Report of Image-to-Image Translation with Conditional Adversarial Networks

Tianlin Yang 40010303
Yunqi Xu 40130514
Jiahui Wang 40070981
Tiancheng Xu 40079681
Tianshu Ji 40043638
Shuo Chi 40051473
Yiying Liu 40085613

***Abstract*— Image-to-image problem is one kind of computer vision and image processing problems. It aims to learn the mapping relationships between input images and output images. And it can be used in wide fields, such as collection style transfer, season transfer, color changing in the pictures and photo enhancement. [1] In this project, we re-implemented the pix2pix algorithm, which is a condition adversarial networks algorithm. We trained 150 and 20 pictures data sets separately. The results shows it works very well, and the processing times are 10 hours to 14 hours for each. For pix2pix algorithm, we accept 256*256 resolution image as input, the algorithm overall has good generalization and regularization ability. However, data set for pix2pix need to be well paired and pre-labeled. And trained model can only be used for the similar test case. To solve the shortages of pix2pix, we find CycleGan as improve algorithm, we applied an author's data set to do experimental test. Which includes 939 horse images to 1177 zebra images as training set, and it taking 30 minutes per epoch of training. Although the result shows it perform well, but still the training speed is extremely slow. To solve speed problem, we tried the simi-supervised CycleGan model and get significantly improved speed. After implemented the gender translate of human face in the images, we realize CycleGan is very powerful for unlabeled data. Unfortunately, it is unfriendly to multi task translation. Therefore, we do research on StarGan, which translate people's hair color, gender and age from original CelebA dataset at the same time. Finally, the Nvidia StyleGan has been researched as to understand current process in this field.**

I. INTRODUCTION

Many problems in computer vision, image processing can be seen as image-to- image translates. But how can a computer translate image-to-image automatically. The answer is deep net, now computer could do the tedious/repetitive work. (see Fig.1) Specifically, conditional adversarial networks is the basic method to solve image translation problems. This network learns the mapping from input images to output, as well as the loss functions. [2]

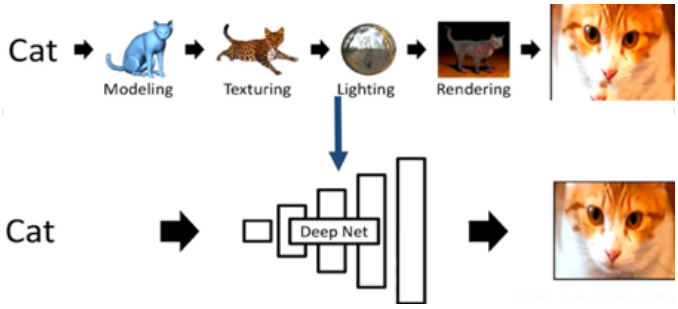

Fig. 1. Deep net example

In this report, we deeply understand many algorithms using to solve image translation problems and we implemented the pix2pix algorithm base on the conditional adversarial nets. We translate black and white images to RGB images. The results have good generalization and regularization ability. We also doing the mustache removing for people in the images. And with limited number of training data, can derive reasonable quality output. However, pix2pix algorithm still have some disadvantages, which makes us to find a better algorithm. CycleGan algorithm, which is an unpaired image-to-image translation. We use CycleGan algorithm to

translate horse images to zebra images as shown in Fig.2. After many days training, results work

