Facial Expression Synthesis

Maheshwari Kotha,Sri Harshitha Karuturi, Shruthi M S and Sowmya Vasuki J
Indian Institute of Information Technology (IIIT) Sri City, A.P - 517646
{maheshwari.k16, sriharshitha.k16, shruthi.ms16, sowmyavasuki.j16}@iiits.in

***Abstract*—Facial expression synthesis is an emerging field in the community of deep learning and computer vision. Facial synthesis is a process in which faces with new expressions are generated without disturbing the distinct facial characteristics of the original face. This paper deals with methodology and results of facial expression synthesis. To achieve this goal, Geometric-Contrastive Generative Adversarial Network (GC-GAN) is used. This is because GC-GAN can generate identity-preserving face with the target expression, given an input face with certain emotion and a target facial expression from another subject. Geometry information is introduced into GANs as continuous conditions to guide the generation of facial expressions. The generated facial expression can be used to enhance various technologies in the present world like existing face identification system, human face recognition.**

***Index Terms*—Facial expression synthesis, GANs, Generative models.**

I. Introduction

Over the past decades, human-computer interaction together with computer vision has been an important field in computer study. Direct communication between the computer and human beings is matter of concern. Lot of research has been conducted on improving and developing the interaction between human and the computer. Facial expression is essential to human communication as well as voice. It includes several kinds of factors which can express non-verbal information as a voluntary or a spontaneous activity. So, recognizing and synthesizing facial expressions can also improve the communication environment between humans and machines.

High-level manipulation of facial expressions in images such as expression synthesis is challenging because facial expression changes are highly non-linear, and vary depending on the facial appearance. Being able to automatically animate the facial expression from a single image would open doors to many new exciting applications in different areas, including the movie industry, photography technologies, fashion and e-commerce business etc. Identity of the person should also be well preserved in the synthesized face. Face is the most exposed part of the body, and allows to use deep learning and computer vision to analyse the expression/emotions. There has been a huge improvement in the facial expression analysis and recognition.

In this project, our aim is to synthesize an output face with the target expression given a input image with a certain emotion without losing its identity. The block diagram of the approach is depicted in Fig.1.

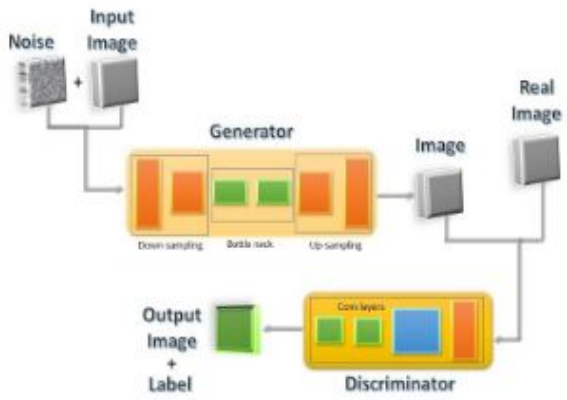

Fig. 1. Block diagram of Approach

The paper is organised as follows - section II explains the related work, section III the dataset used, section IV elucidates the current approach(based on GC-GANs), section V details the implementation, section VI contains the results, section VIII summarizes the paper and throws light on the future work.

II. Related Work

Various methods for face detection can be grouped into four categories. Knowledge based methods, feature invariant approaches, template matching methods and appearance-based methods. Knowledge based methods are rule based methods. These methods try to capture human knowledge of faces, and translate them into a set of rules. The feature invariant approach finds some invariant features for face recognition. The idea is to overcome the limits of our instinctive knowledge of faces. The template matching methods compare input images with stored patterns of faces or features. These appearance-based methods rely on techniques from statistical or probabilistic analysis and machine learning to find the relevant characteristics of face images.

Statistical methods provide a way for estimating missing or uncertain information. The statistics works on a big set of data, we want to analyze that sets in terms of the relationships between the individual points in the data set. Principle Component Analysis (PCA) is the way of identifying patterns in the data, and expressing the data in such a way as to highlight their similarities and differences.

Fig. 2. Sample Images from Dataset

The other main advantage of PCA is data compression, by reducing the number of dimensions, without significant loss of information. Appearance based face recognition methods are PCA, Linear Discriminant Analysis (LDA), Independent Component Analysis(ICA). But with the recent development of GANs, image synthesis has migrated from pixel level manipulations to semantic level.

This paper proposes the use of GC-GAN over other types like Star GAN owing to the fact that StarGAN [3] cannot effectively take into account all the distinctive face related features [1]. This method takes into consideration the geometric features which are important for identity preservation [2] [5]. Other works include use of DC-GAN especially in unsupervised learning scenario [6]. The EmotioNet challenge [4] tested the ability of computer vision algorithm for Action Unit (AU) identification. The algorithms showed poor results for 3D-pose images.

