Summer Research Program in Industrial and Applied Mathematics

SEOUL NATIONAL UNIVERSITY

Sponsor

⟨**Tencent**⟩

Final Report

⟨Sketch to Image Generation⟩

Student Members

⟨Seon Gyu PARK⟩ (Project Manager), ⟨*SNU*⟩,
⟨SunQ0313@gmail.com⟩
⟨Chun Wai WONG⟩, ⟨*HKUST*⟩
⟨Fuzhe LI⟩, ⟨*HKUST*⟩

Academic Mentor

⟨Dr. Ningchen YING⟩, ⟨mancying@ust.hk⟩

Sponsoring Mentors

⟨Prof. Yu-Wing TAI⟩, ⟨yuwingtai@tencent.com⟩

Consultants

⟨Name⟩
⟨Name⟩

⟨Date: 7th August 2018⟩

Abstract

This project aims to build a system that is capable of transforming hand-drawn sketches of human face into a photo-realistic face images. In order to do that, CycleGAN, which is a neural networks model based on GAN(generative adversarial networks) was utilized, having our modifications on structure such as U-net based generators. Besides the Celebrity A human face photograph dataset, considerable amount of sketch images were collected and used to train the model after steps of preprocessing. By experiments, we were able to find that the preprocessing steps were critical to the output image quality, and to build a model that can produce results with acceptable quality. In this paper we provides the processes of experiments we've gone through, and some of possible improvements that is considered to be applied in future works.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The goal of this project is to build a system that can generate photo-realistic images from rough sketch pictures. To serve this purpose, we utilized the Cycle-GAN [19] with sketch images collected from google image search and Celebrity A dataset [9]. The process of collecting and refining data, and detailed information about netrwork architecture are provided at Chapter 2 and at Chapter 3 respectively.

The series of experiments and their result images can be found in Chapter 4 and Chapter 5 with description, which would be considered as the main content of this report. All the code we used has been made available to public at `https://github.com/SunQpark/SPIA2018_cycle_GAN`. The only thing excluded is the input images, which are considered to contain some images that are able to cause some legal issues. Any kind of interest or contribution on our project would be welcomed.

1.1 Related Works

Sketch-to-Photograph generation is a sub-problem of task called Image-to-Image translation. In general, the goal of the Image-to-Image problem is to learn the mapping between distinct domains of image. The problem include some of most important problems in image processing by computer science, such as image segmentation [10] where one domain is the natural image and the other is semantic label of that image, or image colorization [18] where one domain is grayscale image and the other is RGB image, and so on. The recent Image-to-Image translation approach has been hugely improved due to the raise of deep learning in the field of computer vision, especially using the method of **Generative adversarial network(GAN)** models.

Being published by Ian Goodfellow in 2014, GAN is originally developed as a generative model whose purpose is to train a generative model that can make fake data which look as similar as the given set of train data. To achive that object, GAN make use of two-player min-max game setup. One part of GAN is called the Generator, which are supposed to learn to generate images. To be more specific, generator part of GAN takes random gaussian noise vector as input and transform that noise into image features using the set of parameters included in the model. On the other hand Discriminator, the remaining part of GAN, takes image as input, and guesses whether the input image is sampled from the given set of input data(real) or

generated(fake) by the generator. As the training of GAN goes on, the discriminator gets better and better at telling the fake images apart from the real images, which results in the generator to produce realistic images to fool the discriminator.

When GAN was so successful in the field of image generation, it was a model called **Pix-to-Pix** [5] that first succeeded in applying GAN to image conversion, not image creation. While there were approaches before Pix-to-Pix to transform images using supervised deep-learning, with the ordinary cross-entropy or mean-squared-error as loss function tended to produce blurry and weird images rather than realistic images, since that was 'safe' choice of model under loss function that puts penalty on the pixel-wise difference of result image, not the subjective quality of that. Pix-to-Pix model solved this problematic situation by setting up discriminator to penalize the output that does not looks like true result image and succeed in generating realistic images.

Although the Pix2Pix model succeeded in producing realistic images using GAN loss, it still had the disadvantage of requiring a paired image dataset to train the model due to the supervised setup of problem. Where the **CycleGAN** [19] emerges is in this context. CycleGAN is different from Pix-to-Pix in that it consists of two GAN models to learn mapping function between two domains of image one direciton per one model respectively. It differs in that it compares the output image with the images in the output domain while the Pix-to-Pix model only compares with the ground truth label corresponding to the input image, and that it has cycle-consistency loss term in the loss function which helps keep the contents of output image do not differ too much from those of the input image. The authors of CycleGAN successfully built models to transform horse image into zebra image, picture taken in winter into taken in summer, and realistic photo into stylized painting of famous painters.

