COMP4801 Interim Report
Facial Expression Synthesis using Generative Adversarial Network

Wang Hanyan (3035330088)
Supervisor: Dr. KP Chan

January 13, 2020

Abstract

Facial expression synthesis is the process of rendering face images with target expressions while preserving identity information. The generated images can be used in various areas including affective interaction, data augmentation, artificial agent, etc. Existing methods are usually conducted in a sequence to sequence manner which requires the availability of a video of varying facial expressions of a subject. However, the lack of such data in real life limits the application of these methods. Synthesizing facial expressions on limited faces remains an open problem. In recent years, the generative adversarial networks (GANs) [5] have received substantial attention. GAN consists of two models: (1) Generator G that produces images based on random noises and (2) Discriminator D that distinguishes fake images rendered by G from real sample images. GAN establishes a minimax adversarial game between G and D, making the generated images look more and more real. Different variations of GANs have been applied successfully to age progression/regression [16], image super-resolution [8], image to image translation [7], etc. In this project, we use the idea of GAN to synthesize face images, i.e. given a subject I and a facial expression E, an image with identity I and expression E should be generated. We have proposed a GAN model based on Adaptive Instance Normalization (AdaIN) [13] which enables transfer learning. We have trained it on the public dataset CelebA [9] and observe reasonable results. It can alter face images based on a wide range of expressions in terms of action units, which describe the muscle movements of human faces.

Acknowledgements

I want to thank my supervisor Dr. KP Chan for his guidance in the project. I also want to thank mable for her teaching in report writing.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background

Facial expression synthesis aims to transfering a target expression to a source person's face. Successful synthesis has various applications including film production, augmented reality and affective interaction. Traditional methods extract geometric features like facial landmark points of a face and map them to a new face. However, most of them merely solve the discrete emotion problem. In other words, they can only transfer emotions such as happy, angry, disgusted, etc. They fail to generate faces with more complex and subtle expressions. Facial Action Coding System (FACS) [3] was proposed to describe expressions in terms of action units, which are based on the muscle movements of human faces. The 17 commonly used action units can produce a wide range of expressions. In this project, we use action units of images to generate synthesized faces, without using facial landmark points.

Recent years have seen rapid development in generative adversarial networks (GANs) [5]. CycleGan [18] and Pix2Pix [7] are extensively used in facial synthesis tasks. Albert Pumarola [12] proposed GANimation, a model trained on images annotated with action units in an unsupervised manner. It synthesized anatomically coherent facial expressions and achieved state of the art performance.

1.2 Objectives

For this project, our first objective is to reimplement the GANimation model to produce photorealistic faces and synthesize faces on public datasets. We will then focus on modifying the architecture and loss function to achieve better qualitative results. The further goal is to apply the method to a video for face reenactment, which requires transfering facial expressions of a driving video to a source image and generate a video with the same expressions of the driving one while preserving the identity. This objective is more difficult as it needs to generate realistic images as well as alter the expressions smoothly, in order to ensure the continuity of frames in the video. The computing resources will also be higher.

1.3 Current progress

We have proposed a GAN model based on Adaptive Instance Normalization (AdaIN) [13] which enables transfer learning. We have trained it on the public dataset CelebA [9] for 30 epochs. We tested the model on different pairs of source and target images to alter source expressions towards target ones. The image quality gradually got better after more epochs. We linearly interpolated the intermediate results to show the gradual changes of facial expressions. Realistic images could be generated, which indicates that our approach is correct.

1.4 Outline

This report will first review the literature in Chapter 2, namely facial expression synthesis, Generative Adversarial Network and face reenactment. Among the models presented in the literature, our work is based on a model called GANimation [12]. The report will then introduce the dataset, architecture and loss functions used in Chapter 3. Here we propose a variation of GANimation by adding a domain embedding layer, which enables transfer learning. In Chapter 4, it will cover the implementation details and the current results trained on the dataset. It proves the correctness of our approach. Evaluation of the results and further works will be provided. This report will conclude in Chapter 5 and propose future works.

Chapter 2

Related works

This chapter will review the literature related to the project. In 2.1, traditional and recent methods for facial expression synthesis will be reviewed. Section 2.2 will introduce GAN, which is used in this project. Section 2.3 will review papers about the further goal-face reenactment.

