Summer Research Program in Industrial and Applied Mathematics

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

In this project, we investigated sketch-to-image translation by implementing CycleGAN to learn the mapping between human face sketch to realistic photograph. We used U-net to form the Generative Adversarial Networks This makes the model possible to train end to end from very few images and guaranteeing the performance of the mapping.

Acknowledgments

It is appropriate in the Acknowledgments to thank individuals or organizations who made especially noteworthy contributions to your project. Elsewhere, within the body of the report, you can acknowledge more specific contributions where appropriate. These are matters of courtesy and professional ethics. As an example:

> The RIPS LaTeX report template has been developed by Mike Raugh with advice and assistance from Oleg Alexandrov and Shawn Cokus in the early stage of development and general support of IPAM and the System Administration staff. The first RIPS template was based on an early version of the Math Clinic's report template at Harvey Mudd College; there the original template has been improved and is managed by Claire Connelly, the HMC Math Department's system administrator. Claire and her co-authors offer coding advice, a wealth of references, and a note about the origin of the template in their current edition, the `sample-clinic-report.pdf` accessible at `http://www.math.hmc.edu/computing/support/tex/sample-report`. Claire copyedited the third edition of Grätzer's *Math into LaTeX*, most of which work seems to have survived into the fourth edition: *More Math into LaTeX* [**?**].

When acknowledging individuals in this section, it is OK to use the names by which you know and speak to them. Here it is OK to write "Oleg Alexandrov." But you must be formal on the Title page and elsewhere within the report, where it is proper to specify honorifics, e.g., Dr. or Prof. On the Title page you would write "Dr. Oleg Alexandrov," and likewise within the body of the report if you were acknowledging him for a specific contribution, Claire Connelly uses no honorific, so you would use just her name on the title page. When in doubt, check the person's business card or follow usage on the person's web page.

As a result of suggestions from users, this Sample Report and its source are under continual improvement. Please contact the RIPS program director for your suggestions. An up-to-date list of changes is recorded in the ”Revisions” folder for the Master Template Folder.

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 [16] with sketch images collected from google image search and Celebrity A dataset [7]. The process of collecting and refining data, and detailed information about netrwork architecture are provided at Chapter 3 and at Chapter 4 respectively.

The series of experiments and their result images can be found in Chapter 5 and Chapter 6 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 [8] where one domain is the natural image and the other is semantic label of that image, or image colorization [15] 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** [16] 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 [9] 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 [12] 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 [11] 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 4. 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 [10] 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.

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

Chapter 2

Background

2.1 Related Models

There are two main approaches for image generation tasks: generative adversarial nets and variational autoencoder. In this project, only GAN-related methods will be considered and inplemented. The original GAN proposed by Goodfellow can train a generator to learn the real image distribution by training a discriminator together, which can tell the difference between real images and fake images.

In the field of image translation trained with unpaired image datasets, 3 models, CycleGAN, DiscoGAN, DualGAN are doing the same thing from high level perspective. They train two generative models to learn the mapping between two image domains by training two discriminators at each domain together and considering the cycle-consistency loss, which ensures ignorable difference between images and reconstructed images.
In detail, the differences are CycleGAN uses instance normalization, patchGAN discriminator. To stabilize the training, use least square GAN. Replay buffer no random input z no drop out; L1 distance as cycle consistency

DualGAN use generator and discriminator of pix2pix; no random input z but implemented drop out; use wgan; L1 distance as cycle consistency

DiscoGAN generator: conv, deconv and leaky relu; discriminator:conv+leaky relu L2 distance as cycle consistency

Chapter 3

Processing Data

3.1 Collection and Refinement of Data

In this section, we will explain about the progresses of collecting and refining data we've gone through. Table 3.1 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 3.1: Number of images used for training in each component.

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

3.1.1 Photograph

For realistic photographs of human faces, we used Celebrity A [7] 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 3.1.3.

3.1.2 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 3.1.3 After preprocessing steps, these images were aligned as fig 3.1

We also used CUHK(The Chinese University of Hong Kong) face sketch database(CUFS) [14], 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.

3.1.3 Preprocess

Figure 3.1: 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 [6] 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.

