International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-6, March 2020

Enhanced Unsupervised Image Generation using GAN based Convolutional Nets

M Rama Bai, J Sreedevi, B Pragna

Abstract: Generative Adversarial Networks (GANs) use deep learning methods like neural nets for generative modeling. Neural style transferring of images and facial character generation of anime images are previously implemented by applying GAN methods but were not successful in giving a promising output. In this work, Image Processing is applied on the datasets in the mode along with the training of GAN system. The problem of applying GAN to generate specific images is addressed by using a clean and problem specific dataset for anime facial character generation. Modeling is done by applying Convolutional Neural Nets, GANs empirically. Neural style transfer, Automatic Anime characters are generated with high-resolution, and this model tackles the limitations by progressively increasing the resolution of both generated images and structural conditions during training. This model can be used to develop unique anime characters or the image generated can be used as inspiration by artists and graphic designers, can be used as filters in famous apps such as snapchat for style transferring. With different evaluations and result analysis, it is observed that this model is a stable and high-quality model.

***Keywords* : Automated Character Generation, DC GAN, GANs, Generative Adversarial Networks, Image Generation, Image Processing, Neural Style Transfer, Neural Nets.**

I. INTRODUCTION

A powerful class of neural networks was introduced by Ian J. Goodfellow [1], which can be used for unsupervised learning in 2014. These networks are Generative Adversarial Networks (GANs) which typically perform analysis over the dataset. Analysis is done on the given real data and features are noted at every step in the model by comprising two different neural network models. This analysis also includes evaluating quantitative and qualitative methods, analyzing variations in data, pattern changes of dataset by compete with each other. Generating automated images that are unique is difficult using the convolutional neural networks. Other neural nets, such as convolutional neural networks misclassify things when minimal amount of random noise is added to the input data. Further, these networks showed adversary giving wrong predictions with a higher confidence as noise is added than the confidence for correct predictions. Most network models are trained on less quantity of data, due to which when noise is added, the system is manipulated into giving wrong results and thus, over fitting the data. These models have a linear input and output mapping, due to which seemingly linear boundaries of separation lead to misclassification of outputs as the boundaries are composed of linearities. And a minute variation in feature space gives varied results and thus, incorrect predictions. These drawbacks in conventional models can be reduced by developing Generative Adversarial Networks. In this model, given the images, a style image and a content image, it generates new image combining the style image onto content image. Automated Anime Characters are generated using the GAN's.

II. RELATED WORKS

Neural Style Transfer is to migrate the semantic content of one image onto different styles, was done by Gatys et al using Convolutional Neural Networks (CNN). Image-to-Image Translations are done by creating a synthetic image of an existing image with modifications specific to content. These translations require training to be done on large amount of paired dataset, which are difficult to prepare and time-consuming. Previous works on Image-to Image Translations involved paired examples for training the model. Automatic facial images have been generated previously using GANs but these models do not generate facial features accurately.

GANs are applied to image generation, image transfer, super-resolution etc. Metz et al. proposed to render discriminator omniscient wherever applicable in order to train stably. LS-GAN learns loss functions to separate generated data from original data to avoid model collapse.

Manuscript received on March 15, 2020.
Revised Manuscript received on March 24, 2020.
Manuscript published on March 30, 2020.

* Correspondence Author

Dr M Rama Bai, Professor, Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad, India.

Mrs. J. Sreedevi, Assistant Professor**,** Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad, India

Ms. B. Pragna, Student, Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Hyderabad, India.

Other GAN variants have been proposed for generating images. To generate images from latent vector input space, Radford et al. applied convolutional neural network in GAN. But many methods apply adversarial idea to generate images instead of latent vector generation to get more accurate results.