2.1 Facial expression synthesis

Facial expression synthesis is to generate face images with target expressions while preserving identity information. Traditional methods achieved this by extracting facial landmark points [15] from a face and mapping them to a new face. The problem with this approach is that faces generated look unnatural due to the high complexity of human expressions. More recent works adopt the idea of convolutional networks to synthesize facial expressions and have achieved better results. For instance, [2] performed image-to-image translation for multiple domains in the task of facial expression synthesis. But it only handled the case of discrete emotion labels such as smile, angry, disgust, etc. Higher-level descriptions like action units are needed to produce continuous and smooth images.

2.2 Generative Adversarial Network (GAN)

In 2014, Goodfellow [5] introduced the concept of GAN. Since then, GAN has been extensively used in different areas including image generation, super-resolution, text to image, etc. The original GAN consists of two parts: Generator G that produces images based on random noises and Discriminator D that distinguishes real sample images between fake images generated by G. GAN establishes a minimax adversarial game between G and D, making the generated images look more and more real. However, there is little control over the output images because of the random sampling of latent variables during generation. Conditional GAN [7] was proposed to add conditions like labels to control the output of the network. After that, more and more GANs have been used in image tasks. In this project, we use a variation of GAN to synthesize facial expressions.

2.3 Face reenactment

Face reenactment is the process of rendering a video based on a source image and a driving video. The rendered video has the same expressions and muscle movements as the driving one while preserving the identity of the source image. Most of the existing studies are model-based. They capture face movements with RGB or RGB-D camera and fit the movements in a model. Then they render a video frame by frame using the model through morphing [4]. In 2018, GANimation [12] was introduced to generate continuous faces with varying facial expressions based on action units. It used Openface [1] to extract action units from images. This project modifies the GANimation model to train on faces.

Summary

Chapter 2 has reviewed literature related to facial expression synthesis, GAN and face reenactment. It has described a model called GANimation to synthesize faces. The next chapter will introduce the dataset, architecture of GANimation and loss functions used in the project.

Chapter 3

Method

This chapter will introduce the dataset used in our experiment, namely the CelebA dataset. It will also formulate the problem we face and provide the architecture of the model, the modification we made and reasons behind, especially the domain embedding layer. Then loss functions involved will be defined and elaborated.

3.1 Dataset

CelebA [9] is a large scale face dataset containing around 200K images. Faces from different background and lighting conditions are cropped and aligned to $128 * 128 * 3$ size. We previously used VoxCeleb [11], which is a public dataset that contains celebrity interview audio clips from YouTube. However, training results on VoxCeleb was not satisfactory. We speculate that it is due to the different size and image quality of the two datasets. VoxCeleb has only 50K images and some of these images are of low quality. It is also notable that the images in VoxCeleb are not aligned and cropped. Therefore, we prefer to use CelebA. In this project, we sample 180K faces for training and 20K for testing. For each image, Openface [1] is used to extract action units. There are 17 action units in total with each one being a floating number ranging from 0 to 5. These values are normalized between 0 and 1 in the experiment.

3.2 Problem formulation

Let $I_{y_r} \in R^{H \times W \times 3}$ be the input RGB image, the facial expression of I_{y_r} is encoded by a vector of N continuous labels, denoted as y_r. This continuous presentation allows the use of interpolation to generate a wide range of expressions, which extends the scope of the research area.

