UNIVERSITY OF CALIFORNIA

Los Angeles

Automatic Sketch to Photo Translation

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Statistics

by

Zhenli Jiang

2018

ABSTRACT OF THE THESIS

Automatic Sketch to Photo Translation

by

Zhenli Jiang
Master of Science in Statistics
University of California, Los Angeles, 2018
Professor Yingnian Wu, Chair

We present an application of conditional generative adversarial network(cGAN) to produce photo-realistic portraits based on human face sketches.

Our basic U-Net and PatchGAN model architecture is from pix2pix GAN. U-Net is a generator that skips connections between each layer i and layer n-i of neural networks to preserve lower-layer information between inputs and outputs, and PatchGAN is a discriminator modeling on small patches of images to force high-frequency correctness.

Based on U-Net and PatchGAN, we tried different loss functions, including L1+cGAN, L2+cGAN, L1+GAN and L1+WGAN. By training the paired images of sketches and real photos, the results show that the L1+cGAN and L1+WGAN are able to produce pictures of acceptable quality. We even found that our L1+WGAN loss has better performance than the original pix2pix model.

The results of our application are promising: everyone can get any photo-realistic portraits by their own drafts!

The thesis of Zhenli Jiang is approved.

Chad J Hazlett

Hongquan Xu

Yingnian Wu, Committee Chair

University of California, Los Angeles

2018

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

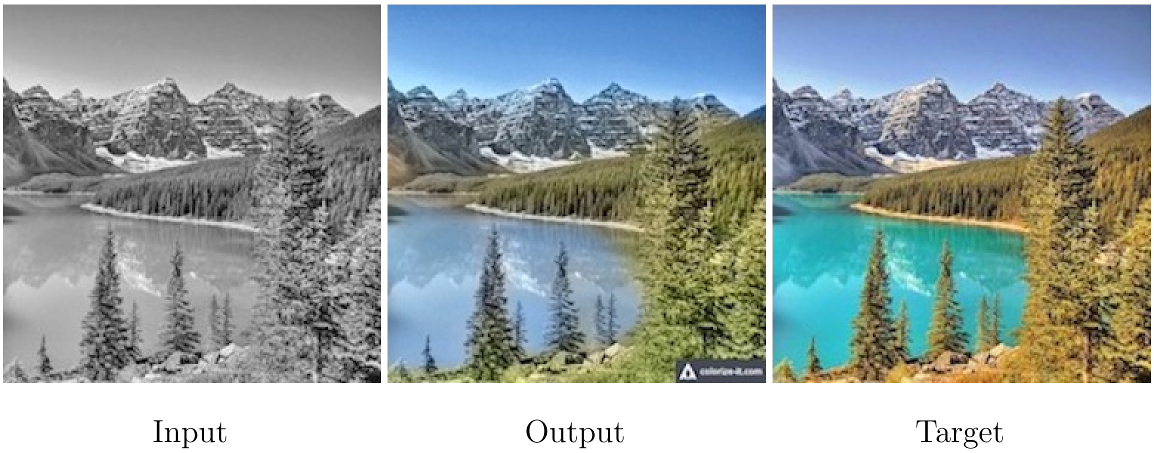
Introduction

The generative adversarial network(GAN) is a framework for training generative models to sidestep the difficulty of approximating many intractable probabilistic computations. It is implemented by two neural networks contesting with each other in a zero-sum game framework[9]. Since it was first introduced by Goodfellow et al.in 2014, different variations of GANs have been introduced. For example, in 2014, Mirza et al. proposed conditional GAN (cGANs), which control on modes of the data being generated. However, GAN is known to be unstable to train and often suffer from the problem of mode collapse. Recently, Arjovsky et al.[6] defined a new form of GAN called Wasserstein GAN (WGAN). WGAN improves the stability of learning and are able to generate images of different modes. [11]

Basically, the model architecture of our experiments is based on the paper ”Image-to-Image Translation with Conditional Adversarial Networks”[14](pix2pix GAN). It trained paired images with the idea of conditional GAN. The model is able to translate the original image into another one by some learned rules. The paper of pix2pix GAN has presented the examples of successfully building connections between Cityscapes and photo, Edges and shoes, Edges and Handbag, Day and night. etc.

Pix2pix GAN is of great practical value and enjoys high popularity since it was invented. There are a number of online applications based on pix2pix GAN now. Such as ”mimic of the facial expression of the German chancellor”[2], ”edge2cats”[3] and tons of automatic colorization applications such as Figure(1.1).

Our goal in this paper is to construct grey-scale facial images based on human face sketches. We produced the ”sketch version” of original photos by Canny edge detector and then trained them in pairs. In addition, we tried different loss functions for the model, in-



Input

Output

Target

Figure 1.1: Automatic colorization by pix2pix[1]

cluding L1+cGAN, L2+cGAN, L1+GAN and L1+WGAN. To our surprise, the L1+WGAN combination presents even better performance than the recommended L1+cGAN objective function. The results show that the pix2pix GAN is quite good at producing realistic pictures from very simple sketches. If you are interested in it, you can even try the model using your own sketches!

CHAPTER 2

Method

2.1 Edge detection

Since sketch-photo pairs are required for training, we applied the canny edge detector[7] to automatically generate the sketch version of the original photo. It is a technique to extract useful structural information from vision objects.

The process of Canny edge detection algorithm:

- Apply Gaussian filter to remove the noise.
- Calculate G_x and G_y by standard sobel edge detector, and get the magnitude $m = \sqrt{G_x^2 + G_y^2}$ and the direction $\theta = arctan(\frac{G_y}{G_x})$
- Compare the edge strength of the current pixel with the edge strength of the pixel in the positive and negative gradient directions. If its value is largest, it will be preserved.
- Strong edge pixel: gradient value > high threshold value.
  Weak edge pixel: low threshold value < gradient value < high threshold value
- Preserve a weak edge pixel as long as one of its 8-connected neighborhood pixels is strong edge pixel.

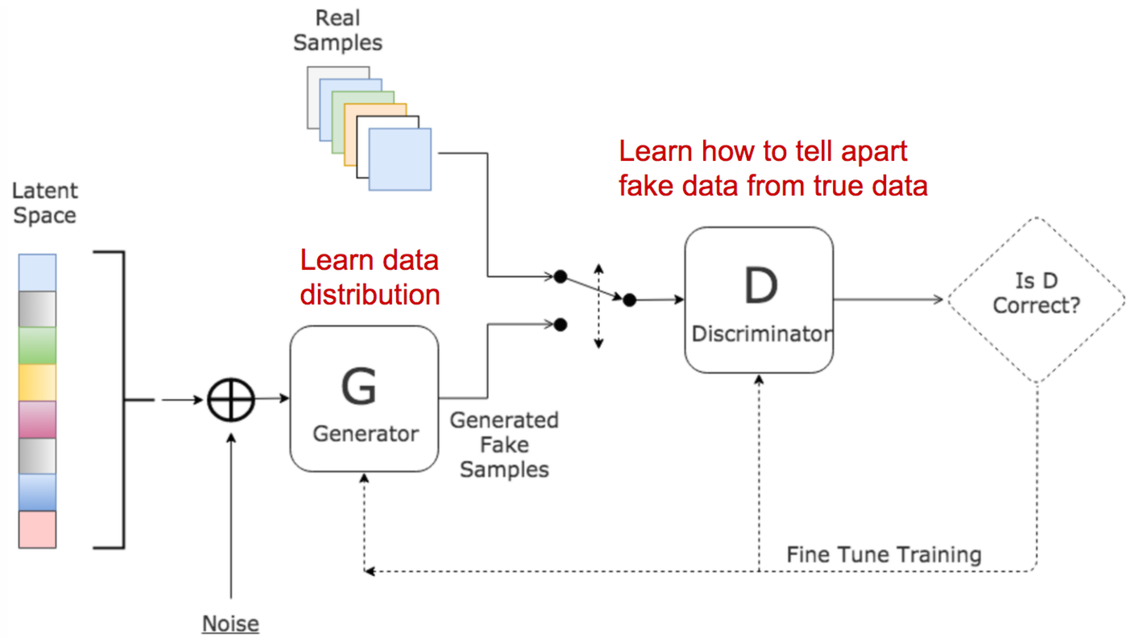


Figure 2.1: Architecture of GAN[4]

2.2 Training Process

GAN

Figure(2.1) presents the architecture and training process of a simplest GAN. The training process of GAN is like a zero-sum game between the generator and the discriminator. The generator G generates images $G(z)$ from random noise z. The discriminator D simultaneously learns a mapping from an image, which might be the real image y or the fake image $G(z)$, to some values between 0 to 1. The values of $D(x)$ indicates the probability that the x is from the true data distribution.

- The goal of discriminator D: maximize $D(y)$ and minimize $D(G(z))$.
- The goal of generator G: maximize $D(G(z))$.

