Realistic Face Generation with Facial Expression Synthesis

A BTP Report

by

K.Upendra Sainath Reddy

Roll No: S20160010038

K.Sai Prasanna Kumar

Roll No: S20160010047

Under the guidance of

Dr. Snehasis Mukherjee

Dr. Shiv Ram Dubey

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SRICITY

AUGUST 2019

Contents

List of Figures

List of Abbreviations and Symbols

ANN Artificial Neural Network.

CNN Convolutional Neural Network.

DCGAN Deep convolutional Generative Adversarial Network.

GAN Generative Adversarial Network.

GPU Graphical Processing Unit.

WGAN Wasserstein Generative Adversarial Network.

Abstract

There are several methods that allows facial expression editing which doesn't involve deep learning, But the recent advancements in deep learning helps us the understand how the deep neural networks would do the similar tasks. Especially, GANs[2] [3] is a recently proposed deep learning method. By using GAN's to different problems, such as synthesization of face expression expression, we can better understand and discover new features. Ongoing investigations have demonstrated noteworthy accomplishment in Image to Image translation for two domains. Existing methodologies have limited scalability and robustness in dealing with multiple domains,since different models should be built independently for every pair of image domains. To generate facial expressions for an image we implement a new GAN model called starGAN [4],It is a scalable approach that can perform image-to-image translations for multiple domains using only a single model.

Chapter 1

Introduction

The goal of this project is to create a facial expression synthesizer that generates face images with several different face expressions using GAN's. The task of generating face expressions from an input image falls under the category of Image to Image translation of deep learning where the main aim of Image-to-Image translation is to learn the mapping between an input image and an output image. Image to Image translation has uses like Image Enhancement, season transformation, face expression synthesis .. etc.

This synthesizer is a variation of GAN which will be trained on various face images and and ideally output the synthesized expressions for the face . The synthesizer takes an image as input and outputs different facial expressions for the given input. Though the synthesizer outputs several facial expressions the deep learning model also has to preserve the identity of the input face features.

StarGAN is used to achieve Image to Image translation for multiple domains using only a single model. StarGAN [4] allows simultaneous training of multiple datasets with different domains(disgusted, angry and fearful, neutral, happy, sad, surprised) within a single network.

Chapter 2

Literature Review

2.1 DCGAN:-

DCGAN stands for Deep convolutional Generative Adversarial Network. The initial GANs that has the regular neural network are only able to generate small images but the inital GAN's have difficulties in handling large images. The DCGAN is proposed to overcome the shortcomings of inital GAN's. The major modication made in DCGAN is to replace the regular neural network with convolutional networks for both discrimination and generator. Now the discriminator resembles a CNN, and the generator resembles a reversed CNN which is achieved by deconvolution (transposed convolution)[10].
Convolutional layer turns a large size image into small size feature maps where as in Deconvolutional layer turns a smaller size features into a set of larger size feature maps. This

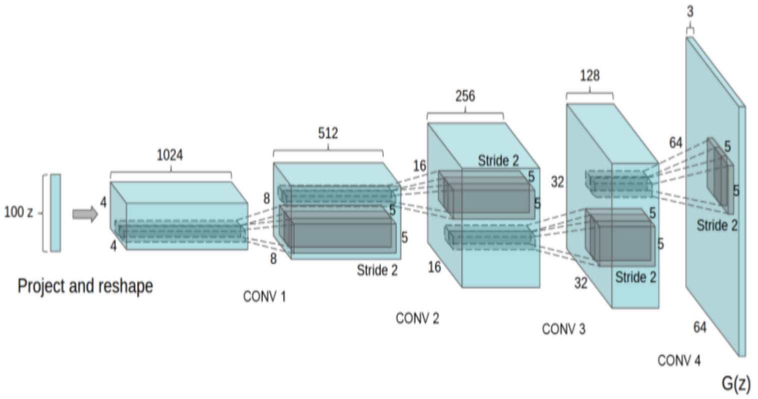

Figure 2.1: Generator of a DcGAN using Deconvolutional Network

is achieved by padding input zeros to input image so that size of input image is larger than size of outer map and apply convolutional layer to get desired output. DCGAN allows higher learning rate for training by using batch normalization technique between layers of discriminator and generator.

