Applying GANs for Generating Image with Varied Facial Attributes from Sketch

W Phusomsai and Y Limpiyakorn
Department of Computer Engineering, Chulalongkorn University, Bangkok 10330, Thailand
Yachai.L@chula.ac.th

Abstract. The rapid development of GANs and its variants has shown remarkable progress in synthesizing realistic images. Image-to-image translation becomes a potent research topic due to its wide application. The translation of face sketch to color images greatly contributes to digital image processing industry. It could also help confirm the identity of the suspect or lost persons. In this paper, a model is implemented by applying GANs for generating an image with varied facial attributes from sketch. The architecture consists of two separate GANs, implemented with Pix2Pix and StarGan2 as bases. The output realistic images can be generated with manipulating a single facial attribute including, wavy hair, straight hair, and wearing glasses.

1. Introduction
In recent years, GANs have been applied in many different research fields such as data augmentation and image-to-image translation. The rapid development of GANs and its variants has shown remarkable progress in synthesizing realistic images. Translation of face sketch to color images greatly contributes to digital image processing industry. In addition, it could help confirm the identity of the suspect or lost persons. In literature, image-to-image translation has become an important research topic in the field of computer vision and deep learning. Most of the research on this topic has focused on improving the efficiency of translation and quality of output images. Isola et al. [1] released software called Pix2Pix to address paired image translation task. The method is effective at synthesizing photos from label maps, reconstructing objects from edge maps, and colorizing images. However, there is a shortage when applied to sketch images as input because the edge and shape might not clear. Thereafter, Chen and Hays [2] developed SketchyGAN, a network to synthesize images from edge detection to generate objects in 50 classes. Photosketcher implemented in [3] also detected edge of input, mapped with the image database, then a composite of mappings is constructed based on the input image. Sometimes the edge of sketch image does not refer to real object or real target, to avoid this problem, Lu et al. [4] applied joint distribution to generate images rather than edge of sketch. Afterwards, GANs was applied to generate image across of domains by using unpaired image. Zhu et al. [5] applied cycle consistency to learn mapping across of domain and used cycle consistency loss to enforce generator.
This paper presents an application of GANs to generate a color realistic image from sketch. The output can be varied with three facial attributes and evaluated by the measure value of FID.
2. Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) was proposed by Goodfellow et al. [6] in 2014. In recent years, GAN has become one of the most popular research areas. The traditional structure of a Generative Adversarial Net consists of two separate neural networks: a generator, G, and a discriminator, D. The generator tries to produce realistic samples that fool the discriminator, whereas the discriminator tries to distinguish target samples from generated ones. As shown in Fig. 1, a generator G takes a random input vector, and outputs synthetic data. A discriminator D needs to classify whether it is synthetic or from the true data distribution by using probability of both target and synthetic samples, assuming the one with the probability value near 1 will be true data. If both of probability values approach 0.5, it denotes the two samples are almost identical. The first GAN [6] uses fully connected layer as its building block. Later, DCGAN [7] proposed to use fully convolutional neural networks which achieves better performance, and since then convolution and transposed convolution layers have become the core components in many GAN models [8].

![Common architecture of Generative Adversarial Networks](image)

**Figure 1.** Common architecture of Generative Adversarial Networks [9].

3. Methodology

3.1. Baselined Networks

3.1.1. Pix2Pix. Isola et al. [1] proposed an image-to-image translation with conditional adversarial networks, called Pix2Pix GAN. The method is addressed as paired image translation. The networks learn to mapping input image with target image and use L1 loss function to optimize during training. This approach is effective on synthesizing image, reconstructing objects from edge, and colorizing black and white images [1]. The authors presented a variant of GANs generator by using U-Net as an architecture.

3.1.2. StarGAN2 [10]. The method added two more networks: Mapping network and Style encoder, to the generator to generate the realistic image combined with the style code randomly generated and mapped with image attributes from mapping network to represent the style of each attribute. Style encoder learns styles from real images and utilizes loss value to reconstruct the style code.

3.2. Network architecture

The model architecture is composed of two separate GANs as shown in figure 2. The first GANs is implemented based on Pix2Pix [1] and responsible for sketch-to-image translation. The output is realistic images generated from the input face sketch images paired with the target or ground truth.
Next, the realistic image combined with style code are used as input passing through the generator of the second GANs implemented based on StarGAN2. By passing the style code of each attribute, the second generator will be able to generate the realistic image in different styles.

![Design of proposed architecture.](image)

Following U-Net architecture, the network architecture of the first generator contains nine layers in each convolution and deconvolution layer. Each layer has stride size and image input size as described in Table 1. The discriminator of the first GANs uses 5 layers of convolution with sigmoid function to classify probability of the generated image. Figure 3 (a), (b), (c), and (d) illustrate the structures of the second GANs' components, including: generator, discriminator, mapping network, and style encoder, respectively. Each residual block is combined with networks using ReLU activation function, instance normalization and 2 convolution layers. Whereas the adaptive residual block is connected to IReLU activation function, adaptive instance normalization and convolution layer.

