# Generating High Quality Facial Images from Sketches Using Conditional GANs

Emmanuel Amaro
amaro@berkeley.edu

Fumika Isono
fumika@berkeley.edu

Philip Jacobson
philip_jacobson@berkeley.edu

## Abstract

*Generating photo-realistic facial images from hand-drawn sketches has been a problem of significant interest in the computer vision literature. Most approaches rely on using conditional GANs, with the best models requiring training using multiple GANs on segmented inputs. We are interested in exploring how to generate high-quality images while reducing the computational complexity of the state-of-the-art models. To this end, we introduce using data augmentation with a variety of sketch generation and simplification methods. Our best model, trained using an augmented dataset of three different sketch types, produces significantly higher quality images than simply relying on one sketch generation method.*

## 1. Introduction

Image-to-image translation has been proposed as a task that translates a representation of one scene, into another scene. Pix2pix [1] previously presented a common framework for these kind of tasks where the use of a Conditional Generative Adversarial Network (cGAN) allows to learn a loss function that adapts to the data. In addition, cGANs classify whether the output is real or fake while at the same time training a generative model that minimizes the learned loss.

In this work, we focus on using cGANs (based on pix2pix) to learn translations from a simple sketch to high quality, photorealistic human faces. Previous approaches that have explored similar goals require more complex network architectures, and pre-training data preparation. For example, DeepFaceDrawing [2] requires pre-label segmentation masks, and an encoder-decoder for each face feature: left eye, right eye, nose, mouth, and remainder. Another example of previous works is Lines2FacePhoto [3] which required computing dense distance fields for the input sketches, and used a conditional self-attention module with multi-scale discriminators with multiple sub networks in its model.

In this work, we focus on using simple sketches as input without any additional preprocessing. We use a single pix2pix model with 1 generator and 1 discriminator. In addition, we explore the effect on output prediction when training with different sketch types. We find that aggregating training datasets with multiple sketch types for the same target face clearly improves human face predictions.

## 2. Sketch Generation

### 2.1. Dataset

For the task of generating photo-realistic images from sketches, we require a dataset of both high-quality facial photographs, as well as associated sketches. For the photographs, we use the dataset CelebAMask-HQ, a dataset of 30000 1024 x 1024 pixel celebrity headshots, chosen to allow for direct comparison with previous work. Because of computation constraints, we use at most 5000 of the images in the dataset for our work, while shrinking them down to 284x284 resolution and discarding the image segmentation masks.

### 2.2. Sketch Generation

All sketches used for training, testing and validation are generated from the aforementioned dataset using a variety of methods, including classical edge detection methods, commercial sketch emulation software, and neural network based methods. Examples of these sketches are shown in figure 1.

#### 2.2.1 Canny Edge Detection

The first of these methods is Canny edge detection, a classical edge detection algorithm. The Canny method can be summarized in the following steps:

1. Apply a Gaussian filter to the images for noise reduction.
2. Calculate the intensity gradient of the pixels.
3. A double threshold, consisting of a min and max gradient value, is applied to the pixels. Gradients below the min threshold are removed whereas gradients above the max threshold are definitively considered edges.
4. Hysteresis thresholding is used on gradients between the two thresholds, meaning the pixel is considered an edge if it is locally connected to a pixel definitively considered an edge.

In the generation of our sketches, threshold values of 70 and 200, respectively, are used in a pixel scale of 0 to 255. After generating the edges, a binary colormap is applied to create sketch-like appearance.

#### 2.2.2 Photoshop Filter

The next method we applied to sketch generation uses commercial software. Adobe Photoshop includes several filters which can be applied to images to achieve various effects, including a photocopy filter, meant to create images emulating hand-drawn sketches. We use this filter to generate realistic sketches, as was similarly used in [2].

#### 2.2.3 Fully Convolutional Network

The last method we applied for generating sketches is based on a fully convolutional neural network model [4], [5]. The model consists of a 23-layer fully-convolutional network based on an encoder-decoder architecture, which is augmented at training time with a discriminator network. To generate our sketches, we deploy a pre-trained version of the network trained using real pencil sketches drawn by an artist. The strength of this model, in comparison with the previous methods, is the ability to capture many of the smaller details present in real pencil sketches, such as dirty and faded pencil lines. For the remainder of the paper, we refer to these sketches as the Pencil sketches.

