International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-8, Issue-3, May 2020
https://doi.org/10.21276/ijircst.2020.8.3.22
www.ijircst.org

A Primer on Generative Adversarial Networks

**Dr. Vikas Thada, Mr. Utpal Shrivastava, Jyotsna Sharma,
Kuwar Prateek Singh, Manda Ranadeep**

ABSTRACT- Generative Adversarial Networks (GANs) is a type of deep neural network architecture that utilizes unsupervised machine learning to generate data. They were presented in 2014, in a paper by Ian Goodfellow, Yoshua Bengio, and Aaron Courville. This paper will introduce the core components of GANs. This will take you through how every part function and the significant ideas and innovation behind GANs. It will likewise give a short outline of the advantages and downsides of utilizing GANs, comparison of architectures of various GANs and knowledge into certain true applications.

KEYWORDS- Deep Learning, Generative Adversarial Networks, Neural Network, Application

1. INTRODUCTION

In the last few years, a type of generative model known as Generative Adversarial Networks (GANs), has achieved tremendous success mainly in the field of computer vision, image classification, speech and language processing, etc[18]..A GAN is a profound neural system design made up of two systems, a generator network and a discriminator network. Through various cycles of generation and discrimination, the two systems train one another, while at the same time attempting to outmanoeuvre one another

Manuscript Received May 21, 2020

Dr. Vikas Thada, Associate Professor, Department of Computer Science & Engineering, Amity School of Engineering & Technology, Amity University Haryana, India (email: vikasthada@gmail.com)

Mr. Utpal Shrivastava, Student, M. Tech, Department of Computer Science & Engineerin, GJUS&T, Haryana and Pursuing Ph.D. from Banasthali University Jaipur.

Jyotsna Sharma, Student, B.Tech, Department of Computer Science & Engineering, Amity University Haryana

Kuwar Prateek Singh, Student, B.Tech, Department of Computer Science & Engineering, Amity University Haryana

Manda Ranadeep, Student, B.Tech, Department of Computer Science & Engineering, Amity University Haryana, India.

A generator network utilizes existing information to produce new information[5]. The generator's essential objective is to produce information, (for example, pictures, video, sound, or content) from an arbitrarily created vector of numbers, called a latent space. While making a generator network, we have to specify the goal of the system. This might be image generation, text generation, audio generation, video generation etc. The discriminator is a network that attempts to differentiate between real information and the information generated by generator. The discriminator network attempts to place the incoming information into predefined classifications. It can either perform multi-class

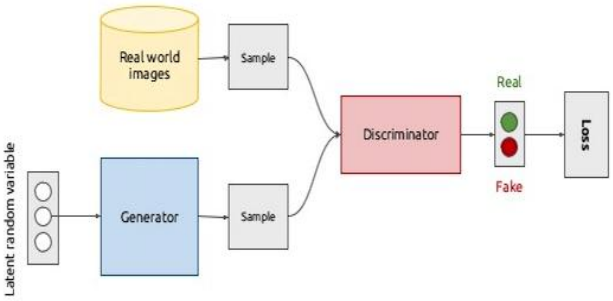

Fig 1: Rough illustration of Generative Adversarial Networks [6]

Characterization or binary classification. Generally, in GANs binary classification is performed.

A. Practical Implementations of GANs

GANs have valuable applications such as:

a. Image generation

Generative adversarial networks can be utilized to produce real images after they have been trained on sample images[8]. Like for instance, in the event that we need to produce new pictures of flowers, we prepare GAN on a great many sample images of flowers. When preparation has been completed, the generator system will have the option to produce new pictures which are unique in relation to images in the training set. Picture generation is utilized in marketing, for generation of logo, purpose of entertainment, social media and so on.

b. Synthesis of Image from Text

Generating pictures from content description is a fascinating use of GANs that can be useful in film business, as GAN is equipped for creating new information dependent on content that we made up. In the comic industry, it's conceivable to consequently create a story sequence [8].

c. Image-to-image translation

Can be utilized to change over pictures taken in day to pictures captured at night, to change from sketches to works of art, style pictures to look just like Picasso, to change over aeronautical pictures to satellite pictures consequently, and to change over pictures of horses to pictures of zebras.

d. Synthesis of videos

GANs likewise can be utilized to produce videos. They are capable of producing content in much less time if somehow we manage to manually make content. They can increase the productivity of film makers and furthermore engage specialists that need to make imaginative recordings in their spare time [8].

