# Generative Model for Real Image Generation

Group : 11
Jay Dangar, Jayveer Gadhvi,
Ravi Patel, Madhav Chavda
School of Engineering and Applied Science

***Abstract*—In this Paper, we are going to throw some light on concept related to Convolutional Neural networks and how it works?, What Generative Model is and how it works in our Application.**

***Index Terms* — Convolutional Neural Network,Estimation Maximization,Probabilistic Principle component Analysis,Generative Adversial Network,ReLU,Pooling,Minimax Game,Generators and Discriminators.**

## I. INTRODUCTION

THIS paper describe the upcoming hot topic in the world of computer vision and that is the Generative Model and Neural Network. In the first part we will give you the overview about What GAN is, and How it works. Then we will give you brief theory about the CNN and their working layers. After that we will tell you our approach towards making the model of GAN with Algorithms like PPCA using EM and also using CNN for the better classification. At the end of this paper we will give you some interesting results that will realize you how important GAN is in Image Generation and also for other applications. At the end of this paper we will have some useful references for better understanding of this paper.

## II. WHAT IS GANS?

GANs are Generative models based on Neural network which automatically generates new images from given Dataset. The Generative Model basically have 2 Modules, the first one is Generators and Second one is Discriminators.

You can think generator as Fraud who generates fake money and discriminator's job is to discriminate fake money generated by generator from original money. By the time, both the generator and discriminator learns from their respective inputs and gets better at their respective fields. This learning process ends when generator becomes so better that discriminator cannot discriminate the difference between real and fake money. That means generator has generated exactly same looking image from any random image.

This Generator and Discriminator's Idea is called minimax game.

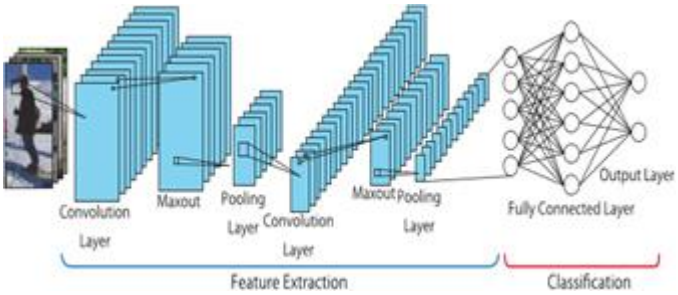
## III. HOW GAN WORKS?

Discriminator Module contains Convolution Neural Network works, which gives a scalar value which is the difference between the original image and generated image and also by the time this difference(Cost) will decrease as the generator learns. This process is called downsampling.

In Generator Module Inverse of the Discriminator's work is done. That means it will do upsampling.

## IV. HOW CNN WORKS?

In CNN basically, there are mainly 4 layers working.




1.Convolution Layer :

- In this Layer Convolution happens between different types of kernels and our original images and that gives us different types of filters.
- This helps in detecting edges and other lower level detections in images.

2.ReLU Layer :

- After Convolution, ReLU will convert negative values of pixels in 0 values and that will remove blck pixels from image.

3.Pooling Layer :

- In this layer, maximum value is selected in order to downsampling and this maximum value is selected based on the windows size.

4.Fully Connected Layer.

- This is the final layer in which the loss is calculated and accordingly backpropogation happens.

## V. HOW OUR GAN MODEL WORKS?

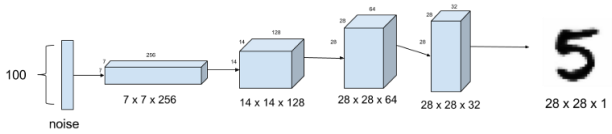
In our GAN model we have used MNIST Dataset and from that data set, we have taken 980 images of different types of zeros. Then, we have converted this images into vectors. We got 980 images and from each of these images we got 784*1 vectors and by appending these all vectors we will get 784*980 matrix.