So far we have briefly introduced the overall structure of our research. Below is a list of the small ideas that did not affect the overall flow but were included in our experiments and had effect on the result.

LSGAN [11] uses least square loss function to replace the original loss function of GAN. By some experiments this is known to improve the stability of GAN training and the quality of generated images. This idea was applied on the original work of CycleGAN authors and so of us.

PatchGAN [14] uses discriminator having receptive field of output vector smaller than the size of input images, which result in discriminator to differentiate images only using small patches of input image. This trick is known to help the generator to generate fine details in the output image. This idea was added in original work by authors of CycleGAN, too.

U-net [13] is a model that is first developed for the semantic segmentation in medical image. We used this U-net structure as the generators of CycleGAN, expecting it to help keeping the details in content of input image till the output layer. More explanation on the model structure and implementation will be provided at Chapter 3. For those who wants more detailed information can refers to the source code on github.

Checkerboard Artifacts on the output image are an well-known problematic phenomenon of using transposed-convolution(or deconvolution or up-convolution). We refered to this article [12] published by Google Brain researchers and applied

the main idea of replacing transposed-convolution with nearest-neighbor upsampling followed by 3 by 3 convolution to remove strange checkerboard patterns in the output images.

Generating Sketch Image does not requires complicated processing and can be implemented simply with image operations like gaussian blur, inverting color, add pixel values and brightness, contrast shift operation. We implementated simple version of sketch effect filters using PIL library but since this result was not so helpful to the final result, the detailed process will provided on appendix for those who are interested.

The team would like to thank **Tencent** for the generous sponsorship of this project.

Chapter 2

Processing Data

In this section, we will explain about the progresses of collecting and refining data we've gone through. Table 2.1 shows the composition of dataset we used in experiments. Unless mentioned otherwise, every dataset used in experiments composed as the proportion as Set1 in table.

Dataset	Sketch			Photo
	Collected	CUFS	Sketch Filter	Celeb A
Set1	920 (30.8%)	571 (19.2%)	0 (0.0%)	1500 (50.1%)
Set2	920 (16.8%)	571 (10.4%)	1000 (18.2%)	3000 (54.6%)

Table 2.1: Number of images used for training in each component.

More details on each sources of data will be provided in following sections.

2.1 Photograph

For realistic photographs of human faces, we used Celebrity A [9] which contains more than 200,000 images of celebrity in the world. In addition to having clean images of human faces, this dataset contained 'attributes', some additional imformation of images. Though those informations were not used directly as input to our neural networks, they were useful in some progress of preprocessing. We will describe about the details in 2.3.

2.2 Sketch

The main portion of sketch image dataset we used was collected from Google Image search engine. However, the collected images were highly inconsistent, which was undisirable. figure shows some bad examples of image which were included in the first collection. Thus, these images has gone through preprocessing steps, which will be described in next section 2.3. After preprocessing steps, these images were aligned as fig 2.1.

We also used CUHK(The Chinese University of Hong Kong) face sketch database(CUFS) [16], containing about 500 face sketches. Although this dataset also provides face photographs paired with the sketch images, we only used the sketch images to train our model to keep it from overfitting to these paired images.

Examples of images in each dataset can be seen in fig.

2.3 Preprocess

Figure 2.1: Preprocessing examples. Original images containing faces(top) and results after preprocessing steps(bottom)

In this section we deals with the preprocessing steps on face sketches and photos, which were applied separately before the training of model. By experiments of a few weeks, we found this part crucial to the performance of model. The first step of data refinement was to detect every faces for each image. By excluding images not containing recognized faces, we could get rid of 20% of bad examples in sketch database. Then, we applied 'facial landmark detection' for the detected faces, in order to get further information concerning the direction of head in the images. We utilized the 'dlib' framework [7] for those two steps. Next, we rotated the images to have faces aligned vertically using the detected locations of two eyes. Finally, the images were cropped to have faces in center and 30% of the width of head margins on both sides. The cropped images have 128 by 128 size in the end of preprocessings. Those steps were applied to both sketch and photo images to make them consistent in any attributes other than it is sketch or photo.