II. TYPES OF GENERATIVE ADVERSARIAL NETWORKS

There are presently many distinctive GANs accessible and this number is expanding at a sensational rate.A few variants are discussed in this paper.

A. Deep convolutional generative adversarial networks [DCGAN]

DCGAN is one of the most famous likewise the best execution of GAN. It is made out of ConvNets instead of multi-layer perceptrons. The ConvNets are actualized without max pooling, which is replaced with convolutional stride. Likewise, the layers are not completely associated.

a. Architecture of a DCGAN

The most interesting part of Generative Adversarial Networks is the design of the Generator network. The Generator network can take up random noises and maps it to images such that discriminator can't tell which image came from the datasets and which came from generator.This is an interesting application of neural networks. Typically neural nets map input into a binary output, (1 or 0), maybe a regression output, (some real-valued number), or even multiple categorical outputs, (such as MNIST or CIFAR-10/100).In this article, we will see how a neural net maps from random noise to an image matrix and how using Convolutional Layers in the generator network produces better results.

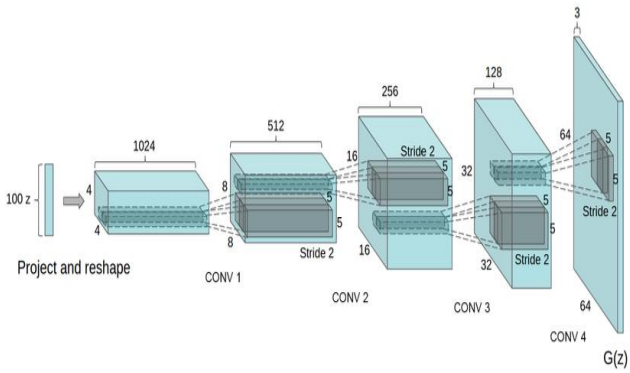

Fig 2: Architecture of DCGAN

Fig 2 shows DCGAN generator was presented in the LSUN scene modeling paper. This network takes 100x1 noise vector as input, denoted by z, and maps it to the G(Z) output that is 64x64x3.This architecture is especially interesting the way first layer expands the random noise[2]. The network varies from 100x1 to 1024x4x4! This layer is labelled as 'project and reshape'. Following this layer are classical convolutional layers which reshape the network with the (N+P — F)/S + 1 equation classically taught with convolutional layers. In the diagram above we can see that the N parameter, (Height/Width), goes from 4 to 8 to 16 to 32, it does not appear that there is any padding, the kernel filter parameter F is 5x5, and the stride is 2. We see the network goes from 100x1 → 1024x4x4 → 512x8x8 → 256x16x16 → 128x32x32 → 64x64x3

b. Practical applications of DCGAN

The various applications of DCGANs include:

- **The generation of anime characters**: Currently, animators physically draw characters with PC programming and sometimes on paper too. This procedure is manual so generally takes more time. With DCGANs, new anime characters can be produced in much less time, thus improving the creative process.
- **The expansion of datasets**: On the off chance that you need to train a supervised machine learning model, to prepare a good model, you would require an enormous dataset. DCGANs can help by expanding the current dataset, along these expanding the size of the dataset required for supervised model training.
- **The generation of MNIST characters**: The MNIST dataset contains 60,000 pictures of manually written digits. To prepare a complex supervised learning model, the MNIST dataset isn't adequate. DCGAN, when prepared, will create new digits that can be added to the first dataset.
- **Human face generation**: DCGANs use convolution neural networks and are quite acceptable at producing practical looking images.
- **Feature extractors**: Once prepared, a discriminator can be utilized to extract futures from a moderate layer. These extracted futures can be useful in tasks, for example, style transfer and face recognition. Style transfer includes creating interior representation of pictures, which are utilized to compute style and content misfortunes.

B. Vanilla GAN

This is the simplest GAN. Here, Generator and Discriminator are straightforward multi-layered perceptrons. In vanilla GAN, the calculation is extremely basic, it attempts to improve the scientific condition utilizing SGD[7].