2.2 WGAN

WGAN is similar to DcGAN model architecture but different objective function . Instead of using the initial Jensen-Shannon divergence to calculate distance between p_{data} and p_{model} we use Wassertein distance to avoid problem during training. According to Wassertein distance minimizes approximation of the Earth Movers distance where EM distance is a method to look at dissimilarity between two multi-dimensional datasets.
Wasserstein distance is measure of distance between two probability distribution. Discriminator's parameter must be clamped in between small range [-c,c]. In new loss function there is no logarithm, Discriminator helps in estimating Wasserstein distance between generated and real data distribution. Use a new loss function derived from the Wasserstein distance, no logarithm anymore. The discriminator model does not play as a direct critic but a helper for estimating the Wasserstein metric between real and generated data distribution Using WGAN, Stability of optimization process is improved and we have a loss metric that correlates with the generators convergence and sample quality.

Chapter 3

Methodology/Design

The main aim of this project is to create a facial expression synthesizer that generates face images with different face expressions using GAN's. So, The initial plan of the project included four modules.They are:-

- In the first module of our project we try to generate faces. We implement DcGAN[3] to generate faces.
- In the second module the model will fetch the facial expression corresponding to an input image using an StarGAN [4] architecture. Given an image the model will generate face images with given facial expressions.
- In the third module we're modifying StarGAN[4] architecture by adding discriminator to help generator in reconstructing original image.

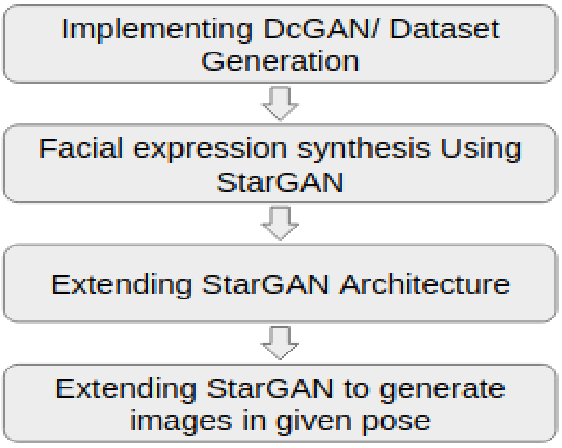
Implementing DcGAN/ Dataset Generation
Facial expression synthesis Using StarGAN
Extending StarGAN Architecture
Extending StarGAN to generate images in given pose

Figure 3.1: Flow diagram of our work

- In the fourth module we'll extend StarGAN architecture to generate images in given pose.

To generate face images we use above models(DcGAN,WGAN). As the main agenda of this project is to generate facial expression modifier from face images. For generating facial expressions we use StarGAN model.

3.1 StarGAN

Existing models for image to image translation for more than two domains have some limitations, different models should be developed for every pair of domains.So, we use StarGAN [4] model to perform image to image translation for multiple domains trained on a single network. Whole network is trained on single Generator and Discriminator. Generally in GANs Generators use deep autoencoders with encoder which encode data into smaller feature vector and decoder which decode the condensed vector obtained and reconstruct the input image. In StarGAN we slightly have variation of autoencoder. Encoder has 3 convolutional layers and Decoder has 3 transposed convolutional layers. Encoder in Generator down-samples the input data into set of feature maps.These feature maps are passed through bottleneck(six residual blocks [convolutional]) before passing through decoder. Decoder uses strided transposed convolutional layers and upsamples the resultant obtained from bottleneck network.

Discriminator classifies an image into corresponding domain with the help of auxiliary classifier and also determines whether input image is real or fake. Generator tries to generate images which are indistinct from real images and should be classified to target domain. And also tries to reconstruct the input image from generated image given original domain label. This model has three main loss functions Adversarial Loss, Classification Loss and Reconstruction Loss.

Adversarial Loss is the common objective of GANs which helps generator to create images indistinct from the real image. Discriminator uses Wasserstien GAN objective for adversarial loss. The discriminator tries to maximize the error while Generator tries to

