Transform Human Face to Simpsons Face by the Modification of CycleGAN

Yu-Hsiang Liu
Oregon State University
liuyuhs@oregonstate.edu

Abstract

CycleGAN is one of the most popular methods in image to image translation. However, this method often struggles with large geometric shifts and shape deformation. The translator can not find the correct corresponding information between two domains and will produce an unexpected result at the end. Human face to Simpson's face is an example of a challenge for it. In this paper, we construct SimpsonsGAN based on CycleGAN which provides a method to address the face collapse problem in facial translation. We add two additional local descriptors into the model. After the image has been translated to the other domain, the images will be cropped to several parts containing face components such as eyes, nose, mouth in each patch. The local discriminators are trained to distinguish the real and fake from the facial components. We evaluate this modification performance by using FID score and compare the final result with CycleGAN to prove the improvement. Furthermore, we also illustrate the effect using different ratios of global and local parameters.

1. Introduction

Image to Image translation is the task of taking an image from one domain and transforming its style to another domain. It is an interesting topic and has been widely used in many real-world applications. In the previous approach, it required collecting tons of paired data before building the model, but finding paired data is an expensive or impossible mission in most cases. Therefore, CycleGAN (Generative Adversarial Networks) [2] became one of the most popular methods in image to image translation. Despite unsupervised image to image translation, the property of cycle consistency makes it able to transfer the texture between image domains and the output images preserve the important information in input images. CycleGAN has achieved great performance in many tasks, such as apple-to-orange and horse-to-zebra conversion. However, this method often struggles with large geometric shifts and shape deformation. The translator can not find the correct corresponding information between two domains and will turn out an unexpected result at the end.

Human face to Simpson's face is an example of a challenge for CycleGAN. Although the Simpson's face has the same amount and corresponding face components compared to humans, the size and shape of eyes, noses and mouth are totally different. In addition, different facial components in the human face and Simpson's face might be more similar to each other. For example, instead of human eyes, the shape of a human's cheeks and forehead is similar to Simpson's eyes. Hence, It is easy to generate an output image with more than two eyes, which could be considered as "face collapse".

As a result, the goal in this paper is to construct a translator by modifying the CycleGAN which is called SimpsonsGAN, to make it successfully transform the human face (input) to the Simpsons-like face (output). In order to avoid the problem of face collapse, we implemented the modification by adding additional local discriminators. After the image has been translated to the other domain, the images will be cropped to several parts containing face components in each patch. The global discriminator will determine the whole image as usual. In contrast, the local discriminators are trained to distinguish the real and fake from the facial components. This would tell the models to focus on the important regions and capture the joint distribution with both global and local. Since some components have big differences between humans and Simpsons, we set a lower weight value in local cycle consistency to encourage the shape deformation. We evaluate this modification performance by using FID (Fréchet inception distance) score [3] and compare the final result with CycleGAN. Moreover, we also illustrate the effect using different ratios (weight) of global and local parameters.

Must to mention that solving this specific problem not only addresses the facial translation difficulty in CycleGAN, but also provides a method to deal with other cases that are not suitable with the original Cycle-

GAN. The source code in this project can be found in https://github.com/tony85212/SimpsonGan

2. Related Works

2.1. Generative Adversarial Networks

Generative adversarial networks (GANs) are an approach to build up creative models using deep learning methods. It has shown decent results in image generation and translation. GANs train the model as a supervised learning problem by conducting two sub-model, generator and discriminator. The discriminator is a binary classifier, which distinguishes between real and generated samples. On the other hand, the generator attempts to fool the discriminator with realistic but fake samples.

The key to success of GANs is that two sub-models are trained together during the adversarial learning procedure. The discriminator would gain a stronger ability to determine whether the image is real or fake after numerous iterations. Therefore, if the generator is able to fool the discriminator, meaning that the model can generate plausible samples.

