Cartoon Character Generation using Generative Adversarial Network

Gourab Guruprasad, Gauri Gakhar, D Vanusha

***Abstract:** Animated faces show up in cartoons, comics and games. They are broadly utilized as profile pictures in online life stages, for example, Facebook and Instagram. Drawing an animated face is work intense. Not just it requires proficient skills but also its time consuming. A lot of time is wasted in creating a cartoon character from scratch, and most of the cases ends up in creating an awkward character having very low polygon intensity. Generative adversarial network (GAN) framework can be trained with a collection of cartoon images. GANs comprises of a generator network and a discriminator network. Because of the ability of deep networks and the competitive training algorithm, GANs produce realistic images, and have great potential in the field of image processing. This method turns out to be a surprisingly handy tool in enhancing blurry images. The underlying idea behind GAN is that it contains two neural networks that compete with each other in a zero-sum game, which constitutes of generator and a discriminator.*

***Keywords:** Generative Adversarial Network*

I. INTRODUCTION

Usually machine learning systems takes complicated input and produce comparatively simpler results. Generative adversarial networks is an Machine Learning model that is used to generate an numerous number of alike samples based on a particular dataset. GANs are based on generator and discriminator, which are two deep networks trained in a competitive manner in a zero-sum game. To generate such samples that will fool the Discriminator to think that it is seeing real images while actually seeing fakes. Discriminator network takes both real and fake image and gives a verdict whether the given image is acceptable or not. Obtaining labelled data is a manual process and is time consuming too. GANs don't make use oflabelled data and thus can be trained using unlabelled data. GAN can be helpful to convert real world images to high definition cartoon images, outperforming other methods.The latent space learned by GAN follows a distributed representation but also perceives the vector arithmetic phenomenon of the output's semantics in latent space. The objective behind the advancement of image synthesis using generative adversarial network is to learn the mapping from a latent space to real data distribution through adversarial training. After learning such a nonlinear mapping, GAN is capable of producing photorealistic images by testing the latent code from any random distribution. The endeavours made to improve GANs lie in different perspectives which are designing better objective functions, improving synthesis diversity, image resolution and also training stability.

Manuscript received on April 02, 2020.
Revised Manuscript received on April 15, 2020.
Manuscript published on May 30, 2020.
***** Correspondence Author

Gourab Guruprasad*, SRM Institute of Science and Technology
Gauri Gakhar, SRM Institute of Science and Technology
D Vanusha, SRM Institute of Science and Technology

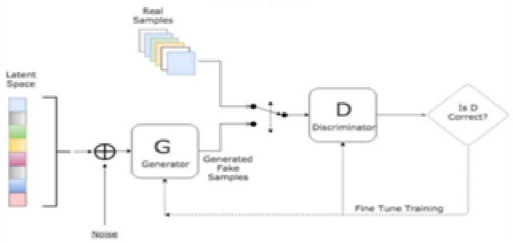

Figure 1: Generative Adversarial Network

II. RELATED WORK

2.1 Generative Latent Flow

Generative Latent Flow Algorithm is used for generative modelling of the data distribution. Generative Latent Flow makes use of an Auto-encoder to learn latent representations of the data, and a normalizing ow to trace the distribution of the latent variables to that of simple noise. It utilizes a deterministic auto-encoder to gain proficiency with a mapping to and from a latentspace, and a normalizing flow that fills in as an invertible transformation between the latent space distribution and a simple noise distribution. The model which implies generative latent flow algorithm accomplishes best in sample quality among competing models, and can coordinate the benchmarks of GANs. In addition, it has a benefit of one stage training and faster convergence.

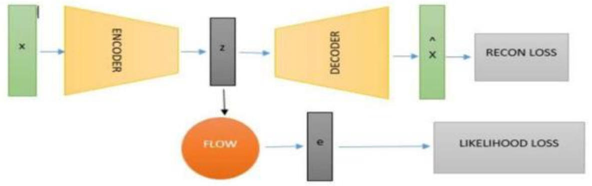

Figure 2: Generative Latent Flow

2.2 Cycle GAN

CycleGAN is a system which can perform image translation with unpaired training data. CycleGAN also includes the programmed training of image to image translation models without paired models. The models are trained in an unsupervised way utilizing an assortment of images from the source and target space that don't need to be connected in any matter. It's straightforward technique is amazing and powerful, accomplishing outwardly

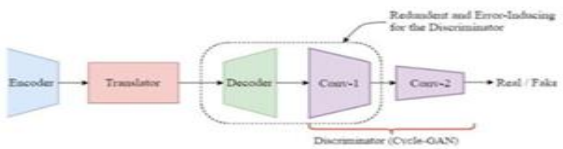
Redundant and Error-Inducing for the Discriminator
Encoder
Translator
Decoder
Conv-1
Conv-2
Real / Fake
Discriminator (Cycle-GAN)

Figure 3: Cycle-GAN

great outcomes on a scope of application domains.

