

# Face Generation using Deep Convolutional Generative Adversarial Neural Network

Devaki P.<sup>1\*</sup>, Prasanna Kumar, C.B.<sup>2</sup>, Kaviraj S<sup>3</sup> and Ramprasath A<sup>4</sup>

<sup>1\*</sup>Professor, <sup>2,3,4</sup>UG Scholars

<sup>1,2,3,4</sup>Department of CSE, Kumaraguru College of Technology, Coimbatore, India

## ABSTRACT

Due to the huge availability of data, it is difficult to classify/process images at a higher speed and accuracy. The first technique was in the field of computer vision and it used image data for face recognition and detection of an object from the image but later Convolutional Neural Networks (CNNs) took place. CNNs are used for feature detections by looking at the image and try to check if certain features are present in the image and then it classifies the image accordingly. Acquiring and processing the dataset for the Machine learning technique is one of the time-consuming processes, so Generative Adversarial Neural Network (GAN) are introduced. GAN typically work with image dataset but they are difficult to train. This paper explores the potential of GAN to generate realistic images. Deep Convolutional Generative Adversarial Networks (DCGAN) is used to generate new images that are not in the dataset. DCGAN has a great success in generating the new images. MNIST (Modified National Institute of Standards and Technology dataset) contains images of handwritten digits dataset and CelebA dataset contains images of celebrities are used, performing the adversarial learning on it and try to generate new images as same as the MNIST and CelebA datasets.

**KEY WORDS:** CELEBA, CONVOLUTIONAL NEURAL NETWORKS, DEEP CONVOLUTIONAL ADVERSARIAL NEURAL NETWORKS, MNIST.

## INTRODUCTION

Machine directed algorithms in the form of Machine Learning were built known as Self-Organizing Networks (SON) (Hughes et al., 2019). The SON was used to train highly co-relevant data. With the scarcity of real labeled data alarming a concern, a need for a new method arose where data was not pure, not easy to segregate, and unsupervised. Hence, a data-driven (model-free) approach built using two interconnected Artificial Neural Networks called Generative Adversarial Networks which filled

the setbacks in SON. This method was able to generate synthetic data that was created by augmenting real data with realistic synthetic data. GANs were able to develop new datasets on historical data without specifying a model or fitting probability distribution.

With GANs being a promising technology in the field of machine learning approach, it played a significant role in non-labeled data (i.e.) data where the details were not sufficient. Thus, the usage of GANs became more popular in the field of semi-supervised and unsupervised learning. (Li et al., 2018) shows that the relation of specific selections of high-level textual representations in linear models was substantial, and they failed to capture complex distributions. An observation of limitation in scalability and inference accuracy proved to be a critical factor in the failure of Gaussian and Non-parametric density models. Therefore, for the joint optimization of the model and variational parameters, stochastic inference algorithms were used. GANs provided an improvement in

## ARTICLE INFORMATION

\*Corresponding Author: [devaki.p.cse@kct.ac.in](mailto:devaki.p.cse@kct.ac.in)

Received 5th Dec 2020 Accepted after revision 27th Nov 2020

Print ISSN: 0974-6455 Online ISSN: 2321-4007 CODEN: BBRBCA

Thomson Reuters ISI Web of Science Clarivate Analytics USA and Crossref Indexed Journal



NAAS Journal Score 2020 (4.31) SJIF: 2020 (7.728)

A Society of Science and Nature Publication,  
Bhopal India 2020. All rights reserved.

Online Contents Available at: <http://www.bbrc.in/>

Doi: <http://dx.doi.org/10.21786/bbrc/13.11/5>

the learning of the classifiers by incorporating adversarial objectives that made it more robust and efficient.

(Ledig et al., 2017) Application of GAN's plays a role in converting the image from Low Resolution (LR) image to High Resolution (HR) image. High MSE results in Peak Signal to Noise Ratio (PSNR), but they often lack high-frequency details and perceptual unsatisfying. Structural Similarity Index (SSIM) and PSNR are the metrics used to determine the quality of the image, and MSE is the error calculation commonly used in encoding and decoding of the image. The MSE based solution appears overly smooth image due to the pixel-wise average of possible solutions in the pixel space. At the same time, GAN drives the reconstruction towards the original image manifold producing perceptually more convincing solutions. Since MSE struggles to handle the uncertainty in recovering the high-frequency details such as texture, a loss function based on the Euclidean distance between feature maps are used to provide perceptually more satisfying result.

