A FACIAL EXPRESSION GENERATION SYSTEM

Shucheng Zhong, Tongan Cai, Qiucheng Wu, Chengcheng Jin
{joezhong,johncta,wuqiuche,jccccccc}@umich.edu

Abstract

Face, as one of the most important first-glance impressions of human nature, has always been the interest of Computer Vision. Hundreds of thousands of face related databases that specialists in Computer Vision are constructing and utilizing. In this project, a system for facial expression generation is proposed with the help of Haar-Cascades Face Detection and Generative Adverbial Networks trained on both Cohn-Kanade Dataset and Japanese Female Facial Expression (JAFFE) datasets. Tests on both modules are showing good performance and the system can achieve fair generation results from input. The potential of the system can be real-time facial expression generation, and has been tested on several real world scenarios. This system shows its promising value in generating facial expression in an interactive and efficient way, and a good prospect to be utilized on large scale for different purposes.

1. Background & Impact: Why do this

Emoticons have been a fashion in online communication for a long time. Old-fashioned emoticons consist of only simple symbols and can't express detailed emotions. With the development of the digital technology, the emoticons are evolving as well, from symbols to emoji and even "Animoji" proposed by Apple Inc. Still, the form of emoticons is limited and not "natural" enough. Among the social communication website, users are provided standard emotions, which seems hard to distinguish themselves. A natural thought is to replace the emoticons with real faces of users, yet it will be tedious for users to change their facial expressions and take photos every time.

With the development of Computer Vision, the need for better quantity and quality of image sources is growing. Making new dataset is in demands of intense labor and has great problem of variation. One dataset could fit some certain situation, yet may perform poorly on other settings. Such issue well exists among facial expression datasets, with low amount of high quality datasets and relatively small size for each one.

This project purpose a Facial Expression Generation System as a general way of synthesizing multiple human facial expressions. Cooperating with web cam face detection, the users can expect the system to generate a series of other human emotions within a short time. Potentially, this project can fulfill the needs of online emoticons, as well as producing facial expression dataset with high efficiency.

2. Related works

2.1. Face Detection

Ever since the 90s, when digital cameras just came into the market, face detection soon became a hot topic of Computer Vision. How to let machine detect human faces accurately and efficiently has always been interesting scientists a lot. The main ideas in face detection came first as feature-based methods, i.e. manipulate distance, angles, and area measurements of the visual features derived from the scene. Common implementations like Edge detection [6] and color [7]. One other way is subspace method, and basis method is just one of such example. In recent years, learning methods also inspire works that cooperate with pattern recognition with neural network. For the proposed system in this project, a feature-based method evolved from traditional feature detection way, Harris-cascade, was used.

2.2. Facial Expression Generation

Generation of facial expression goes well behind the development of detection and recognition. Traditional methods include feature modification and refinement, as well as basis methods (EigenFace [8] can be used as one such example). Recent years, many research have been done with regarding to image-to-image Generative Adversarial Network (GAN) [2], showing great power and potential. Traditional GANs(cGANs) have been studied by providing class information to both generator and discriminator in order to generate samples conditioned on the class and one model for each generation task, while in this project, a unified generative adversarial network was implemented that works in "one-for-many" pattern.

3. Method

3.1. Haar-Cascade Face Detection

3.1.1 Overall Advantages

The system is designed to be real-time so that fast computation feature becomes a necessary requirement of the face detection algorithm in this system. Therefore, Haar-Cascade Face Detection [9] is chosen. Three main features can be boiled down to account for why Haar-Cascade Face Detection can finish detecting task fast.

The first one is combining successive classifiers in a cascade structure, more complex processing is reserved only for them. The second one is selecting a small number of important features using AdaBoost. The third one is that Haar-Cascade Face Detection can take advantage of integral image method to make the calculation of a certain region in constant time. This enables the system to only compute several operators for each pixel once and get arbitrary region's value in constant time.

3.1.2 Cascade feature classifiers

For this system, the most critical motivation for features is that feature based system operates much faster than a pixel-based one. As shown in picture below, different kinds of feature kernels are used to convolve with target image to find out various features of a general face. In addition, to speed up the detection process, as long as a certain block of classifiers fail during the detection process, the system will not continue to do the following judgment but regard the input as negative directly for the sake of computation speed.

