

Image Generation for Real Time Application Using DCGAN (Deep Convolutional Generative Adversarial Neural Network)

Dr.V.Vijeya Kaveri ¹, V.Meenakshi ², Deepan.T ³, Dharnish.C.M ⁴, Haarish.S.L ⁵

¹Professor , Department of CSE ,Sri Krishna College of Engineering and Technology, Coimbatore,India.

²AP, Department of EEE , Sathyabama Institute of Science and Technology, Chennai, India.

³Department of CSE ,Sri Krishna College of Engineering and Technology, Coimbatore,India.

⁴Department of CSE ,Sri Krishna College of Engineering and Technology, Coimbatore,India.

⁵Department of CSE ,Sri Krishna College of Engineering and Technology, Coimbatore,India.

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 10 May 2021

ABSTRACT

As the technology keeps developing the unimaginable possibilities keep happening. And it leads to easy use of our daily life. In image processing when the CNNs came to our life it makes the world to turn around and makes the human work easier in all organization. Convolutional Neural Network were mainly used in computer vision, mainly in face recognition, image classification, action recognition, and document analysis, but these gets difficult when comes to dataset. Gathering dataset for machine learning is time consuming operation, at that point the new technique called GAN were introduced. It can predict that whether the image is real or not, which is a next level improvement of machine learning techniques. Our aim is to improve the creativity of the machine and generate different type of images which will be useful in the fields like animation and designing. Here in our paper, we will use the Deep Convolutional Generative Adversarial Networks (DCGAN) where it will be used to generate new images that are not in the dataset. And it's been a huge success in terms of creating new images. MNIST dataset and Anime dataset are used here, by using the DCGAN in it and try to create pictures that are similar to the datasets.

Keywords: Convolutional Neural Network, MNIST, Deep Convolutional Generative Adversarial Neural Network, Conditional Generative Adversarial Neural Network.

I. INTRODUCTION

GANs were introduced in 2014 by Goodfellow as an impressive technology in the field of machine learning techniques, and they played a key role in non-labeled data. As a result, the use of GANs in semi-supervised and unsupervised learning has grown in popularity.[2]

The GAN can be explained in a simpler way and it is structurally inspired by two-person game. The generator's goal is to understand and capture the potential distribution in the current data samples as possible before generating new data samples. Discriminator is a binary classifier whose aim is to decide if the input data comes from the generator or from the actual data. The two players must continually develop their ability to generate and discriminate in order to win the game. The aim is to reach a Nash equilibrium between the two sides so that the generator can estimate the data sample distribution.[1]

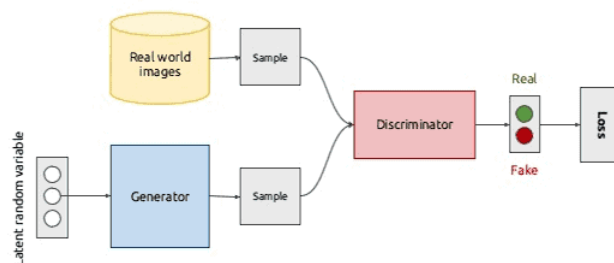


Figure 1: GAN Architecture

As the generator and discriminator do their job the optimization process of Generative Adversarial Network is to minimize the process, the optimization main role is to reach the Nash equilibrium. When it reaches only it is presumed that the generator has identified the distribution of real samples. In Discriminator the sigmoid output is a scalar value representing the possibility of the image being true (0.0 is certainly fake, 1.0 is certainly real, anything in between is a grey area). For downsampling, strided convolution is used. Each CNN layer uses a leaky ReLU as an activation mechanism. An exclusion between layers between 0.4 and 0.7 prevents overfitting and

memorization. In generator it generates fake images. Transposed convolution, the inverse of convolution, is used to create the fake image from a 100-dimensional noise. In the first three layers, upsampling is used; in the layers between, batch normalisation stabilises learning, and the activation function is ReLU. The output of Sigmoid final layer is where the fake image gets created. Overfitting is prevented by a dropout of 0.3 to 0.5 at the first layer. A spectacular GAN called Deep Convolutional Generative Adversarial Network (DCGAN) is being used in unsupervised data to achieve quite efficient images. The DCGAN is a deep artificial neural network that combines the Generator and Discriminator.