2.2. CycleGAN

The CycleGAN is an extension of the GAN architecture, which includes two generators and two discriminators (one paired). Generator $X \rightarrow Y$ takes images from the X domain as input and outputs images for the Y domain, and the other Generator $Y \rightarrow X$ takes images from the Y domain as input and generates images for the X domain. Two discriminator models are used to determine the real or fake in domain X, Y respectively.

Cycle Consistency. Cycle consistency is a special property in the CycleGAN framework. The concept is from machine translation where a phrase translated from English to Chinese should translate from Chinese back to English and be as similar as the original phrase. In the image perspective, an image after passing through both generators $X \rightarrow Y$ and $Y \rightarrow X$, the output should match to the initial image. Adding cycle consistency into the total loss will encourage the model to preserve the important information in input images.

3. Method

3.1. Previous Work

In the article "Simpsonize Yourself using CycleGAN" [1], the author has tried to build a CycleGAN translator to transform a human face to Simpson's face. Nonetheless, an unexpected result came out at the end. From Figure 1, the shape of the face and clothes is fine, but we can see that face components are not in the right place (face collapse). The output images have more than two eyes, even with the correct amount of eyes, they are not in the same horizon

Figure 1. The example from previous work, using CycleGAN to transform a human face to Simpson's face. It turns out to be an unexpected result.

Figure 2. Real image from human domain. Generated image is produced by Generator $X \rightarrow Y$ and takes the real image as input. Reconstructed image is produced by Generator $Y \rightarrow X$ and takes the generated image as input. The specific regions become brighter after reconstruction.

compared to the original image. This presents that CycleGAN is unable to find the right relationship while it comes to large geometric shifts.

3.2. Observation

Instead of human eyes, if the cheeks, forehead or nose on the human face are bright, the model would often consider it as eyes. By observing the reconstructed image in Figure 2, we can easily see that some bright regions are more obvious than the real image. It is because no matter the shape or size, for the generator model, they are more likely to be Simpson's eyes. Another reason is that the face components in the Simpson domain are extremely dense. The generator believes that eyes, nose and mouth are supposed to be in the small range of regions. Therefore, the model would completely learns the wrong direction in the training episodes.

3.3. Approach

To address the problem of face collapse, the efficient way is to tell the model the corresponding facial components in both human and Simpson's domain. Since our goal is to match the face components from humans and Simpsons, we could crop the face images into several parts, including eyes, nose, mouth. Next, adding local discriminators to determine each region that we extract from the image. The structure and mission of the local discriminator is the same as the global one. The fake images and the reconstructed images will be tested on both global and local discriminators. For example, the local mouth discriminator is trained to distinguish the real and fake mouth in the Y domain. After generator $X \rightarrow Y$ generates a fake sample, it will be an input of global discriminator first, then the cropped sample

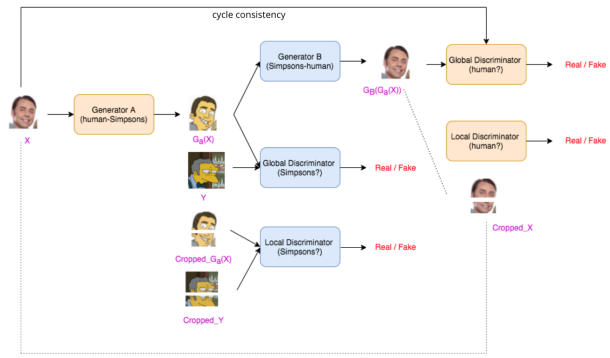

Figure 3. Illustration of method. With input in the human domain image, generator Human-Simpsons generate a fake image from Simpson domain. With input in a fake image, generator Simpsons-Human generates a fake image back to the human domain. The whole fake and reconstructed image will go through a global discriminator and the cropped images will be tested on local discriminators. (Reserve the architecture if input is in the Simpsons domain)

will be an input of local mouth discriminator.

This would force the generator model to generate a corresponding facial component to satisfy the local discriminator. Therefore, the generator would try to generate a plausible image that can fool the global and local discriminators at the same time. In addition, same as the original CycleGAN, we have to add the local cycle consistency to preserve the local feature in the input image. Figure 3 demonstrates the flow chart from human domain to Simpsons domain.