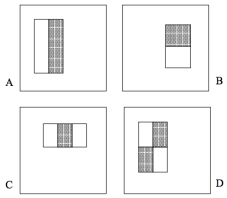

Figure 1. Feature Classifer for Cascade

3.2. GAN-Based Facial Expression Generation

3.2.1 Generative Adversarial Network

Generative adversarial networks (GANs) are a class of artificial intelligence algorithms used in unsupervised machine learning, implemented by a system of two neural networks contesting with each other in a zero-sum game framework. [2] GANs are widely used to generate photorealistic fake images from the input of real images, which can be served as the tool of generating human facial expression. This system takes the reference of StarGAN, a unified GAN for multi-domain image-to-image translation. StarGAN only needs to train 1 pair of generator and discriminator to generate k different expressions from only one image, while tradition cGANs requires k pairs of generator and discriminator, which greatly saves time and computing power.

3.2.2 Training of StarGAN

This part briefly summarizes the principals of StarGAN. Details can be find in the original paper. [1]

In StarGAN, the input of the generator G is a real image x and the expression label c, the output of G is a generated image: $G : \{x, c\} \rightarrow x'$. The input of discriminator D is an image x and the output is two probability distributions over both sources(generated or not) and classified labels(which expression the image belongs): $D : x \rightarrow \{D_{src}(x), D_{cls}(x)\}$. Then, the adversarial loss can be defined as

$$L_{adv} = \mathbb{E}_x[\log D_{src}(x)] + \mathbb{E}_{x,c}[\log(1 - D_{src}(G(x, c))]$$

The generator G tries to minimize this loss to fool the discriminator D, while the discriminator D tries to maximize it to distinguish the generated images from the real ones.

Classification loss is introduced for both generator and discriminator. For the generator G, the classification loss is defined as

$$L^G_{cls} = \mathbb{E}_{x,c}[-\log D_{cls}(c|G(x, c))],$$

where c is the target expression we want to generate. By minimizing it, generator can generate more realistic image representing target expression c. For discriminator D, it is trained by the classification loss defined as

$$L^D_{cls} = \mathbb{E}_{x,c}[-\log D_{cls}(c|x)],$$

where x is the real images and c is the original expression provided in dataset.

Additionally, the reconstruction loss is applied to preserve the contents of its input image from generator. Let (x, c') is the input image and original expressions and c is the target expression. Generator G first generates the image with regarding to target expression c and then reconstructs the image using the original expression c'. The loss is defined as the $L - 1$ difference between the reconstructed image and original image,

$$L_{rec} = \mathbb{E}_{x,c,c'}[||x - G(G(x, c), c')||_1]$$

Finally, the objective functions to optimize for G and D are given as

$$L_D = -L_{adv} + \lambda_{cls} L^D_{cls}$$
$$L_G = L_{adv} + \lambda_{cls} L^G_{cls} + \lambda_{rec} L_{rec},$$

where λ_{cls} and λ_{rec} are parameters to tune. The work-flow is shown in figure below.

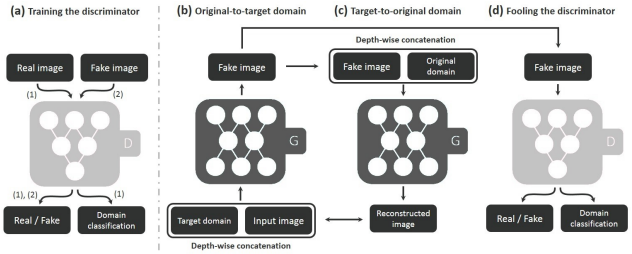

Figure 2. WorkFlow of StarGAN

4. Implementation & Prototype

4.1. System Flow & Logic

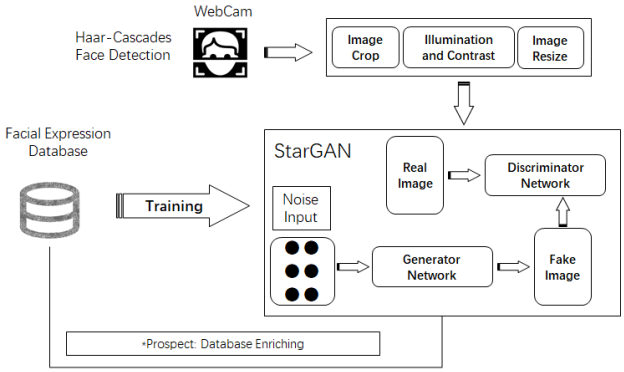

Figure 3. The system flow

The system flow of the system is shown as above. Haar-Cascade Face Detection was applied with WebCam, and input source images are cropped. After making necessary size, illumination and contrast changes, the image is then fed to the StarGAN. The StarGAN model was already trained by multiple facial expression datasets including JAFFE [5] and Cohn-Kanade [3].