C. Conditional GAN [CGAN]

CGAN can be depicted as a profound learning technique in which some contingent parameters are established. In CGAN, an extra parameter 'y' is added to the Generator for producing the relating information. Names i.e. labels are additionally taken care of into the contribution to the Discriminator for the Discriminator to help recognize the difference between real generated information and non-real generated information.[11]

a. The CGAN Architecture

The CGan consists of four networks: an encoder, the FaceNet, one generator and one discriminator. Encoder helps learn inverse mapping of input face images and the age condition with the latent vector.FaceNet is a network for face acknowledgment that learns the distinction between an input image x and a reproduced image. A generator takes hidden representation consisting of a face image and a condition vector and generates an image. The discriminator discriminates between real images and fake images.Primary goal of the encoder network is to generate a latent vector of the provided images. The encoder network is a deep convolutional neural network. The network consists of four convolutional blocks and two dense layers. Each convolutional block consists of one convolutional layer, one batch normalization layer and an activation function. In every convolutional block, each convolutional layer is followed by a batch normalization layer apart from the first convolutional block, which doesn't have the batch normalization layer. **The generator** is a deep convolutional neural network[13] too. It consists of dense, upsampling, and convolutional layers. It takes two input values: a noise vector and a conditioning value. The conditioning value is the additional information provided to the network. **The discriminator** identifies whether the provided image is fake or real. It does this by passing the image through a series of downsampling layers and some classification layers. In other words, it predicts if the image is real or fake. Like the other networks, the discriminator network is another deep convolutional network. It contains several convolutional blocks. **Face recognition network** recognizes a person's identity in a given image.

b. Practical applications of Age-cGAN

- **Cross-age face recognition**: This can be consolidated into security applications, for example, cell phone unlocking or laptop unlocking. With Age-cGAN systems, the life expectancy of cross-age face recognition frameworks will be longer.
- **Finding lost children**: This is a useful application of an Age-cGAN. As the age of a kid expands, their facial highlights change, and it turns out to be a lot harder to recognize them. An Age-cGAN can reproduce an individual's face at a specified age.
- **Visual effects in movies**: It's repetitive and lengthy procedure to physically stimulate an individual's face when they are more aged. Age-cGANs can accelerate this procedure and reduces the expenses of creating and reproducing faces.

D. StackGAN

A StackGAN comprises of two GANs stacked together framing a system which is equipped for creating high-quality pictures. It comprises of two stages[4]. The first stage produces low-resolution pictures with essential hues and harsh portrayals, conditioned on embedding of text while the second stage takes picture created by first stage and produces a high-quality picture which is conditioned on embedding of text. Fundamentally, subsequent system amends deformities and includes convincing subtleties, yielding a progressively practical high-resolution picture.[1,3]

a. Architecture of StackGAN

StackGAN is a two-stage network. Each stage has two generators and two discriminators. StackGAN consists of many networks which are Stack-I GAN: text encoder, Conditioning Augmentation network, generator network, discriminator network, embedding compressor network Stack-II GAN: text encoder, Conditioning Augmentation network, generator network, discriminator network, embedding compressor network[10].The model architecture of StackGAN consists of mainly the following components:

- Embedding that converts the input variable length text into a fixed length vector.
- Conditioning Augmentation (CA)
- Stage I Generator that generates Low resolution images
- Stage I Discriminator
- Residual Blocks
- Stage II Generator that generates high-resolution images
- Stage II Discriminator

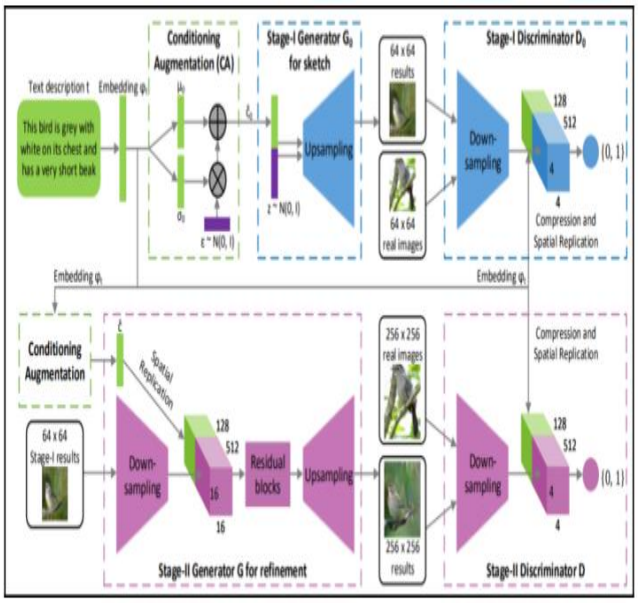

Fig 3: The architecture of StackGAN.