After some experiments we found that the model was learning to put smile on the generated photo from sketch images even when the original sketch images were not smiling. This seemed to be originated from the fact that images in photograph have smile in most case, while sketch images haven't. Since this was undisired effect, we reduced the ratio of smiling faces in the photograph dataset. This process was done

by checking attributes file which Celebrity A dataset provided to see wether given image is smiling or not and using only one images per 20 smiling images.

Chapter 3

Model Structure

3.1 Generator

As we mentioned earlier, baseline structure mainly used in this project is Cycle-GAN. The Cycle-GAN consists of two GAN models, which learn to translate images from one domain to the other, one direction per each respectively.

The first architecture we selected as generators is the *Residual Network* [4]. Being published in 2015, The skip connection of ResNet is considered one of most important idea in the field of deep learning. What the skip connection is doing is nothing more than copy early features of layer and paste it to later layers, and known to help the training process of neural networks a lot. There are many explanation why this helps, one of the most significant one is that it makes the optimization problem that deep networks faces by making loss function to be optimized over smoother manifold [8].

The ResNet consists of small modules called 'residual module' which has two 3 by 3 convolutions and one skip connection. This topology makes ResNet model to be stacked deep without making optimization too difficult. While this structure make it suitable for classification and also for other purposes, we decided to use U-net for our case because it is known to keep details in images better than ResNet does.

Fig 3.1 shows overall structure of U-net. While the U-net also features the skip connection as the ResNet, U-net differs from it in that it has skip connection from symmetrical shape. The skip connections connecting early layers with the later layers transport the low-level features to the high-level layers which significantly reduces spatial infomation loss and ensures feedback on internal structures of both inputs and outputs.

While the original U-net only used central part of images abandoning the boundary features, we used input padding on the boundary inputs to keep images to be invariant in size. In addition, different from the original U-net, we replaced the transposed-convolution with nearest-neighbor upsample and convolution of kernel size 3, which are known to prevent output images from generating checkerboard-like artifacts by removing the overlap between filters [12].

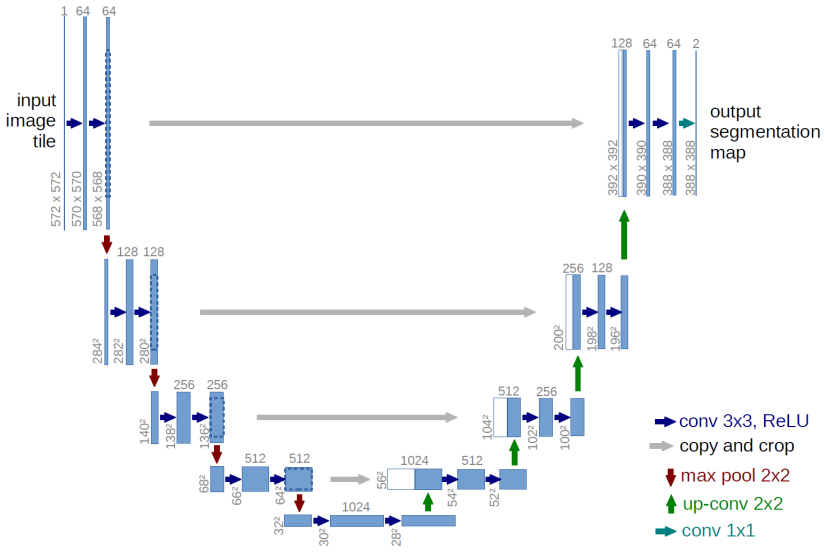

Figure 3.1: Structure of original U-net [13].

3.2 Discriminator

For the discriminators we used the 70 by 70 PatchGAN discriminator with instance normalizations, which is same as the original CycleGAN article. For stability of the train process, we added some noise to the real / fake label and flipped the labels once in a hundred.

Finally, we utilized 'grouped convolutions [17]' to the bottleneck(most deepest part of U-net) with 4 groups, which makes convolution operation happens separately on the 4 separated groups of intermediate features in networks. For more detailed information about model structure, the reader can refers to our published code.

Chapter 4

Result

Figure 4.1 shows some examples of result before the data refinement steps were applied. The results seems to contain some face-like objects, but the original faces in the sketch images were not properly translated, resulting in weired colorization in the result. The result of photograph to sketch translation also shows severe artifacts due to the mismatch of locations of faces between datasets.