| Generator | Discriminator |
|-----------|---------------|
| 2×2 Conv (256, 256, 3) | 2×2 Conv (256, 256, 3) |
| 2×2 Conv (128, 128, 64) | 2×2 Conv (128, 128, 64) |
| 2×2 Conv (64, 64, 128) | 2×2 Conv (64, 64, 128) |
| 2×2 Conv (32, 32, 256) | 2×2 Conv (32, 32, 256) |
| 2×2 Conv (16, 16, 256) | 2×2 Conv (31, 31, 512) |
| 2×2 Conv (8, 8, 256) | 2×2 Conv (8, 8, 256) |
| 2×2 Conv (4, 4, 256) | 2×2 Conv (4, 4, 256) |
| 2×2 Conv (2, 2, 256) | 2×2 Conv (2, 2, 256) |
| 2×2 Conv (1, 1, 256) | 2×2 Conv (1, 1, 256) |
| 2×2 DeConv (1, 1, 256) | 2×2 DeConv (1, 1, 256) |
| 2×2 DeConv (1, 1, 256) | 2×2 DeConv (1, 1, 256) |
| 2×2 DeConv (2, 2, 256) | 2×2 DeConv (2, 2, 256) |
| 2×2 DeConv (4, 4, 256) | 2×2 DeConv (4, 4, 256) |
| 2×2 DeConv (8, 8, 256) | 2×2 DeConv (8, 8, 256) |
| 2×2 DeConv (16, 16, 256) | 2×2 DeConv (16, 16, 256) |
| 2×2 DeConv (32, 32, 256) | 2×2 DeConv (32, 32, 256) |
| 2×2 DeConv (64, 64, 128) | 2×2 DeConv (64, 64, 128) |
| 2×2 DeConv (128, 128, 64) | 2×2 DeConv (128, 128, 64) |

**Figure 2.** Design of proposed architecture.
3.3. Model Training

We trained on 256×256 pixels of 2000 RGB color JPEG image files from CelebA dataset. All images are separate into 2 groups: female and male. Each image was labelled denoting three containing attributes, 1 as existing and 0 as none. Prior to training, all CelebA colored images has been converted to sketch images using xDoG [11]. The ratio of training and testing data is 90:10 or 1800:200 images. The models were trained on Colab Pro environments: 1) CPU Intel(R) Xeon(R) CPU @ 2.20GHz, and 2) 1 GPU Tesla P100-PCIE 16GB. The development tools used for model training are Python 3.6.9 and TensorFlow 1.5.0.

The network structure produces the output of a 3D tensor with RGB channel size 256×256. The input of face sketch is passed through the first generator containing 9 convolution layers using lrelu as activation function for encoder. The encoder convolution layer uses kernel size of 4, 2x2 stride and no padding. The decoder contains 9 deconvolution layers with kernel size of 4, 1x1 stride, no padding and uses relu as activation function. Both encoder and decoder use batch normalization, setting parameters: axis=3, epsilon=1e-15, and momentum=0.1. For generator and discriminator optimization, Adam optimizer is used with learning rate set to 0.0001. Both generators own their discriminator which learn to distinguish between a realistic image and its target image using the loss value of classification for reconstruction during training. Figure 4 and figure 5 depict the loss values of the generators and discriminators of the first and the second GANs during training the two datasets of male and female. During 150000 epochs training, validation was taken place every 1000 epochs and measured by FID denoting the discrepancy between the generated realistic and the target image as shown in figure 6. Based on the minimum value of FID and how visually realistic the synthesized images are, the best models of female and male are selected at the epoch of 147000 and 122000, respectively.
Figure 5. Loss values of generator and discriminator of 2\textsuperscript{nd} GANs during training.

Figure 6. FID scores for model evaluation during training.

Figure 7. Example realistic images varied by three attributes.
4. Results
Figure 7 illustrates the output of realistic images diversified by three attributes. The first column depicts the input face sketch. The second column is the output realistic image. The variants of output: wearing glasses, straight hair, wavy hair, are shown in the sequencing columns. The model is evaluated using Fréchet Inception Distance (FID) value. We separated the test set into three subsets associated with each attribute. The evaluation reported the FID values of female dataset varied by wavy hair, straight hair, wear glasses are 65.77, 79.35, 75.85, respectively, while they are 63.55, 71.42, 72.82 on male dataset.

5. Conclusion
This research carried out the experiments of the face sketch to image translation with manipulating a single facial attribute, wavy hair, straight hair, wear glasses. The quality of the variants of realistic images, measured by FID, still requires more improvement of the model. However, the model can synthesize visually realistic facial images compared to ground truth data, and the generated images look natural. Further research direction would be investigation of network architecture and techniques to produce more realistic and natural images with manipulating multiple facial attributes as well as controlling facial attribute change.

6. References
[1] Isola P, Zhu J, Zhou T and Efros AA 2018 Image-to-image translation with conditional adversarial networks. arXiv:1611.07004v3
[2] Chen W and Hays J 2018 SketchyGAN: Towards diverse and realistic sketch to image synthesis. arXiv:1703.10593v6
[3] Eitz M, Richter R, Hildebrand K, Boubekeur T and Alexa M 2011 Photosketcher: interactive sketch-based image synthesis. Computer Graphics and Applications, pp 56-66
[4] Lu Y, Wu S, Tai Y and Tang C 2018 Image generation from sketch constraint using contextual GAN. arXiv:1711.08972v2
[5] Zhu J, Park T, Isola P and Efros AA 2018 Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv:1703.10593v6
[6] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2014 Generative Adversarial Networks. arXiv:1406.2661v1
[7] Radford A, Metz L and Chintala S 2015 Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv:1511.06434
[8] Cheng K, Tahir R, Eric LK and Li M 2020 An analysis of generative adversarial networks and variants for image synthesis on MNIST dataset. Springer Multimedia Tools and Applications, pp 13725-13752
[9] Brownlee J 2019 A Gentle Introduction to Generative Adversarial Networks (GANs). https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/
[10] Choi Y, Uh Y, Yoo J and Ha J 2019 StarGAN v2: diverse image synthesis for multiple domains. arXiv:1912.01865v2
[11] Winnemoller H, Kyriyanis J and Olsen S 2012 XDoG: An eXtended difference-of-Gaussians compendium including advanced image stylization. Computer and Graphics, pp 740-753