The original objective function proposed by Goodfellow et al.is:

$$\underset{G}{min}\,\underset{D}{max}\,\mathscr{L}_{cGAN(G,D)} = \mathbb{E}_y[\log D(y)] + \mathbb{E}_z[\log(1 - D(G(z))]$$

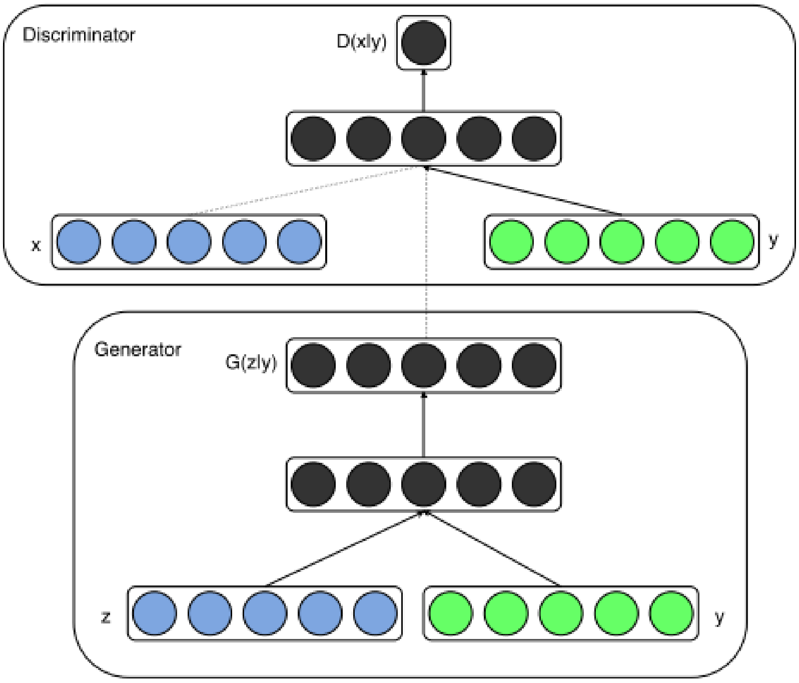


Figure 2.2: Discriminator and Generator of Conditional GAN

Hence we have two loss functions, one for discriminator and another for generator. In each iteration, generator generates the image and then discriminator will judge the possibility that the input(fake image and real image) is real. The network parameters will be updated by these two loss functions. Note that sometimes we may need to train the generator more times than the discriminator(eg: the original GAN), or vice versa(eg: the WGAN).

Since GAN was first introduced in 2014, different variations of loss functions(L1 penalty, L2 penalty, LSGAN, WGAN GAN, etc) of GAN were introduced. We will discuss some of them in the next section.

Our Model

Our model is based on conditional GAN(cGAN)[15]. As Figure(2.2) shows, cGAN provides the extra information while generating or discriminating images.

Initially, the generator generates an output image. The discriminator looks at the input/target pair and the input/output pair and produces its guess about how realistic they look. Then the parameters will be updated in the same way as the traditional GAN.

2.3 Objective functions

The pix2pix model has illustrated the combination of loss functions :L_1 + CGAN should be the optimal by the measure of "FCN score". The objective is:

$$arg\,\underset{G}{min}\,\underset{D}{max}\,\mathscr{L}_{cGAN}(G,D) + \lambda\mathscr{L}_{L1}(G) \tag{2.1}$$

In this paper, we will also show the training results of different combinations of loss functions other than L_1+cGAN.

GAN

The original objective for GAN should be:

$$\underset{G}{min}\,\underset{D}{max}\,\mathscr{L}_{cGAN(G,D)} = \mathbb{E}_y[\log D(y)] + \mathbb{E}_z[\log(1 - D(G(z))] \tag{2.2}$$

y: sampled data z: input noise, D: discriminator, G: generator.

We train D to maximize the probability of assigning the correct label to both training examples and samples from G. We simultaneously train G to minimize $\log(1 - D(G(z))$ [10].

Conditional GAN

Conditioned on the extra information x[15], the objective of conditional GAN becomes:

$$\mathscr{L}_{cGAN(G,D)} = \mathbb{E}_{x,y}[\log D(x,y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x,z))] \tag{2.3}$$

L1 Loss

An L_1 term can be added to the loss function.

$$\mathscr{L}_{L_1(G)} = \mathbb{E}_{x,y,z}[||y - G(x,z)||_1] \tag{2.4}$$

L2 Loss

L_2 loss can also be applied to training GAN as part of the loss function[16]. It encourages blurriness in comparison with the L_1 loss.

$$\mathscr{L}_{L_2(G)} = \mathbb{E}_{x,y,z}[||y - G(x,z)||_2^2] \tag{2.5}$$

Wasserstein GAN

There are some experiments trying to optimize the similar architecture of pix2pixGAN based on Wasserstein GAN(WGAN) loss, such as the paper dualGAN[19]. Theoretically, with the use of EM distance, WGAN will stabilize the training process of GAN.
The EM distance is a measure of distance between two distributions:

$$W(P_r, P_g) = \inf_{x,y\sim\gamma} \mathbb{E}_{x,y}[||x - y||] \tag{2.6}$$

After some transformations, our goal becomes:

$$max_{w\in W}\, \mathbb{E}_y[f_w(y)] - \mathbb{E}_z[f_w(g_\theta(z))] \tag{2.7}$$

where $\{f_w(x)\}_{w\in W}$ are all K-Lipchitz for some K.
Objective of conditional WGAN:

$$min_{w\in W}\{-\,\mathbb{E}_y[f_w(x,y)] + \mathbb{E}_z[f_w(x, g_\theta(x,z))]\} \tag{2.8}$$

Based on the original GAN, to implement WGAN, it requires us to:

1. Change the loss function to be in accordance with the EM distance
2. Remove the sigmoid function from the discriminator
3. Use a trick of weight clipping to enforce the Lipchitz constraint.
4. Change the optimization function to RMSProp

It has been illustrated that weight clipping method exhibits some unexpected behavior by [12]. An alternative better method to enforce Lipchitz constraint is Gradient Penalty.

$$L = \mathbb{E}_{\tilde{x}\sim P_g}[D(\tilde{x})] - \mathbb{E}_{x\sim \mathrm{Pr}}[D(x)] + \lambda\, \mathbb{E}_{\hat{x}\sim P_{\hat{x}}}\, \mathbb{E}[(||\Delta_{\hat{x}} D(\hat{x})||_2)^2] \tag{2.9}$$

Where $P_{\hat{x}}$ sampling uniformly along straight lines between pairs of points sampled from the data distribution Pr and the generator distribution Pg.

2.4 Network architecture

The network architecture of our model, including the number of layers, the size of filters, etc, is exactly the same as the pix2pix model. More details are in [14], appendix.

The U-Net

For the generator part, unlike the traditional encoder-decoder structure, the pix2pix GAN[14] invented the U-Net under the consideration that "For many image translation problems, there is a great deal of low-level information shared between the input and output, and it would be desirable to shuttle this information directly across the net".

The U-Net skips connections between each layer i and layer n-i, where n is the total number of layers. Each skip connection simply concatenates all channels at layer i with those at layer n-i. Figure(2.3) shows the simplified architecture of our U-Net.

Encoder:

C64-C128-C256-C512-C512-C512-C512-C512

U-Net decoder:

CD512-CD1024-CD1024-C1024-C1024-C512-C256-C128

After the last layer in the decoder, a convolution is applied to map to the number of output channels, followed by a Tanh function. As an exception to the above notation, Batch Norm is not applied to the first C64 layer in the encoder. All ReLUs in the encoder are leaky, with slope 0.2, while ReLUs in the decoder are not leaky.

PatchGAN

The discriminator only penalizes structure at the scale of patches. It tries to classify if each patch is real or fake. Then run this discriminator convolutionally across the image, averaging all responses

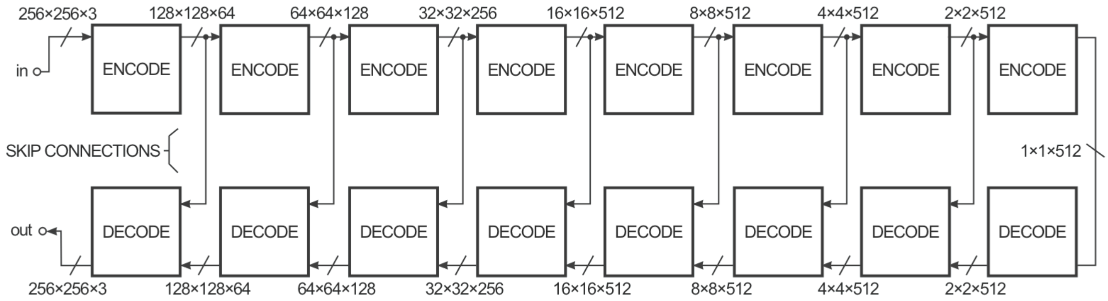


Figure 2.3: A simplified form of U-Net[13]

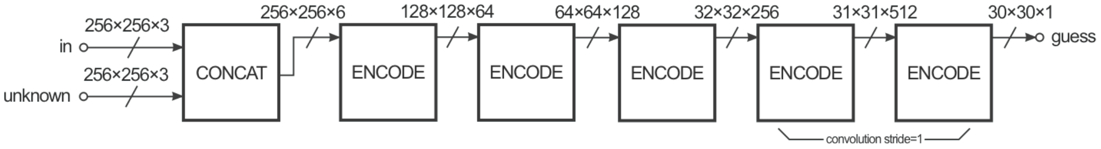


Figure 2.4: PatchGAN[13]

to provide the ultimate output of D. Here we used the patch of size 70*70 pixels.