Figure 4.1: Original input images(left) and result of translations(right) by networks trained before input data is not aligned. Image size is 256 by 256 and model is trained for 64 epochs.

The result seems to be improved considerably with preprocessing steps. Fig 4.2

shows aligned faces helped the model tell the face and background apart, resulting it to generate better volumes, colors on the paintings.

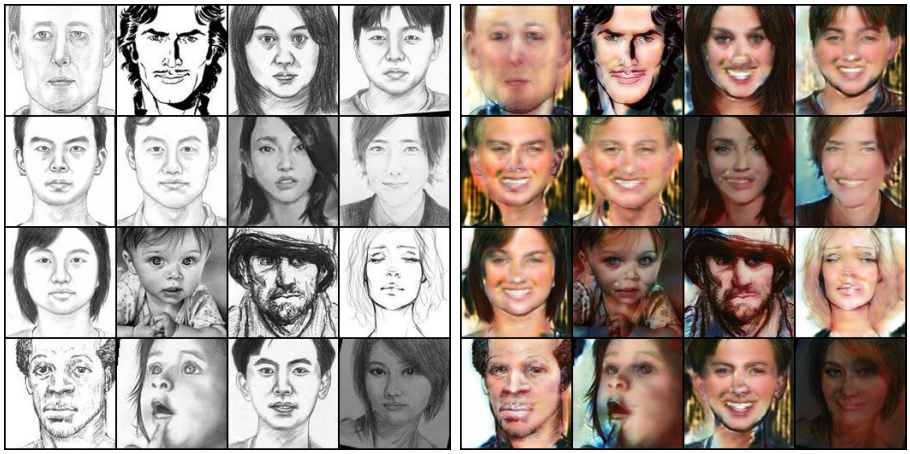

Figure 4.2: Original input images(left) and result of translations(right) by networks trained on datasets after face alignment, but before number of smiling faces was reduced. Image size is 128 by 128 and model is trained for 128 epochs.

However, the results still looks not realistic, mostly due to mistakenly generated smile on the translated images. After some investigation we could find out that is due to the difference of distribution of train data that people smiled much more on the photos rather than on the paintings. After reducing the ratio of smile in the photograph dataset used as in previous chapter 2, we could get following results 4.3.

While some of the sketch-to-photo results seems quite realistic(3^{rd} image of 2^{nd} column, 2^{nd}, 3^{rd} of 3^{rd}), images generated from low-quality inputs does not seems to be properly translated(2^{nd} image of 2^{nd} column, last ones in 3^{rd}, 4^{th} column). On the other hand, despite the general quality looks fine, some of results(1^{st}, 2^{nd} diagonal positions) resembles average sample images of CUFS datasets rather than contents of input.