E. Laplacian Pyramid GAN [LAPGAN]

The Laplacian pyramid is a direct invertible picture portrayal comprising of band-pass pictures, separated an octave separated, in addition to a low-recurrence leftover. This methodology utilizes numerous quantities of Generator and Discriminator systems and various degrees of the Laplacian Pyramid. This methodology is for the most part utilized in light of the fact that it creates top notch pictures. The picture is down-inspected from the outset at each layer of the pyramid and afterward it is again up-scaled at each layer in a regressive pass where the picture gains some clamour from the Conditional GAN at these layers unless it arrives at its earlier size.

F. CycleGANs

A CycleGAN is kind of GAN that interprets a picture from one space X to another area Y, without requirement for combined pictures. CycleGAN attempts to become familiar with generator which, thus, memorises two mappings. Rather than preparing one solitary generator utilized in the vast majority of GANs, CycleGANs train two generators as well as two discriminators.[14]

a. The architecture of a CycleGAN

CycleGAN consists of overall two architectures: one generator and one discriminator. The architecture of generator creates two models- Generator A and Generator B. The architecture of discriminator creates another two models- Discriminator A and Discriminator B.

b. Architecture of the generator

The generator network is an auto-encoder network type. It takes an image as an input and gives another image as output. It comprises of two parts: encoder and decoder. The encoder consists of convolutional layers with capabilities of downsampling and transforms an input of a shape of 128x128x3 to an internal representation. The decoder consists of two upsampling blocks and a final convolution layer, which transforms the internal representation to a shape of 128x128x3 as output.

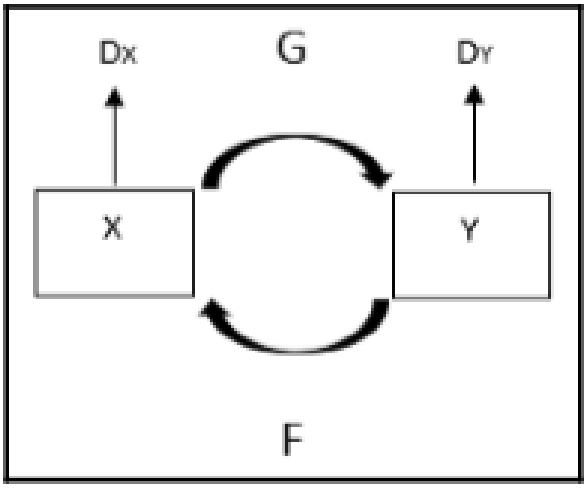

Fig 4: Illustration of a CycleGAN with two generator and two adversarial discriminator networks.

c. The architecture of the discriminator

It is a deep convolutional neural network and consists of several convolution blocks. Basically, it takes an image of a shape of (128, 128, 3) as input and guesses if the input image is real or not. It consists of many Zero Padding 2D layers.

G. 3D Generative Adversarial Networks (3D-GAN)

Same as vanilla GAN, it has one generator and one discriminator[16]. Both networks use 3D convolutional layers instead of 2D convolutions. Given sufficient data, it figures out how to produce high quality 3D-shapes.

a. The architecture of a 3D-GAN

Both of the networks in a 3D-GAN are deep convolutional neural networks. The generator network is an upsampling network. It upsamples a noise vector (a vector from probabilistic latent space) to produce a 3D image with shape similar to the input image in terms of its length, breadth, height, and channels. The discriminator network is a downsampling network. Using a series of 3D convolution operations and a dense layer, it identifies whether the input data provided to it is real or fake[8,13].

b. Applicatios of 3D-GAN

3D-GANs can potentially be used in a wide variety of industries, such as:

- **Manufacturing**: 3D-GANs can help create prototypes quickly. They can cook up creative ideas and help in visualizing 3D models[16].
- **3D printing:** 3D pictures created by 3D-GANs can be utilized to print protests in 3D printing. The manual procedure of making 3D models is exceptionally long.
- **Design processes**: 3D generated models can provide a good estimate of the eventual outcome of a particular process. They can show us what is going to get built.
- **New samples**: Similar to different GANs, 3D-GANs can create pictures to prepare a supervised model.