Figure(2.4) shows the architecture of the 70 × 70 discriminator PatchGAN, it should be:

C64-C128-C256-C512

After the last layer, a convolution is applied to map to a 1 dimensional output, followed by a Sigmoid function. As an exception to the above notation, BatchNorm is not applied to the first C64 layer. All ReLUs are leaky, with slope 0.2.

PatchGAN motivates restricting the GAN discriminator to only model high-frequency structure, relying on an L1 term to force low-frequency correctness.

2.5 Evaluation Methods

Visual Evaluation

We have several principles of our visual evaluation:

(1)Whether they have unacceptable defects according to our common sense. A picture can be ruined by a black stain, no matter how perfect it is without the stain. On the other hand, flaws such as uneven skin tone might not be regarded as fatal.

(2)Whether they are photo-realistic. It's quite intuitive: as long as we have the feeling that "this picture must be generated by computer", then it fails to meet our standard.

Simple and straightforward as it is, visual evaluation turns out to be the most reliable approach.

Structural Similarity

The Structural Similarity (SSIM) index is a method for measuring the similarity between two images[18]. It is a perception-based model that considers image degradation as perceived change in structural information, while also incorporating important perceptual phenomena, including both luminance masking and contrast masking terms.

The SSIM formula is based on three comparison measurements between the samples of x and y: luminance (l), contrast (c) and structure (s). The individual comparison functions are:

$$l(x,y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}$$

$$s(x,y) = \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3}$$

with:

$c_1 = (k_1L)^2$ and $c_2 = (k_2L)^2$ two variables to stabilize the division with weak denominator

$c_3 = c_2/2$

L: the dynamic range of the pixel-values (typically this is $2^{\#bits\ per\ pixel} - 1$)

$k_1 = 0.01$ and $k_2 = 0.03$ by default.

SSIM is then a weighted combination of those comparative measures:

$$SSIM(x,y) = \left[l(x,y)^{\alpha} \cdot c(x,y)^{\beta} \cdot s(x,y)^{\gamma}\right]$$

Gradient Magnitude

One disadvantage of SSIM is that blurring images are likely to score higher.

Therefore, to counter the effect of SSIM, our evaluation method should take the sharpness into account. The average gradient is a good way to measure sharpness.

The gradient of an image is a vector of its partials:

$$\nabla f = \begin{bmatrix} g_x \\ \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

To calculate $\frac{\partial f}{\partial y}$ we can apply a 1-dimensional filter to the image A by convolution:

$$\frac{\partial f}{\partial y} = \begin{bmatrix} -1 \\ \\ +1 \end{bmatrix} * \mathbf{A}$$

where $*$ denotes the 1-dimensional convolution operation. The magnitude is given by:

$$\sqrt{g_y^2 + g_x^2}$$

.

CHAPTER 3

Experiments

The primary goal of our experiments is to produce realistic pictures according to face sketches. To train the model, we utilized the facial images in the color-feret-database. Then we generated the face sketches using the Canny edge detector algorithm. During the training process, we tried different loss functions on the basis of original pix2pix GAN. Finally, we evaluated our models by checking the facial images(SSIM, sharpness, visual evaluation) produced by the models.

3.1 Dataset

The training and testing images are from the color-feret-database[8]. The database contains a total of 11338 facial images as in Figure(3.1). They were collected by photographing 994 subjects at various angles, over the course of 15 sessions between 1993 and 1996.

Prepossessing

- Convert the original color images to the grey scale images.
- To produce the sketch of a photo, we applied the Canny edge detector[7] with low threshold value 100 and high threshold value 200.
- For some photos that are too obscure or too complicated, canny edge detector cannot depict the outline of them accurately. Therefore, we only preserved the sketches that satisfy: [1]

$$0.03 < \frac{black - pixels}{total - pixels} < 0.06$$

[1] Most sketches filtered by this range can reflect the original portrait properly. However, some of them still have defects such as no lines for eyes, too many lines for the face muscles, etc. We eliminated those flawed images in the testing set, but preserved them in the training set since we can get reasonable models even with them.

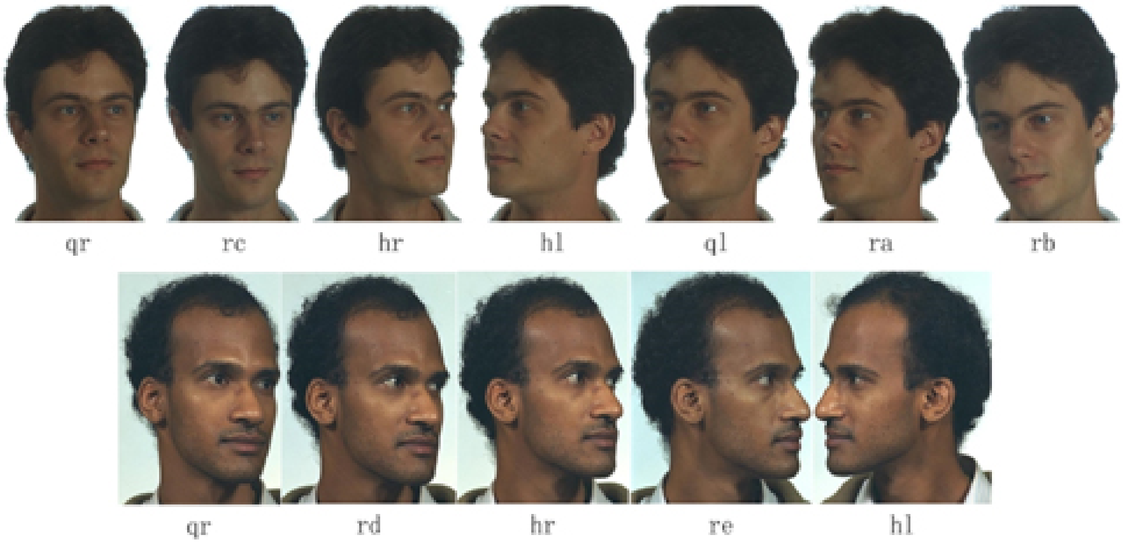


Figure 3.1: Images in color feret database

- Re-size the paired images to be 256*256 pixels.

One example of our training pairs is in Figure(3.2). After the prepossessing, we have 3997 images left.

We tried to train the model with different training sets: one consists of 3200 images, and a relatively smaller one with randomly selected 511 images. The testing set contains 122 images.

Since from [14] batch size 1 can produce better result for the U-net, the batch size for our training process is 1.

3.2 Results

3.2.1 L1 + cGAN

Firstly, we implemented the optimal combination of loss functions demonstrated by [14]: L1 loss on generator loss + cGAN. The L1 loss weight : Generator loss weight = 99 : 1.

We tried training 3200 images as well as a subset of it containing 511 images. Table(3.1) exhibits significant improvement in the result image quality for the first 100 epochs. However, the quality for results stays almost the same after the number of epochs is larger than 150. Therefore, we set the number of training epochs to be 200. It will be fruitless to train more epochs but risky to train

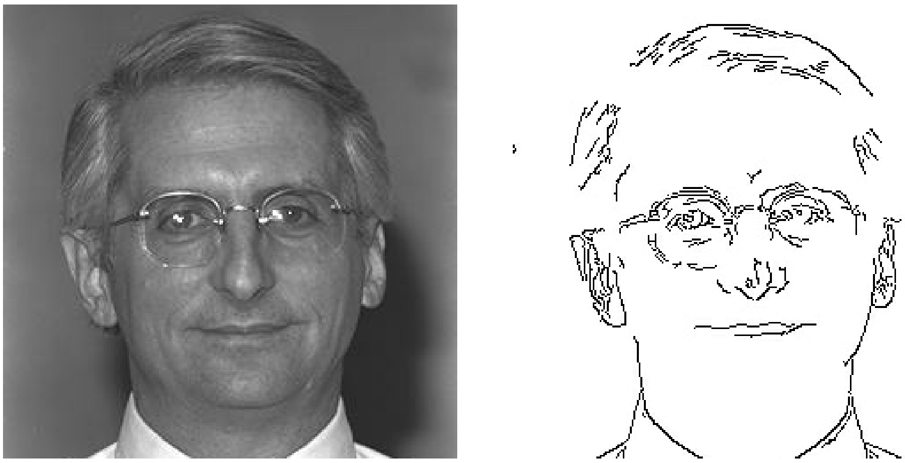


Figure 3.2: Example of paired training images

less.