In this project, we aim to learn a mapping function M to produce a fake image

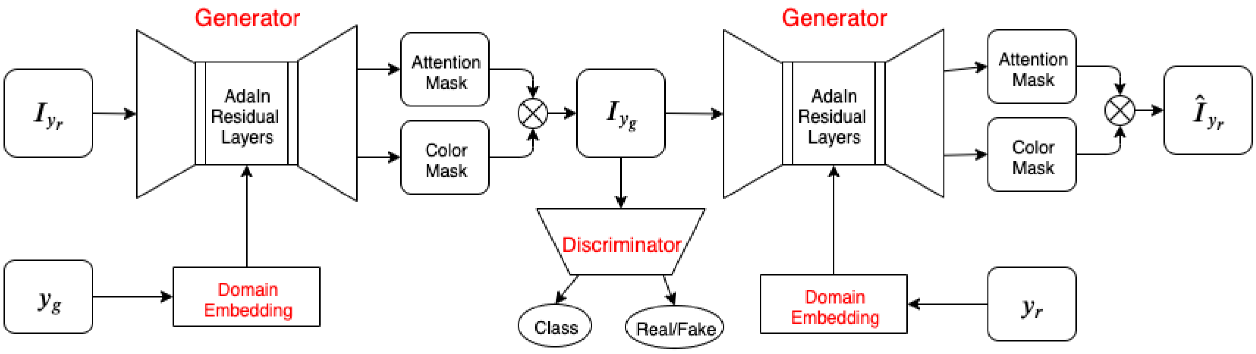

Figure 3.1: Our Model. The model consists of three modules (marked in red): generator G to get attention mask G_A and color mask G_C, domain embedding layer to embed the target labels and discriminator D to evaluate photorealism loss and facial expression loss.

given the input I_{y_r} and target expression y_g. Formally:

$$M(I_{y_r}, y_g) \rightarrow I_{y_g} \tag{3.1}$$

This is conducted in an unsupervised manner, since we have no access to the ground truth I_{y_g}.

3.3 Architecture

The architecture of our model is shown in Fig. 3.1. It is based on GANimation model [12]. We modify the model by adding a domain embedding layer which converts the target label to a fixed size vector and feeding it into Adaptive Instance Normalization (AdaIn) layers in the generator. This step makes the generator agnostic to the size of target label, which is desirable in transfer learning as our target label may not be action units in other tasks. The input is an RGB color image $I_{y_r} \in R^{H \times W \times 3}$. H and W denote the height and width of the image. y_r is the facial expression features of the image denoted by a set of 17 action units. y_g is the target facial expression. The model takes an input image I_{y_r} and a target expression y_g and generates I_{y_g} with the target expression while preserving the identity information. The model consists of three modules: a generator G, a domain embedding layer and a discriminator D.

3.3.1 Generator

Generator G [6] is applied twice in the model. First, it maps an input image I_{y_r} to a target image I_{y_g}. Then, it renders an image $\hat{I}_{y_r}$ which has the same identity and expression as the input. This idea is adopted from [18] to enhance the performance of generator. Given an input $I_{y_r} \in R^{H \times W \times 3}$, it is fed into G. Then, in order to generate an image with only expression altered, an attention mask A and a color

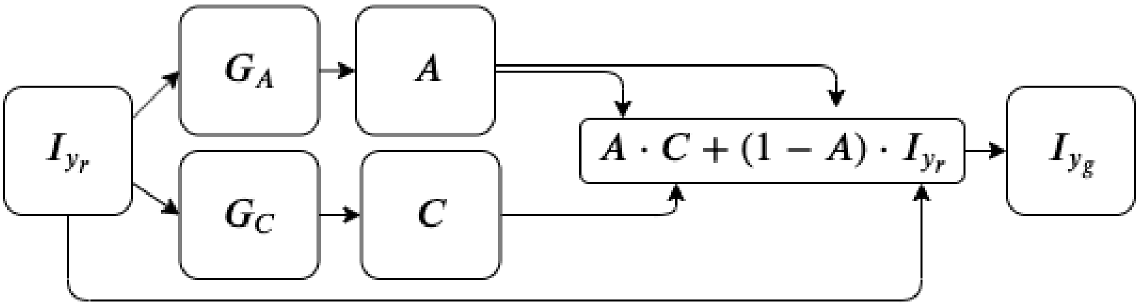

Figure 3.2: Attention-based generator. The generator gets a source image as input and outputs an attention mask A and a color mask C. They are combined with the input to generate the target expression.

mask C are used. A, C, and I_{y_g} are defined as:

$$\begin{aligned} A &= G_A(I_{y_r}|y_g) \in \{0, ..., 1\}^{H \times W} \\ C &= G_C(I_{y_r}|y_g) \in R^{H \times W \times 3} \\ I_{y_g} &= A \cdot C + (1 - A) \cdot I_{y_r} \end{aligned}$$

The mask A measures the amount each pixel in C contributes to I_{y_g}. It captures the expression change while ignoring the identity information. We use $(1 - A) \cdot I_{y_r}$ to get the identity of the original image and combine it with the target expression, thus rendering I_{y_g}. The whole process is shown in Fig. 3.2.

