Forensic Sketch-to-Face Image Transformation Using CycleGAN

Nischal Tonthanahal[1], Sourab B R[1], Dr Sharon Christa[2]

[1]Dept. of Information Science and Engineering, Dayananda Sagar College of Engineering, Bengaluru, Karnataka, India
[2]Assistant Professor, Dept. of Computer Science and Engineering, RV Institute of Technology and Management, Bengaluru, Karnataka, India

---***---

Abstract - *Would it be possible to generate a realistic looking image of a person's face by using only a forensic sketch? This paper considers the practical application of converting sketches to faces, specifically, processing a facial sketch of a person to generate a realistic image of their face. This task involves learning the mappings between these two different styles or domains of images. There are several methods to achieve this using Generative Adversarial Network (GAN) models such as Pix2Pix, CycleGAN, and BicycleGAN, to mention a few. One such method discussed is Cycle Consistent Generative Adversarial Networks (CycleGAN). This paper examines the viability of a CycleGAN model to produce faces from sketches. A diverse dataset is used to train and validate the model.*

***Key Words*: GANs, CycleGAN, Image Transformation, Hyperparameter tuning, Sketch to face, Forensic Sketch**

1. INTRODUCTION

Forensic sketch to face transformation refers to processing the sketch of a person's face to generate a realistic image of it. This is an example of image-to-image transformation. This technique can be used for many applications in a number of domains, be it artistic or forensic. In the case of a forensic application, it can be utilized in order to better visualize the face of a missing person or criminal based on the sketches drawn by forensic artists.

Transforming an image of one style to another style and texture is challenging when using the methods of Conventional Image Processing techniques. However, it can be achieved using Adversarial Style Transfer by learning the mappings between two high-level domains. GANs (Generative Adversarial Networks) can be used to generate images. CycleGAN is used for this application. It is a variant of Generative Adversarial Networks. It works without the requirement of paired examples of source and target images. Methods such as Pix2PixGAN [1] depend on the availability of training examples where the same subject is represented in both domains.

The nature of CycleGAN is advantageous as it can learn such inter-domain transformations without an injective mapping (which requires paired images having a one-to-one relation) between the training data in source and target domains. The requirement for paired images of the target is overcome with a two-step transformation process. First, the image from the source domain is transformed by mapping it to the target domain. Next, it is mapped back to the original domain, thereby trying to replicate the input by learning an inverse mapping. Mapping of images from source to target domain can be performed with a generator network, whereas, by pitching the generator network against an opposing discriminator network, there is a progressive improvement in the generated image quality.

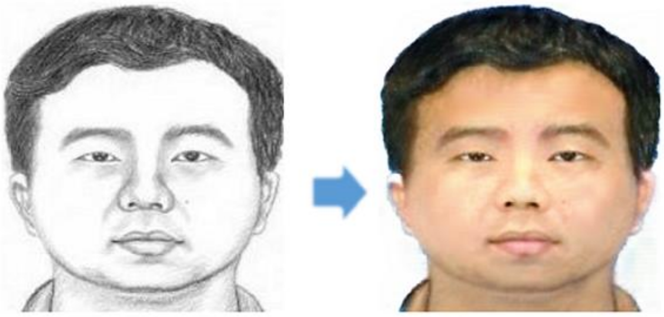

Fig 1: Sketch to Face result of the application

There are a few systems existing which can transfer certain characteristics of one domain to another domain using CycleGAN [2]. Some of the models are:

- Generating realistic landscape images from paintings (in the style of Van Gogh, Monet, etc) and vice versa.
- Season transfer - Using an image of a landscape during a particular season, that landscape could be viewed instantly from a different season perspective.
- Photo enhancement - Converting low-resolution image into high-resolution image.

2. RELATED WORK

Since the inception of GANs [3], there has been a lot of progress in the field of image-to-image translation. Pix2Pix [1] is a "paired" approach of image-to-image translation. Pix2Pix networks not only learn the mapping from input to output image, but also learn a loss function to train the mapping. These networks are applicable to a wide range of image-to-image translations such as synthesizing photos from label maps, colorizing images, day to night image transformation, and many more as they learn a loss adapted to the task and data at hand. CycleGAN [2] and DualGAN [4] performed unpaired image-to-image translation. GANILLA [5] improved on existing methods for the application of image-to-illustration translation that balance transfer of both style and content of the input image, as they are measured separately in two Convolutional Neural Networks. Like CycleGAN, GANILLA is also an "unpaired" approach where there is no one-to-one mapping between the source and the target image.