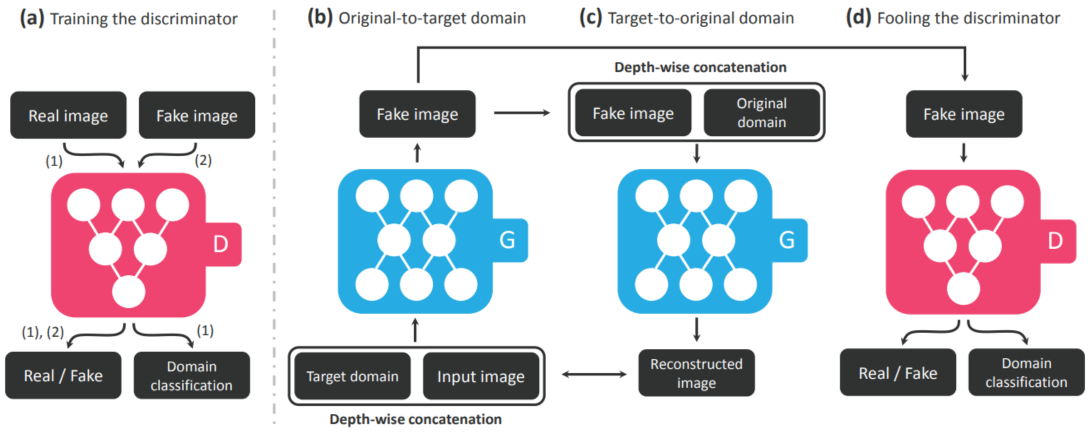

Figure 3.2: Training of a StarGAN

minimize.

$$L_{advers} = E_x\ [Log(D_{source}(x))] + E_{x,c}\ [Log(1 - D_{source}(G(x,c))]$$

Classification Loss is divided into two terms.

Loss related with how good our discriminator classifies and generating image. according to specific domain provided. Discriminator(D) minimize this objective in-order to classify a real image to its original domain The Second loss function is related to how D classifies fake images into given target domain. Generator tries to minimize this loss function.

$$L^r{}_{class} = E_{x,c^`}\ [\text{-}Log(D_{class}(c^`|x))]$$

$$L^f{}_{class} = E_{x,c}\ [\text{-}Log(D_{class}(c|G(x,c)))]$$

where $D_{class}(c^`|x)$ represents the probability that given a real image $c^`$.

Reconstruction Loss is to avoid reconstruction errors after changing particular domain.To prevent these reconstruction errors we use cycle consistency loss [5]. It reconstructs the original image from generated image and find the loss between reconstructed image and original image.

$$L_{rec} = E_{x,c,c^`}\ [||x - G(G(x,c),c^`)||_1]$$

Full Objective: Below is the Full objective function to optimize G and D.

$$L_D = \text{-}L_{advers} + \lambda_{class}\ L^r{}_{class}$$

$$L_G = \text{-}L_{advers} + \lambda_{class}\ L^f{}_{class} + \lambda_{rec}\ L_{rec}$$

Chapter 4

Experimental Analysis/Simulation Setup

The above deep learning model is evaluated on Tesla K80 GPU for feature extraction, training, and testing.The GPU is 70 times faster compared to normal CPU. The operating system of the machine is Ubuntu 16.04. For the deep learning framework py-Torch is used.

The evaluations are performed on face expression dataset. The model is trained using adam optimizer and batch size of 16 is used. when training with face expression dataset the models are trained for 100 iterations and has learning rate of 0.0001 for first 100 epochs and linear decay was applied for next 100 epochs. For training on CelebA dataset, a learning rate of 0.0001 for the first 10 epochs is used and decay for the learning rate to 0 linearly over the next 10 epochs is used

For extracting the frontal face of the image a frontal face recoginzer is used. The frontal face detector uses Dlib library and python is used to code the script. To crop the images Opencv along with Numpy libraries are used.

4.1 Dataset

Facial Expression Dataset is a mix of different datasets like cohn-kanade facial expression dataset, KDEF, MUG facial expression dataset. The dataset contains 2100 images of seven different facial expressions from different persons under different angles. There are total seven facial expressions which include neutral, happy, sad, surprised,disgusted,angry and fear. The dataset is preprocessed to get 256 x 256 cropped image of the frontal face. A dlib frontal face detector is used to detect the frontal face from the images in the dataset and the detected images are cropped into 256 x 256 size images. The dataset is splitted into 80% train set and 20% test set.

CelebA Dataset The CelebFaces Attributes (CelebA) dataset contains 202,599 face images of celebrities, each annotated with 40 binary attributes. The images are cropped initially to 178 x 178 size, then resize them as 128 x 128. We randomly select 2,000 images as test set and use all remaining images for training data.

Chapter 5

Results and Discussion

The model is trained on both face expression dataset and the CelebA dataset. The dataset is splitted into 80% training data and 20% test data. The model is trained for 100 iterations with learning rate of 0.0001 and with linear decay rate. The figure 5.1 shows the face expressions generated by the model for a given input neutral image. The figure 5.1 shows the seven expressions and the model has done a good job of preserving input features of the face in the generated image. The figure 5.2 shows the generated images for the CelebA dataset. The model has generated images with different features like Black, Gray, Brown hair and male/female gender and old/young Age on a given input image. The model struggled to generate the Aged feature and there is more loss of input image feature for the generated aged image for Aged feature image.