2.3 DC GAN

DCGAN is an immediate expansion of the GAN, except the fact that it explicitly utilizes convolutional layer in the discriminator and convolutional-transpose layer in the generator. It replaces all maximum pooling with discriminator and furthermore utilizes generator for up sampling. They generally eliminate most part of the completely connected layers. DCGAN utilizes batch normalization, but this is not implemented inthe output layer for the generator as well as the input layer of the discriminator.

2.4 Conditional GAN

Conditional GANs are extend the functionality of normal GAN by storing extra data for the generator as well as discriminator. This takes into account the capacity to control the created yield picture, in order to deliver progressively adaptable manufactured information. This data is ordinarily a mark applied to the subsequent yield picture, for instance, skin composition, enthusiastic state or hair colour. In differentiation to conventional convolution activities acted in profound learning, depth-wise distinguishable convolution carry out convolutions on subsets of the information tensor and total the data over the subsets utilizing a solitary point-wise convolutions.

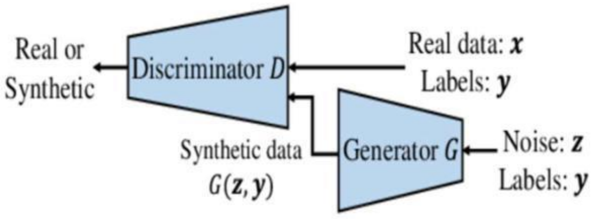

Figure 4: Conditional GAN

III. ALGORITHMS

3.1 Generative adversarial networks - GAN

GANs are algorithmic structures that utilizes two neural networks, setting one in opposition to the next in order to create new, synthetic instances of data that can pass for real data. They are generally utilized in video creation, voice creation and image creation. Unsupervised learning includes Generative modelling that involves consequently identifying and learning the regularities or patterns in input data in so that the model can be utilized to produce and yield new examples that conceivably could have been drawn from the legit dataset. GANs trace the latent code sampled from a random distribution into a realistic image. The latent space learned follows a distributed representation and also perceives the vector arithmetic phenomenon of the output's semantics in latent space.

GAN is capable of producing photo-realistic images by testing the latent code from any random distribution. The endeavours made to improve GANs lie in different perspectives which are designing better objective functions, improving synthesis diversity, image resolution and also training stability.

GANs comprises of a generator and a discriminator that contend in a zero-sum game: The discriminator is used to recognize actual training data fromsynthetic or inappropriate images, and the generator is used to trick or fool the discriminator as the training goes on.

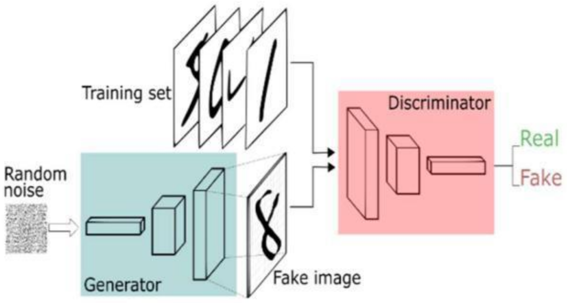

Figure 5: GAN

3.1.1 Discriminator

The discriminator is classified form of GAN. It attempts to distinguish real data from the data made by the generator. It utilizes any network architecture appropriate to the type of data it's classifying.

Discriminator's training data originates from Real data instances that includes real images of entities. The discriminator uses these occurrences as appropriate sample or positive examples during training and also makes use of the Fake data produced by the generator as inappropriate samples or negative models during training. The discriminator mainly utilizes the discriminator loss and ignores the generator loss while training.

3.1.2 Generator

In a GAN the main purpose of generator is to figures out how to make fake data by incorporating input from the discriminator. Generator mainly aims at how to make the discriminator predict its output as real. Generator takes random noise as an input and generates samples as and output. It uses the de-convolution technique for up-sampling the input images. The main objective is to generate such samples that will trick the Discriminator to think that it is seeing real images while actually seeing fake images.Training a generator network requires a tighter integration between the generator and the discriminator than training a discriminator network requires.

Retrieval Number: F7639038620/2020©BEIESP
DOI:10.35940/ijrte.F7639.059120
Journal Website: www.ijrte.org

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

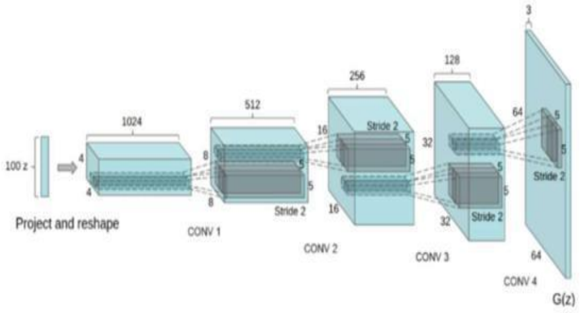

Figure 6: Generator

3.2 Convolutional Neural Networks

This network is specially used and designed to handle pixel data; it is kind of an artificial neural network which has applications in _led like image processing and recognition. The make of CNN includes many layers which are the input layer, output layer and a multilayer perceptron layer included as the hidden layer, then we have the fully connected layers and normal layers.