3.3.2 Domain Embedding

The most notable difference between our model and the GANimation model is that we add a domain embedding layer. This is inspired by [13]. In GANimation, they simply used the concatenation of the source image I_{y_r} and the target label y_g as input. However, in facial expression synthesis tasks, target labels may be of different sizes, so the model will be label dependent. In our model, a domain embedding layer, i.e. a fully connected layer is used to map the target label to a fixed size vector, then it is fed into the AdaIn layers of the generator. Using this approach, we can indirectly incorporate the label information into the model without the constraint of label size. It makes the pre-trained model applicable to other tasks, assisting researchers in the real world.

3.3.3 Discriminator

The disciminator D has two outputs (Fig. 3.1): class and real/fake. Class is used to test whether the facial expression of I_{y_g} is the same as target y_g. It ensures that the wanted expression is embedded in the generated image. The second output is to evaluate whether the rendered image is realistic. We adopt the idea from [7] to test on a patch rather than a single image. This makes our output photorealistic.

3.4 Loss function

We use four loss functions from [12] to train the model. They are adversarial loss that simulates the generated image distribution to training image distribution, attention loss that prevents attention mask from saturating, identity loss that preserves the identity information and expression loss that makes the expression of generated image similar to the target.

3.4.1 Adversarial loss

The adversarial loss is useful in rendering realistic images and helping G to learn parameters. Given an input image I_{y_r} following the distribution x of training images, the adversarial loss can be expressed as:

$$L_{adv} = E_x[D_I(G(I_{y_r}|y_g))] - E_x[D_I(I_{y_r})] \tag{3.2}$$

3.4.2 Attention loss

The attention mask A is used to get the facial expression change between input and output while ignoring the identity information. In real experiment, A can easily saturate to 1, which leads to $I_{y_g} = C$. In order to avoid that, we regularize A with l_2 penalty and perform $Total\ variational\ regularization$. Formally:

$$L_{attention} = E_x[\sum_{i,j}^{H,W}[(A_{i+1,j} - A_{i,j})^2 + (A_{i,j+1} - A_{i,j})^2]] - E_x[\|A\|_2] \tag{3.3}$$

$A_{i,j}$ is the i, j th entry of A.

3.4.3 Identity loss

With the losses defined above, we can generate realistic images. However, the identity of the image may not be preserved. Identity loss is used to mantain the identity between the input image and the output image. We compare I_{y_r} and $G(G(I_{y_r}|y_g)|y_r)$ using l_1 loss to ensure they correspond to the same person. More formally:

$$L_{id} = E_x[\|G(G(I_{y_r}|y_g)|y_r) - I_{y_r}\|_1] \tag{3.4}$$

3.4.4 Expression loss

While rendering the output, G should not only preserve the input identity but also satisfy the target expression. This loss consists of two parts: the action unit regression loss of fake images and real images. D_y is used to extract the expression $\hat{y}$ of an image, and then the extracted expression is evaluated against ground truth expression using l_2 loss. It can be defined as:

$$L_{expression} = E_x[\|D_y(G(I_{y_r}|y_g)) - y_g\|_2{}^2] + E_x[\|D_y(I_{y_r}) - y_r\|_2{}^2] \tag{3.5}$$

3.4.5 Full loss

Combining the four partial losses defined above, the full loss can be expressed as:

$$L = L_{adv} + \lambda_1 L_{attention} + \lambda_2 L_{id} + \lambda_3 L_{expression} \tag{3.6}$$

$\lambda_1, \lambda_2, \lambda_3$ are weights of different terms that balance the training process. This loss function aims at rendering a realistic image with desired expression original identity information.

Summary

This chapter has introduced the dataset and architecture used in the project. It has emphasized the notable modification made to enable transfer learning. It has also defined four partial loss functions and the full loss used in the experiment. The next chapter will discuss the experiment implementation details and the results.

Chapter 4

Current progress

This chapter will cover the current progress of this project. We have trained the dataset based on the model in Chapter 3. It will first explain the implementation details and then discuss the experiment results. Limitations of the project will be analyzed and a schedule will be proposed for future works.