3.4. Fomulation

In the details of modification structure, we first define two types of mapping: $G : X \rightarrow Y$ and $F : Y \rightarrow X$, where X is human domain and Y is Simpsons domain. The global discriminator D_X distinguishes between *x* and *F(y)*, and D_Y distinguishes between *y* and *G(x)*. The local discriminators D_{X_l} distinguishes between x_l and $F(y_l)$. The local discriminators D_{Y_l} distinguishes between y_l and $G(x_l)$.

Adversarial Loss. The objective of the global and local facial adversarial learning of G, F can be expressed as

$$\begin{aligned} L_{GAN}(G, D_Y, D_{Y_l}, X, Y) = & \mathop{\mathbb{E}}_{y \sim p_y(y)} [\log D_Y(y)] \\ & + \mathop{\mathbb{E}}_{x \sim p_x(x)} [1 - \log D_Y(G(x))] \\ & + \mathop{\mathbb{E}}_{y \sim p_y(y)} [\log D_{Y_l}(y_l)] \cdot \beta \\ & + \mathop{\mathbb{E}}_{x \sim p_x(x)} [1 - \log D_{Y_l}(G(x_l))] \cdot \beta \end{aligned}$$

$$\begin{aligned} L_{GAN}(F, D_x, D_{X_l}, Y, X) = & \mathop{\mathbb{E}}_{x \sim p_x(x)} [\log D_X(x)] \\ & + \mathop{\mathbb{E}}_{y \sim p_y(y)} [1 - \log D_X(F(y))] \\ & + \mathop{\mathbb{E}}_{x \sim p_x(x)} [\log D_{X_l})(x_l)] \cdot \beta \\ & + \mathop{\mathbb{E}}_{y \sim p_y(y)} [1 - \log D_{X_l}(F(y_l))] \cdot \beta \end{aligned}$$

Note that β is a weight parameter, which stands for the ratio between global and local images. With higher value, the generated image will focus more on local features. With values equal to zero, the generator model is the same as the original CycleGAN.

Cycle Consistency Loss. As mentioned above, the purpose of cycle consistency loss is to preserve the feature from input image. With larger weight in cycle consistency, the generated image would be more similar to the original image. If the weight of the cycle consistency is zero, the model will discard all information in the input and generate images from the other domain. The objective of the cycle consistency loss of global and local can be expressed as

$$\begin{aligned} L_{cyc}(G, F) = & \mathop{\mathbb{E}}_{x \sim p_x(x)} [||F(G(x) - x)||_1] \\ & + \mathop{\mathbb{E}}_{y \sim p_y(y)} [||G(F(y) - y)||_1] \end{aligned}$$

$$\begin{aligned} L_{cyc_{local}}(G, F) = & \mathop{\mathbb{E}}_{x \sim p_x(x)} [||F(G(x_l) - x_l)||_1] \\ & + \mathop{\mathbb{E}}_{y \sim p_y(y)} [||G(F(y_l) - y_l)||_1] \end{aligned}$$

Combining the adversarial losses and the cycle consistency loss, we obtain the full objective function, which can be expressed as

$$\begin{aligned} L(G, F, D_X, D_Y, D_{X_l}, D_{Y_l}) = & L_{GAN}(G, D_Y, D_{Y_l}, X, Y) \\ & + L_{GAN}(F, D_x, D_{X_l}, Y, X) \\ & + L_{cyc}(G, F) \cdot \alpha \\ & + L_{cyc_{local}}(G, F) \cdot \gamma \end{aligned}$$

Note that α and γ are the weight of cycle consistency of local and global and local respectively. Since the shape deformation is large in local features, we would set a lower value in than to give the generator more flexibility on local transforming.

In the training process, the parameters in G, F, D_X, D_Y, D_{X_l} and D_{X_l} are estimated by optimizing the full objective function.