III. Dataset

The FERA 2013 dataset has been used for the facial expression synthesis. This dataset consists of test set, train set and validation set. Each of the sets contain folders corresponding to the various facial expressions. Face expressions include the contempt, sadness, anger, disgust, surprise, fear, neutral and happiness. Fig.2. shows the samples for each emotion from the dataset.

We have merged the training and validation set before training. The test set and train set consist of 4000 and 500 images respectively corresponding to each emotion. Before training, we need to preprocess the images. In this pre-processing step, initially the images are flipped horizontally and all the images along with the flipped ones are resized to a resolution of 256*256. Then the dataset is fit to the model for training.

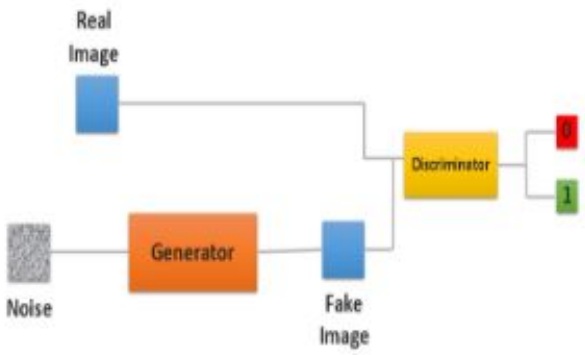

Fig. 3. General Structure of GAN

IV. Approach

A. Introduction to GANs

GANs [7]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. One neural network, called the generator, generates new data instances, while the other one , discriminator, evaluates them for authenticity; i.e. the discriminator decides whether each instance of data it reviews belongs to the actual training dataset or not. Fig.3 shows the genral structure of GAN with generator and discriminator. GANs potential is huge, because they can learn to mimic any distribution of data. That is, GANs can be taught to create worlds eerily similar to our own in any domain: images, music, speech, prose. They are robot artists in a sense, and their output is impressive poignant even.GANs are being extensively used for various purposes like text to image synthesis [8].

The discriminator network is a standard convolutional network that can categorize the images fed to it, a binomial classifier labelling images as real or fake. The generator is an inverse convolutional network, in a sense: While a standard convolutional classifier takes an image and down-samples it to produce a probability, the generator takes a vector of random noise and up-samples it to an image. The first throws away data through down-sampling techniques like maxpooling, and the second generates new data. In original GAN there is no control over the output (completely dependant on the random noise), so Conditional GAN, in particular GC -GAN are used, wherein the obtained output capures the geometric information which is fed as the condition [9].

The steps involved in GAN:

1) The generator takes in an image as input and generates fakes using it.
2) These generated images along with the actual images in the dataset are fed to the discriminator.
3) The discriminator tries to predict whether the given image is real or fake.

4) The discriminator is in feedback loop with ground truth of the images and the generator is in feedback loop with the discriminator.

B. Use of GAN in implementation

Some parameters are defined before start of training. We define num_iters=200000, batch_size = 16 and g_lr = d_lr = 0.0001. It means we train the model for 200000 epochs with a batch size of 16 and learning rates of generator and discriminator as 0.0001.

A generator is designed to map the latent space vector to the data-space. This is accomplished through a series of strided two-dimensional convolutional transpose layers, each paired with a 2d batch Instance Norm layer and a Relu activation. The output of the generator is fed through tanh function to return it to the input data range. The generator consists of 3 layers: down-sampling layer, bottleneck layer and the up-sampling layers. In a sequential order the training set is fed to the generator. The input image passes through the down-sampling and up-sampling layer twice and once through the bottleneck layer. The discriminator takes the input image, processes it through a series of Conv2d and LeakyReLU layers. LeakyRelu functions promote healthy gradient flow which is critical for the learning process. Same as the in generator the input image is fed to the network in a sequential order. For every 10000 epochs, the model is being saved along with the sample outputs. PyTorch and TensorFlow have been used to implement the above neural networks.

V. Implementation

Implementation involves two major steps viz., Training and Testing

A. Training

The training is split into two parts i.e., training the discriminator and the generator.

1) Training of the Discriminator: It is trained to classify correctly between the real and fake images. Firstly, the discriminator is trained with the real images of the data set and the loss is calculated followed by calculation of gradients in backward propagation. Now the same is done again with fake samples generated by the generator. Loss is calculated along with the gradients with a backward pass. with the gradients accumulated from both the all-real and all-fake batches, we call the discriminator optimizer.
2) Training of the Generator: The main aim of the generator is the make fake images. The output images of the discriminator is fed to the generator computing Gs loss using real labels, computing Gs gradients in a backward pass, and finally updating Gs parameters with an optimizer step.