International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-8, Issue-3, May 2020
https://doi.org/10.21276/ijircst.2020.8.3.22
www.ijircst.org

Table 1: Comparison of GANs [19]

VANILLA GAN	CGAN	DCGAN	LAPGAN[17]
Learning is supervised[19]	Learning is supervised	Learning is not supervised	Learning is not supervised
Follows Multilayer perceptron Structure	Follows Multilayer Perceptron Structure	Uses Convolutional networks but there are certain constraints	Uses Laplacian pyramid in addition to sequential convolutional networks
Stochastic Gradient Descent with k steps for D and 1 step for G is used for optimization	Stochastic Gradient Descent with k steps for D and 1 step for G is used for optimization[19]	Stochastic Gradient Descent With Adam optimizer for both G and D is used for optimization	Nothing like this is used
High error rate is wanted by G and low error rate by D	G desires high rate of error of D, end D desires low rate of error conditioned on extra information	Hierarchy of representations is learned from object parts to scenes in G and D both	At each level of pyramid, generation of Images takes place in grainy-to-fine fashion with sequential CNN

III. CHALLENGES AND LIMITATIONS OF GAN

GANs face various challenges and limitations but all these are an area of research. While none of the problems have been solved, we have mentioned some possible solutions that have been tried.

A. Vanishing Gradients

Research has proposed that in the event that your discriminator is excessively acceptable, at that point generator preparing can be unsuccessful because of vanishing gradients. As a result, an ideal discriminator doesn't give enough data to the generator to gain ground.

Possible solutions

- **Wasserstein loss**: The Wasserstein loss is intended to forestall disappearing gradients in any event when training the discriminator for optimality.
- **Modified minimax loss**: The original GAN paper[2] proposed an alteration[3] to minimax loss to manage vanishing gradients.

B. Mode Collapse

Typically GAN needs to create a wide range of yields. For instance, an alternate face for each irregular contribution to the face generator has to be obtained. Notwithstanding, if a generator delivers a particularly conceivable yield, the generator may figure out how to create just that yield. Actually, the generator is continually attempting to locate the one yield that appears to be generally conceivable to the discriminator. In the event that the generator begins creating
a similar yield (or a little arrangement of yields) again and again, the discriminator's best procedure is to figure out how to consistently dismiss that yield. In any case, if the up and coming age of discriminator stalls out in a nearby least and doesn't locate the best methodology, at that point it's unreasonably simple for the following generator to locate the most conceivable yield for the present discriminator. Every cycle of generator over-enhances for a specific discriminator, and the discriminator never figures out how to become familiar with out of the snare. Accordingly the generators pivot through a little arrangement of yield types. This type of GAN failure is called mode collapse[15].

Possible Solutions

The following methods in general power the generator to widen its scope by keeping it from improving for a single fixed discriminator:

- **Wasserstein loss**: The Wasserstein loss lightens mode collapse by letting you train the discriminator to optimality without agonizing over vanishing gradients[9]. On the off chance that the discriminator doesn't stall out in nearby minima, it figures out how to

dismiss the outputs that the generator balances on. So the generator needs to have a go at something new.

- **Unrolled GANs**: Unrolled GANs use a generator loss function that consolidates the present discriminator's orders, yet additionally the outputs of future discriminator renditions. So, the generator can't over-advance for a solitary discriminator.

C. Convergence Failure

GANs often fail to converge. As the generator improves on the go with training, the discriminator execution deteriorates in light of the fact that the discriminator can only with significant effort differentiate among genuine and counterfeit[9]. On the off chance that the generator succeeds impeccably, at that point the discriminator has a half precision. This poses a problem for convergence of GAN as the discriminator feedback becomes less important over time. If the GAN keeps preparing past the moment where discriminator is giving totally arbitrary criticism, at that point the generator begins to prepare on garbage input, and its own quality may fall. For a GAN, convergence is fleeting, as opposed to stable state.