4. Implementation

4.1. Data Collection and preprocessing

In the training phrase, we need two face dataset from Human and Simpsons domain. For real faces, we use the

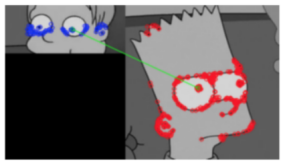

Figure 4. The cropping and padding action for local adversarial learning. The input for local discriminators are extracted from the input by feature matching and padded as the size of the input. Two specific local parts including eyes and mouth

Flickr-Faces-HQ Dataset https://github.com/NVlabs/ffhq-dataset which is a high-quality image dataset (resolution 1024x1024 pixels) of human faces. It contains considerable variation in terms of age, ethnicity and image background. For Simpsons, we found the ONLY open source on https://www.kaggle.com/kostastokis/simpsons-faces, which contains Simpsons faces extracted from seasons 25-28 (resolution 200x200 pixels). In the implementation, we use 2000 images of each dataset for training and 500 images of each for testing.

In the experiment, we resize the images size to 256x256 pixels for training convenience. For the requirement of local discriminators, we use eyes and mouth as our local feature. The size of the cropped image is 64x256 pixels which is one-fourth of the original image. After extracted from the input, it will be padded as the size of the input.

In this case, we must know the specific position of face components (eyes, mouth) so that we could crop the image as input. Since the face in Flickr-Faces-HQ Dataset is in the center of the image, it is simple to find a constant position for local regions that contain eyes or mouth. However, the face is not in the center in the Simpsons dataset. To cope with this sub-problem, we use a keypoint detector in OpenCV to find out the eyes's position in the image. The eyes of Simpsons are strongly closed to each other in different images, so only a few images cropped the wrong region. Since the face component is dense, we find Simpson's mouth by adding 64 pixels.

4.2. Model

Our generator architecture is based on the ResNet, using a residual block makes it more effective to learn the image translation function. The generator model has three convolution blocks and three deconvolution blocks with six residual blocks in the middle (1,965,059 trainable parameters). The amount of filters references the original CycleGAN implementation.

Both global and local discriminators use the same architecture with five convolution layers (2,764,737 trainable parameters). The discriminators take 256x256 pixels (64x256 pixels and 192x256 padding) as input and then return the

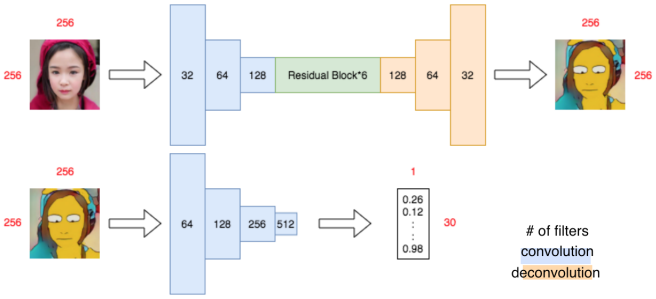

Figure 5. Illustration of generator and discriminator model, including the input and output value. The generator is based on ResNet-6 and the discriminator is a binary classifier.

Method	FID scores
CycleGAN	151.28
Our Approach	148.05

Table 1. Show the FID scores of generated images and real Simpsons images.

tensor size with 30x30. In detail, the discriminators check whether 70x70 pixels area (overlap) seems real or fake.

To sum up, there are totally 2 generators, including Human-Simpsons, Simpsons-Human. And 6 discriminators, including global-Human, global-Simpson, eyes-Human, eyes-Simpsons, mouth- Human, and mouth-Simpsons.

5. Result

In Figure 6 and Figure 7, it is the result with (α, β, γ) = (10, 0.5 ,5) after 100 training epochs, where α and γ represent the global and local cycle consistency weight, β represents the weight of local regions. We can see that the eye and mouth are now in the right place. In addition, the eyes are extremely close to Simpson's domain. Although the output is not perfect, it is reasonable and acceptable.

5.1. Evaluation

The FID scores can measure the distance between real and fake datasets. The lower scores means the fake datasets are close to real datasets, which is positive. We use both CycleGAN and our implementation to generate 500 plausible images from a human testing dataset. Then, calculating the FID scores with the Simpson testing dataset.