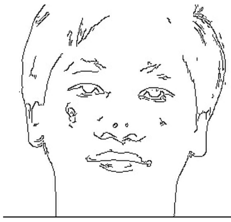
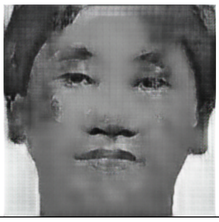
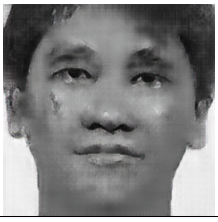
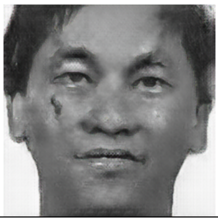
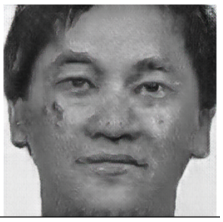
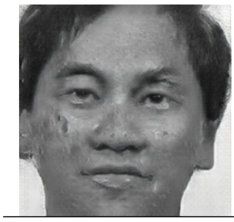
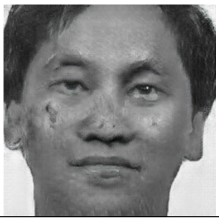
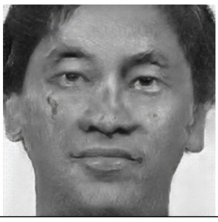
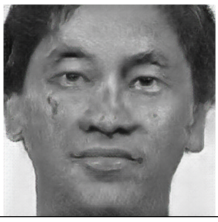
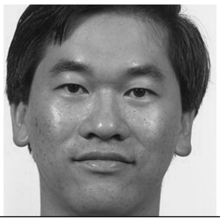
Input	10 epochs	20 epochs	50 epochs	80 epochs
				
100 epochs	120 epochs	160 epochs	200 epochs	target
				

Table 3.1: Training steps for L1 + cGAN, 511 images

Figure(3.3) shows the comparison between two different training groups. Intuitively, the results from the 3200 training group are more 'colorful', with evener skin tone and darker hair color. In addition, they are with more details if you examine people's eyes in the pictures. The result images from the 511 group look more like old and faded portraits since there are several noticeable defects in them. On the other hand, there are few unnatural things in the results for the 3200 group. One of them might be people's lips.

The pix2pix GAN has demonstrated its ability to obtain decent results for some small datasets

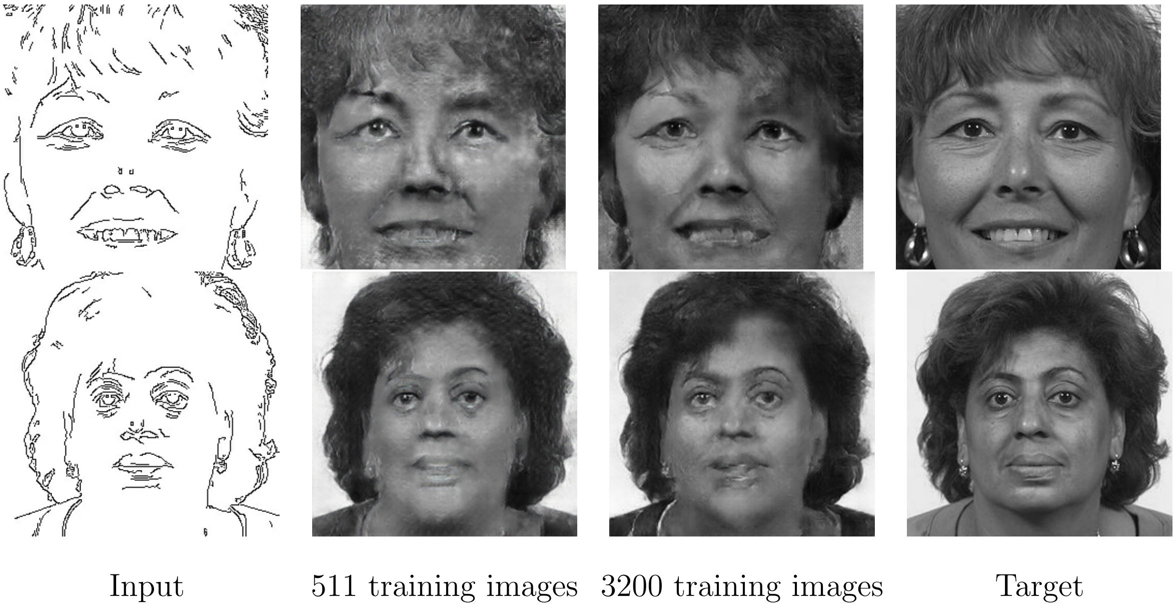


Figure 3.3: 511 training images vs 3000 training images

such as the example of facades[14], which consists of just 400 images. In some extent, the results of our 511 training group might also be regarded as ”decent”. But it’s undeniable that, in our situation, more training images is a quite straightforward and effective approach to improve the quality of results.

However, it will take too much time to train a model with 3000+ images. We did most of the experiments for various of trials using the 511 image training set, since our goal was to compare the effects of different loss functions at some certain stage. After all, if the sketch used for testing is clear enough, the model(L1 + cGAN , 511 images) can produce the satisfying portraits for different angles, hairstyles, etc as shown in Table(3.2).

3.2.2 L1 + GAN

To test the importance of conditioning the discriminator, we also compare to an unconditional variant in which the discriminator does not observe x:

$$\mathscr{L}_{cGAN(G,D)} = \mathbb{E}_{x,y}[\log D(y)] + \mathbb{E}_{x,z}[\log(1 - D(G(x,z))] \tag{3.1}$$

Most ’area’ of the result images are of good quality. However, the generator seems to make

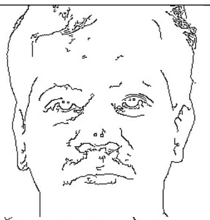
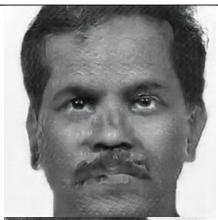
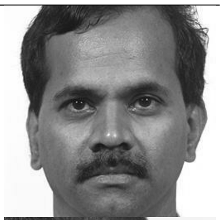
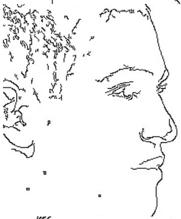
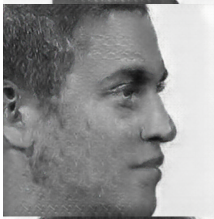
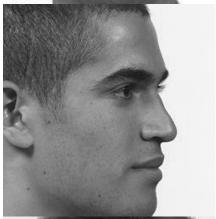
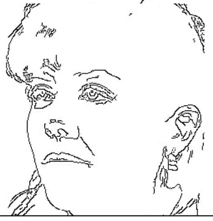
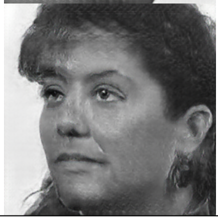
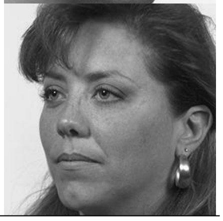
	Input	Output	Target
Regular frontal			
Head right			
Quarter left			

Table 3.2: More results for L1 + cGAN, 511 training images

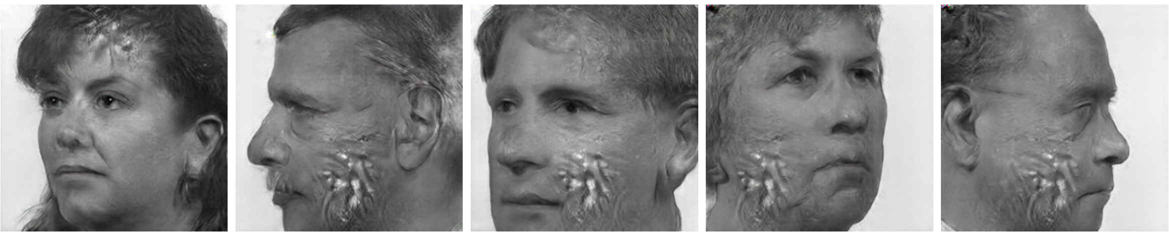


Figure 3.4: Flawed images generated by L1+GAN

stupid mistakes quite often. There are some pretty eye-catching defects in our results(eg: the hair in the first picture and the forehead in the second picture of Figure(3.5) and Figure(3.4)). Without conditioning on the discriminator, the generator didn't know how to deal with blank space, went out of control, and hence produced some weird patterns for it.

Even though without those defects, the rest part of the images look pretty good, the result of GAN+L1 is still unacceptable.