3.3 Autoencoder

Auto-encoder is a type of neural network is used to learn the data encodings in an efficient manner by an artificial neural network in an unsupervised way. What the auto encoder aims to learn is the encoding or the representation of the data set, which is typically done for the task of dimensionality reduction. What is being done is that we are ignoring the signal of the network which is the outlier noise.

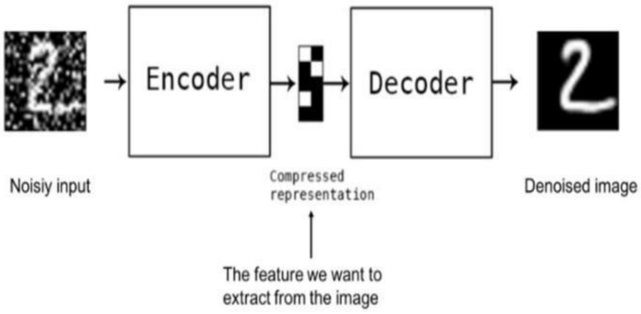
Encoder
Decoder
Noisy input
Compressed representation
Denoised image
The feature we want to extract from the image

Figure 7: Auto encoder

3.4 Stochastic Gradient descent

Gradient Descent is a mainstream enhancement strategy in AI and deep-Learning and it very well may be utilized with most, if not all, of the learning calculations. An angle is essentially the incline of a capacity; the level of progress of a parameter with the measure of progress in another parameter. Numerically, it very well may be portrayed as the incomplete subordinates of a lot of parameters regarding its sources of info. The more the inclination, the more extreme the slant. Gradient Descent is a convex function. This function can be seen as an iterative method which is used to find out the predictions of the parameters of a capacity that limits the cost capacity. The parameters are characterized a specific worth at the beginning and from that, Gradient Descent is run continuously to locate the ideal estimations of the parameters, utilizing analytics, to map the base conceivable cost estimation.

3.5 Loss Function

Generative adversarial networks attempt to replicate a probability distribution. Therefore, they use loss functions to show the gap in the data distribution produced by the GAN and that of realworld. GAN mainly constitutes of two loss functions, one which is for the training of discriminator and the other for the training of generator. The generator loss and discriminator loss derive from a single measurement of separation among the probability distributions. In any case, the generator can just influence one term in the distance measure that will react the distribution of the counterfeit data. Hence, while training the generator the other term is dropped, which will react the distribution of the real data.

The generator loss and discriminator loss appear to be unique at the end, despite the fact that they derive from a single formula, the generator at-tempts to limit the function whereas the discriminator attempts to expand it:

$$Ex[\log(D(x))] + Ez[\log(1 - D(G(z)))]$$

D(x) Estimation given by discriminator that x is real. Ex- Instances of real world.

IV. EXPERIMENTATION

The following experiments were conducted in order to derive the accurate results:

4.1 Cartoon character generation using GAN

Using GAN we can create characters by taking human faces as input and then processing it to get a high density polygon which saves a lot of time in creating cartoon characters from scratch. GAN model creates a high polygon density image of the picture taken as input with the technology used up scaling. GAN model can generate a cartoon character from taken input in lesser time when provided with adequate CPUs and GPUs.

GAN takes a set of photos in the form of a cartoon images for training. (CartoonSet100K). The input taken is then passed into the discriminator network. At first, we take a D-dimensional noise vector and the pass it into the Generator network, where the generator produces a fake image. The fake image is then passed into the discriminator network where the discriminator compares the fake image to the real image from the data set, if the image resembles to the data set then its classified as a predicted image else it returns back to the generator network for retraining. This sequence continues as a zero-sum game till the fake image is able to deceive the discriminator as a real once which is practically very much time consuming.So, when the probability to predict the real image exceeds around 60 to 70 percent, we predict the image as a required output.

4.2 Cartoonist

In the cartoonist application we take real world images taken by the user and process them. The dataset is first cleaned and outlier data is removed. Then the dataset is checked for some missing values and inappropriate values.

The dataset is then passed through the discriminator and compared with the random noise generated by the generator. After the process secures a handsome accuracy the output is generated.