Jun-Yan Zhu proposed implementations [6] of Pix2Pix GAN and CycleGAN, using which, there are a few implementations which have the application of sketch-to-face image generation. Richard Liao proposed one such implementation [7] based on the Pix2PixGAN model by Christopher Hesse [8]. This is a TensorFlow based implementation in which the Generator and Discriminator networks are configured to take 64x64 images from the CUHK Face Sketch Database [9] dataset as input.

The conclusion ensued from this implementation is that if the network size and image resolution is increased, then the output images would be of better quality. Furthermore, since the model is trained exclusively on the CUHK dataset, it does not generalize well on input sketches outside this dataset.

3. MODELS

The model considered in developing this application is a variant of GANs called CycleGAN. The detailed explanation of these models is given further in this section.

3.1. Generative Adversarial Networks

Generative adversarial networks (GANs) [3] are deep neural network [10] architectures consisting of two neural networks, competing against each other. One neural network, called the generator, generates new data instances, while the other, the discriminator, evaluates their authenticity i.e., it decides whether each instance of data it reviews belongs to the actual training dataset or not. This training procedure corresponds to a min-max two-player game between generator G and discriminator D. A vector z is mapped using the function G and p(z) is used to sample the prior. This is formulated as:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))]$$

A combination of Rectifier linear activations and sigmoid activations are used for the generator network [11] and maxout activations [12] are used for the discriminator network. The "Dropout technique" [13] is used to regularize the discriminator network. The discriminator is trained to distinguish between a real and generated sample and the generator is trained to fool the discriminator with the generated samples.

3.2. CycleGAN

The goal of image-to-image translation is to learn the mapping between an input image and an output image by using a training set consisting of aligned pairs of images. However, for many applications, paired data would not be available. Zhu at el. proposed CycleGAN [2] which learns the mappings between two types of image groups with no paired examples. CycleGAN considers additional terms to generate outputs close to input images using class label space [14] and image pixel space [15]. The system must learn two mapping functions G: X->Y and F: X->Y. For this, the system is trained using two loss functions, an adversarial loss,

$$\mathcal{L}_{LSGAN}(G, D_Y, X, Y) = \mathbb{E}_{y\sim p_{data}(y)}[(D_Y(y)-1)^2] + \mathbb{E}_{x\sim p_{data}(x)}[D_Y(G(x))^2]$$

and a cyclic consistency loss,

$$\mathcal{L}_{cyc}(G,F) = \mathbb{E}_{x\sim p_{data}(x)}[\|F(G(x))-x\|_1] + \mathbb{E}_{y\sim p_{data}(y)}[\|F(G(y))-y\|_1]$$

CycleGAN's full objective function is expressed as:

$$\mathcal{L}(G,F,D_X,D_Y) = \mathcal{L}_{LSGAN}(G,D_Y,X,Y) + \mathcal{L}_{LSGAN}(F,D_X,Y,X) + \lambda\mathcal{L}_{cyc}(G,F)$$

CycleGAN learns by transforming an image from domain A to B and then re-transforming the generated image from domain B to A. So, each time it learns not only to transform the image to the other domain, but also with what efficacy it can reproduce the original image from the transformed image. The drawback of this model includes a lack of accountability on why it generates specific image properties. There also remains no objective metric to measure the accuracy of the transformation for unseen or test data, as this quality is entirely subjective.

4. DATA PREPARATION

This section describes the dataset and pre-processing of the dataset for the model.

4.1. Dataset

The CUHK Face Sketch Database [9] and IIIT-D Sketch Database [16] are used for training the model. We used a subset of these datasets as depicted in Table 1.

Table 1: Dataset Specification

Data Source	Number of images used	Resolution
IIIT-D: LFW and FG-NET	166	1287x1848
IIIT-D: Student & Staff	72	2413x3352
CUHK Student Data Set	88	200x250

4.2. Data Preprocessing

The dataset containing sketch images and face images are converted to non-background or white background images. The face images' backgrounds are removed manually for every image. They are further cropped to fit the top of the hair, bottom of the chin, and the sides of the head, to eliminate as much of the background as possible, and finally resized to a resolution of 256x256 pixels. This is done in order to match the shape of the input sketch and the output face image as closely as possible, thereby reducing the adverse effects of any mismatched pairings and to keep the dataset uniform with respect to shape. It is important to note that resizing of images is done in such a way that the original aspect ratio of facial features is kept unchanged. This is based on the observation that, style-transfer using GANs is not capable of accurately transforming the shape of the image, as it learns only to transform the texture and colour.