III. METHODOLOGY

Generative Adversarial Networks take random noise as input along with training data to generate new data from the same input distribution. These networks have a framework containing two distinct models- generator, and discriminator as in figure 3.1. The generator takes random noise of D-dimension and generates images, these images are sent to discriminator along with real images. The discriminator compares both the images and gives labels for generated images classifying them as real or fake. Discriminator also gives the loss function values of generator and itself. The generator tries to maximize the chance of discriminator classifying a fake image as real i.e., to generate a fake image which is exactly like a real one. Whereas, the discriminator tries to classify the input image correctly. This is known as adversarial loss which is applied to both generator and discriminator in actual training process, and gives realistic outputs. This training process, if not optimized, leads to model collapse where the generator produces the same image repeatedly.

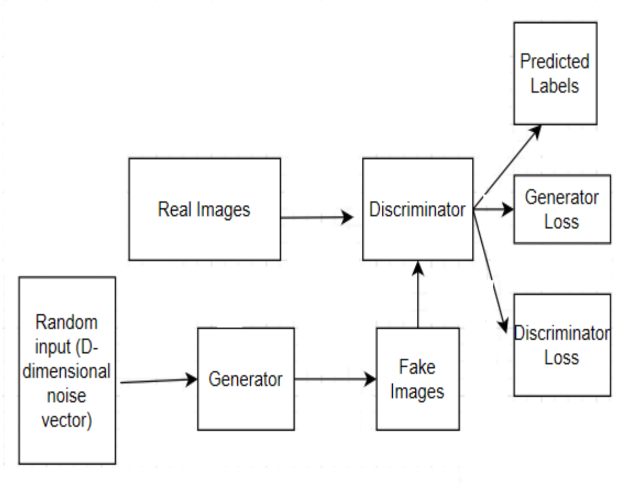

Figure 3.1 Framework of Generative Adversarial Networks

The random noise input to the generator is called latent sample on which training is done. The discriminator uses binary notation is classify if an image is real(1) or fake(0) in supervised manner. GANs are developed based on certain characteristics mentioned below.

1. GANS learn joint probability P(x,f(x)), x is input and gives inference based on P(x/f(x)). Real data is f(x).
2. While training, given real data, GANs learn it by identifying real data latent feature representation variable i.e., construction features of images.
3. They build a probability distribution to avoid outliers like normal distribution. As outliers are rare in distribution, generating it is also rare, and thus GANs work better with outlier data samples.

Generative Adversarial Networks work on maximum likelihood estimation principle. If z is data representing an image. D(z) is the discriminator network that gives probability of z from training data rather than the generator network. D(z), a binary classifier, is to be high when z is from training data and low if it is from generator. Let q be a latent space vector or noise. G(q) gives the generator network function that maps q vector to data space. G estimates the distribution that the training data comes from (pdata) to generate fake images from that estimated distribution (pg).

Also, D(G(q)) is the probability (scalar) that the output of the generator G is a real image. The two network models discriminator and generator are bounded in a minimax game where the probability to correctly classify images (logD(z)) is maximized by the discriminator, and the generator tries to minimize the probability that this discriminator's prediction of outputs are fake (log(1−D(G(z)))) [2]. The GAN loss function is

$$\min_G \max_D V(D,G) = E_{z\sim pdata(z)}[\log D(z)] + E_{q\sim pq(q)}[\log(1-D(G(q)))] \quad (1)$$

This equation(1) is solved when pg=pdata, and the discriminator gives output randomly. Moreover, the models are not trained till the convergence point in GANs. While training a GAN, three major steps are to be followed.

1. Random noise is used in generator to create fake images as inputs to discriminator.
2. Discriminator is to be trained with both real and fake images simultaneously or one after the other.
3. Training the model comprising of both generator and discriminator.

IV. PROPOSED WORK

In this model, Generative Adversarial Networks are applied to implement three applications- Neural Style Transfer and Anime Character Generation (DCGAN).

A. Neural Style Transfer

Neural style transfer (Figure 4.1.1) takes a content image, a input image and a references style image to blend them together such that the input image is transformed into the style of reference style image and content image.