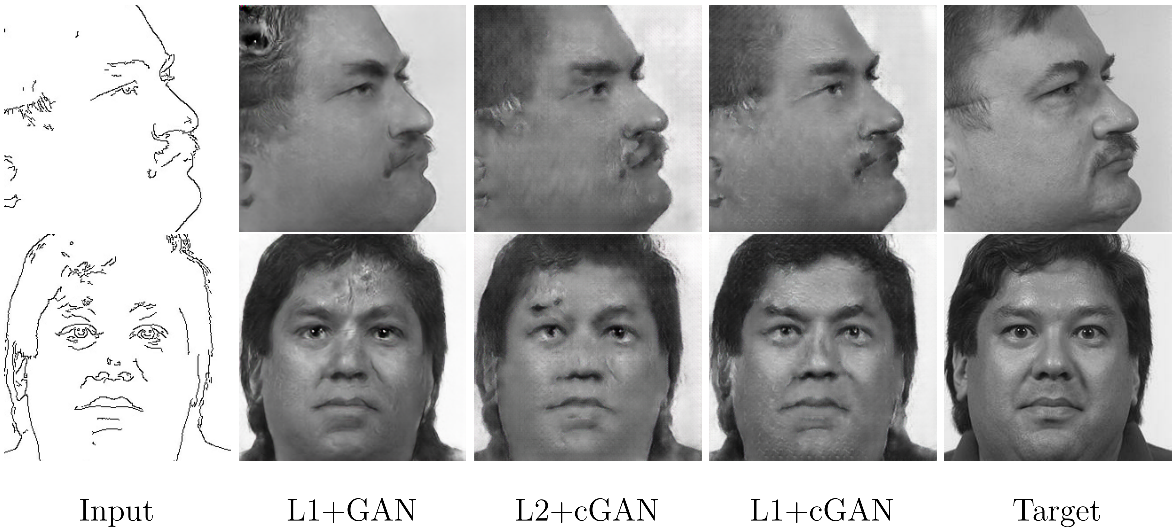


Figure 3.5: Comparisons among L2+cGAN, L1+cGAN and L1+GAN

3.2.3 L2 + cGAN

We tried the combination of L2+cGAN. Generally, L1 loss is more preferable for image translation and can generate more photo-realistic pictures. L2 loss fails to capture multimodal distributions, while L1 loss encourages sparseness and is invariant to the noisy inputs.

As Figure(3.5) shows, L2 loss indeed blurred the images as we have expected. If the original sketches are good enough, L2 loss can still reconstruct acceptable pictures.

From Figure(3.6) the discriminator loss and generator loss of L1 and L2 are of the same shape. If we evaluate the results in a quantitative way by the discriminator loss, the Figure(3.6) also reflects that the L2's discriminator loss is a little bit higher than L1's. The mean discriminator loss for the last 25 epochs is 0.175 for L1 and 0.181 for L2.

3.2.4 WGAN

There are works which make use of pix2pix GAN's network architecture(U-Net+PatchGAN) and perform some modifications. One of them is the paper of DualGAN: it absorbed the idea of WGAN Loss[19] and tried optimizing the U-Net and PatchGAN based on it.

Even though after the paper has been published, the author of DualGAN found that the usual loss function of GAN could have better performance than WGAN [5], it is still a good idea for us to give a try. WGAN has a number of attractive properties: theoretically, it overcomes the problems

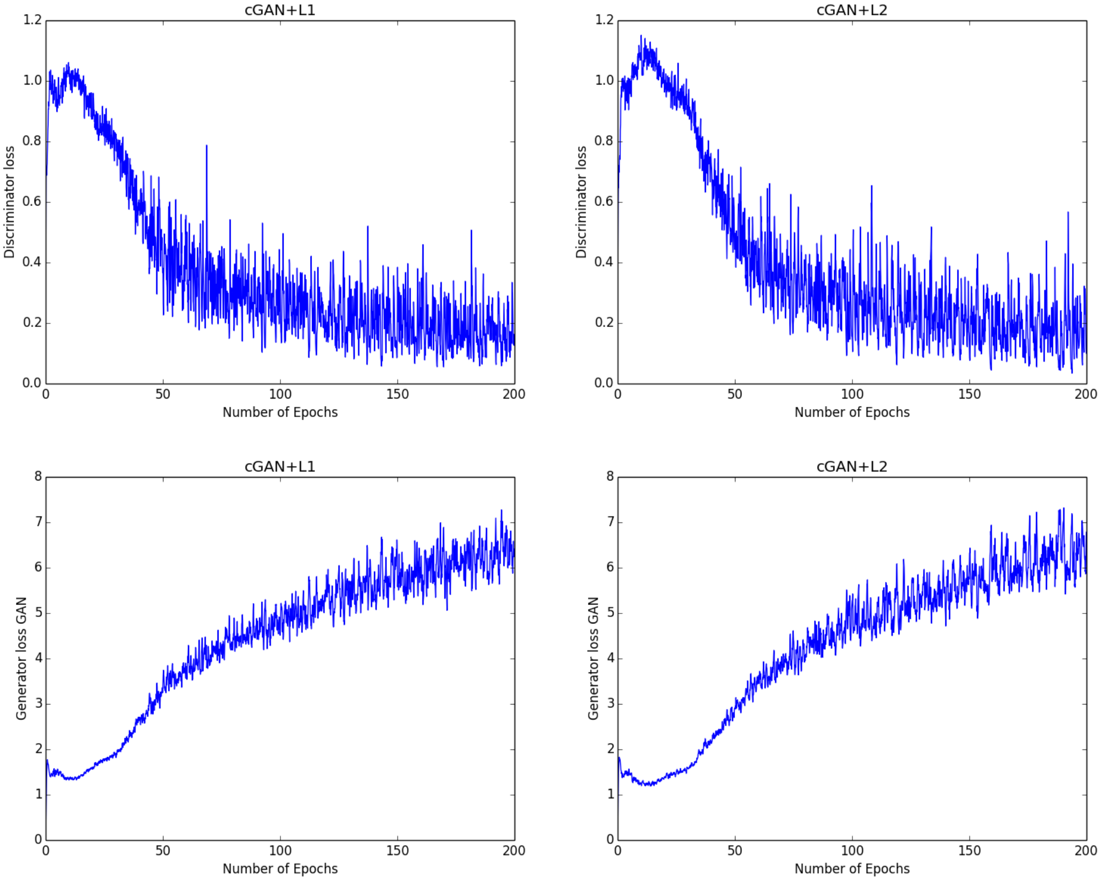


Figure 3.6: Discriminator and Generator Loss

such as instability and mode collapse of traditional GAN.

Our objective function is (2.8) with L1 Loss on generator. We implemented the WGAN with the weight clipping method, and tried to tune the range of clipping parameter as well as the learning rate of RMSPropOptimizer. We also tried to train the discriminator more times than the generator(5:1, 3:1, 2:1) as suggested by [6], but it doesn't seem to be a key factor here.

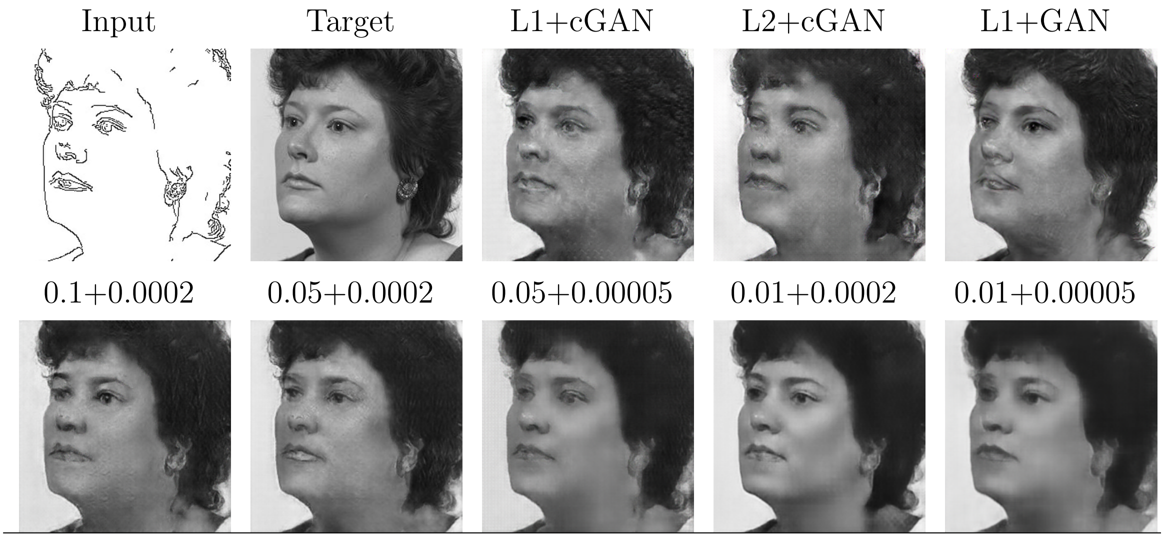


Table 3.3: The training results for WGAN. The second row: the first value is clipping parameter, and the second value is learning rate.

Table(3.3) shows the results for WGAN. The default clipping parameter and learning rate recommended by [6] is 0.01 and 0.00005, but the corresponding results are not the best.

If we fix the clipping parameter at 0.01 and increase the learning rate to be 0.0002, the model generates extraordinarily beautiful portraits: skin unblemished and raven haired, just like someone wears perfect make-up. On the other hand, however, if you look at the texture of hair and compare it with the L1+cGAN version, you may assert that the WGAN loss has blurred the images more or less. But unlike the "L2 loss blurring" with mixed colors and unclear boundaries, it captures the whole outline accurately and just blurs some textures. The results are amazing, even though in the sense of "image to image translation", it's not as promising as L1+cGAN loss as they abandon some details.