5. IMPLEMENTATION

The following section presents the details of the implementation of the CycleGAN algorithm for the sketch to face conversion. The code-level implementation is performed using Python3. The model is trained on a virtual machine with the configuration specified in Table 2. The process of implementation, the training phase, and the hyperparameters involved, along with the system configuration, is presented further.

Table 2: System Configuration

CPU	1 core, 2 threads Xeon Processors @2.3Ghz 45MB Cache
RAM	~12.7 GB
DISK	~358.27 GB
GPU	1xTesla K80, having 2496 CUDA cores, compute 3.7
VRAM	12GB (11.439GB Usable) GDDR5

5.1. Training

The model is initially trained using PyTorch [17] library on the datasets (CUHK, IIIT-D) [16] individually. It takes in unpaired images of sketches and faces for training, where each input image is 256x256 pixel size. Each dataset is divided into "train" and "test" sets. The images found to be subjectively subpar sketches are discarded. Finally, all datasets are combined and shuffled for a more varied dataset consisting of multiple face types, colours and shapes in order to prevent model bias.

5.2. Algorithm Hyperparameter List

The parameter and configuration used for training the model is depicted in Table 3.

Table 3: Hyperparameter List

Parameter	Value
aspect_ratio	1.0
batch_size	1
image_load_size	256
image_crop_size	256
gain	0.02
input_img_dim	3
output_img_dim	3
no_gen_filters	64
no_disc_filters	64
gen_arch	resnet_9blocks
disc_arch	basic (70x70 PatchGAN)
no_dropout	true
no_threads	4
learning_rate	0.0002
epochs	300
dataset_size	280

The *aspect_ratio* specifies the ratio of height to width of the image. *batch_size* is the input batch size. Images are scaled to the size mentioned by *image_load_size*. Images are cropped to the size mentioned by *image_crop_size*. *gain* specifies the scaling factor of the network initialization. *input_img_dim* and *output_img_dim* specify the input and output dimension of the image. *no_gen_filters* is the number of generator filters in the last convolution layer. *no_disc_filters* is the number of discriminator filters in the first convolution layer. *gen_arch* and *disc_arch* specify the generator and discriminator architecture to be used. *no_dropout* takes a Boolean value indicating whether to have dropout for the generator or not. *no_threads* specifies the number of threads to be used for loading the data. *learning_rate* is the initial learning rate for Adam optimizer. *epochs* is the number of epochs over which the model is trained. *dataset_size* specifies the number of images in the training set.

6. EXPERIMENTAL RESULTS

The CycleGAN model is trained for 294 epochs. The results observed are shown in Figure 2 and Figure 3.

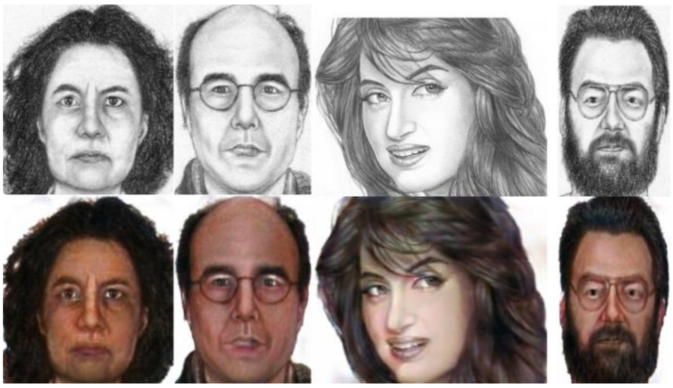

Fig 2: Sketch to Face photo result of Validation set

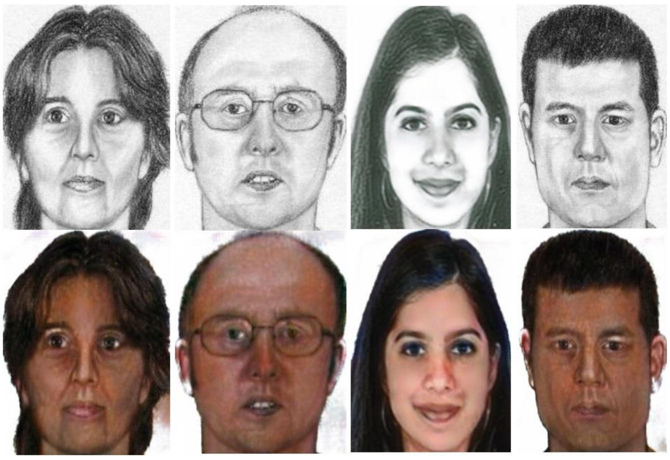

Fig 3: Sketch to Face photo result of unseen sketches

The benefit of using CycleGAN model is that it is cycle-consistent, that is, it can not only transform sketches to faces but also can learn to transform faces to sketches, as shown in Figure 4.

