INVESTIGATION OF DIFFERENT GENERATIVE ADVERSARIAL NETWORKS TECHNIQUES FOR IMAGE RESTORATION

Tammineni Shanmukha Prasanthi[1], Swaraiya Madhuri Rayavarapu[2], Gottapu Sasibhushana Rao[3], Rajkumar Goswami[4]

[1]Department of Electronics and Communication Engineering, Research Scholar, Andhra University college of engineering, Visakhapatnam, India.

[2]Department of Electronics and Communication Engineering, Research Scholar, Andhra University college of engineering, Visakhapatnam, India.

[3]Department of Electronics and Communication Engineering, Senior Professor, Andhra University college of engineering, Visakhapatnam, India.

[4]Department of Electronics and Communication Engineering, Professor, Gayatri vidya parishad college of engineering for women, Visakhapatnam, India

***Abstract*— Generative Adversarial Networks are artificial neural networks that pit two different sets of neural networks against one another in order to generate data that isn't part of the training set. The Generative Adversarial Network (GAN) produces good outcomes when it is trained on image data that comes from the actual world. The generator and the discriminator make up the Generative Adversarial Network (GAN), which stands for "generative adversarial network." The parameters that were utilized to generate the data are completely arbitrary. The information is evaluated, and erroneous information is distinguished from true information by the discriminator. Several researchers has investigated various types of GANs but comprehensive analysis and comparison of different types of recent GAN's has not been done in literature. The article concludes with a discussion of the possible uses of GANs in a variety of settings, as well as how these applications constitute a fascinating new area of research and prospective expansion.**

Keywords*— Artificial Intelligence, Image Inpainting, Deep leaning, Generative Adversarial Networks (GAN), Generator, Discriminator, Neural Networks, Unsupervised Learning.

I. INTRODUCTION

Unsupervised learning has become a prominent area of research, although supervised machine learning has gained much prominence in most artificial intelligence methods. One of the best-known examples is Generative Adversarial Networks (GANs). The integration of generative modelling with Deep Learning methods has led to a significant advancement in unsupervised learning in recent years. Goodfellow et al. (2014) demonstrated the use of a generative adversarial network (GAN) [1]. Generative Adversarial Networks (GANs) are extensively used in both supervised and unsupervised learning tasks, and their widespread acceptance and usage is steadily increasing. Generative Adversarial Networks (GANs) might potentially enable unsupervised learning inside a supervised learning framework by using either real-world or generated data. The simultaneous training of both the generator (G) and the discriminator (D) networks is crucial for the Generative Adversarial Network (GAN). The discriminator acquires the ability to accurately categorize both synthetic and real data, therefore functioning as a binary classifier. On the other hand, G confuses the D by presenting convincing evidence. Network G sometimes provides accurate data, whereas network D has enhanced its capability to predict falsified data.

GAN, besides its application in generating handwritten anime characters, has demonstrated its utility in diverse domains including object recognition, text synthesis, face ageing, image manipulation applications, image overpainting, image stitching, human pose synthesis, visual salience prediction, stenographic applications and numerous others. Some examples of applications include texture generation, facial reconstruction, facial recognition, high-resolution imaging, music composition, drawing creation, cosmetic enhancements, image transformation, voice synthesis, medical diagnostics, and video editing.

This paper explores an analysis of the modified classical variance of GAN models and a comprehensive investigation of the many applications of GAN in different areas of computer vision. However, the main motive of this work is to discuss the disadvantages associated with training the GAN model and possible strategies for overcoming them. This article describes the findings of a study on the challenges faced during the training of models. In addition, this paper presents a Generative Adversarial Network (GAN) model along with details on its training process and resulting outcomes.

This article consists of many sections. The study includes several GAN models, which are outlined in Section II. The subsequent sections include of the experimental results and discussions (Section III) followed by the conclusions (Section IV).

II. VARIOUS TYPES OF GAN

Machine learning practitioners are increasingly using generative adversarial networks (GANs) to enhance image processing. Applications that greatly benefit from the use of Generative Adversarial Networks (GANs) include generating artistic and photographic representations based on textual descriptions, enhancing the resolution of pictures, moving images between distinct domains (such as transitioning from daytime to nighttime settings), and several more applications. In order to get such outcomes, many enhanced GAN architectures have been devised, each possessing distinct characteristics to address certain image processing challenges.