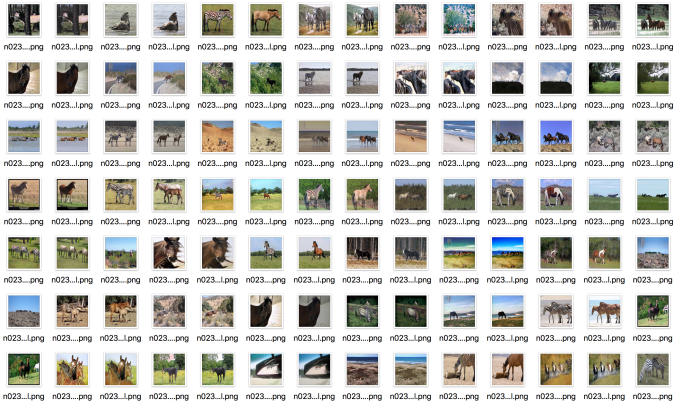

Fig. 2. Data set example

very well but training process is really slow. After dig-in, we found the solution (simi-supervised CycleGan) to solve speed problem. We use simi-supervised CycleGan changing the gender of people in photos. Finally, the speed is indeed increased (only 12 mins/epoch) and get much better results. Furthermore, we also did the research on the StarGan method, which is used for multi-domain image-to-image translation. With this method, it offer a stronger ability to do the multi-attributes translate (hair color, gender, age) for human face photo.

II. Various Methods

Some of the various methods being practiced worldwide are discussed below.

A. Generative Adversarial Nets

Generative Adversarial Nets is a new method for training generative models proposed by Goodfellow [3], which includes two "adversarial" models Eq.1: the generative model (G) is used to capture the data distribution, and the discriminative model (D) is used to estimate the probability that a sample comes from real data instead of generating a sample. In order to learn the generative distribution $P(g)$ on the real data set x, the generative model (G) constructs a mapping function $G(z;g)$ from the prior distribution $Pz(z)$ to the data space. The input of the discriminative model D is a real image or a generated image. $D(x;d)$ outputs a scalar, which indicates the probability that the input sample comes from the training sample.

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

There are two models (Generative and Discriminative) trained at the same time with following processes: Fixed discriminant model D, adjust the parameters of G to minimize the expectation of $log(1D(G(z))$, fixed generate model G, adjust the parameters of D so that $logD(X) + log(1D(G(z)))$. This optimization process can be reduced to a 'minimax two-player game' problem.

B. Conditional generative adversarial network

Conditional generative adversarial network is an extension of the original GAN as shown in Eq.2. Both the generator and the discriminator add additional information y as a condition. y can make any information, such as category information, or other modal data. the conditional GAN is implemented by feeding additional information y to the discriminant model and the generation model as part of the input layer. In the generative model, the prior input noise $p(z)$ and the condition information y form a joint hidden layer representation. The adversarial training framework is quite flexible in how the hidden layer representation is composed. Similarly, the objective function of a conditional GAN is a two-player minimax game with conditional probabilities.For training model, the data should be paired.

$$\min_G \max_D V(D,G) = E_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathrm{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))] \quad (2)$$

C. CycleGAN

CycleGAN as shown in Eq.3 can't train with the loss function of cGAN alone. In response, it propose called cycle consistency loss. The CycleGAN has some loss function as GAN, so the Loss function for first part could be defined as:

$$L_{\mathrm{GAN}}\left(F, D_Y, X, Y\right) = E_{y \sim p_{data}(y)}\left[\log D_Y(y)\right] + \mathrm{E}_{x \sim p_{data}(x)}\left[\log\left(1 - D_Y(F(x))\right)\right] \quad (3)$$

Assuming a mapping G, which can transform the picture y in space Y into the picture $G(y)$ in X. CycleGAN learns both F and G mappings at the same time. In other words, after converting the picture of X to space Y, it should still be able to be converted back. This prevents the model from converting all X pictures to the same picture in space Y. The loss of cyclic consistency is defined as shown in Eq.4:

$$L_{cyc}(F, G, X, Y) = E_{x \sim p_{data}(x)}\left[\|G(F(x)) - x\|_1\right] + \mathrm{E}_{y \sim p_{data}(y)}\left[\|F(G(y)) - y\|_1\right] \quad (4)$$

At the same time, the discriminator Dx for G, which can also define a GAN loss $L_{GAN}(G, D_Y, X, Y)$, and the final loss consists of three parts as shown below in Eq.5:

$$L = L_{\mathrm{GAN}}\left(F, D_Y, X, Y\right) + L_{\mathrm{GAN}}\left(G, D_X, X, Y\right) + \lambda L_{\mathrm{cyc}}(F, G, X, Y) \quad (5)$$

D. StarGAN

The StarGan figured out unpaired data sets image to image transformation, and apply multiple domains as shown in Fig.8.