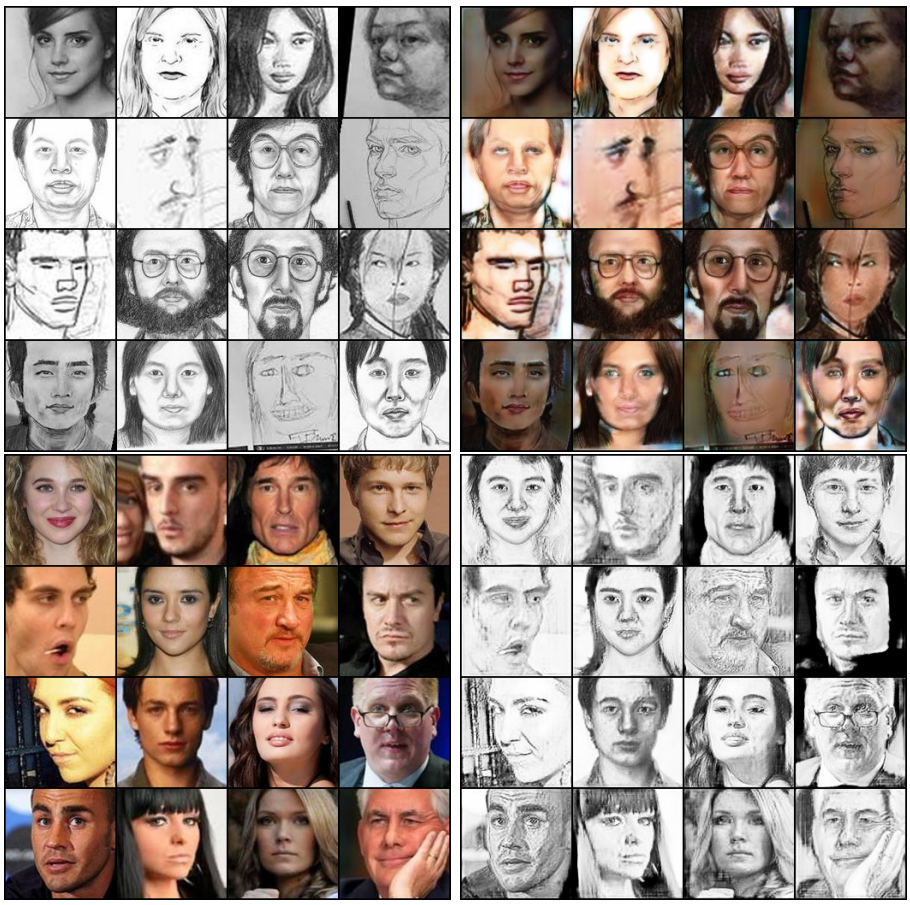

Figure 4.3: Original input images(left) and result of translations(right) by networks trained on datasets after face alignment and number of smiling faces was reduced. Image size is 128 by 128 and model is trained for 128 epochs.

Chapter 5

Conclusion

By far, we have introduced our works done and examples of results. The result of output images are not as satisfying but considering the quantity and quality of sketch images collected from google image search, this amount of dependency on input quality might be considered acceptible. Building more generally appliable model is considered possible but may require more time and resources. In our opinion, there are still several schemes remaining that deserve experiments.

We experimented utilizing the 'grouped convolution' to improve U-net structure and got results which are considered better by us. However, the absent of objective method for evaluation made it impossible to properly compare our models and choose better ones. So, getting proper measure for the performance of generated image would be most immediate goal for the case when this work is continued. After that, we expect to evaluate the value of ideas applied on our model and properly select better model, which would make possible to get images with much better quality.

Although our work, done as SPIA 2018 program, did not finished with so called 'state of the art' result, as undergraduate students interested in deep learning and current state of technology, we expect this experience would someday help us to push the boundary of the mankind knowledge.

5.1 Acknowledgements

Our team would like to acknowledge Professor Yu-Wing TAI and Dr. Ningchen YING for helpful discussion. Professor Yu-Wing TAI offered a lots of helpful suggestions to model development such as implementing U-net as well as suggestion to data collection. Dr Ningchen YING offered a general introduction and outlining to the project and his patience to guide the project. We would also like to thank Professor Shingyu Leung, Professor Avery CHING, HKUST and SNU MATH department for providing computational resources and a sight seeing trip to Macau for stimulating our creativity.

Bibliography

[1] S. Barratt and R. Sharma, *A note on the inception score*, 2018.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.

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

[4] K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, 2015.

[5] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, *Image-to-image translation with conditional adversarial networks*, (2016).

[6] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, *Learning to discover cross-domain relations with generative adversarial networks*, 2017.

[7] D. E. King, *Dlib-ml: A machine learning toolkit*, Journal of Machine Learning Research, 10 (2009), pp. 1755–1758.

[8] H. Li, Z. Xu, G. Taylor, and T. Goldstein, *Visualizing the loss landscape of neural nets*, 2018.

[9] Z. Liu, P. Luo, X. Wang, and X. Tang, *Deep learning face attributes in the wild*, in Proceedings of International Conference on Computer Vision (ICCV), 2015.

[10] J. Long, E. Shelhamer, and T. Darrell, *Fully convolutional networks for semantic segmentation*, 2014.

[11] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, *Least squares generative adversarial networks*, 2016.

[12] A. Odena, V. Dumoulin, and C. Olah, *Deconvolution and checkerboard artifacts*, Distill, (2016).

[13] O. Ronneberger, P. Fischer, and T. Brox, *U-net: Convolutional networks for biomedical image segmentation*, 2015.

[14] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, *Improved techniques for training gans*, 2016.

[15] D. Ulyanov, A. Vedaldi, and V. Lempitsky, *Instance normalization: The missing ingredient for fast stylization*, 2016.

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

[17] S. Yi, J. Ju, M.-K. Yoon, and J. Choi, *Grouped convolutional neural networks for multivariate time series*, 2017.

[18] R. Zhang, P. Isola, and A. A. Efros, *Colorful image colorization*, 2016.

[19] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, *Unpaired image-to-image translation using cycle-consistent adversarial networks*, 2017.