II. RELATED WORK

GENERATIVE ADVERSARIAL NETWORK (GAN):

Initially, we gather a large number of fake images created by several GANs, referred to as generative models. Based on the proposed contrastive loss, real images were used to learn the jointly discriminative features X1. Then a discriminator X2 will be added to the X1 to help differentiate fake images. Where in the test phase it will be easy to find whether it is fake or not by X1 and X2. But when we compare to our proposed model here the image dataset training will be hard and it fails to overfit the training data.[3]

CONDITIONAL GAN (CGAN):

CGANs are permitted to create images with specific attributes or conditions. Here the generator and discriminator will have extra conditioning input information. A new layer with the values of one hot encoded image which will be inserted. Here the Discriminator in a Conditional GAN does not learn to distinguish between the different classes. It learns to accept only valid, matching pairs while rejecting mismatched pairs and pairs with a fake context. It detected notable results in similar faces while training the network, but it still failed to overfit the data. When the generator keeps developing, but the discriminator continues to fail. When comparing to this approach, DCGAN will be more effective deep learning technique.[4]

III. PROPOSED SYSTEM

Below figure shows the process of proposed system. Here Deep Convolutional Generative Adversarial Networks are used in the proposed solution to produce artificial (which look alike original image) photos. DCGANs are an excellent choice for this scenario because they have previously performed well with unlabelled data. Two datasets were used to test the model: MNIST Dataset which contains 60,000 handwritten images, and Anime image dataset contains 92,300 Anime faces.

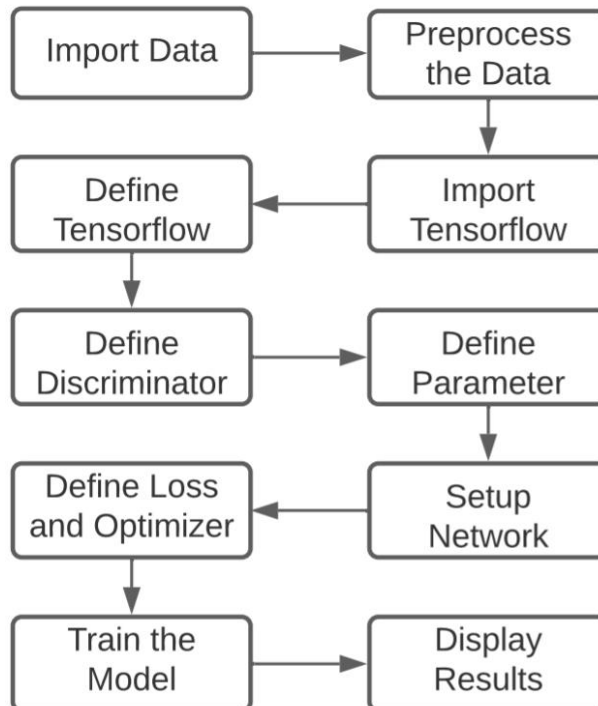


Figure 2: - Process Flowchart

A. ARCHITECTURE

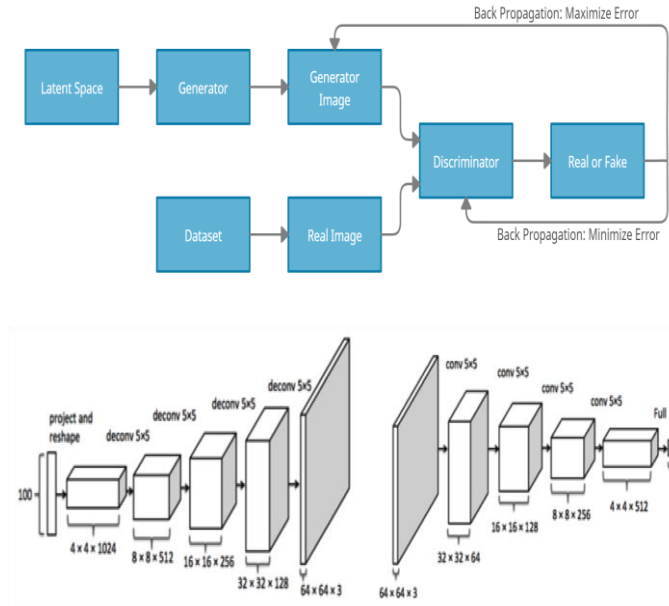


Figure 3: Architecture of DCGAN (Left-Generator, Right- Discriminator)