4.1 Implementation details

CelebA [9] is used to train the model. We adopted Adam optimizer with learning rate 0.0001, batch size 25, $\beta 1$ 0.5, $\beta 2$ 0.999. The weights in equation 3.6 were set to $\lambda_1 = 0.1, \lambda_2 = 10, \lambda_3 = 4000$ as indicated in [12]. We performed an optimization step of G every five steps of D. Since training requires substantial computing resources, we used HKU GPU farm to handle the task. It took 60 hours to train the dataset for 30 epochs with a GTX 1080Ti GPU.

4.2 Experiment results

We trained the model for 30 epochs. The results after 10, 20 and 30 epochs are shown in Fig. 4.1 on the first, second and third row respectively. The leftmost image is the source face while the rightmost image is the target face. The middle image on each row is the generated image with the identity of the source face and the expression of the target image. As is shown in the result, the smile of the man gradually becomes weaker towards neutral expression. Some artifacts are observable around the mouth after 10 epochs. While after 20 and 30 epochs, the image gradually becomes better and artifacts are reduced.

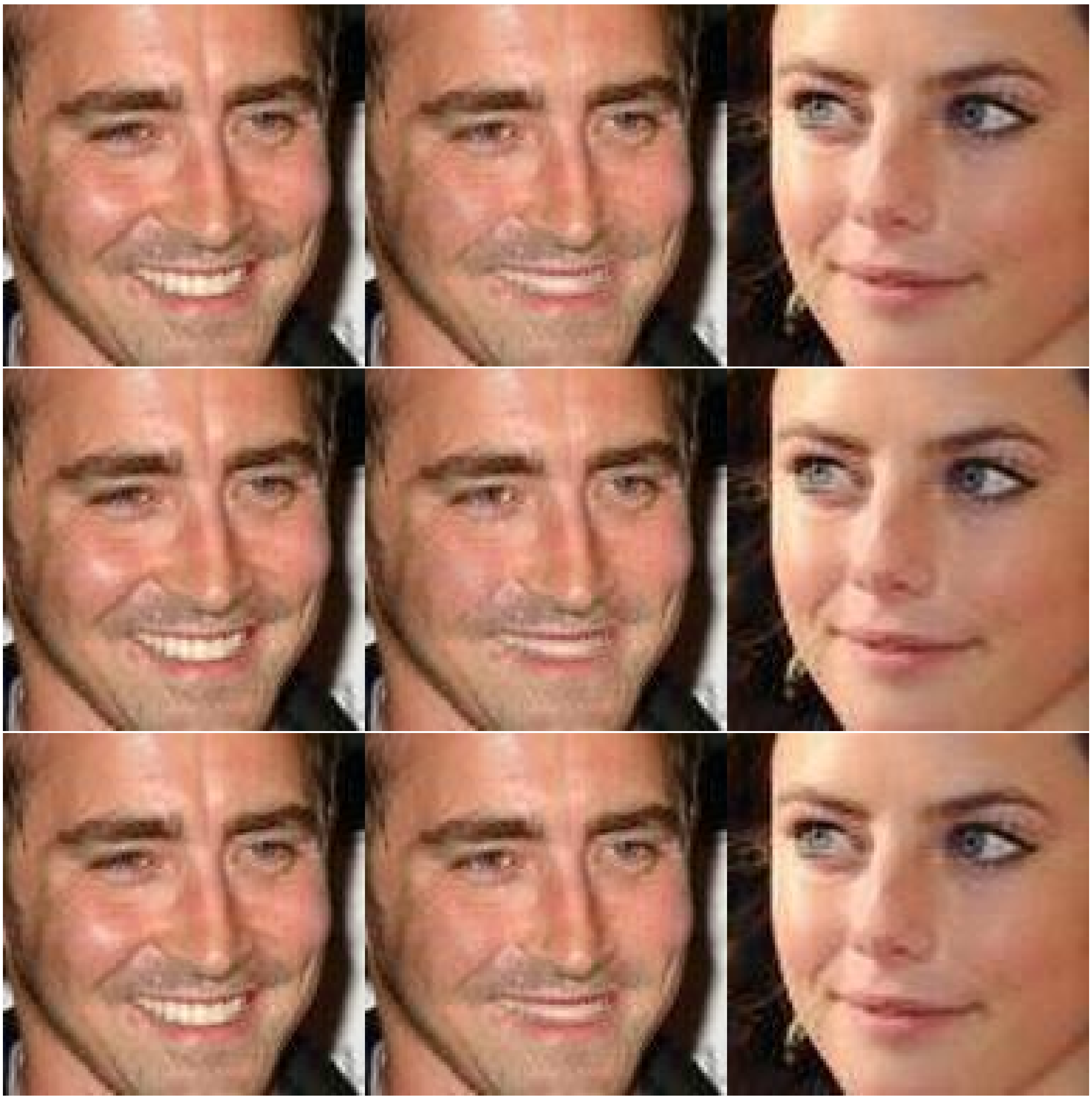

Figure 4.1: Results after different epochs