### 2.3. Sketch Simplification

Although we have presented a variety of ways for generating sketches from photographs, these methods all tend to generate very detailed and noisy sketches. To further augment our dataset and add to the photo-generation challenge, we would also like to be able to generate more simplistic, less noisy sketches. To do this, we use a sketch simplification model based on the same principles as the model in 2.2.3 [5]. As in this previous section, this model consists of a fully-convolutional network pre-trained using sketches drawn by five different artists. The artists first drew "clean" sketches, before afterward adding extra lines, scratches, and smudges to produce the "rough" training sketches. Applying this sketch simplification routine to our own generated sketches significantly reduces the detail of the sketches, as shown in Figure 1.

## 3. Model

### 3.1. Conditional GAN Theory

Generative adversarial networks (GANs) have emerged as a compelling new current in computer vision for creating generative models [6]. In contrast, conditional GAN (cGAN) is a particular class of GANs that have recently been shown to perform extremely well on a variety of image to image translation tasks, such as the coloring of black and white images, or the applying of an artistic style to an image [1]. GANs learn a mapping from random noise vector $z$ to output image $y$, $G : z \rightarrow y$, while cGANs learn a mapping from observed image $x$ and random noise vector $z$, to $y$, $G : x, z \rightarrow y$. The method consists of a generator G, which translates semantic label maps to realistic-looking images, and discriminator D, which aims to distinguish real images from the translated ones. It aims to model the conditional distribution of real images with the following minimax game:

$$G^* = \arg \min_G \max_D \mathcal{L}_{cGAN}(G, D) \qquad (1)$$

where the objective $\mathcal{L}_{cGAN}(G, D)$ is given by

(a) Original

(b) Canny

(c) Photoshop

(d) Pencil

(e) Simplified Canny

(f) Simplified Photoshop

(g) Simplified Pencil

Figure 1: Comparison between different sketch generation/simplification methods for a given photograph

$$\mathcal{L}_{cGAN}(G,D) = \mathbb{E}_y[\log D(y)] + \mathbb{E}_{x,z}[\log(1 - D(G(x,z))] \quad (2)$$

### 3.2. pix2pix

The original model of Image-to-Image Translation with cGANs, pix2pix framework by Isola et al.[1], gained large popularity due to its simple and user friendly architecture without any need of hand-engineering mapping functions or loss functions. Its objective has an additional component $\lambda\mathcal{L}_{L1}(G)$ added to Eq.2 to encourage the generator to make the output to be near the ground truth. Here $\mathcal{L}_{L1}(G)$ is shown as

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x,y)\|_1]. \quad (3)$$

The generator uses 'U-net' [7] architecture, which is an encoder-decoder with skip connections between mirrored layers in the encoder-decoder stacks. Whereas the discriminator uses a patch-based fully convolutional network. The resolution of the generated images is up to 256 $\times$ 256.

### 3.3. pix2pixHD

Since the release of the pix2pix framework, newer architectures have been developed by others, for example, pix2pixHD[8], and DeepFraceDrawing [2]. Pix2pixHD introduced by Wang et al.[8] improved the pix2pix framework by using a coarse-to-fine generator, a multi-scale discriminator architecture and a robust adversarial learning objective function in order to generate high-resolution images. In Pix2pixHD, the generator is decomposed into two sub-networks: the global generator network $G_1$ and the local enhancer network $G_2$. During training, they first train the global generator at a resolution of 1024 x 512 and then train the local enhancer which outputs an image with a resolution 4x the output size of the previous one. It then jointly fine-tunes all the networks together.

For the discriminator to have a large receptive field, the model uses multi-scale discriminators operating at different image scales. They also improved the GAN loss in Eq.2 by incorporating a feature matching loss based on the discriminator. Accommodating three discriminators $D_1, D_2$ and $D_3$ and the feature matching loss $\mathcal{L}_{\text{FM}}$, learning problem becomes as following

$$G^* = \arg \min_G \left( \left( \max_{D_1,D_2,D_3} \sum_{k=1,2,3} \mathcal{L}_{cGAN}(G, D_k) \right) + \lambda \sum_{k=1,2,3} \mathcal{L}_{\text{FM}}(G, D_k) \right) \quad (4)$$

where $\lambda$ controls the importance of the two terms.