Hence we proceeded to increase the clipping parameter. For the 0.05+0.0002 combination, the sharpness becomes acceptable now. When it comes to 0.1+0.0002, the pictures become sharper but lose some accuracy.

3.3 Evaluation and Comparison

Table(3.4) is the quantitative analysis of our results.

	Structural Similarity	Average Gradient
Target	1.000	6.39
cGAN + L1,3000 imgs	0.594	5.86
cGAN + L1	0.565	6.57
cGAN + L2	0.541	6.48
GAN + L1	0.579	5.56
WGAN, 0.01+0.00005	0.660	3.28
WGAN, 0.01+0.0002	0.682	3.60
WGAN, 0.05+0.00005	0.639	4.08
WGAN, 0.05+0.0002	0.623	5.08
WGAN, 0.1+0.0002	0.584	5.83

Table 3.4: Structural similarity and average gradient magnitude for different models.

Structural Similarity

To begin with, we need to make sure that the structural similarity should be a reliable quantitative method to measure the quality of our results.

By checking the paired output-target images and their corresponding SSIM scores(Table(3.5) and Table(3.6)) one by one, we found when at the same sharpness level, SSIM can reflect the goodness for pictures to some extent, but is not very accurate. For example, according to our perception(visual evaluation), the 0.7 pair in Table(3.5) is not as good as the 0.65 pair in Table(3.5), and two pictures of the 0.77 pair in Table(3.6) look quite different but have the highest SSIM score.

Generalization of limitations of structural similarity:

- Generally, lower sharpness, which is not desired by our intention, will contribute to high

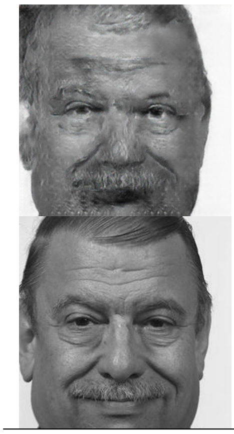
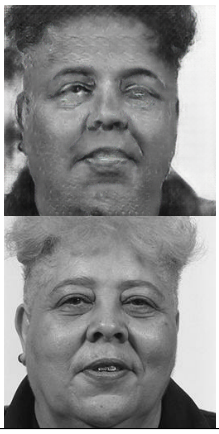
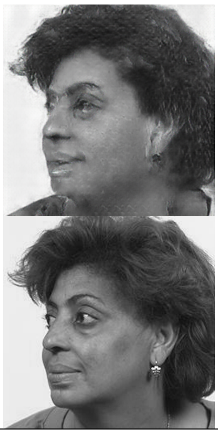
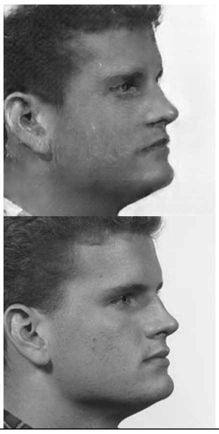
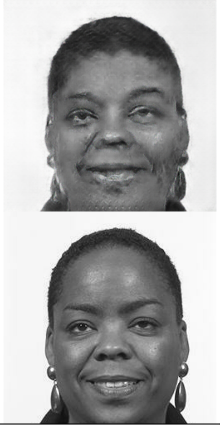
∼ 0.45	∼ 0.50	∼ 0.55	∼ 0.65	∼ 0.70
				

Table 3.5: The SSIM score and corresponding paired images in L1+cGAN group. The gradient magnitude for this group is 0.657.

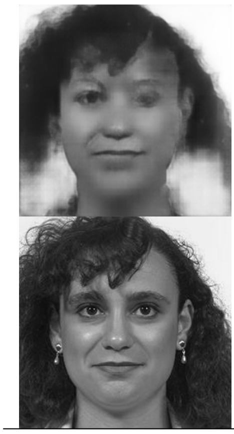
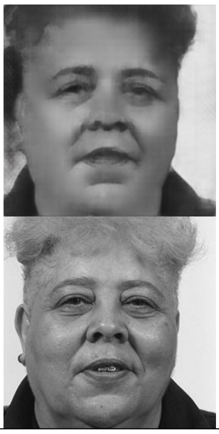
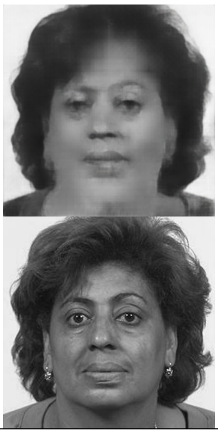
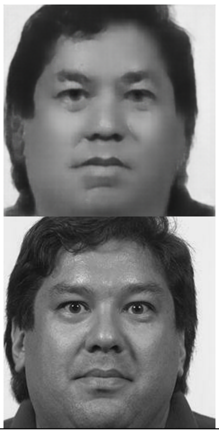
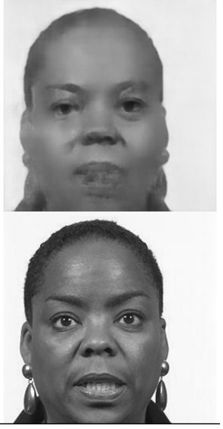
∼ 0.53	∼ 0.60	∼ 0.65	∼ 0.70	∼ 0.77
				

Table 3.6: The SSIM score and corresponding paired images in WGAN,0.01+0.00005 group. The gradient magnitude for this group is 0.32.

SSIM score (compare the average score of Table(3.5) and Table(3.6)). The forth picture in Table(3.5) and the third picture in Table(3.6) are of the same SSIM score. However, it's obvious that the one in Table(3.6) is of much better quality.

- Some inconsistencies are underestimated. For example, L1+GAN scores higher than L1+cGAN. However, the conspicuous stains generated by L1+GAN, which may not exert too much negative influence on SSIM, are not bearable at all according to our cognition.

- 'Similarity' is not what we really want. Our goal is not "resembling the original one as much as possible". Instead, a generated picture is good as long as it looks realistic and is in accordance with the sketch.
- From the perspective of testing individuals, the simpler the original image is, the more likely the sketch-photo pair will score high. But this will not become an issue if we use the same testing set for all models.

In all, it's save to say that, the SSIM score can't reveal which of the two pairs is better if the gap between them is not large(smaller 0.1). But much higher SSIM score strongly indicates better result quality. On the other hand, the average SSIM score of the whole testing set seems to better reflect the result quality than the SSIM for individual pairs.

It's extremely hard to evaluate generative models by a simple method. We even need to train a neural network to tell whether a image is a portrait, not to mention the more complicated sketch to photo translation judgment. Therefore, SSIM appears to be a relatively effective way that can indeed provide us some useful information.

Average Gradient Magnitude

The examples in the section above illustrates that SSIM alone can't be a standard, we should attach much importance to the sharpness of images, too. Images with different gradient magnitudes are in Table(3.9). By checking the average gradient magnitudes of result image sets, if average gradient magnitude for the targets is around 6.3, we have the following conclusions:

- When average gradient magnitude is below 3.5, the model can never be good.
- Average gradient magnitude with the value of 3.6 seems to be acceptable(WGAN, 0.01+0.002)
- Average value of 5.0 can satisfy most requirements.

Meanwhile, higher sharpness is not equivalent to better quality. One output image tends to be a generalization of all faces in the training set. Therefore, by intuition, the ideal average gradient magnitude should be a little bit smaller than the original images(which is 6.39). The L1+cGAN with 3200 images has recognizable improvement over the L1+cGAN 511 images group, with less sharpness but higher SSIM.