Content Image Style Image Output Image

Figure 4.1.1 Example of Neural Style Transfer

Retrieval Number: F9856038620/2020©BEIESP
DOI: 10.35940/ijrte.F9856.038620
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

To implement neural style transfer, intermediate layers are necessary. These layers represent feature maps which increase as the model gets deeper. A network architecture VGG19, which is a pretrained image classification network is used where the intermediate layers define representation of content, style from the given input images. These layers match the corresponding style and content targets to the input image. This optimization technique depends on the principle to define two distance functions, Lcontent and Lstyle. The difference between two input images in terms of style is given by Lstyle and content difference between the images is given by Lcontent. The input images are transformed in such a way that the content distance and style distance are minimized.

B. Content Loss

Network is given desired content image and base input image as input, which returns the intermediate layer outputs. Then, euclidean distance is calculated between these intermediate outputs of the images which gives the content loss function value.

Content loss function which gives the distance of content from given input image x and content image p. If C_{nn} is a pre-trained deep convolutional neural network, VGG19 architecture is used in this case. If X is any image, $C_{nn}(x)$ is the network fed by X, $F^l_{ij}(x) \in C_{nn}(x)$ and $P^l_{ij}(x) \in C_{nn}(x)$ give the intermediate feature representations of the network with inputs x and p at layer l respectively, then the content distance (loss) formally is given as in equation(2).

$$L^l_{content}(p,x) = \sum_{i,j}(F^l_{ij}(x) - P^l_{ij}(p))^2 \tag{2}$$

C. Style Loss

Given an input image x, style image a, Style Loss is the distance between the style representation given in gram matrices for these images. Consider G^l_{ij} as the inner product between the vectorized feature map i and j in layer l. Correlation between different Gram matrix filter responses G^l is given by this. The correlation between feature maps i and j is represented by G^l_{ij} generated over the feature map [2].

Gradient descent from content image is done to transform input image into an image that matches the style representation of the original image. This is done by minimizing the mean squared distance between the feature correlation map of the style image and the input image. The contribution of each layer to the total style loss is described by below equation(3).

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j}(G^l_{ij} - A^l_{ij})^2 \tag{3}$$

The respective style representation in layer l of input image x and style image a I sgiven by G and A. The number of feature maps each of size Ml=height∗width is given by Nl. Thus, the total style loss across each layer is as in equation(4).

$$L_{style}(a,x) = \sum_{l \in L} w_l E_l \tag{4}$$

where the contribution of each layer's loss is given by a factor wl. Then, weighting of layers is done equally as in equation(5).

$$(w_l = \frac{1}{||L||}) \tag{5}$$

D. Anime Character Generation (DCGAN)

In animation industry, creating new unique character images is a difficult task. But if the generation of new anime characters is automated, then this problem can be solved. These automated facial images of anime characters need not require specialized skills from professionals and these images can also be taken as inspiration by the professionals. This generation of images is done by GANs using convolutional layers in the network. These layers process the images and identify features, like edges, shapes, and complex objects. Neural Networks like Inception, AlexNet, Visual Geometry Group (VGG), and ResNet are usually used for processing images.

E. Dataset

The dataset used in this model consists of anime facial images that are cropped from various manga websites, and are resized to a standard size. There are total 21552 images in the folder as shown in figure 4.2.1. Automated Character generation has to undergo the following steps mentioned below [3].

