Generative models using Probabilistic Principal Component Analysis

Group - 2
Bhavya Patwa - 1401063, Harsh Mehta - 1401086
Vidit Shah - 1401078, Yash Turakhia - 1401118
School of Engineering and Applied Science
Ahmedabad University

***Abstract*—In computer vision, Supervised learning with CNN helps to obtain accurate results but the amount of data required for this approach is very large. To solve this problem, we use the concept of Generative Adversarial Models(GAN) to generate similar data from currently available data. In this paper we use semi-supervised learning with Deep Convolution Generative Adversarial Network Model (DCGAN). In later section of the paper, we show PPCA generation and Image Generation from given input data-set. To reduce computation time and dimensionality we use the approach of PPCA.**

Key Word - Computer Vision, Convolution Neural Network (CNN), Generative Adversarial Network (GAN)

I. Introduction And Theory

A. Generative Model

Generative models are models that randomly generates observed data values with some given latent parameters. It is a joint probability distribution where generative models are used either for modelling data or as an intermediate step to generate data. It is a very broad concept which has various application such as image denoising, super-resolution, inpainting, structured prediction etc. "Make my smile wider" in Photoshop++ uses the concept of generative models.

B. Generative Adversarial Network (GAN)

Generative Adversarial Network (GAN) are type of artificial intelligence algorithms that are used in unsupervised machine learning environment to generate data like photo, music or movies by itself. This idea was developed by in 2014 by a group of researchers at the University of Montreal lead by Ian Goodfellow. GAN are powerful approach for probabilistic modelling that enables to generate data which is similar to given input data. GAN has two competing neural network models.

Generator - This model converts random noise into image.
Discriminator - This model tries to differentiate between real and generated images. Where real image comes from our training data.

In this model, generator and discriminator works simultaneously. Discriminator has real image from training data set and fake image generated by generator. Now the discriminator says NO to fake image and YES to real image and optimizes the generator to let the discriminator believe that fake images are real.

C. Dataset

We have used data set of 35,887 images of various facial expressions. Dimension of each image is (48 X 48). This dimension reduces to 48 X 20 when we apply Principal Component to this dataset. This dataset is comprehensive enough as it contains some of the missing values and hence PPCAs can be incorporated in this model.

II. Model

Generative Adversarial Network(GAN) is able to learn by itself how to generate new images from given input data set. Deep convolutional GAN (DCGAN) is one of the way to build a practical GAN. Here, we implemented DCGAN to learn how to generate principal component from some new synthesized image.

A. Generator

Generator model is used to synthesize fake image. We generate a fake image of dimension 48 X 20 from noise having dimension 16 X 100. The technique used for this is inverse convolution. Instead of fractionally-strided convolution as suggested in DCGAN, upsampling between the first three layers is used since it synthesizes more realistic handwriting images.We have used ReLu as the activation function after each layer and batch normalization to stabilize learning.To prevent overfitting, a dropout of 0.4 is used.

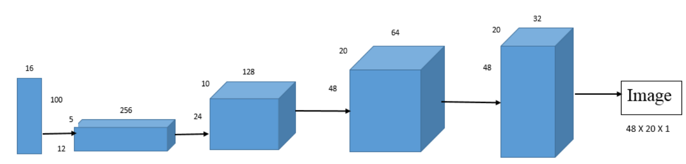

Figure 1.1 : Generator and Deconvolution

B. Discriminator

Discriminator part of the model tells the how much synthesized image is related to real image. We have used strided convolution for down sampling and leaky ReLU(Rectified Linear Unit) as an activation function in each CNN layer. Sigmoid function of a discriminator gives the probability of how real the image is. To avoid over fitting we use a dropout of 0.4.

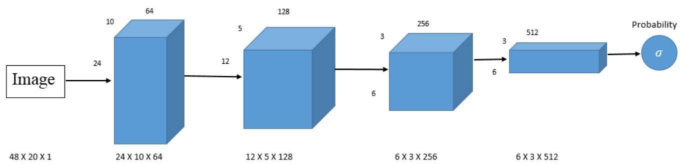

Figure 1.2 : Discriminator and Strided Convolution

PCA Generation - Figure 1.3 shows our approach for PCA generation. The reason behind using PPCA instead of using original image is to reduce dimensionality of image. In this approach, we have created a data-set which contains only the 20 principal component of each image from the data-set.Here, we have trained our model using PPCA (probabilistic principal component) instead of using original image. Output of generator is synthesized PCA.