### 3.4. DeepFaceDrawing

The DeepFaceDrawing framework was proposed by Chen et al.[2] to overcome the overfitting seen during the training of the above methods.It consists of three main modules, CE(Component Embedding), FM(Feature Mapping), and IS (Image Synthesis). The CE module learns five feature descriptors from the face sketch data, 'left-eye', 'right-eye', 'nose', 'mouth', and 'remainder' separately. The two other modules together form another deep learning sub-network for conditional image generation, and map component feature vectors to 32-channel feature maps instead of 1-channel sketches in order to improve the information flow and provide more flexibility to combine individual face components for higher-quality synthesis results. Its generative network is mask-guided, and semantic label masks are required for training.

### 3.5. Our Model

Even though pix2pixHD can accommodate higher resolution output images and DeepFaceDrawing can produce more realistic images without overfitting, their larger networks and the requirement of the label masks for DeepFaceDrawing give huge drawbacks when applying it to large datasets. We especially found that pix2pixHD model requires 20 times more computation than pix2pix even with the same input images with the same resolution. Thus instead of using these slow models, we decided to use a light and simple pix2pix framework to tackle the problem of highly stochastic outputs of the conditional GAN.

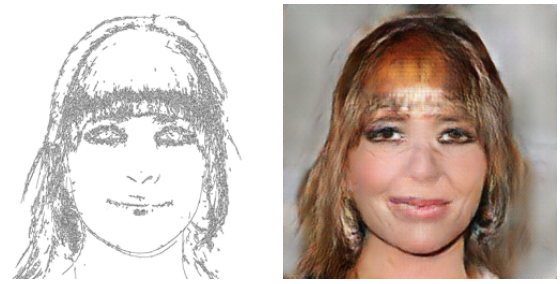
## 4. Results

### 4.1. Unsimplified Sketches

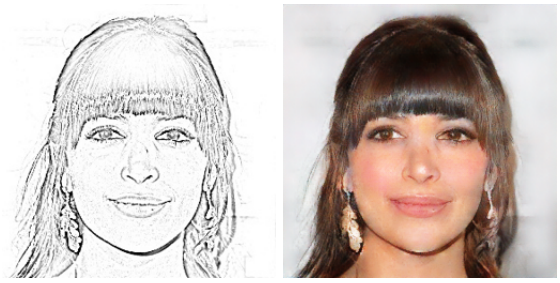
The first models we trained used the unsimplified versions of the sketches as a training set. We trained three models: one for each type of sketch generation technique, with each model trained using only one type of sketch. Using the unsimplified sketches to test the models, we find that the Photoshop model produces the best results, as the Photoshop filter consistently produces the most detailed sketches. However, we find that all three of these models generalize poorly to a test set composed of simplified sketches, as shown in Models 1, 2, and 3 in Figure 4.

### 4.2. Simplified Sketches

Next we trained the model with simplified sketches, with other conditions kept exactly same as in the previous section. Comparing Figure 2 and Figure 3,



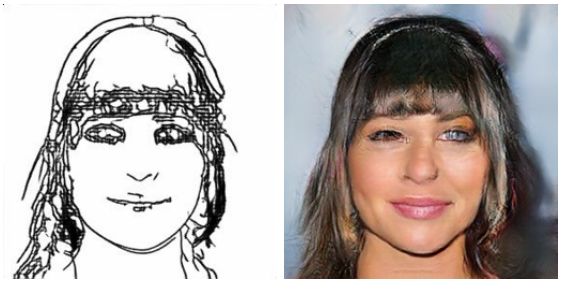
(a) Canny



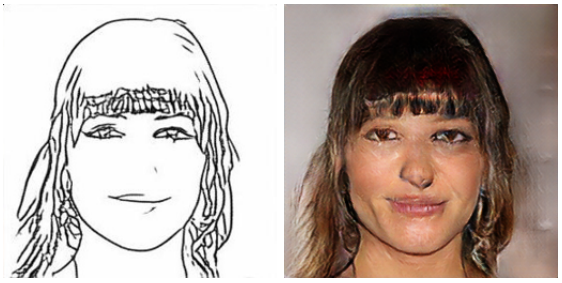
(b) Photoshop

(c) Pencil

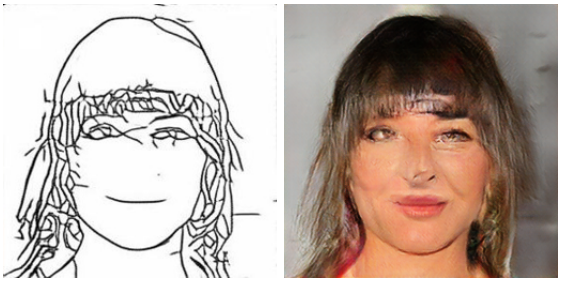
Figure 2: Comparison of test set images (right) generated from unsimplified Canny, Photoshop, and pencil sketches (left).