1. Generate random normal noise for input
2. Real data is collected from dataset and is to be concatenated with generated noise
3. Add noise to the input label
4. Training only the generator
5. Training only the discriminator
6. Train the combined GAN
7. Saving instances of generator and discriminator

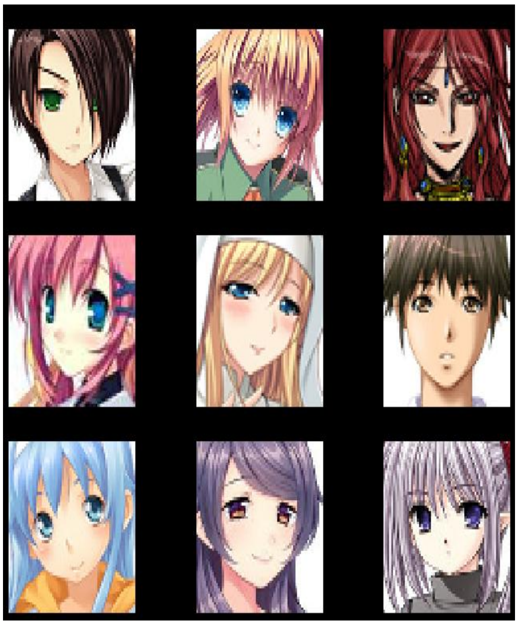

Figure 4.2.1 Dataset sample for Anime Character Generation

F. The Generator

As shown in figure 4.2.2, the generator contains convolutional transpose layers followed by batch normalization and a leaky ReLU activation function for up sampling. Strides parameter is used in the layers to avoid unstable training of the dataset. Dying ReLU problem can be solved by using Leaky ReLU, where the function will have a small negative slope of 0.01 or lesser value when x < 0 instead of value equal to zero.

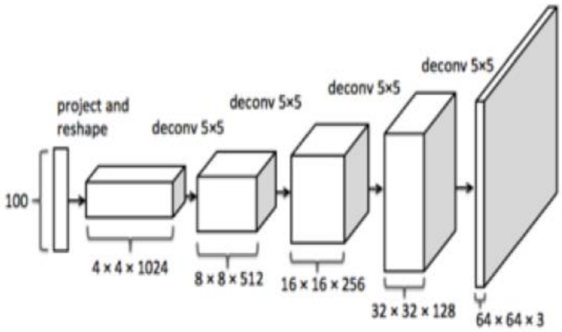

Figure 4.2.2 Generator for DC -GAN

G. The Discriminator

The discriminator (figure 4.2.3) also consists of convolution layers where we use strides to do down sampling and batch normalization for stability.

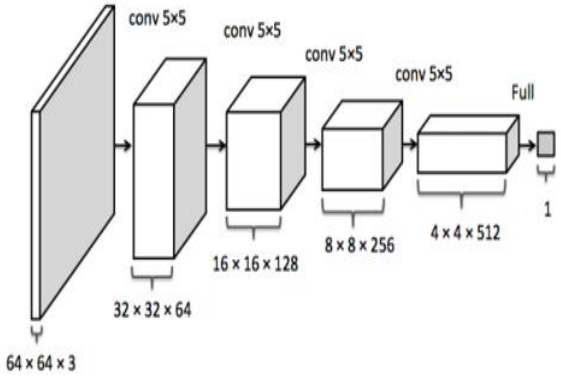

Figure 4.2.3 Discriminator for DC-GAN

H. The Compiled GAN

The GAN is compiled by compiling a network in Keras-generator followed by discriminator. It is done to implement proper backpropagation and to get correct outputs. In this model, random noise is taken as input by generator and its output is fed to the discriminator. To avoid adversarial collapse, the discriminator's weights are kept as frozen.

I. Loss versus training iteration

Below is a plot (figure 4.2.4) of Discriminator's(D) & Generator's(G) losses versus training iterations. As the training iterations are increasing, the generator and discriminator loss is decreasing. This is because the generative adversarial networks keeps improving the deep convolutional neural nets after every iteration and so the loss function value decreases.