Since the score does not consider the similarity between original image and output, in Figure 8, we also demonstrate the comparison of the same input image and we can easily see a big improvement between CycleGAN and our modified version.

Figure 6. On the left hand side is the real face in the human domain. On the right side are fake samples created by SimpsonsGAN.

Figure 7. On the left hand side is the real face in Simpsons domain, On the right side are fake samples created by SimpsonsGAN.

5.2. Discussion

In this section, we are going to discuss the effect of local discriminator weight and local cycle consistency weight. We test with the different values of β and γ, and fixed α (10). In the Figure 9.A, smaller β get a similar result with original CycleGAN, where the model misunderstands the face component in different domains. With larger β, the model starts to ignore the global condition. We can see that the girl's face on the left side and the outline of the man's face disappear. In Figure 9.B, the smaller γ represent model is more flexible, since the restriction between original image and reconstructed image is lower. With larger γ, the output will be more likely to input, but might not be the result we expected.

6. Conclusion

In this paper, we have constructed SimpsonsGAN based on the modification of Cyclegan and provided a method to translate human faces to Simpson faces. Since CycleGAN can not handle the datasets with large geometric shifts. Cropping images to several parts and adding additional local discriminate can improve the performance of CycleGAN. The most difficult part in this paper is finding a balance between global and local weight. Observing the result from output is the only way to tune the hyperparameter.

For future works, we can use a pre-trained model of facial component alignments to crop the local feature more accurately. Since Simpson's face components are dense, with this pre-trained model, we can tell more precisely and small range regions to model.

Last but not least, the method applies to different tasks that are not suitable for CycleGAN.

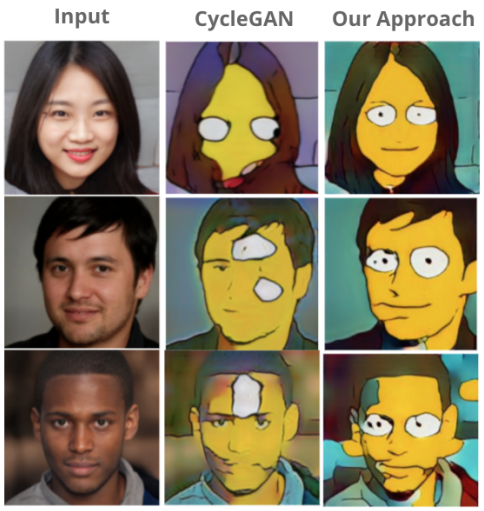

Figure 8. Demonstrate the difference between CycleGAN and our approach using the same inputs.

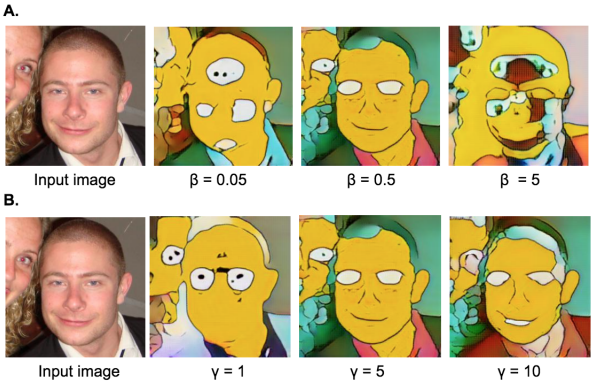

Figure 9. Demonstrate the difference between CycleGAN and our approach using the same inputs.

References

[1] N. Iyer. Simpsonize yourself using cyclegan and pytorch. *https://towardsdatascience.com/simpsonize-yourself-*

using-cyclegan-and-pytorch-cea94ee199ca, 2020.

[2] P. I. Jun-Yan Zhu*, Taesung Park* and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial network. *in IEEE International Conference on Computer Vision (ICCV)*, 2017.

[3] M. Seitzer. pytorch-fid: FID Score for PyTorch. `https://github.com/mseitzer/pytorch-fid`, August 2020. Version 0.1.1.