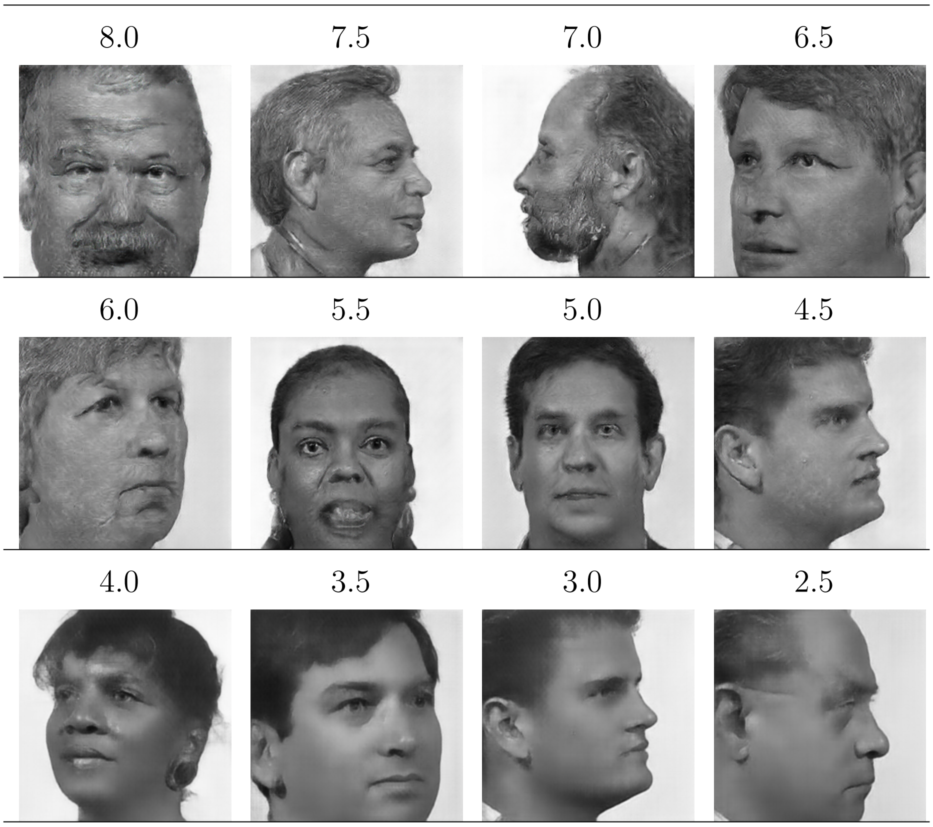


Table 3.7: Pictures with different gradient magnitude

Note that high frequency might even be a signal of poor picture quality. Bad models can generate meaningless colors with very high variation.

Model Comparisons

Obviously, L2+cGAN, L1+GAN, WGAN(0.01+0.00005, 0.1+0.0002, 0.5+0.00005) are not great choices. They neither stand out in SSIM score nor in average gradient. More specifically:

- L2 loss leads to blurring pictures in comparison with L1 loss. Meanwhile, it doesn't produce results with less defects than L1 as it also has lower SSIM score.
- In Figure(3.4), the results from GAN+L1 are really horrible. It can never be the optimal combination, even though its results are of high SSIM score and high average gradient magnitude.

- WGAN with learning rate 0.00005 is inferior to 0.0002 if we fix the clipping value at 0.01 or 0.05.

Then the problem becomes **a tradeoff between perfectness and sharpness**. Any of the three models –WGAN(0.01+0.0002), WGAN(0.05+0.0002) and L1+cGAN can be adoptable in certain situations. More examples of our results are in Table(3.8).

- WGAN with clipping value 0.01 and learning rate 0.0002 can generate flawless pictures. Even though we collected blurring images by this means, unlike the L2 blurring, it seems that WGAN is just insensitive to the dense lines, and can produce results with clear edges. Despite that they do not look so realistic, it is desirable in many situations. Especially when the training set is small, or high frequency and high quality results can never be achieved by other methods.
- L1+cGAN is of highest frequency. Though we have demonstrated in previous sections that we shouldn't blindly pursue high frequency results, this model has shown much of its potential. By tuning its parameters or enlarging the training set, the L1+cGAN model might present amazing results.
- WGAN with clipping parameter 0.05 and learning rate 0.0002 seems to be a good compromise between sharpness and perfectness. It has much better SSIM score than L1+cGAN, as well as acceptable average gradient magnitude.

Color Examples

We also trained color images using the three models with the same parameter value as described above.

From Table(3.9), we have no reason to choose the L1 + cGAN model this time. Generally, the WGAN(0.05+0.0002) should be the best loss function considering both sharpness and perfectness. WGAN(0.01+0.0002) generated very beautiful figures as before, it should be adoptable when high sharpness is not required.

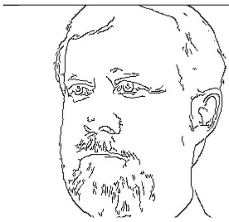
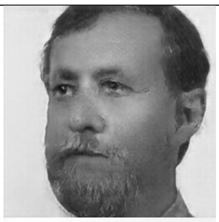
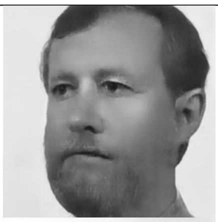
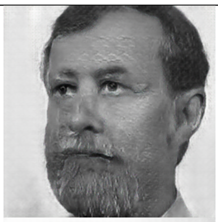
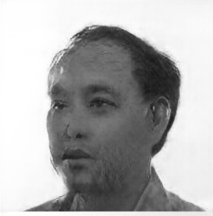
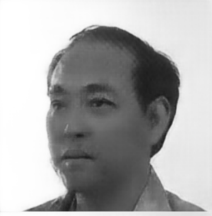
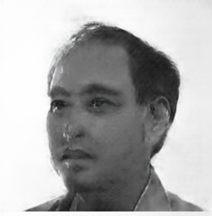
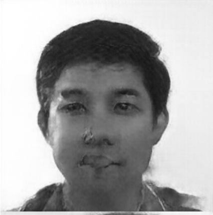
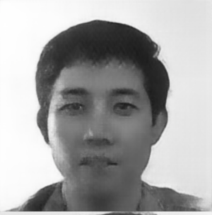
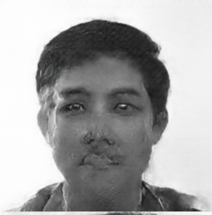
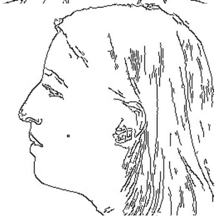
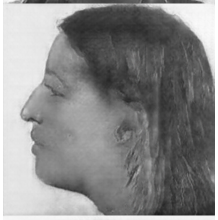
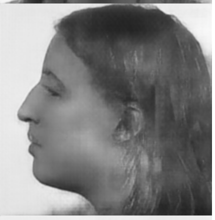
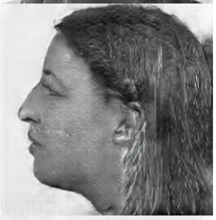
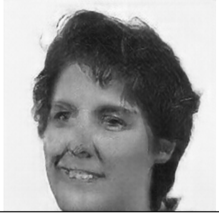
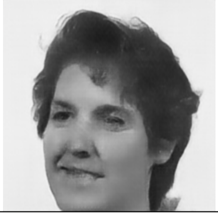
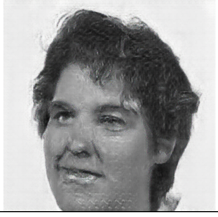
Input	0.05+0.0002	0.01+0.0002	cGAN+L1	Target
				
				
				
				
				