2.1 Variational Auto Encoder (VAE)

Data compression is a crucial step in the process of training a network. Data compression is the process of reducing the number of bits required so that equivalent data may be shown while maintaining the same level of data quality. This also mitigates the issue of the curse of dimensionality. Training a dataset with several characteristics poses a challenge due to its tendency to produce overfitting in the model. Hence, it is necessary to use dimensionality reduction methods prior to using the dataset for training purposes. The Autoencoder (AE) and the Variational Autoencoder (VAE) are relevant in this context. The Autoencoder and the Variational Autoencoder are both used to compress data by converting it from a higher-dimensional space to a lower-dimensional one[2].

The Variational Autoencoder (VAE) tackles the issue of uncontrolled latent space in Autoencoders (AE), allowing it to create data by sampling vectors from the latent space in a random manner. The encoder in AE generates latent vectors as output. The vectors in latent space are not generated directly by the VAE encoder in this regard. Instead, it determines, for each input, the parameters of a pre-defined distribution in latent space and applies those parameters to the model. This underlying distribution is then subjected to a limitation that is imposed by VAE, which forces it to conform to a Gaussian distribution. This limitation helps to guarantee that the latent space is properly controlled [3].

The encoder processes a picture and generates two vectors, which respectively reflect the average (μ_x) and variability (σ_x). The mean vector and the standard deviation vector are added together, with the standard deviation vector first being multiplied by a tiny random value (ϵ) as noise. This results in a modified vector (Z)(eq.1) that has the same size.

$$Z = \mu_x + \sigma_x \epsilon, \quad \epsilon \sim N(0, I) \tag{1}$$

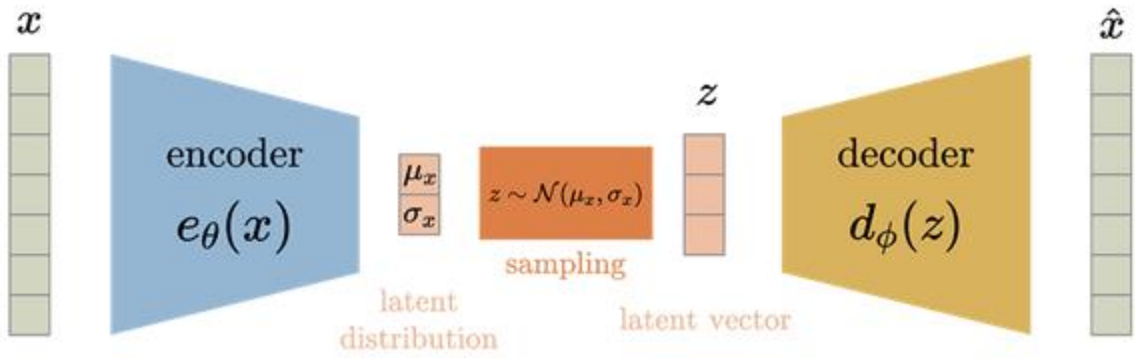


Fig.1 Architecture of variational auto encoder[19]

The loss function for a Variational Autoencoder (VAE) typically comprises two components: the reconstruction loss (Eq. 2), which is similar to the loss used in traditional autoencoders, and the Kullback–Leibler divergence loss (KL). The difference between the input that was originally provided and the output that was generated by the decoder may be quantified using reconstruction loss. The KL divergence loss (Eq. 4) quantifies the discrepancy between the learned probability distribution and the predefined reference distribution.

$$\text{Reconstruction loss} = ||x - y||_2 = ||x - d_\emptyset(z)||_2 = ||x - d_\emptyset(\mu_x + \sigma_x \epsilon)||_2 \tag{2}$$

Where x is input and $\hat{y}$ is reconstructed output

$$\mu_x, \sigma_x = e_\theta(x), \epsilon \sim N\,(0, I) \tag{3}$$

$$\text{Similarity Loss} = \text{KL Divergence} = D_{KL}\,(N\,(\mu_x, \sigma_x)||N\,(0, I)) \tag{4}$$

$$\text{Loss} = \text{Reconstruction Loss} + \text{Similarity Loss} \tag{5}$$

2.2 Deep Convolutional Generative Adversarial Networks (DCGAN)

DCGAN is a network design for Generative Adversarial Networks (GAN) that has received a lot of attention for its efficiency. The architecture mostly comprises convolutional layers, excluding max-pooling or fully linked layers[4]. The method employs convolutional stride and transposed convolution to reduce and increase the resolution, respectively. Generator and Discriminator loss are calculated using eq. 6 and eq.7

The architecture of DCGAN involves the following steps:

- Replacing every occurrence of max-pooling with convolution stride.
- Utilize transposed convolution for the purpose of up sampling.
- Remove the layers that are fully interconnected.
- Apply batch normalization to all layers of the generator, except for the output layer. For the discriminator, employ batch normalization for all levels, except for the input layer.
- Utilize the hyperbolic tangent (tanh) activation function specifically for the output layer, while using the Rectified Linear Unit (ReLU) activation function for the remaining layers in the generator.
- Use the Leaky Rectified Linear Unit (ReLU) activation function in the discriminator.