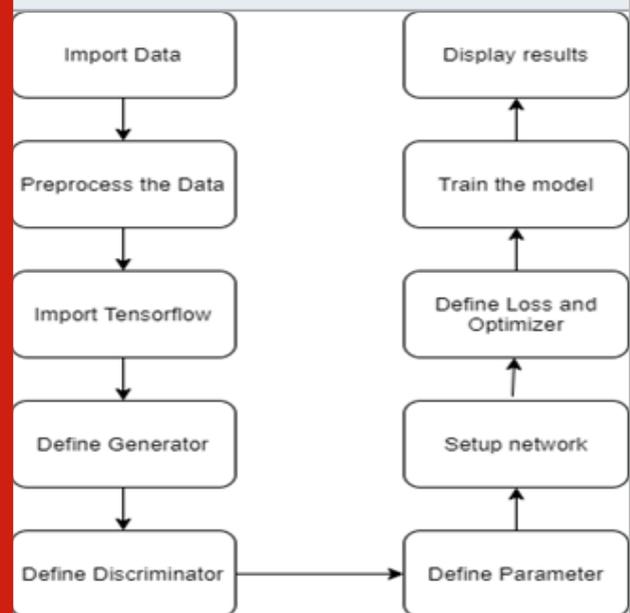
In order to obtain more efficient and clean images, batch-normalization of layers and ParametricReLU, which uses LeakyReLU as activation function, are used to obtain a sigmoid function in the two deep neural networks used respectively. This method is not optimized for video and can be used only for images. A special GAN called the Deep Convolutional Generative Adversarial Network (DCGAN) finds its usage in unsupervised data. The DCGAN is a combination of two deep artificial neural networks called the Generator and Discriminator. (Liu X et al.,2018) Uses "Age-DCGAN," where input is the image of a person's face, which is passed to the encoder and learn its personality and age features. The Discriminator network uses conditions to generate the corresponding aging face and arrive at information that shall act as base values in determining synthetic images. Thereby, Discriminator imposes on the Generator to discriminate input images and generated images until they are real. The method makes use of two sub-encoders to separate personality and age-features in simulating and predicting the aging of the human face. Results show that using two vectors in the Generator shall generate a more accurate and realistic image to match the input samples.

DCGAN find their scope extending to a greater area when it comes to images. (Arora et al.,2019) Inpainting is filling the neighboring pixels to remove an object from the image or removing a lousy effect like the red-eye effect or removing the watermark. However, in outpainting, there is a lot to explore, and the implementation of new techniques are needed. The sole aim of outpainting is to produce millions of new pixels needed to generate a similar one to the original image. DCGAN helps to outpaint the image recursively up to a greater extent, by obtaining larger extended realistic images. By giving contextual and perceptual information to the algorithm, the generator can generate the images by repeating Recursive painting (up to a maximum of 3 times) to output more clear and high-quality images that have a lesser pixel loss.

(Nataraj et al.,2019) Detecting Fake images using Co-occurrence matrices is GAN based technique to identify fake images such as Deep fakes, Image-to-Image transitions. It is inspired by classical Steganalysis, to detect manipulation of images. In this, Co-occurrence matrices are extracted on 3-color channels of a pixel domain and trained using CNN. The nature of ML used is similar to a Game Theory optimization problem that imbibes min-max to compete between generator and discriminator. Initially, using image residuals, these matrices were computed, passing it through several filters to obtain differences over which a classifier was built to check the authenticity of the image. However, authors provide a method of building convolutional layers over a neural network by using pixels of RGB from an image, thereby allowing the network to learn features from the matrix itself. The optimizer used is of stochastic gradient descent.

With all said, for further improvement (Salimans et.al.,2016) says that training GANs requires finding NASH equilibrium of a non-convex game with continuous, high dimensional parameters. Even though gradient descents provide the best outcomes to the present situations, they fail to converge sometimes. In order to improvise techniques such as Feature matching, Minibatch normalization, Historical averaging, Virtual batch normalization are some of the techniques the author suggests to improve GANs. Though GANs have shown promising development, it still lacks proper metrics and unstable training.

Figure 1: System Flow Diagram



## MATERIAL AND METHODS

**Vanilla Gans:** These are the simplest of GANs. The generator and discriminator are only multi-layer perceptron. The Vanilla GANs function using stochastic

gradient descent algorithms. From (Gauthier Jon 2014) , by considering four arbitrary noise samples, images are mapped in rows, and the number of epochs it takes to generate a proper image is represented in columns and observed a high of 131 epochs, which is a considerably very high cost. They observed Mosaic patterns and discolorization in raw images and struggle in learning data when there is an overlap. Therefore, Vanilla GANs failed to overfit the training data.