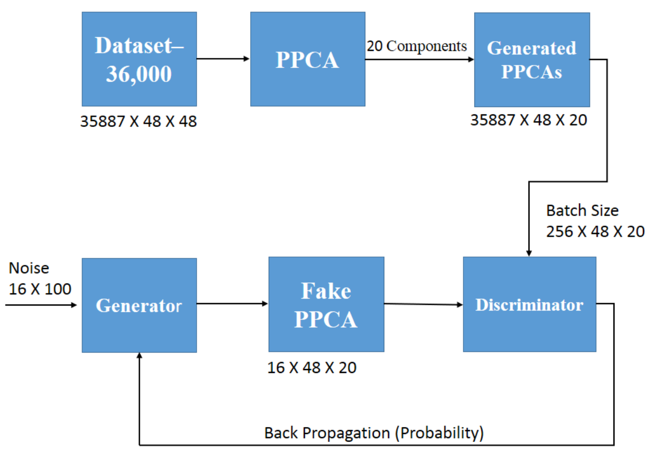

Figure 1.3 : Generative Adversarial Network For PCA Generation

Image Generation - In this aproach we obtain PPCA from given input image. From obtained PPCA we extract PCA(W) and latent variable(M X_n). Both this component are provided as input to two different discriminator. Generator will generate W and M X_n from noise and recreates image.[1]

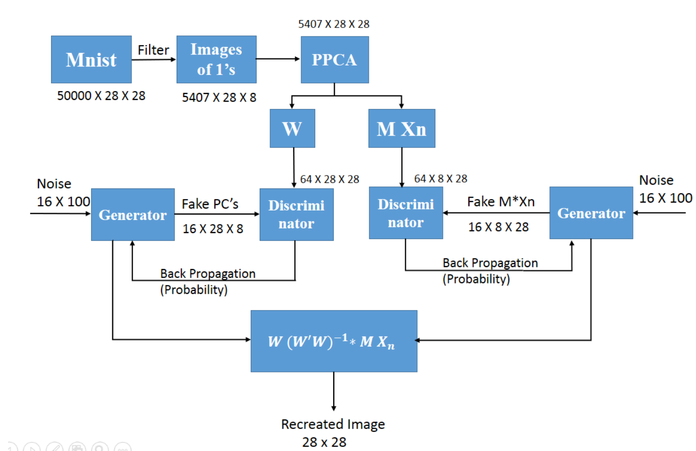

Figure 1.4 : Generative Adversarial Network For Image Generation

III. Results

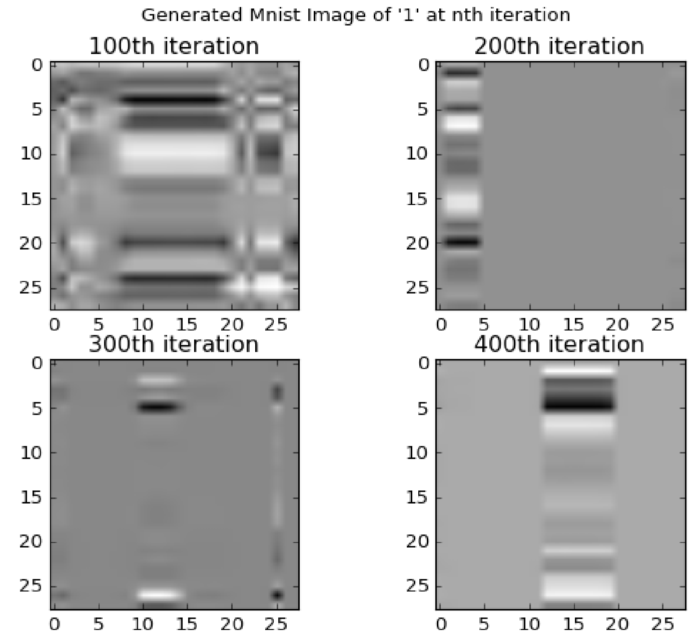

Figure 1.5 : Generated 1's image from MNIST dataset

No. of Iteration	Norms
100	1564.72
200	22734.41
300	1203.46
400	1075909.60

Table - 1. Calculated Norms (MNIST)

In the above table, we have calculated norm between generated image of '1' and the original image of '1' of MNIST dataset for different number of iterations.

No. of Iteration	Norms
1000	50.05
5000	50.06
8000	50.02

Table - 2. Calculated Norms (FER2013)

IV. Conclusion

In PCA generation we were sucessful in generating PCA on FER2013 data-set. The technique to verify this PCA is not known to us. Also, image generation using PPCA did not gave a satisfactory output image. For generating relavant images we have to train our model with different features, which is not possible using PPCA.

References

[1] Tipping, M. E., & Bishop, C. M. (1999). Probabilistic principal component analysis. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 61(3), 611-622.

[2] Rowel Atienza, (2017, March 30) GAN by Example using Keras on Tensorflow Backend Retrieved from http://www.medium.com

[3] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).