(a) Canny Simplified



(b) Photoshop Simplified



(c) Pencil Simplified

Figure 3: Comparison of test set images (right) generated from simplified Canny, Photoshop, and pencil sketches (left).

we see that output images look more similar to each other when training with the simplified sketches. Due to the limited information in the input images, the model seems to have less biases, as you can see the model successfully produced noses even without the its sketch in the input.

### 4.3. Aggregation of Simplified and Unsimplified Sketches

Our first attempt to improve output quality when using simplified sketches as input was to aggregate the training data set by using Simplified and Unsimplified versions of the same sketch type. Model 5 in Figure 5 was trained using Photoshop Simplified sketches, whereas Model 4 in Figure 4 was trained with an aggregate data set of Photoshop Simplified and Photoshop Unsimplified. As we can see, aggregating Simplified and Unsimplified sketches does not significantly increase the quality of the prediction, although Image 2 (from left to right) shows slightly improved results, particularly in her hair and eyes.

### 4.4. Aggregation of Distinct Simplified Sketch Types

Slight improvements obtained by aggregating Simplified and Unsimplified sketches of the same type (i.e., previous subsection) led us to aggregate training data sets using distinct Simplified sketch types. Model 8 from Figure 5 shows the best predictions we obtained in this work. This model was trained with an aggregated dataset from Simplified Canny, Simplified

Photoshop and Simplified Pencil sketches for a total of 2400 images (where each sketch type had 800 images). The predictions are still not perfect, for example, the eye colors do not match the ground truth, and the eye gaze direction is also tilted left.

We attempted to explain the high quality predictions of Model 8 by producing a new training data set that averaged Simplified Canny, Simplified Photoshop and Simplified Pencil into a single input. Model 7 was trained with the averaged images and a corresponding data set size of 800. However, as Figure 5 shows, we were not able to match the high quality outputs of Model 8.

### 4.5. Effect of Number of Filters

In the pix2pix model we used, $ngf$ is the number of filters in the generator's last convolutional layer, and $ndf$ is the number of filters in the discriminator's first convolutional layer. We explored two sets of values for these: $(ngf = 64, ndf = 64)$ and $(ngf = 128, ndf = 128)$.

Models 5 and 6 from Figure 5 allow us to directly compare the effect of the number of filters. Model 5 used a network with $(ngf = 64, ndf = 64)$, and Model 6 used $(ngf = 128, ndf = 128)$. We can see that overall Model 6, with the higher number of filters produces less blurry outputs with finer grained details.

### 4.6. Exhaustive Model Comparison for Simplified Sketches

Figure 4 and Figure 5 show an exhaustive comparison of the different models and their outputs when using 3 Photoshop Simplified inputs. As mentioned in §4.4, Model 8 produces the best predictions across all our attempts.

### 4.7. Discussion

Observing the results, we noticed several clear trends emerging which we will briefly discuss, and have the potential to serve as the basis for future work. One shortcoming arising from our chosen photo database is the inclusion of stray artifacts in many of the photos (hats, sunglasses, etc). In nearly all of the sketches in which one of these artifacts is present, the generated image is of poor quality. One possible improvement might come from manually pruning these bad pictures from the training dataset. Furthermore, our most successful model still struggles to generate high-fidelity photos when input sketches contain conspicuous missing edges, which we can see in the constitently better images generated by photoshop sketches, which tend to have the most complete edges. Lastly, we also have noticed our models contain clear biases when determining color from black and white sketches. In regards to hair, eye, and skin color, the model seems to generate better images when inputs contained lighter colors (e.g. blonde hair, blue/green eyes, light skin). One reason for this issue arises from the method used to generate the sketches, which tend to produce sharper edges for lighter tones. Additionally, the dataset we use contains significantly more examples of lighter-skinned faces than dark-skinned ones, contributing to the poorer reconstruction of non-white faces.

## 5. Conclusion

Using conditional GAN model pix2pix, we have investigated the effect of the variation of simple input sketches to the output facial images. We adopted three ways of sketch generation: Canny edge detection, a Photoshop filter, and a sketching scheme based on a fully-convolutional network. Additionally, we also applied a simplification scheme similarly based on a fully-convolutional network to generate a mixture of both high and low-quality sketches. Exploring various schemes for data augmentation, we found that we were able to generate the highest-quality images when training using a augmented dataset containing simplified sketches from all three generation algorithms. The current shortcomings of our method, such as inability to deal with artifacts in sketches and bias in determining color present interesting future directions for this work.