**Conditional GAN:** These are the GANs with conditional parameters. From (Gauthier Jon 2014), observed that by considering a conditional parameter 'y' as face attributes in the generator, they train the network using the image data as 'x'. Only proper visual attributes are considered as attributes and the remaining attributes as noise. Training the network, they found considerable changes in similar faces, but it still failed to overfit the data. Though the system observed strong positive correlations with the generator, the cost of generator is improving overtime, but the discriminator keeps on failing. These methods led us to use a more efficient deep learning technique called DCGANs proposed in this paper.

**Proposed Solution:** Figure 1 is the proposed System Flow. The proposed solution seeks to generate artificial images using Deep Convolutional Generative Adversarial Networks (DCGAN). Having performed well with unlabeled (unsupervised) data in the past, DCGANs make a perfect option to this scenario. The model was run against two datasets:

1. MNIST dataset containing images of handwritten digits of 60,000 images
2. CelebA dataset containing 200,000 images of celebrity faces.

The designed solution is in a way that it adapts both the datasets with only the number of channels varying as 1(L) and 3(RGB), respectively. The system is constructed using Tensorflow v1.0.

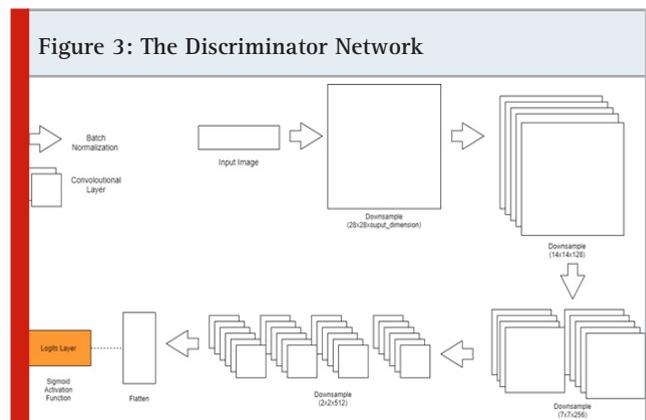
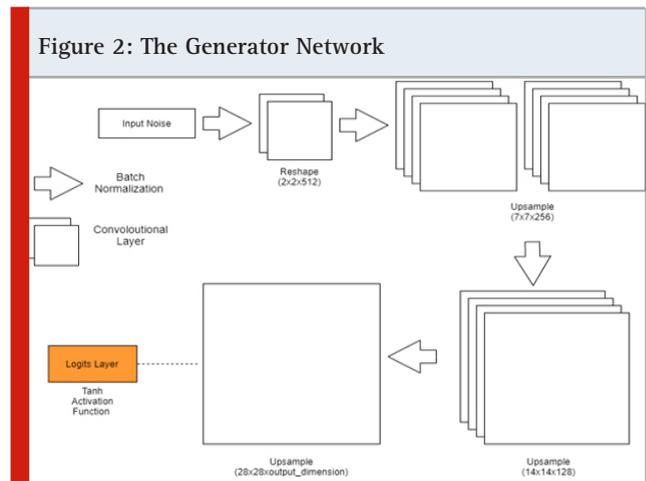
**Module 1(Generator):** In the generator network Figure 2, along with every input, a random input (noise generated from Gaussian distribution) is added to scramble the original image, thereby generating a new image. This is performed for all the images provided as input. The generator also performs up sampling, where it combines a broader set of smaller images together to form a single large image. In this system, 2 hidden layers are present. Xavier initializer is used to initialize weights to make sure that neuron activation functions do not happen in zero or dead regions. By doing batch normalization in every layer for standardization, it also decreases computation cost by reducing the number of epochs. 'Tanh' is used as the activation function at the logits layer (Output layer of the network) as they work well along with Xavier initializers.

$$\text{Tanh}(x) = 2/1+e^{-2x} - 1.$$

Tanh activation function is preferred over sigmoid

functions because the gradients get stronger and steeper over time.

**Module 2: (Discriminator):** The Discriminator network Figure 3 is the reverse of the generator structure. The Discriminator performs down sampling (i.e.) breaks down the large image received due to up sampling in the generator into smaller fragments. Similarly, 2 hidden layers are there in the Discriminator and 'Sigmoid function' as the activation function in the output layer to determine whether the generated image is real or fake.



**Figure 4: Sigmoid Cross-entropy loss**

$$H(y, \hat{y}) = -\frac{1}{M} \sum_{j=1}^M [y_j \log \hat{y}_j + (1 - y_j) \log(1 - \hat{y}_j)]$$

$$\hat{y}_j = \sigma(z_j) = \frac{1}{1+e^{-z_j}} \text{ for score } z_j$$

**Module 3: ( Loss And Optimization):** Generator and Discriminator loss computed are using Sigmoid cross entropy Figure 4 from the computed logits.