Possible Solutions

Researchers have attempted to utilize different types of regularization to improve convergence of GAN which includes:

- Embedding noise in discriminator inputs
- Penalizing weights of discriminator

IV. CONCLUSION

In this paper, we review about developing and well-known network, Generative Adversarial Networks. We likewise research about the different kinds of GANs, their applications, and current future advancements in this field. As we would see it, GANs could be one of the profoundly potential fields in the special territory of neural systems, which is still unexplored and has not arrived at its maximum capacity in research development. It could develop out to be one of the most dominant generative models accessible.

REFERENCES

[1] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, Dimitris Metaxas(19 Oct 2017). StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks

[2] Alec Radford, Luke Metz, Soumith Chintala (Submitted on 19 Nov 2015 (v1)). Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

[3] Jonathan Shapiro (April 2016). Text to Image Synthesis Using Generative Adversarial Networks

[4] Tingting Qiao, Jing Zhang, Duanqing Xu, Dacheng Tao**:** MirrorGAN: Learning Text-to- image Generation by Redescription (14 Mar 2019)

[5] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde- Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio: Generative Adversarial Networks(10 Jun 2014)

[6] https://www.slideshare.net/xavigiro/deep-learning-for-computer-vision-generative-models-and-adversarial-training-upc-2016

[7] https://www.geeksforgeeks.com/Generative-adhersarial-networks

[8] (Ahirwar, 2019). Generative Adversarial Networks Projects

[9] (Google, n.d.) https://developers.google.com/machine-learning/gan

[10] (htt).https://medium.com/@rangerscience/lets-read-science-stackgan-text-to-photo-realistic-image-synthesis-4562b2b14059

[11] Augustus Odena, Christopher Olah, Jonathon Shlens: Conditional Image Synthesis With Auxiliary Classifier GANs (20 July 2017)

[12] Andrew Brock, Jeff Donahue, Karen Simonyan: Large Scale GAN Training for High Fidelity Natural Image Synthesis (25 Feb 2019)

[13] https://arxiv.org/pdf/1508.06576.pdf

[14] Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros: Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks (15 Nov 2018)

[15] https://arxiv.org/pdf/1910.00927.pdf

[16] https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781789136678/2/ch02lvl1sec18/introduction-to-3d-gans

[17] Emily Denton, Soumith Chintala, Arthur Szlam, Rob Fergus: Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks (18 Jun 2015)

[18] https://www.researchgate.net/publication/326914856_Review_on_Generative_Adversarial_Networks

[19] https://link.springer.com/article/10.1007/s11831-019-09388-y/tables/1

ABOUT THE AUTHORS

Dr. Vikas Thada has doctoral degree in Computer Science & Engineering from Dr. K.N.M University Newai, Jaipur, India, Master's Degree in Computer Science & Engineering from Rajasthan Technical University Kota, Rajasthan, India and Bachelor's Degree also in Computer Science & Engineering from Engineering College Kota, Rajasthan, India. He is currently serving as Associate Professor in the Department of Computer Science & Engineering, Amity School of Engineering & Technology, Amity University Haryana. He has more than 19 years of teaching experience with around 8 years of research experience. He has many publications in international journals and is author of number of books on programming, data structures etc. His research interests are genetic algorithm, cryptography and network security, machine learning and natural language processing.

Mr. Utpal Shrivastava has M. Tech in computer science and engineering form GJUS&T, Haryana. He is pursuing his doctoral degree from Banasthali University Jaipur. He has 12 years of teaching experience and currently serving as assistant professor at Amity University, Haryana in the department of computer science and engineering. His research interest is in machine learning, genetic algorithm, image processing and security attacks.

Jyotsna Sharma is currently pursuing B.Tech in computer science & Engineering(2016-2020) from Amity University Haryana and is student of Dr. Vikas Thada & Mr. Utpal Shrivastava.

Kuwar Prateek Singh, is currently pursuing B.Tech in computer science & Engineering(2016-2020) from Amity University Haryana and is student of Dr. Vikas Thada & Mr. Utpal Shrivastava.

Manda Ranadeep, is currently pursuing B.Tech in computer science & Engineering(2016-2020) from Amity University Haryana and is student of Dr. Vikas Thada & Mr. Utpal Shrivastava.