Table 3.8: Comparisons among our three selected models

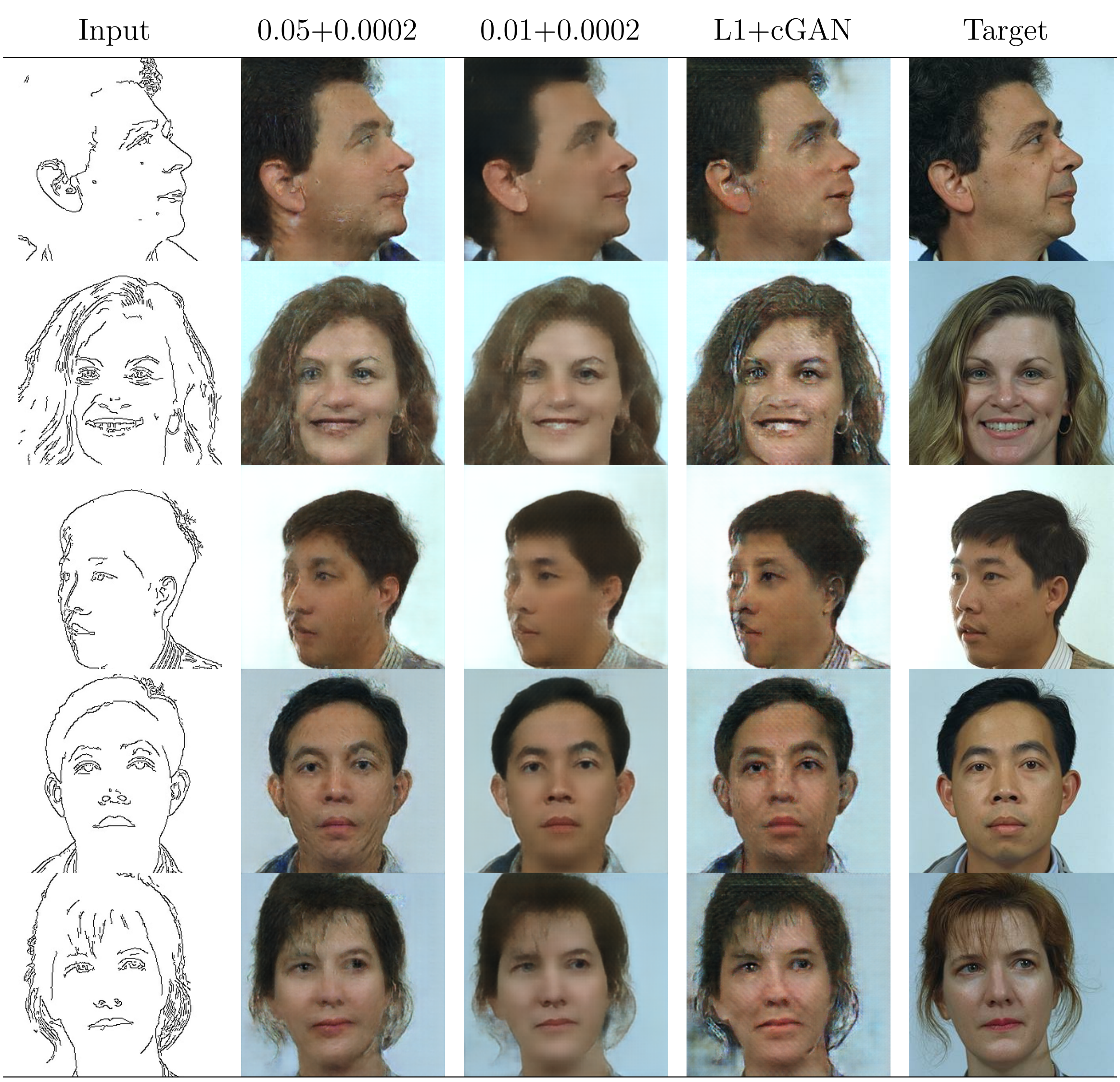


Table 3.9: Color examples for our three selected models

CHAPTER 4

Conclusion and Future Work

Conclusion

Our experiments demonstrate that the pix2pix model is a promising approach to produce photo-realistic pictures from facial sketches.

We explored several combinations of loss functions. It came as no surprise that the L1+cGAN loss proposed by the original paper works well for the sketch to photo translation.

In addition, we found that the L1+WGAN loss leads to the satisfying results, too. With the clipping parameter 0.01 and learning rate 0.0002, the WGAN results are amazing and extraordinarily beautiful, though the sharpness may not be as good as expected. Moreover, by tuning the clipping parameter and learning rate of WGAN, you can even have the choice of obtaining more perfect and beautiful photos(low clipping parameter) or more realistic photos(high clipping parameter).

Limitations

Some limitations of our experiments:

1. We don't have a satisfying metric to quantify the goodness of our results. SSIM and gradient magnitude can provide us some useful information, but are inappropriate to be decisive factors. Our experiments are mainly based on visual evaluation instead of quantitative methods. While it's hard to evaluate generative models, there are some methods should be desirable such as [17].

2. There is strong evidence that larger dataset(3000+) will improve the performance of the model. Yet most of our experiments are on a small dataset(of size 500+), which means that we may not have exhausted the potential of each model.

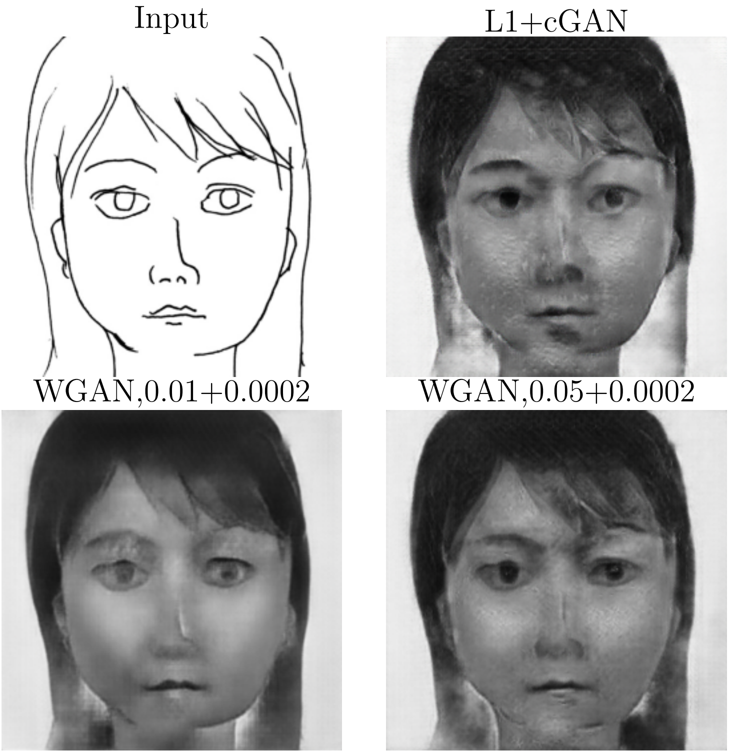


Figure 4.1: Results based on our own sketch. The third and fourth pictures were produced by L1+WGAN, the first value is clipping parameter and the second is learning rate.

3. We didn't try many combinations of parameters of WGAN. Moreover, the weight clipping method has been proved to be an approach with unexpected issues to optimize WGAN. Methods such as gradient penalty might be more preferable.

4. Our training sketches are automatically generated by Canny edge detector. Since Canny edge detector doesn't have any knowledge of face recognition, it sometimes preserves meaningless redundant lines or does not capture the key elements of human faces such as eyes. This may lead to the instability of our model.

Further Application

Figure(4.1) exhibits the pictures generated by the models according to our own sketch.

We didn't try to imitate the drawing style of Canny edge detector, which is inferior to ours. So our sketch is quite different from the sketches in the training set: ours is of fewer lines but more accurate.

Considering the huge difference between the training sketches and our own sketch, the WGAN results are quite good, while the L1+cGAN combination looks terrible.

We believe that, if we have the training set of real sketches and corresponding photos with reasonable large size, the model will be able to produce pictures of better quality. The application of this model will be of great fun because everyone can get any photo-realistic portraits based on their own drafts.

[1] https://ml4a.github.io/guides/Pix2Pix/. Pix2Pix; Accessed: 2018-6-6.

[2] https://towardsdatascience.com/face2face-a-pix2pix-demo-that-mimics-the-facial-expression-of-the-german-chancellor-b6771d65bf66. A Pix2Pix demo that mimics the facial expression of the German chancellor; Accessed: 2018-6-6.

[3] https://affinelayer.com/pixsrv/. Image-to-Image Demo; Accessed: 2018-6-6.

[4] https://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html. Generative Adversarial Networks – Hot Topic in Machine Learning; Accessed: 2018-6-9.

[5] https://github.com/duxingren14/DualGAN/issues/4. DualGAN–Do you use WGAN?; Accessed: 2018-6-2.

[6] M. Arjovsky, S. Chintala, and L. Bottou. "Wasserstein GAN". In: *ArXiv e-prints* (Jan. 2017). arXiv: 1701.07875 [stat.ML].

[7] J. Canny. "A Computational Approach to Edge Detection". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-8.6 (Nov. 1986), pp. 679–698. ISSN: 0162-8828. DOI: 10.1109/TPAMI.1986.4767851.

[8] *color-feret-database*. https://www.nist.gov/itl/iad/image-group/color-feret-database. Accessed: 2018-5-20.

[9] *Generative adversarial network*. https://en.wikipedia.org/wiki/Generative_adversarial_network. Accessed: 2018-6-6.

[10] I. J. Goodfellow et al. "Generative Adversarial Networks". In: *ArXiv e-prints* (June 2014). arXiv: 1406.2661 [stat.ML].

[11] Yuan Chen Guanyang Wang Ying Chen. *Chinese Painting Generation Using Generative Adversarial Networks*. http://cs231n.stanford.edu/reports/2017/pdfs/311.pdf. Accessed: 2018-6-6.

[12] I. Gulrajani et al. “Improved Training of Wasserstein GANs”. In: *ArXiv e-prints* (Mar. 2017). arXiv: 1704.00028 [cs.LG].

[13] *Image-to-Image Translation in Tensorflow.* https://affinelayer.com/pix2pix/. Accessed: 2018-6-6.

[14] P. Isola et al. “Image-to-Image Translation with Conditional Adversarial Networks”. In: *ArXiv e-prints* (Nov. 2016). arXiv: 1611.07004 [cs.CV].

[15] M. Mirza and S. Osindero. “Conditional Generative Adversarial Nets”. In: *ArXiv e-prints* (Nov. 2014). arXiv: 1411.1784 [cs.LG].

[16] D. Pathak et al. “Context Encoders: Feature Learning by Inpainting”. In: *ArXiv e-prints* (Apr. 2016). arXiv: 1604.07379 [cs.CV].

[17] Tim Salimans et al. “Improved Techniques for Training GANs”. In: *Advances in Neural Information Processing Systems 29*. Ed. by D. D. Lee et al. Curran Associates, Inc., 2016, pp. 2234–2242. URL: http://papers.nips.cc/paper/6125-improved-techniques-for-training-gans.pdf.

[18] Zhou Wang et al. “Image Quality Assessment: From Error Visibility to Structural Similarity”. In: *Trans. Img. Proc.* 13.4 (Apr. 2004), pp. 600–612. ISSN: 1057-7149. DOI: 10.1109/TIP.2003.819861. URL: http://dx.doi.org/10.1109/TIP.2003.819861.

[19] Z. Yi et al. “DualGAN: Unsupervised Dual Learning for Image-to-Image Translation”. In: *ArXiv e-prints* (Apr. 2017). arXiv: 1704.02510 [cs.CV].