3.1.4 Generating Sketch Images

Chapter 4

Model Structure

4.1 Network Architectures

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.
Each generator consists of convoulutional layers as decoder layers and deconvoulutional layers as encoder layers and bottleneck layers. Pooling layers and batch normalization are also implemented in decoders and encoders. At the beginning, we implemented resnet to connect the decoder and encoder as suggested by the original paper. Later, we configured the generators to be u-shape nets where skip connections between downsampling layers and upsampling layers are enabled. In this way, low-level information is shared with upsampling layers without passing through bottleneck layers, which significantly reduces infomation loss and ensures feedback on internal structures of both inputs and outputs.
"For the discriminator networks we use 70 70 PatchGANs , which aim to classify whether 70 70 overlapping image patches are real or fake. Such a patch-level discriminator architecture has fewer parameters than a full-image discriminator, and can be applied to arbitrarily-sized images in a fully convolutional fashion ."

4.2 Stabilization of GAN Training

wGAN
dualGAN
Instance / Batch normalization

Chapter 5

Result

Figure 5.1 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 5.1: 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 5.2

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

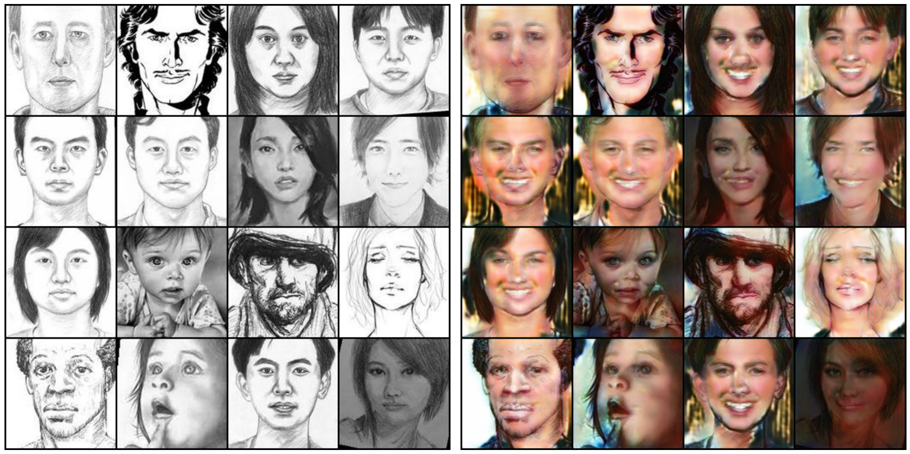

Figure 5.2: Original input images(left) and result of translations(right) by networks trained on datasets after face alignment, but before number of smiling faces was reduced. Image size is 128 by 128 and model is trained for 128 epochs.

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 3, we could get following results 5.3.

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.

Considering the quantity and quality of sketch images collected from google image search, this amount of dependency on input might be considered acceptible.

We have also tried to replace batch normalization by instance normalization to improve the training quality, but the results are not so satisfactory as shown in The mode collapse problem indicates that instance normalization causes instability of training in this case.