In order to make image feature transfer to k features multiple domains, traditional GAN need establish $k * (k-1)$ generators, and the labels for data sets can't be reused. Therefore, StarGan inputs is domain information and the image, which convenient for multiple training data sets as combined one training process. This cross-domain training process via mask vector technique get successfully result on human emotional AI changing and face features automatically transformation.

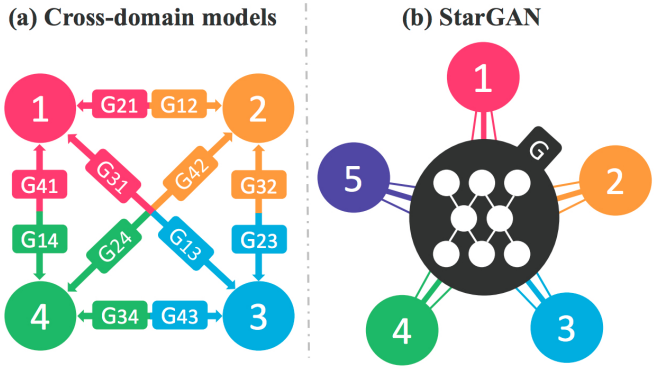

Figure 2. Comparison between cross-domain models and our proposed model, StarGAN. (a) To handle multiple domains, cross-domain models should be built for every pair of image domains. (b) StarGAN is capable of learning mappings among multiple domains using a single generator. The figure represents a star topology connecting multi-domains.

Fig. 3. StarGan

E. Nvidia StlyeGAN

The Style Generative Adversarial Network[7] is focus on generator improvement. Using mapping networks to map points in latent space to an intermediate latent space. The use of the intermediate latent space to control style at each point in the generator model shown in Fig.9, and the introduction to noise shown in Fig.10 as a source of variation at each point in the generator model.

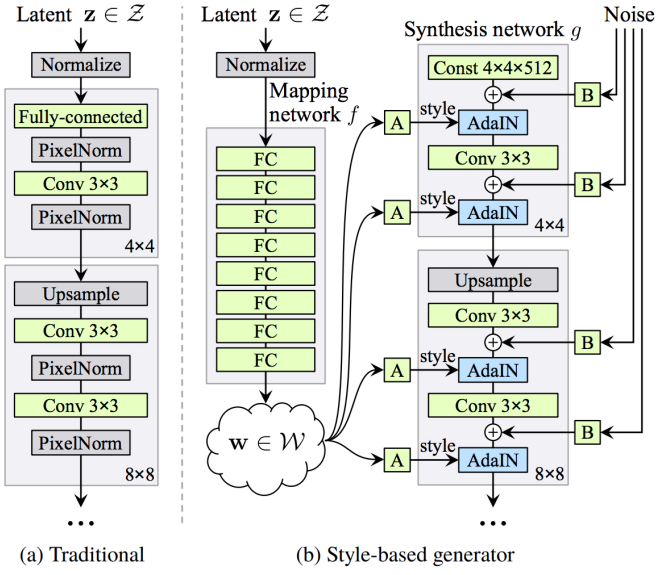

Fig. 4. StyleGan

The resulting model is capable not only of generating impressively photorealistic high-quality photos of faces, but also offers control over the style of the generated image at different levels of detail through varying the style vectors and noise.