We tested more images using the checkpoints of the model after 30 epochs. As is shown in Fig. 4.2, we linearly interpolated five images to show the gradual change of facial expressions from the source image on the left to the target expression on the right. Our model ignores unrelated elements like hair and background and focuses on altering the expression. As for the last row, we use the same source and target faces to check the behavior of the model.

4.3 Evaluation

The results of testing are quite satisfactory. The model can alter expressions smoothly from the source to the target while still preserving the identity of the person. In the first row of Fig. 4.2, the smile of the man becomes weaker and weaker, but his mouth does not close in the end. This is understandable because the difference

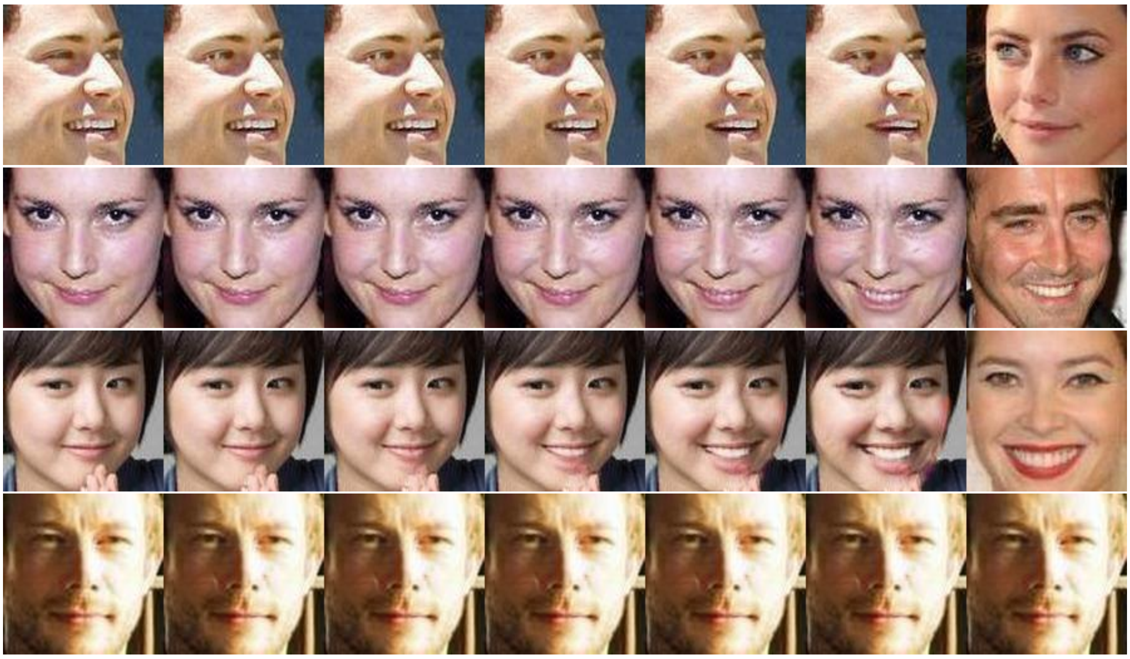

Figure 4.2: More results after 30 epochs, we interpolate the images to show the gradual change of facial expressions.

between the source expression and target expression is large. In the last row of Fig. 4.2, we use the same source and target faces. Intuitively, the model should not make any change since the source image already meets the requirement. Our model does not modify the expression as expected, which indicates that our approach is correct.