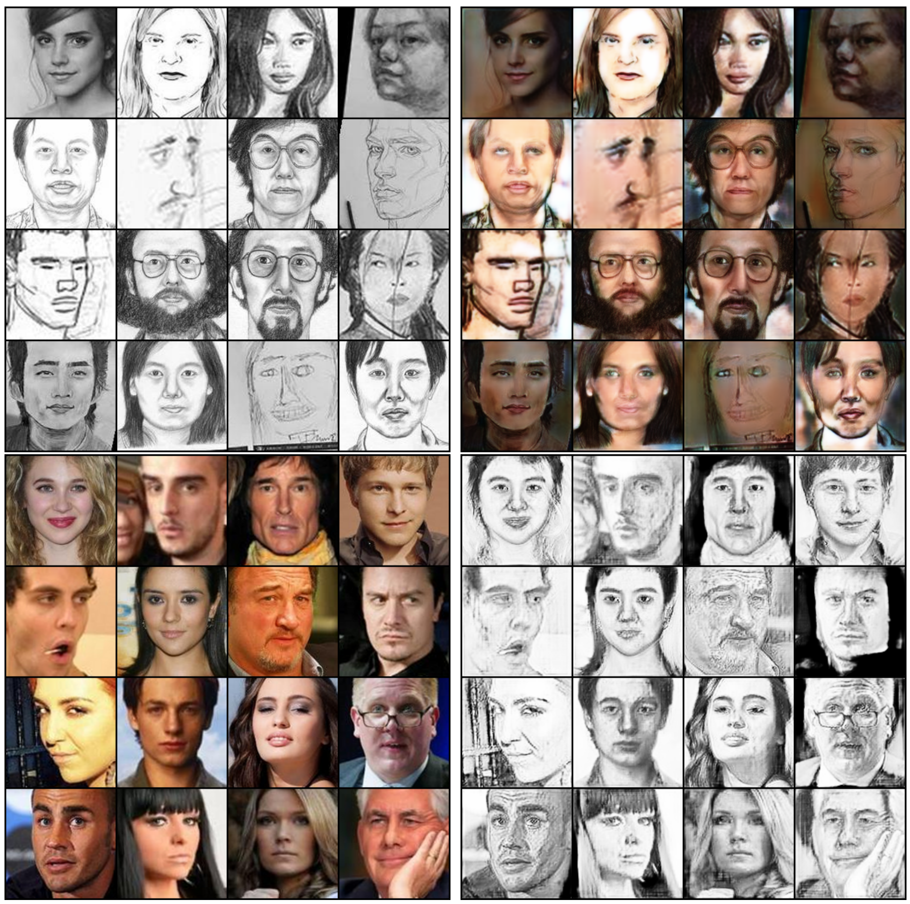

Figure 5.3: Original input images(left) and result of translations(right) by networks trained on datasets after face alignment and number of smiling faces was reduced. Image size is 128 by 128 and model is trained for 128 epochs.

Chapter 6

Conclusion

The point was made in the Acknowledgments section of this Sample Report that it is important to credit others whose work you use—it is a matter of professional ethics and courtesy. In addition to acknowledgment of broad assistance or contributions that you put into the Acknowledgments, you may also need to reference more specific contributions elsewhere in your text. Wherever a distinction is needed, make it clear which part of your work you have borrowed or adapted from others, and provide a reference to the source.

6.1 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.

Chapter 7

Reference

Chapter 8

Appendix

Appendix A

BibTEX Sample Records, Record Types and Fields

Appendix B

Where to find this sample RIPS report?

Read-only LaTeX source code for the RIPS Report Template, sample Beamer slide presentations, and other LaTeX supporting materials are available at,

```
Computer ->  IPAM RIPS FOLDER -> on the R Drive under under "Templates-etc"
```

Your report will be "copyedited", i.e., edited for conformance to the RIPS *House Style*. For reference, a table of proofreader's marks that may be used for markup of your draft is included. It was copied from *The Chicago Manual of Style, 16th ed.* (See original source at: `www.chicagomanualofstyle.org/tools_proof.html`.)

Appendix C

Glossary

Page vs Leaf: In bookbinding, a trimmed sheet of paper bound in a book; each side of a leaf is a **page**.

Opening: The two pages you see when you open a book. The right-hand **page** is the **recto**—and the left-hand page is the **verso**.

Recto: The front side of a **leaf**; in a book or journal, a right-hand page. To **start recto** is to begin on a recto page, as any major section—e.g., title page, table of contents, preface, chapter, appendix, bibliography—normally does. Contrast **verso**.

Verso: The back side of a **leaf**; the **page** on the left-hand side of an **opening**.

Front matter: As applied to this report, the material that appears in the front of the document, including title page, the abstract, acknowledgments, table of contents, list of figures, list of tables, usually numbered with lowercase roman numerals. RIPS reports initiate pagination with 1 in the front matter and proceed throughout with arabic numerals. This variation of usage is allowed because modern typesetting permits easy re-pagination after pages have been added to the front matter, something not easily done—after completion of the main matter—when typesetting was done by hand.

Main matter: The main part of the document, including the appendixes. **Page** numbers start from 1 using arabic numerals if front matter is enumerated using roman numerals.

Back matter: Material that appears at the back of the document, which in our report includes only the Bibliography.

Appendix D

Abbreviations

IPAM. Institute for Pure and Applied Mathematics. An institute of the National Science Foundation, located at UCLA.

RIPS. Research in Industrial Projects for Students. A regular summer program at IPAM, in which teams of undergraduate (or fresh graduate) students participate in sponsored team research projects.

UCLA. The University of California at Los Angeles.

Selected Bibliography Including Cited Works

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

[7] 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.

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

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

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

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

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

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

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

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

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