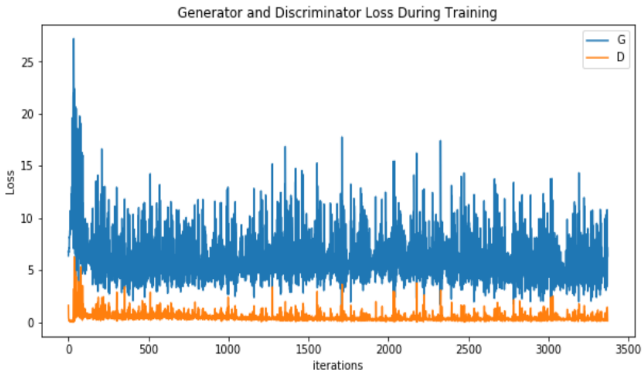

Figure 4.2.4 Generator and discriminator loss during training

J. Output

The figure 4.2.5 shows the fake images generated by generator for the input images from figure 4.2.1.

Figure 4.2.5 Output Anime Character Images

V. CONCLUSION

In this model, Generative Adversarial networks are used to successfully build a model for neural style transfer images, anime character generations.

The model generated a good resolution images in both the applications. There is a need to further research about this topic in GANs. Improving the GAN model in cases where the class labels of training data are not distributed evenly can be done. Furthermore, quantitatively evaluating methods applied in this situation should be carefully noted and results are to be evaluated. Class labels in the training dataset become unbalanced as it would lead to a measure bias. Final resolution of generated images should be improved. More analyzing on data need to be done. Super-resolution might be a possible solution to the problem, but it needs to be tested and implemented carefully.

REFERENCES

1. Yanghua Jin, Jiakai Zhang, Minjun Li, Yingtao Tian, Huachun Zhu, Zhihao Fang, "Towards the Automatic Anime Characters Creation with Generative Adversarial Networks," arXiv:1708.05509
2. Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," arXiv:1703.10593.

3. Yanhua Li, Jianping Wang, Xiaomei Zhang, Yangjie Cao, "FittingGAN: Fitting image Generation Based on Conditional Generative Adversarial Networks," 2019 14th International Conference on Computer Science & Education (ICCSE)
4. Yoshiharu Kawai, Masataka Seo, Yen-Wei Chen, "Automatic Generation of Facial Expression Using Generative Adversarial Nets," 2018 IEEE 7th Global Conference on Consumer Electronics (GCCE)
5. Zheng Xu, Michael Wilber, Chen Fang, Aaron Hertzmann, and Hailin Jin, "Learning from Multi-domain Artistic Images for Arbitrary Style Transfer," arXiv:1805.09987
6. Zhiyao Zhou, Yi Yang, Zhuohao Cai, Yusi Yang, Lan Lin, "Combined Layer GAN for Image Style Transfer," 2019 IEEE International Conference on Computational Electromagnetics (ICCEM)
7. Yu Li, Sheng Tang, Rui Zhang, Yongdong Zhang, Jintao Li, Shuicheng Yan, "Asymmetric GAN for Unpaired Image-to-Image Translation," arXiv:1912.11660
8. Hao Dou, Chen Chen, Xiyuan Hu, Silong Peng, "Asymmetric Cyclegan for Unpaired NIR-to-RGB Face Image Translation," ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)
9. Meng Li, Jian Wang, Yi Yang, Weixing Huang, Wenjuan Du, "Improving GAN-Based Calligraphy Character Generation using Graph Matching" 2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)

AUTHORS PROFILE

Dr M Rama Bai, is working as Professor in Department of CSE at MGIT, Hyderabad, India. She has published more than 30 papers in international journals. Her research areas of interest include Image Processing, Machine Learning, Artificial **Intelligence and Data Science.**

Mrs.J.Sreedevi, working as Assistant Professor in Department of CSE at MGIT, Hyderabad, India. Her research areas of Interest include Machine Learning, Artificial Intelligence and Image Processing.

Ms. B.Pragna, currently pursuing B.Tech final year in Computer Science & Engineering at MGIT, Hyderabad, India. Her areas of interest include Image Processing, Data Science and Natural Language Processing.