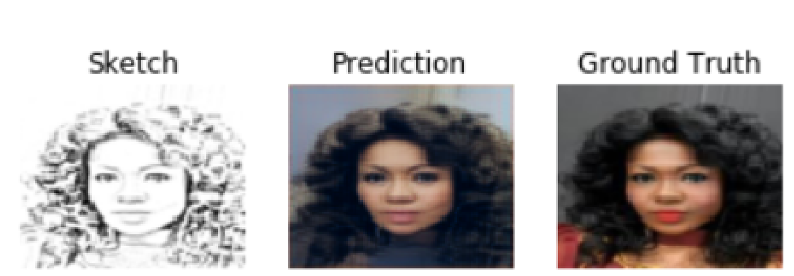


Sketching

The datasets were simulated, i.e the sketches were generated from images using the following methods (with the exception of the CUHK dataset, which contains sketches and the corresponding images):

- XDoG (Extended Difference of Gaussians)
- Pencil Sketchify
- Neural Style Transfer

Furthermore, due to the low number of images of buildings available, we applied various augmentations on the ZuBuD dataset to produce more images.



Network Architecture

We used the same network architecture as [1], as shown below. For model optimization we used Adam, using learning rate 1e-4.

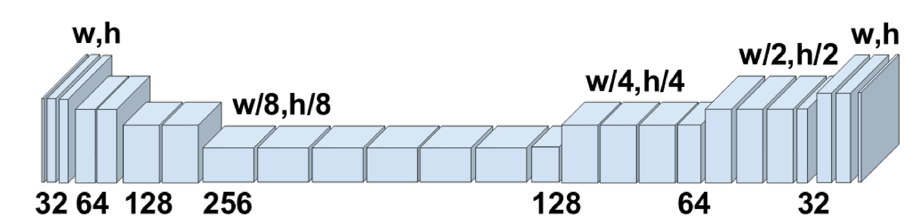


Figure 1: The encoder-decoder generator design was used, with down-sampling steps, followed by residual layers, followed by up-sampling steps.

Loss Functions

The Loss function was computed as the weighted average of three loss functions; namely: the pixel loss, the total variation loss, and the feature loss.
The pixel loss was computed as:

$$L_p = \sqrt{\frac{\left(\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{c}(t_{i,j,k} - p_{i,j,k})^2\right)}{nmc}} \quad (1)$$

Where t is the true image, p is the predicted image, and n,m,c are the height, width, and number of color channels respectively.
The feature loss was computed as:

$$L_f = \sqrt{\frac{\left(\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{c}(\phi(t_{i,j,k}) - \phi(p_{i,j,k}))^2\right)}{nmc}} \quad (2)$$

The total variation loss [3] was used to encourage smoothness of the output, and was computed as:

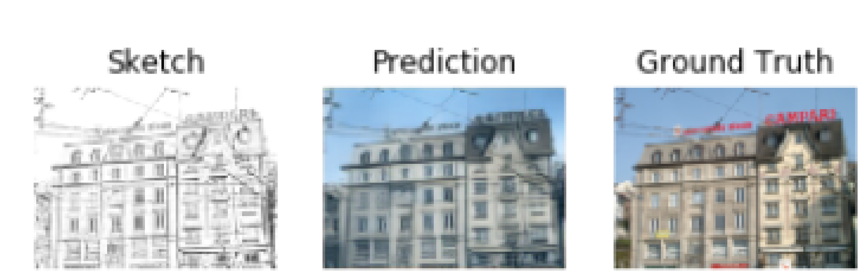
$$L_v = \sum_{i,j}(p_{i+1,j} - p_{i,j})^2 + (p_{i,j+1} - p_{i,j})^2 \quad (3)$$

Where $\phi(x)$ is the output of the fourth layer in a pre-trained model (VGG16 relu_2_2) to feature transform the targets and predictions.
The total loss is then computed as

$$L_t = w_p L_p + w_f L_f + w_v L_v \quad (4)$$

For the present application,

$$w_f = 0.001, w_p = 1, w_v = 0.00001 \quad (5)$$



Results

Training the network for around 17 epochs on the CelebA dataset, and then fine-tuning on the ZuBuD dataset for 25 epochs, we were able to produce realistic images from sketches produced using the same style / method as the the training data. The network was also hand-tested on several testing hand-drawn sketches of buildings, and realistic results were produced, albeit less realistic than those produced by sketchification. It was also observed that of hand-drawn sketches, those of similar style to the sketchified images produced the more realistic results.

Forthcoming Research

The network's results could possibly be improved in several ways in the future.

- Adding adversarial loss to the network.
- Using sketch anti-roughing to unify the styles of the training and input sketches.
- Passing the sketch results to a super-resolution network to improve image clarity.
- Increasing the image size of the training data.
- Training with a larger building dataset with a variety of sketch styles to improve the generality of the network

References

[1] Yagmur Güçlütürk, Umut Güçlü, Rob van Lier, and Marcel A. J. van Gerven.
Convolutional sketch inversion.
CoRR, abs/1606.03073, 2016.

[2] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays.
Scribbler: Controlling deep image synthesis with sketch and color.
CoRR, abs/1612.00835, 2016.

[3] Justin Johnson, Alexandre Alahi, and Fei-Fei Li.
Perceptual losses for real-time style transfer and super-resolution.
CoRR, abs/1603.08155, 2016.