5.1 Evaluation on Face Expression Dataset

A face expression classifier is trained to classify the generated face expression with the target expressions. The face expression classifier uses ResNet architecture and is trained on the different face expression datasets like (Cohn-kanade, MUG and KDEF) with 80% 10% 10% as are splitted among training set, validation set, testing set. The classifier has performed well in classifying the face expressions and has 93.54% accuracy for classifying expressions on the testing data.

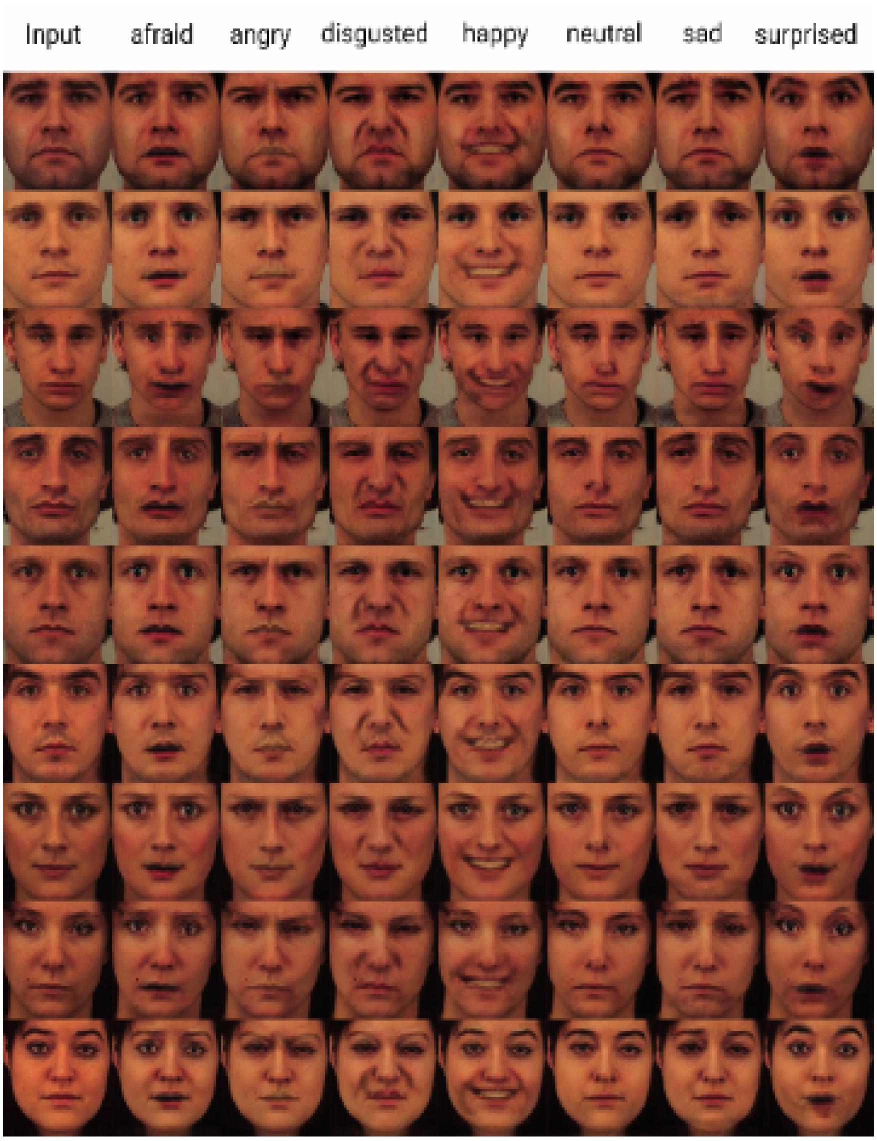

Figure 5.1: Results obtained for facial expression synthesis

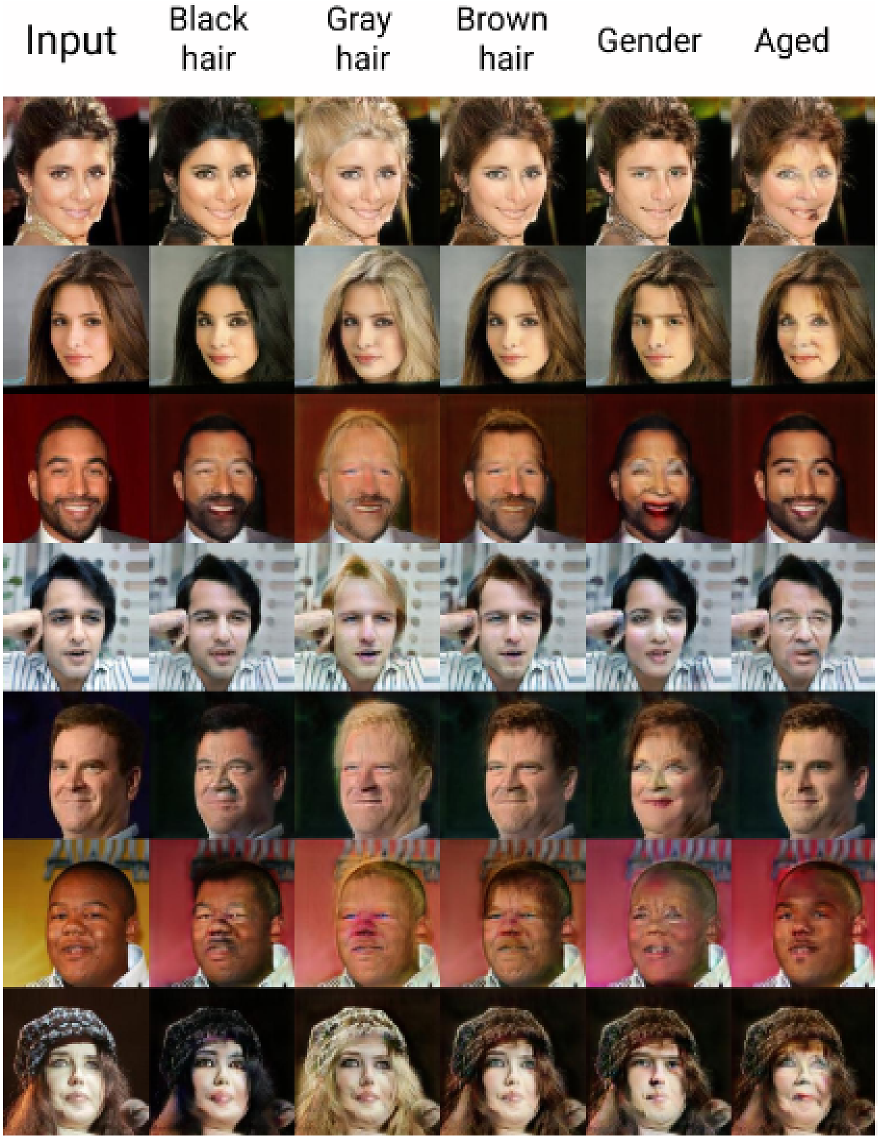

Figure 5.2: Results obtained for facial expression synthesis

Chapter 6

Summary and Conclusion

This work mainly focuses on generating several different facial expressions from a neutral image. From the results, we can see that the model can produce high-quality and precise pictures in one of the seven face expressions when a neutral Image is given as input. Surprised, however, is the hardest expression to produce, which often leads to fear and the model is successful in preserving the identity of input image from the generated image. One of the bigger challenges is preserving the data of the input image features in the reconstructed image. In future the identity of the input image in the generated image can be improved by adding a discriminator between the input generator and reconstruction image generator.

References