B. GENERATOR

In Figure 4, a random input (noise generated) is applied to each input to scramble the original image and generate a new image. This is done with all of the photos that are presented as data. The generator also does upsampling, which is the process of combining a larger number of smaller images into a single large image. There are two secret layers in this technique. To ensure that neuron activation functions do not occur in zero or dead areas, the Xavier initializer is used to initialise weights.

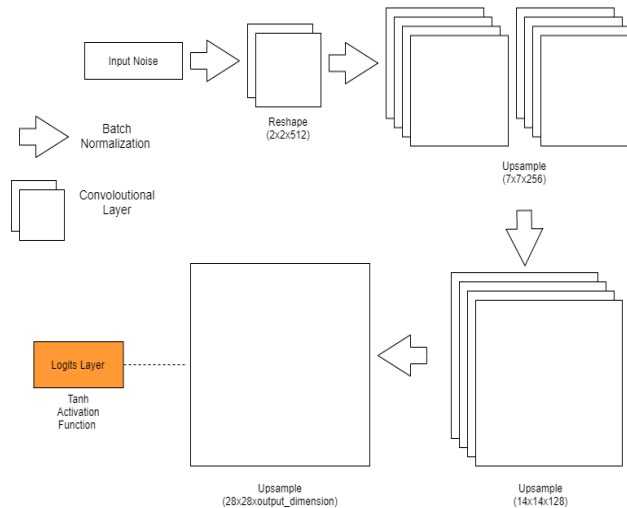


Figure 4: Generator Implementation

By performing batch normalisation in each layer for standardisation, the number of epochs is reduced, lowering computation costs. Since they fit well with Xavier initializers, 'Tanh' is used as the activation feature at the logits layer (output layer of the network).

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

Since the gradients get higher and steeper over time, the Tanh activation function is favored over sigmoid functions.

C. DISCRIMINATOR

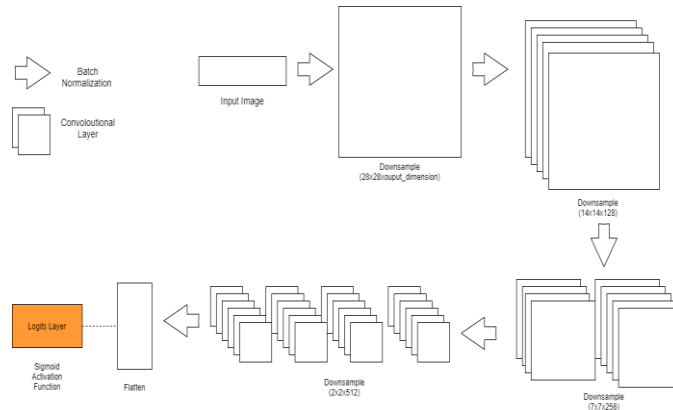
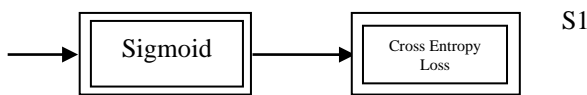


Figure 5: Discriminator Implementation

The Discriminator network above image is the opposite of the Generator. The Discriminator downsamples (divides) the large image obtained from the generator due to upsampling into smaller pieces. To decide whether the created image is real or false, the Discriminator has two hidden layers and uses the ‘Sigmoid function’ as an activation function in the output layer.

D. LOSS AND OPTIMIZATION

The Sigmoid activation with a loss of Cross-Entropy is Sigmoid cross entropy. The generator and discriminator losses are computed using the Sigmoid cross entropy from the measured logits.



$$CE = -t_1 \log(f(s_1)) - (1 - t_1) \log(1 - f(s_1))$$

$$f(s_i) = \frac{1}{1 + e^{-s_i}}$$

Figure 6: Sigmoid Cross Entropy

Where $f(s_i)$ is the i^{th} scalar value in the model output. It is independent for each vector component. Other component values have no effect on the loss computed for each CNN output vector component. It is also called as Binary Cross-Entropy Loss because it sets up binary classification problem.