In Figure 4, M is the number of classes. z subscript j is the raw output of the convolutional network. yhat subscript

$j$  is the predicted probabilities calculated using sigmoid function applied to the  $z_j$ .  $y_j$  is the ground truth of the input images.

The implementation of cross entropy is done with reference to <https://github.com/carpedm20/DCGAN-tensorflow/blob/master/ops.py#L35> Adam Optimizer, is used to provide better individual learning rates for each parameter within the network. The tolerance level of loss of generator is ensured that is lesser than the discriminator loss or close to 0.

## RESULTS

By using DCGAN, after every 10 batches, the generator and discriminator loss values is observed. The generator output is also observed for every 100 batches. With batch size as 64, the MNIST dataset was trained with two learning rates 0.001 and 0.0005. Figure 5 and Figure 6 shows the outputs for the learning rates 0.001 and 0.0005 respectively. The network is trained for 2 epochs with each learning rates.

Figure 5: Output for MNIST dataset after 2 epoch with learning rate of 0.001

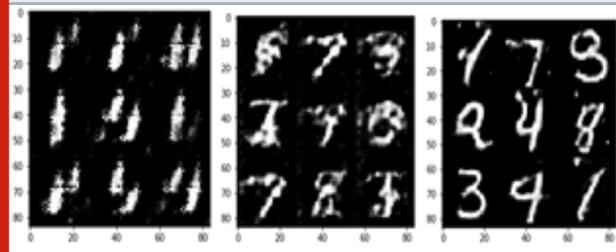
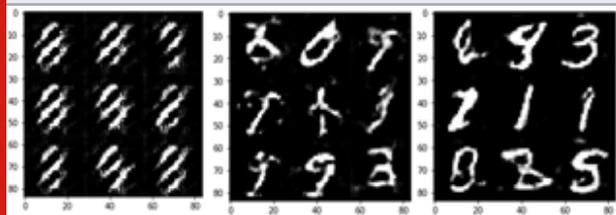


Figure 6: Output for MNIST dataset after 2 epoch with learning rate of 0.0005

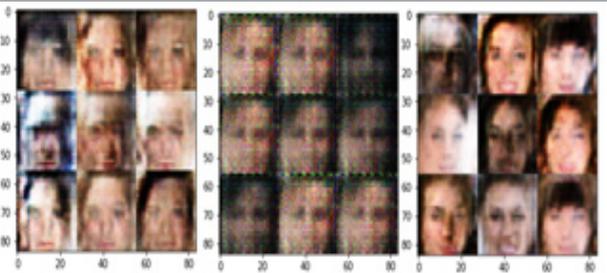


Also, the results for the CelebA dataset, which contains the images of faces is observed by setting the batch size = 64 and learning rates at 0.0005. Figure 7 is the output obtained from training the network.

## CONCLUSION

By training for only two epochs, we were only able to attain the current output efficiency. By increasing the

Figure 7: Output for CelebA dataset after 2 epoch with learning rate of 0.0005



number of epochs and optimizing the neural layers and learning rate, better results could be obtained. By improving the designed network, this model shall prove to be efficient in unsupervised learning problems and can achieve state-of-the-art results with similar datasets as well.

## REFERENCES

- Arora H Jain S Anand S and Rajpoot D S (2019) Augmentation of Images through DCGANs 12th International Conference on Contemporary Computing (IC3) Noida India pp. 1-6.
- Gauthier Jon (2014) Conditional generative adversarial nets for convolutional face generation Class Project for Stanford CS231N: Convolutional Neural Networks for Visual Recognition, Winter semester 2014.5 (2014): 2
- Hughes B Bothe S Farooq H and Imran A (2019) Generative Adversarial Learning for Machine Learning empowered Self Organizing 5G Networks International Conference on Computing, Networking and Communications (ICNC) Honolulu HI USA pp. 282-286.
- Ledig C et al. (2017) Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Honolulu HI pp. 105-114.
- Li T Liu X and Su S (2018) Semi-supervised Text Regression with Conditional Generative Adversarial Networks IEEE International Conference on Big Data (Big Data), Seattle WA USA pp. 5375-5377.
- Liu X Xie C Kuang H and Ma X (2018) Face Aging Simulation with Deep Convolutional Generative Adversarial Networks 10th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA) Changsha pp. 220-224.
- Nataraj L et al.(2019) Detecting GAN generated Fake Images using Co-occurrence Matrices
- Salimans Tim Ian J et.al., (2016) Improved Techniques for Training GANs NIPS.