At the end of each epoch we will push fixed noise batch through the generator to visually track the progress of Gs training. The training statistics reported are:

Fig. 4. Results for a set of four emotions i.e. anger, contempt, disgust and fear

1) Loss_D - discriminator loss calculated as the sum of losses for the all real and all fake batches (log(D(x))+log(D(G(z)))).
2) Loss_G - generator loss calculated as log(D(G(z)))
3) D(x) - the average output (across the batch) of the discriminator for the all real batch. This should start close to 1 then theoretically converge to 0.5 when G gets better.
4) D(G(z)) - average discriminator outputs for the all fake batch. The first number is before D is updated and the second number is after D is updated. These numbers should start near 0 and converge to 0.5 as G gets better.

B. Testing

An image from the testing set is fed to the generator, along with some noise function. The generator generates the fake images i.e., image with trained emotions and is fed to the discriminator. The discriminator identifies the class of the emotion and gives the output as images with emotion name.

VI. Results

In the training process, we save the model after every 10000 epochs and after that we take an image from the test set to test the model and save the output to the samples folder. After the entire training process is completed, the final result is saved to the results folder. The outputs thus saved are as shown in Fig.4 (Anger, Contempt, Disgust, Fear) and Fig.5 (Happiness, Sad, Surprise, Neutral)

VII. Summary & Further steps

Development of an automated system that accomplishes facial expression synthesis is difficult. Various approaches have been made towards robust facial expression synthesis. In this paper, we have presented a GAN model for facial expression synthesis that can be trained in a fully unsupervised

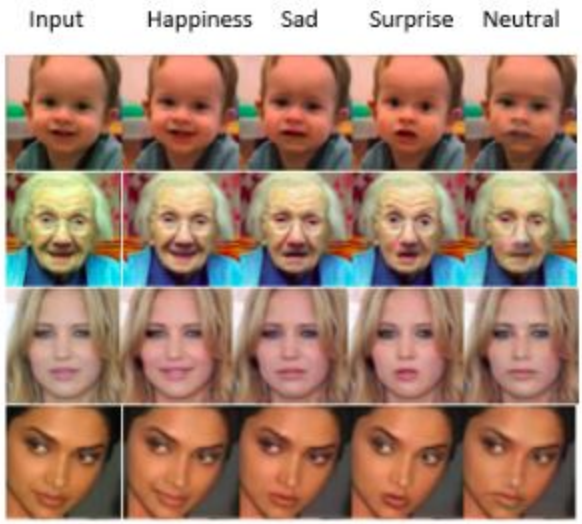

Fig. 5. Results for a set of four emotions i.e. Happiness, sad, surprise and neutral

manner. The results are very promising, and show smooth transitions between different expressions. In the future, we can try implementing this for video sequences as well.

VIII. Acknowledgements

We would like to acknowledge the work done by Choi, Yunjey, et al. "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation." arXiv preprint 1711 (2017) as this is the approach used in our project. We also like to thank all others who have contributed to the field of image synthesis and GANs. Lastly, we also like to thank Dr.Snehasis Mukerjee for providing us the opportunity to work and learn about with GANs.

References

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

[2] Zhang, Qingshan, et al. "Geometry-driven photorealistic facial expression synthesis." *IEEE Transactions on Visualization and Computer Graphics 12.1 (2006): 48-60*

[3] Choi, Yunjey, et al. "Stargan: Unified generative adversarial networks for multi-domain image-to-image translation." *arXiv preprint 1711 (2017).*

[4] Benitez-Quiroz, C. Fabian, et al. "EmotioNet Challenge: Recognition of facial expressions of emotion in the wild." *arXiv preprint arXiv:1703.01210 (2017).*

[5] Song, Lingxiao, et al. "Geometry guided adversarial facial expression synthesis." *ACM Multimedia Conference on Multimedia Conference. ACM, 2018.*

[6] Radford, Alec, Luke Metz, and Soumith Chintala. "Unsupervised representation learning with deep convolutional generative adversarial networks." *arXiv preprint arXiv:1511.06434, 2015*

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In NIPS, pages 2672 2680. 2014.

[8] Reed, Scott, et al. "Generative adversarial text to image synthesis." *arXiv preprint arXiv:1605.05396, 2016.*

[9] Huang, He, Phillip S. Yu, and Changhu Wang. "An Introduction to Image Synthesis with Generative Adversarial Nets." *arXiv preprint arXiv:1803.04469, 2018.*