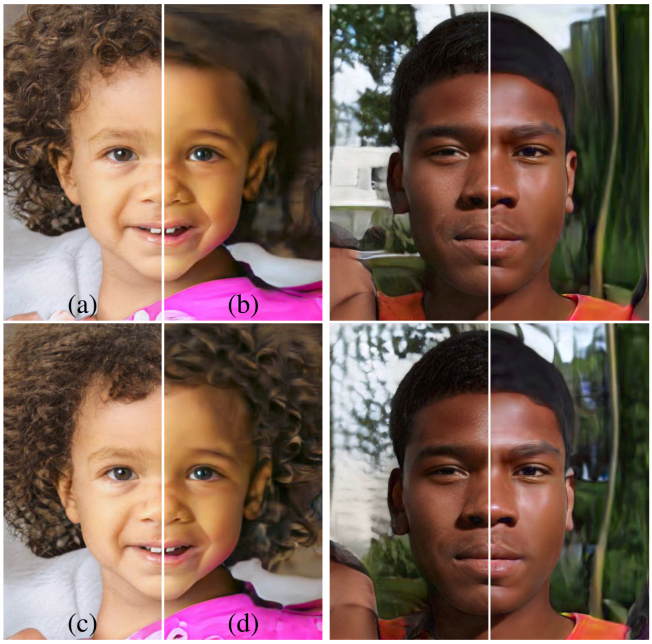

Figure 5. Effect of noise inputs at different layers of our generator. (a) Noise is applied to all layers. (b) No noise. (c) Noise in fine layers only (64^2 – 1024^2). (d) Noise in coarse layers only (4^2 – 32^2). We can see that the artificial omission of noise leads to featureless "painterly" look. Coarse noise causes large-scale curling of hair and appearance of larger background features, while the fine noise brings out the finer curls of hair, finer background detail, and skin pores.

Fig. 5. StyleGan noise

III. Experimental Process

In our project, we reproduced and analysed the results of Image-to-Image Translation with Conditional Adversarial Networks [4] which we chose as our main research. In the process of our research, we collected and explored several related academic papers. In these extending resources, we found an improvement technology about our main research, that is named Cycle-consistent Adversarial Networks [5], and we were pleasantly surprised that this paper is from an identical group with our chosen paper. Additionally, we introduced an pretty new technology in our project, which is named StarGAN [6], to study more about the current developments and performance. In the below, we separate these three technologies and illustrate the process of our experiments visually in details. Furthermore, we use the results of our experiments to do comparison of these three technologies.

A. *pix2pix: Grey2Color*

First of all, we explored the generality of conditional GANs [4], we selected two deferent applications to apply this Image-to-Image Translation technology:

- GreyColor
- Removing the mustache

Overall, the selection of the training datasets in generality of conditional GANs is really strict and inflexible, the training data must be the pair to appear in the one picture (shows in Fig.6) However, in the following researches, we clearly

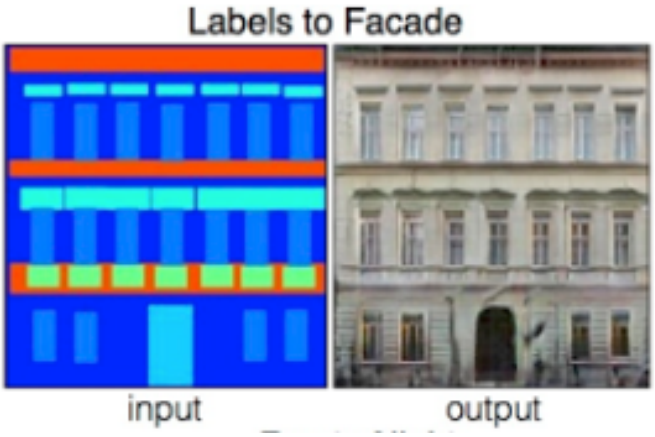

Fig. 6. The author's dataset Facede

knew this feature has indeed become a limiting factor of this perfect theory. And because of this shortage, researchers in this group proposed another new technology based on this and made data set more general and simple. We introduced this improvements in the below.