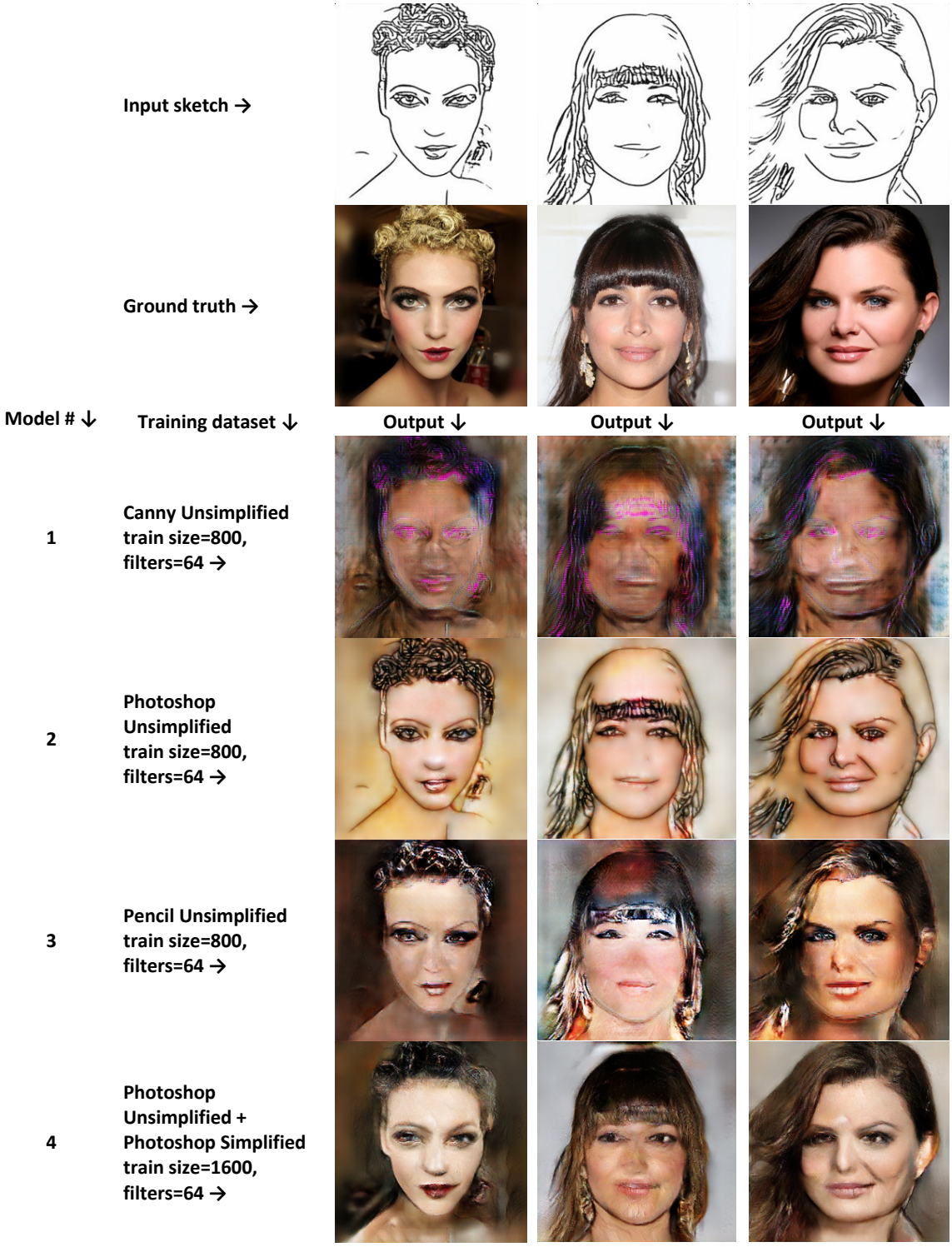



Figure 4: Predictions from models 1 through 4 when using 3 Photoshop Simplified sketches as input.