The Adam is the optimization algorithm which will update its network weights iteratively depending on its training data. The generator loss tolerance level is set to be less than or equal to the discriminator loss.

IV. DATASET

In our Model we have used two dataset which is MNIST and Anime face. (Fig 7).



Figure 7: Sample Image for MNIST and Anime

There are 50,000 handwritten pictures in the MNIST (Modified National Institute of Standards and Technology) dataset. This dataset is still being used to analyze classification algorithms. MNIST remains a reliable platform for developers and learners alike as modern machine learning methods arise. Then Anime Face dataset which consists of 92,300 images. The image is in 256*256-pixel JPEG format. The different between the dataset are number of channels as MNIST is Greyscale 1(L) and Anime Face is 3(RGB).

V. RESULT

In our Model we have used both MNIST and Anime Dataset. Here we have calculated the generator and discriminator loss for every 10 batches. And the generator final output is assessed for 200 batches each.

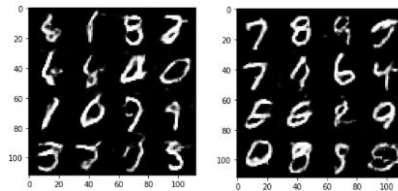


Figure 8: Output for MNIST Dataset

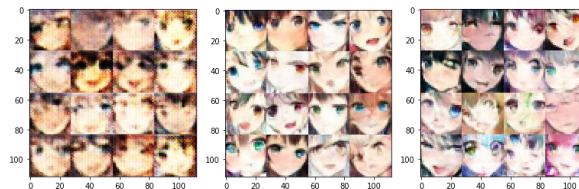


Figure 9: Output for Anime dataset

And our model is trained for 10 epochs with each learning rate as 0.001 and batch size as 128 for both MNIST and Anime datasets. The Fig 8 Shows the output generated for MNIST Dataset and Fig 9 shows the output generated for Anime Dataset.

CONCLUSION

Here we have trained our model for less epochs and we have only able to reach this efficiency. Better performance could be achieved by raising the size of epochs and improving the neural layers and learning rate. Which will be same as the original image.

VI. FUTURE WORK

For the future work we have planned to develop a UI to provide service for the customers where the customers have to give sample class of images for the discriminator. so, the user will be able to generate similar images.

VII. REFERENCES

- [1] "A Review: Generative Adversarial Networks", Liang Gonog and Yimin Zhou. Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China.
- [2]. "Generative adversarial nets," Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Advances in Neural Information Processing Systems 27, Montreal, Quebec, Canada, 2014.
- [3]. "Learning to Detect Fake Face Images in the Wild", Chih-Chung Hsu# , Chia-Yen Lee* , Yi-Xiu Zhuang. Department of Management Information Systems, National Pingtung University of Science and Technology, 1, Shuefu Road, Neipu, Pingtung 91201, Taiwan,2018.
- [4]. V.Vijeya Kaveri, V.Maheswari , "Mining Social Data to Identifying User Behavior in Medhelp Forum on Health-Related Topics ", International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-2S3, July 2019, PP.No.1306-1310 .
- [5]. "MULTI-VIEW FRAME RECONSTRUCTION WITH CONDITIONAL GAN", Tahmida Mahmud Mohammad Billah Amit K. Roy-Chowdhury. University of California, Riverside,2018.
- [6].Kaveri, V. Vijeya, and V. Maheswari. "A framework for recommending health-related topics based on topic modeling in conversational data (Twitter)." Cluster Computing 22, no. 5 (2019): 10963-10968.
- [7]. "Generative Adversarial Networks: Introduction and Outlook", Kurfeng Wang, Member, IEEE, Chao Gou, Yanjie Duan, Yilun Lin, Xinhua Zheng, and Fei-Yue Wang, Fellow,2017.
- [8]. "DCGAN Based Data Generation for Process Monitoring", Yu Du, Wenqian Zhang, Jing Wang, Haiyan Wu. College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, China,2019.
- [9]. "Semi-Supervised Learning with Generative Adversarial Networks", Augustus Odena, 2016.