4.4 Difficulties and limitations

One difficulty lies in the setting of hyperparameters. There are three weights in equation 3.6 which control the balance of attention, identity and facial expression. They need to be set properly to generate an image with source identity, target expression and photorealism. Currently, we use the weights from [12] to train the model.
Finetuning hyperparameters requires huge computing power, but we are limited by the hardware resources. We only have access to one GPU while each training process of 30 epochs takes nearly 60 hours.

4.5 Next steps

Currently we have trained our model and show visual results of the synthesis process, but we have no idea how this model compares to other models in the literature. In the future, we will do feature, qualitative and quantitative comparison of our model with state of the art models. Feature comparison is to analyze the usage of the approach in different tasks. Qualitative comparison presents synthesis results of models regarding same faces. Quantitative comparison provides a more accurate way to evaluate the performance of different models.

Summary

This chapter has introduced the implementation details of the experiment and the results after training. It has evaluated the current results and shown our approach is correct. It has also discussed the difficulties and limitations of the project. A project schedule has been proposed for future work. We will conclude the project in the next chapter.

Chapter 5

Conclusion

Facial expression synthesis renders face images with target expressions while preserving the identity information. It is widely used in affective interaction, data augmentation, film production, etc. The first objective of this project is to synthesize faces based on different expressions annotated in action units and generate realistic images. The further objective is to apply the model to videos and solve the face reenactment problem. We have currently proposed a model based on GANimation [12] and trained it on a public dataset called CelebA [9]. It takes a source image and a target expression as input and generates a face with the target expression and the original identity. The whole training process is performed in an unsupervised manner. As is suggested by the experiment results, the model manages to alter the expression smoothly from the source expression to the target expression. For the next step, we will compare the model with other models in the literature. This will open the possibility of a state of the art facial expression synthesis method and face reenactment in the future.

References

[1] Tadas Baltrušaitis, Peter Robinson, and Louis-Philippe Morency. "Openface: an open source facial behavior analysis toolkit". In: *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE. 2016, pp. 1–10.

[2] Yunjey 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, pp. 8789–8797.

[3] E Friesen and Paul Ekman. "Facial action coding system: a technique for the measurement of facial movement". In: *Palo Alto* 3 (1978).

[4] Pablo Garrido et al. "Automatic face reenactment". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 4217–4224.

[5] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems*. 2014, pp. 2672–2680.

[6] Ishaan Gulrajani et al. "Improved training of wasserstein gans". In: *Advances in neural information processing systems*. 2017, pp. 5767–5777.

[7] Phillip Isola et al. "Image-to-image translation with conditional adversarial networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1125–1134.

[8] Christian Ledig et al. "Photo-realistic single image super-resolution using a generative adversarial network". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4681–4690.

[9] Ziwei Liu et al. "Large-scale celebfaces attributes (celeba) dataset". In: *Retrieved August* 15 (2018), p. 2018.

[10] Arsha Nagrani, Samuel Albanie, and Andrew Zisserman. "Seeing voices and hearing faces: Cross-modal biometric matching". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8427–8436.

[11] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. "Voxceleb: a large-scale speaker identification dataset". In: *arXiv preprint arXiv:1706.08612* (2017).

[12] Albert Pumarola et al. "Ganimation: Anatomically-aware facial animation from a single image". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 818–833.

[13] Andrés Romero et al. "SMIT: Stochastic multi-label image-to-image translation". In: *Proceedings of the IEEE International Conference on Computer Vision Workshops*. 2019, pp. 0–0.

[14] Soumya Tripathy, Juho Kannala, and Esa Rahtu. "ICface: Interpretable and Controllable Face Reenactment Using GANs". In: *arXiv preprint arXiv:1904.01909* (2019).

[15] Qingshan Zhang et al. “Geometry-driven photorealistic facial expression synthesis”. In: *IEEE Transactions on Visualization and Computer Graphics* 12.1 (2005), pp. 48–60.

[16] Zhifei Zhang, Yang Song, and Hairong Qi. “Age progression/regression by conditional adversarial autoencoder”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017, pp. 5810–5818.

[17] Yuqian Zhou and Bertram Emil Shi. “Photorealistic facial expression synthesis by the conditional difference adversarial autoencoder”. In: *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*. IEEE. 2017, pp. 370–376.

[18] Jun-Yan Zhu et al. “Unpaired image-to-image translation using cycle-consistent adversarial networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232.