The first application that we applied Image-to-Image is on the transform from the grey picture to the color picture. We implemented a python program to generate pair pictures automatically, and finally because of the performance of our machines, we generated 150 pair 256*256 pictures to make up our gery2color dataset (shows in Fig.7)

After prepare the training dataset, we configured

Fig. 7. The grey2color dataset

all environments about this project including torch, nngraph, etc. All requirements of configuration show in the author's GitHub project repository https://github.com/phillipi/pix2pix. In the next, we trained the conditional GANs model. As for grey2color training dataset, although the size of dataset is really small (only 150 pictures), the machine of one member in our group does not has GPU. Therefore, this training process under the CPU mode spent 14 hours to iterate 120 epochs. Because of the capacity of the machine and time issue, we stopped training process in 120 epochs. And we selected a dataset to test the performance of this trained model. The final result by 120 epochs model shows in the below (Fig.8) and the more details and comparisons are on the results part.

Fig. 8. Results of gery2color with 120 epochs

B. pix2pix: Removing the mustache

The second application that we applied Image-to-Image is removing the mustache in the pictures. The reason that we did this interesting application is inspired from some dominate picture editing applications. The collecting of training dataset is extremely difficult, we need to find the pair picture from one guy with and without mustache and the angel and external environment such as main color and decorations should be as similar as possible. Due to this, we only generated 50 pairs pictures as our training dataset in this project. (shows in Fig.9.) Due to the size of dataset for this project

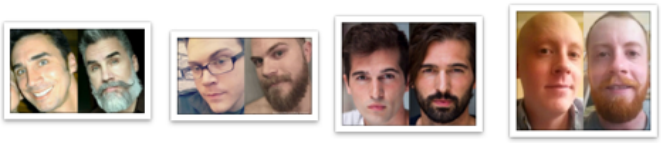

Fig. 9. From the mustache dataset

is pretty small, we trained model to iterate 200 epochs but only used 10 hours. Then we tested our project, final result shows in below (Fig.10):

Fig. 10. Results of removing mustache

Due to some reasons, the results are a little vague but the overall trend can be seen, we analysed in results part.

It is obviously that the training dataset is not simple to generate and collect. Therefore, we found an improvement technology can fix it out.

C. CycleGan: Switching gender

The most surprised found in this technology is the training dataset can be not only the pair pictures but also some general photos. This improvement is obviously useful not only for the research but also in the industrial side, and greatly reduced our time costs for the works in training dataset generating and collecting (appearance of training data shows in Fig.11).

Therefore, we can use more normal photos to train our model, we applied this technology in one application.

Due to the requirements of this technology to the machine are pretty demanding, one of our group member used his machine with GPU to train this model and even used GPU he still trained more than 20 hours to obtain 100 epochs trained model eventually. The results show in below (Fig.12).

Fig. 11. zebra2house data set

Fig. 12. The results of Switching gender

D. The results of Switching gender

This technology is based on the CycleGan which requires two generators and discriminators for each pair of two different domains [6]. StarGan can not only transfer one feature but also it can change several features together such as hair colors, ages (belong to CelebA) and emotions (belongs to RaFD). We applied this technology in our previous Switching gender, trained on the SwitchGender dataset. From the beginning, we have still configured our environment including pytorch, visdom and dominate etc. All the dependencies explain in the pytorch-CycleGAN-and-pix2pix project in the GitHub https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix. After that, we prepared our SwitchGender training data set, we found 2000 photos (include 1000 women's and 1000 men's photo) to make up this data set.

Own data set but due to the process of training in this technology is extremely long and the requirements of machine are pretty high, we just finished CelebA parts of this technology which spent more than 20 hours to obtain results. As for RaFD part which is used to change emotions, we show in below the results from the author's pre-trained model (Fig.13).