[1] A. Odena, C. Olah, and J. Shlens. Conditional image synthesis with auxiliary classier gans. arXiv preprint arXiv:1610.09585, 2016.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D.Warde-Farley,S.Ozair,A.Courville,andY.Bengio. Generative adversarial nets. In Advances in Neural Information Processing Systems (NIPS), pages 26722680, 2014.

[3] M. Mirza and S. Osindero. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784, 2014

[4] StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, Jaegul Choo

[5] J.-Y.Zhu,T.Park,P.Isola,and A.A.Efros. Unpaired image to image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017

[6] Dev Nag. Generative adversarial networks (gans) in 50 lines of code (pytorch), Feb 2017.

[7] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. CoRR, abs/1511.06434, 2015.

[8] T. Kanade, J. F. Cohn, and Yingli Tian. Comprehensive database for facial expression analysis. In Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition

[9] Sergey Ioe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the 32Nd International Conference

on International Conference on Machine Learning - Volume 37, ICML15, pages 448456. JMLR.org, 2015

[10] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin A. Riedmiller. Striving for simplicity: The all convolutional net. CoRR, abs/1412.6806, 2014.

[11] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In Advances in Neural Information Processing Systems 30 (NIPS 2017), pages 57695779. Curran Associates, Inc., December 2017. arxiv: 1704.00028