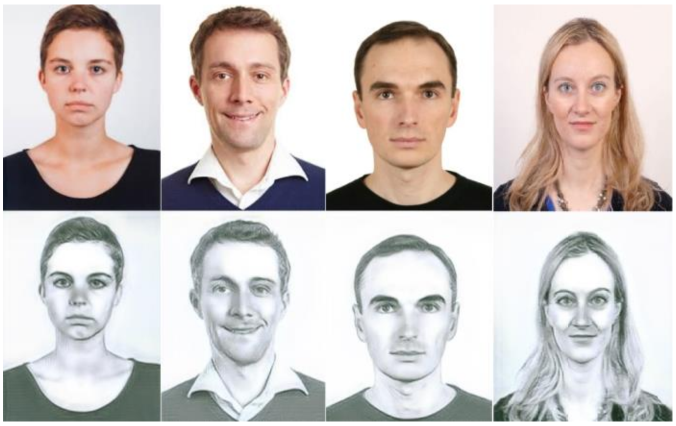

Fig 4: Face photo to Sketch result of unseen faces

In the Sketch-to-Face transformation, the obtained face photos map the sketches accurately. It is seen that the facial tone is sensitive to the shades provided in sketches. In the Face-to-Sketch transformation, the sketches produced by face photos resemble actual pencil sketches, hence this can be used to generate reference sketches for sketch artists.

7. CONCLUSION

From the results it is observed that CycleGAN generates realistic features of coloured face images from sketches by learning the mappings between sketches and faces. This CycleGAN model performed well with relatively few diverse training examples. This application can be used for generating reasonably realistic looking human faces for forensic analysis, or conversely for generating artistic sketches of a person's face. It can be further enhanced to generate higher resolution images and adapted for other applications such as face recognition.

ACKNOWLEDGEMENT

We thank IIIT-Delhi for providing their "Sketch Database" [16] and CUHK for providing their "Face Sketch Database" [9].

REFERENCES

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

[2] J.-Y. Zhu, T. Park, P. Isola, A. A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2223–2232.

[3] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672– 2680.

[4] Z. Yi, H. Zhang, P. Tan, M. Gong, Dualgan: Unsupervised dual learning for image-to-image translation (2017). arXiv:1704.02510.

[5] S. Hicsonmez, N. Samet, E. Akbas, P. Duygulu, Ganilla: Generative adversarial networks for image to illustration translation, Image and Vision Computing (2020) 103886.

[6] J. yan zhu, pytorch cyclegan and pix2pix 2017 github repository, https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix, accessed: 2018-12-21.

[7] R. Liao, Sketchtoface, 2017, github repository, https://github.com/richliao/SketchToFace, accessed: 2018-11-9.

[8] C. Hesse, Pix2pix-tensorflow, 2017, github repository, https://github.com/affinelayer/ pix2pix-tensorflow, accessed: 2018-11-9.

[9] X. Wang, X. Tang, Face photo-sketch synthesis and recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (11) (2009) 1955–1967.

[10] X. Du, Y. Cai, S. Wang, L. Zhang, Overview of deep learning, in: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), IEEE, 2016, pp. 159–164.

[11] K. Jarrett, K. Kavukcuoglu, Y. LeCun, et al., What is the best multi-stage architecture for object recognition?, in: 2009 IEEE 12th international conference on computer vision, IEEE, 2009, pp. 2146– 2153.

[12] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, Y. Bengio, Maxout networks, arXiv preprint arXiv:1302.4389.

[13] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, arXiv preprint arXiv:1207.0580.

[14] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, D. Krishnan, Unsupervised pixel-level domain adaptation with generative adversarial networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 3722–3731.

[15] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, R. Webb, Learning from simulated and unsupervised images through adversarial training, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2107–2116.

[16] H. S. Bhatt, S. Bharadwaj, R. Singh, M. Vatsa, Memetic approach for matching sketches with digital face images, Tech. rep. (2012).

[17] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, pytorch, https://github.com/pytorch, accessed: 2019-01-9.