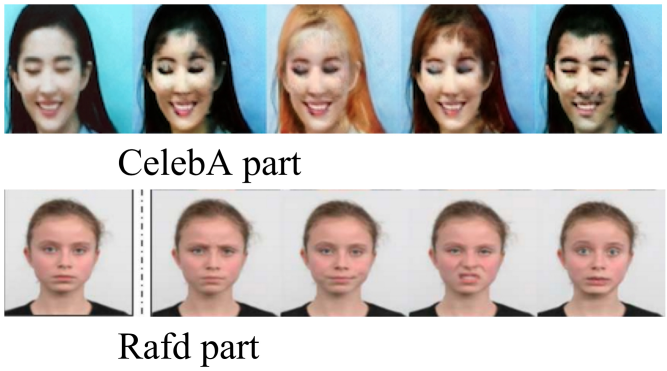

Fig. 13. Results from StarGan

IV. Results Discussion

In this part, the comparison with our results with //we displayed some of our results and compare with author's results to do some researches. We separate to two parts, conditional Gan and Cycle-Gan to do horizontal and vertical comparison.

A. Conditional Adversarial Networks

We implemented two cases to study this technology, Grey-to-Color and mustache removing. Both of them get excellent results finally.

In the below (Fig.14) we displayed the results from 5 epochs to 120 epochs, we found an interesting point. Due to the dataset we chose for training, the results are close to that even reverse the facts. From the figure we can know, the lady's suit is color for red but after our trained the color goes black. And the sports outfit goes to red rather than

green for the same reason. But we can know the quality of results is generally better.

Fig. 14. Red suit goes black

B. Cycle-consistent Adversarial Networks

This algorithm requires higher machine configuration and large data set for training. Therefore, in our results, some of them get really perfect quality but some are not (Fig.15). And from the

Fig. 15. Gender Change Comparison

author's final results (Fig.16) we found our results are vaguer and we think that is because the training time is not enough and size of training dataset is not sufficient.

Fig. 16. Style changing

TABLE I
OUR TEST RESULTS

Dataset	Dataset size	Epoch	Data requirements
Multi attributes	200000	278	Pre-labeled all attributes
Gender switch	6000	100	Label the gender
Facades	400	100	Complete pattern Well paired
Gray2Color	150	120	Well paired
Remove Mustache	50	200	Label mustache

V. Conclusion

By expanding our research paper, we deeply understand cGAN progress. We start from implementing pix2pix, improving our results by CycleGAN and researching StarGan to know the increasing amount of papers from 2014 to 2018 and the development pattern classification process that from pix2pix in 2016 to CycleGAN 2017, pix2pixHD, StarGAN in 2018 and now NVIDIA styleGAN.

This is a good experiment in the understanding of the paper by implementing the pix2pix algorithm and we got different results as shown in Table.I by running different models and targets but after researching this paper we also found some shortages. So, we dug out the subsequent algorithm CycleGan to generalize the datasets and we used StarGan to apply single training process to multiple attributes. At the end we compared all models' outputs and revealed the pros and cons.

Appendix

The results for pix2pix, CycleGAN and StarGan can be seen in folder "result". All codes included in the folder "codes". The data sets CelebA is too large and it should download from: http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

Put the data in correctly positions and the detailed steps can be found in "README".

This projects needs NVIDIA CUDA GPU support to accelerate the speed of training process, the

hardware we are using is: GTX-980m 8GB, 64GB RAM, i5 6600k(4 cores OC 5Ghz).

REFERENCES

[1] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In ICLR, 2015.

[2] P. Isola et al, "Image-to-image translation with conditional adversarial networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[3] Goodfellow Ian, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[C]//Advances in Neural Information Processing Systems. 2014: 2672-2680.

[4] P. Isola et al, "Image-to-image translation with conditional adversarial networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[5] J. Zhu et al, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in Proceedings of the IEEE International Conference on Computer Vision, 2017.

[6] Y. Choi et al, "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.

[7] T. Karras et al, A Style-Based Generator Architecture for Generative Adversarial Networks, CVPR2019, 2018