The first phase, the core of Haar-Cascade Face Detection was implemented by Open-CV and benefited from hardware optimization. A feed logic was created so that it could automatically be forwarded to the generation phase. The StarGAN was then implemented with customized loader and solver for our own training of facial expression datasets. The program originally gives choice for train or test mode, but for the production stage it will always be in test logic. If choose to train, it will load data and train the GAN from scratch. Checkpoint models would be saved for later generation purpose. Else if in test mode, the system load the saved model and run generation from saved model. For the output file, the first column will be the original image, and the latter will be the synthesize image with facial expressions specified by users. Note that the features in test mode would need to be the subset of training mode.

4.2. Prototype Demo

Direct to the base directory of project, excute `python main.py`. The interface will give the real-time feedback for the detection, and will allow 10 seconds before saving the image. The program will then crop, resize and will go for generation, and save the result in the given directory.

Figure 4. Demonstration of Prototype Functionality

5. Result

5.1. Training Result

The result of training the model on Cohn-Kanade Dataset with 200,000 iterations and batch size 16 is shown as follows (selected results with good features):

`{Original | Neutral, Angry, Contempt,Disgust, Fear, Happy, Sad, Surprise}`

Figure 5. Result of training on Cohn-Kanade

The result of training the model on JAFFE Dataset with 100,000 iterations and batch size 16 is shown as follows (selected results with good features):

`{Original|Neutral Happy Sad Surprise Angry Disgust Fear}`

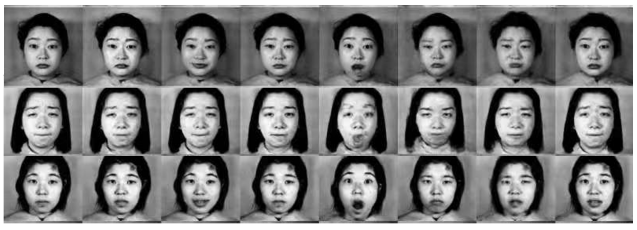

Figure 6. Result of training on JAFFE

5.2. Generation Result

Considered the quality of the images, the model trained on JAFFE was adopted for the generation phase of the system. The generation results on the input images from the

webcam are shown as follows. The last three rows are taken by mobile phone front camera for comparative study.

{Original|Neutral Happy Sad Surprise Angry Disgust Fear}

Figure 7. Result of generation phase

Result of the generation phase demonstrated the capability of the system, and the change of different facial expressions can be observed.

6. Conclusion, Prospect & Discussion

This Facial Expression Generation System shows the idea in which single image is analyzed and synthesized for multiple facial expression fake images. The result in the previous part demonstrated the system's in generating multiple facial expressions from single input. There's still great potential to improve its functionality. The output features of the image are rather less clear than expected, and there are great portions of noise inside the generated images. This is because of the lack of training data and the variation of input quality. If more images with similar settings (random illumination condition, different face characteristics, higher resolution, color options), a better effect of generation–with color and better quality–will be expected.

It's noticed that the generation phase with the trained model will be within 3 seconds[1]. There is need for more powerful platform, and potentially cloud computing would make it way more efficient, for which the system is expected to give real-time and large scale generation capability.

When group members are working on the report, one article "A Style-Based Generator Architecture for Generative Adversarial Networks" just got published [4]. Researchers from NVIDIA improved the structure of GAN and declared their structure is able to synthesis and generate images of great quality for almost everything. Such work will potentially support the idea like the proposed system in this project by improving generation quality, and it seems promised to be really commercialized someday.

[1]In this project, the platform for training is Intel Core i7-4790, with 16G RAM and NVIDIA GTX 1070-8G, while the prototype was demonstrated on a device with Intel Core i7-3635, 8G RAM and Integrated Graphic Card

References

[1] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image- to-Image Translation. *arXiv e-prints*, page arXiv:1711.09020, Nov. 2017.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[3] T. Kanade, J. F. Cohn, and Y. Tian. Comprehensive database for facial expression analysis. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, pages 46–53, March 2000.

[4] T. Karras, S. Laine, and T. Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. *arXiv e-prints*, page arXiv:1812.04948, Dec. 2018.

[5] M. Lyons, S. Akamatsu, M. Kamachi, and J. Gyoba. Coding facial expressions with gabor wavelets. In *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 200–205, April 1998.

[6] T. Sakai, M. Nagao, and T. Kanade. Computer analysis and classification of photographs of human faces. In *Proc. First USA-JAPAN Computer Conference*, pages 55–62, January 1972.

[7] J. Terrillon, M. N. Shirazi, H. Fukamachi, and S. Akamatsu. Comparative performance of different skin chrominance models and chrominance spaces for the automatic detection of human faces in color images. In *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition (Cat. No. PR00580)*, pages 54–61, March 2000.

[8] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1):71–86, Jan. 1991.

[9] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, Dec 2001.