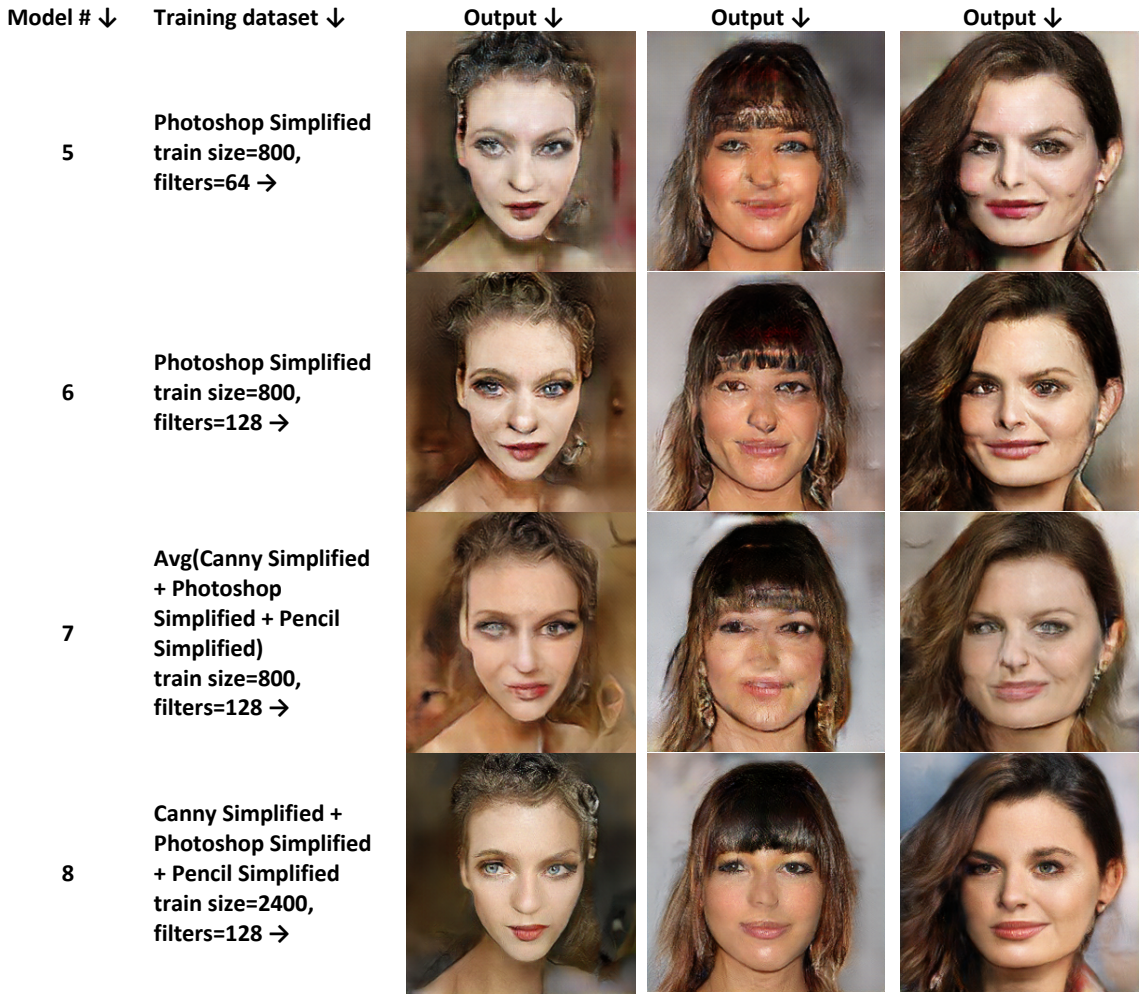



Figure 5: Predictions from models 5 through 8 when using 3 Photoshop Simplified sketches as input. Model 8 is the best model we produced in this work.

## References


[1] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

[2] S.-Y. Chen, W. Su, L. Gao, S. Xia, and H. Fu, "Deepfacedrawing: Deep generation of face images from sketches," *ACM Transactions on Graphics (TOG)*, vol. 39, no. 4, pp. 72–1, 2020.

[3] Y. Li, X. Chen, F. Wu, and Z.-J. Zha, "Linestofacephoto: Face photo generation from lines with conditional self-attention generative adversarial networks," in *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 2323–2331, 2019.

[4] E. Simo-Serra, S. Iizuka, and H. Ishikawa, "Mastering sketching: Adversarial augmentation for structured prediction," *arXiv preprint arXiv:1703.08966*, 2017.

[5] E. Simo-Serra, S. Iizuka, K. Sasaki, and H. Ishikawa, "Learning to simplify: Fully convolutional networks for rough sketch cleanup," *ACM Trans. Graph.*, vol. 35, July 2016.

[6] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

[7] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[8] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8798–8807, 2018.