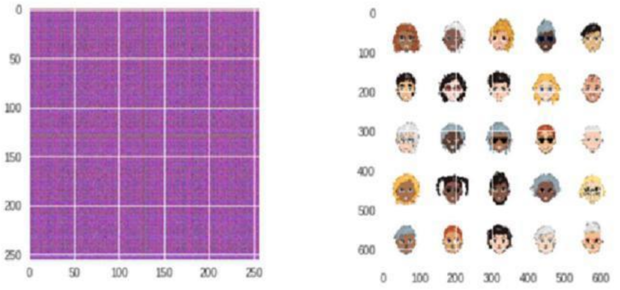

Figure 8: Input noise vector and outputgenerated.

V. ACKNOWLEDGEMENT

It is a great honour to acknowledge Andrew Ng, Ian Goodfellow, Yann LeCunn for their deep learning courses and GAN courses in Udacity, Coursera, and YouTube which was a starting point of our exploration of deep learning and Artificial Intelligence in general. We would also like to give a hearty thanks to S.D choudhary for proofreading, providing important inputs to the project, and invigorating our innovation. At the end we would extend our acknowledgements to Rusheel Gupta for his support in LATEX typesetting.

VI. CONCLUSION

Results show that this technique can produce excellent animation pictures from genuine world photographs which have explicit craftsmen's styles and with legible edges and smooth concealing and outflanks best in class strategies.

Obtaining labeled data is a manual process and is time consuming too. GANs don't make use of labeled data and thus can be trained using unlabeled data. GAN can be helpful to transform real world photos to high quality cartoon images, outperforming other methods.

GAN can help to create a cartoon face just by training the noise vector again and again till a picture is generated which resembles more than 70 percent to that of actual image.

The current project deals with a dataset of 1 lakh cartoon images. The current quality of images generated are moderate to clear. In order to improve the project, we can develop a larger dataset which consist of more than 10 lakh images. We can also improve the processing time with the help of new and advanced deep learning algorithms. The current system utilizes simple GAN which can be replaced with cycleGAN or DCGAN with the help of auto-encoder, stochastic gradient descent, generative latent flow and other algorithms to ensure maximum performance and minimum running time. This project can also be implemented to large scale video editing software.

REFERENCES

1. Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengi; Generative Adversarial Nets, University of Montreal, JUN 2014. 2018.
2. Zhisheng Xiao, Qing Yan, Yali Amit; Generative Latent Flow, University of Chicago, SEP 2019.
3. Yujun Shen, Jinjin Gu, Xiaoou Tang, Bolei Zhou; Interpreting the Latent Space of GANs for Semantic Face Editing, University of Hongkong, JUL 2019.
4. Yang Chen, Yu-Kun Lai, Yong-Jin Liu; CartoonGAN: Generative Adversarial Networks for Photo Cartoonization, Cardifi University, UK, Tsinghua University, China, 2018.
5. Hsu-Yung Cheng, Chih-Chang Yu;LEARNING TO CREATE CARTOON IMAGES FROM A VERY SMALL DATASET, National Central University, Taiwan, 2019.
6. Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, Honglak Lee; Generative Adversarial Text to Image Synthesis, University of Michigan, 2016.
7. Jon Gauthier; Conditional generative adversarial nets for convolutional face generation, Stanford University, 2019.
8. Ruizheng Wu, Xiaodong Gu, Xin Tao, Xiaoyong Shen, Yu-Wing Tai, and Jiaya Jia; Landmark Assisted CycleGAN for Cartoon Face Generation, The Chinese University of Hong Kong, JUL 2019.
9. Mkhuseli Ngxande, Jules-Raymond Tapamo, Michael Burke; Depthwise GANs: Fast Training Generative Adversarial Networks for Realistic Image Synthesis, School of Computer Engineering University of Kwa-Zulu Natal, Durban, South Africa, JAN 2019
10. Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, Brendan Frey; Adversarial Autoencoders, University of Toronto, MAY 2016.
11. Thomas Lucas, Konstantin Shmelkov, Karteek Alahari, Cordelia Schmid, Jakob Verbeek; Adversarial training of partially invertible variational autoencoders, MAR 2019.
12. [12] Uiwon Hwang, Jaewoo Park, Hyemi Jang, Sungroh Yoon and Nam Ik Cho; PuVAE: A Variational Autoencoder to Purify Adversarial Examples, Seoul National University, MAR 2019.
13. Lucas Theis, Wenzhe Shi, Andrew Cunningham and Ferenc Huszar; LOSSY IMAGE COMPRESSION WITH COMPRESSIVE AUTOENCODERS, London, MAR 2017
14. Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte Luc Van Gool; Generative Adversarial Networks for Extreme Learned Image Compression, Zurich, Switzerland, AUG 2019.
15. Byeongkeun Kang, Subarna Tripathi, and Truong Q. Nguyen; Toward Joint Image Generation and Compression using Generative Adversarial Networks, JAN 2019.