From this matrix we will get Principle Components based on Estimation Maximization With PPCA algorithm and these PCA's will be given to our Generative model to generate new set of images. This Generative Model then play minimax game between generator and discriminator. Discriminator fool and generator gets better at each iterations. After sufficient amount of training the discriminator will give you the Output of an image which looks like generated zero.

Block Diagram of GAN Model


*A. Generator*

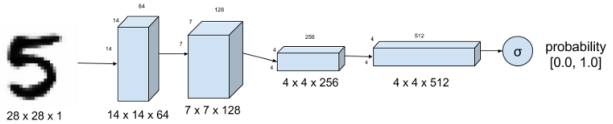
Generator model is used to synthesize fake image. You can think of the generator as being a kind of reverse ConvNet. With CNNs, the goal is to transform a 2 or 3 dimensional matrix of pixel values into a single probability. A generator, however, seeks to take a d-dimensional noise vector and upsample it to become a 28 x 28 image. This upsampling is done through a convolutional transpose (or deconvolution) layer. 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 an activation function after each layer and batch normalization to stabilize learning.




Generator and Deconvolution

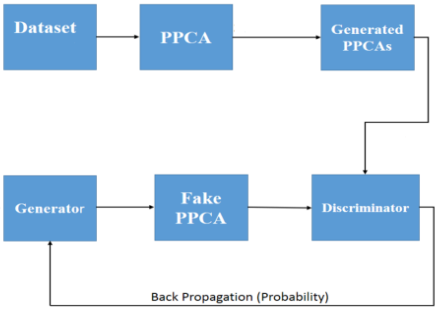
*B. Discriminator*

Discriminator part of the model tells the how much synthesized image is related to real image. In our case we define a CNN classifier function that takes in an image (of size 28 x 28 x 1) as input. The output will be a single scalar number activation that describes whether or not the input image is real or not. In a simpler way it is a reverse process of generator. We have used strided convolution for down sampling and leaky ReLu as an activation function in each CNN layer. The output will be a single scalar number activation that describes whether or not the input image is real or not.



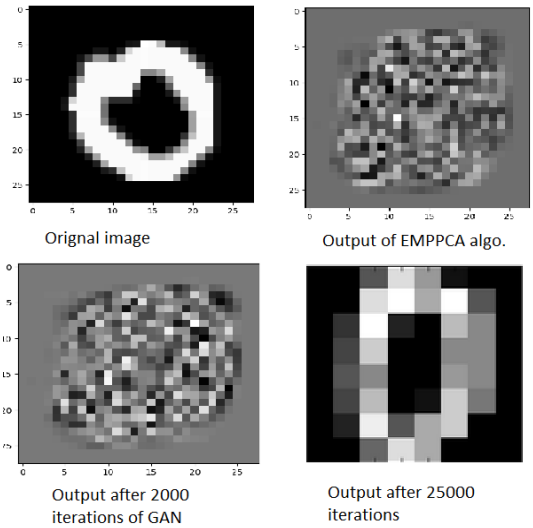

Discriminator and Convolution

**PCA Generation** – Below figure shows our approach of the PCA generation. The main reason of using the PPCA is because of its reduced dimensionality from the original image. And because of this we have trained our model using PPCA instead of using original image.




## VI. Simulation Results

Here are the simulation results of our project. We have taken the MINIST dataset for the project.




First is original image , second is image generated using EMPPCA algorithm after 25 iterations, third is output after 2000 iter of GAN, forth is the image generated after 25000 iters.

## VII. Conclusion

we learned how two networks were able to play a minimax game in order to capture the data distribution of the MNIST digits and generate similar looking samples. With applications in video frame prediction, text-image mappings, and more, GANs are definitely the best topic in deep learning.

## VIII. References


1. https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/
2. https://adeshpande3.github.io/adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html
   http://blog.evjang.com/2016/06/generative-adversarial-nets-in.html
3. https://github.com/myleott/mnist_png