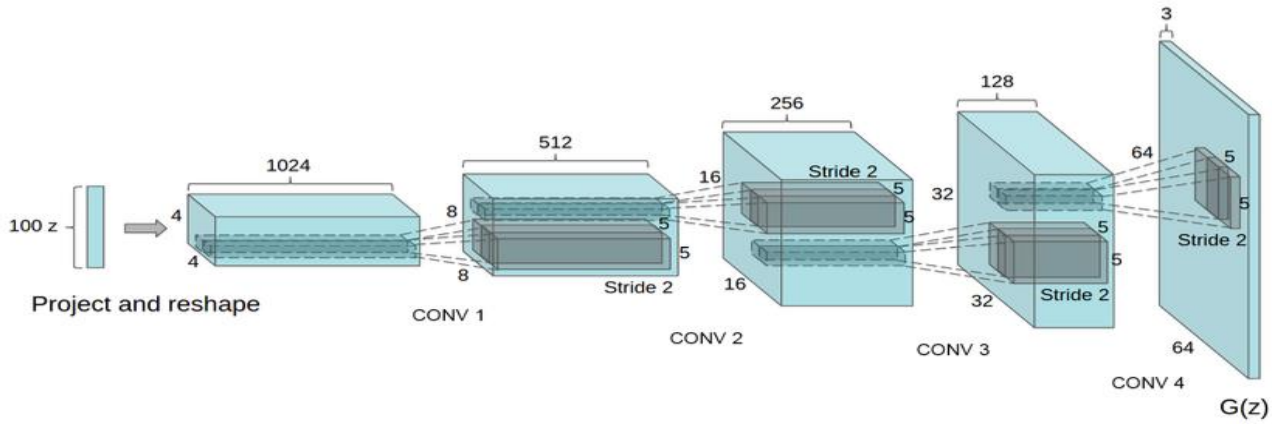


Fig2. **Architecture of DCGAN (Deep Convolutional Generative Adversarial Networks)[17]**

$$L(\text{Generator})=\min[\log(1-D_1(G_1(Z)))] \quad (6)$$

$$L(\text{Discriminator})= \min[-\log(D_1(x))+\log(1-D_1(G_1(Z)))] \quad (7)$$

2.3 Wasserstein GAN(WGAN)

WGAN provides increased stability in training the model as compared to basic GAN designs[5]. The use of the loss function in WGAN provides a termination condition for assessing the model. Despite the sometimes lengthier duration of the instruction, it remains one of the superior choices for achieving more effective outcomes. Nevertheless, the use of the basic GAN approach encounters issues related to gradient manipulation, which might result in precarious training. Hence, we use Wasserstein distance to tackle these persistent issues[6]. The mathematical formula is shown in eq.8.

$$L_{critic}(\boldsymbol{\omega}) = \max_{\boldsymbol{\omega} \in W} \boldsymbol{E}_{x \sim \mathbb{P}_r}[f_\omega(x)] - \boldsymbol{E}_{z \sim Z}[f_\omega(g_\theta(z))] \quad (8)$$

The greatest value in the equation indicates the limitation imposed by the discriminator. Within the WGAN framework, the discriminator is referred to as the critic. One rationale for this practice is the absence of a sigmoidal activation function that restricts values to either 0 or 1, representing genuine or false. Contrarily, the discriminator networks in WGAN produce a value within a certain range, enabling them to function with less strictness as critics.

The latter half of the equation represents data generated by the generator, while the former component represents the actual data. In order to effectively differentiate between the generated data and the authentic data, the discriminator (or critic) in the aforementioned equation aims to maximize the disparity between the two data sets. The objective of the generator network is to minimize the discrepancy between the real data and the artificially generated data in order to enhance the genuineness of the generated data.

2.4 Progressive GAN(Pro GAN)

This research introduces a progressive picture creation model that utilizes multi-level characteristics to cater to the need for high-quality visual goods[7]. The method divides the work of picture production into many phases, with each step dedicated to producing the characteristics of a certain level of abstraction. Commence with the overarching characteristics, and thereafter add the more detailed characteristics gradually via the training process. The iterative generation technique enables the model to initially emphasize abstract high-level semantic knowledge and enormous scale of structural information. Subsequently, it progressively redirects its attention to ever smaller sizes without acquiring knowledge of all scales concurrently. The algorithm's design is based on the methodology used by painters while making a painting.

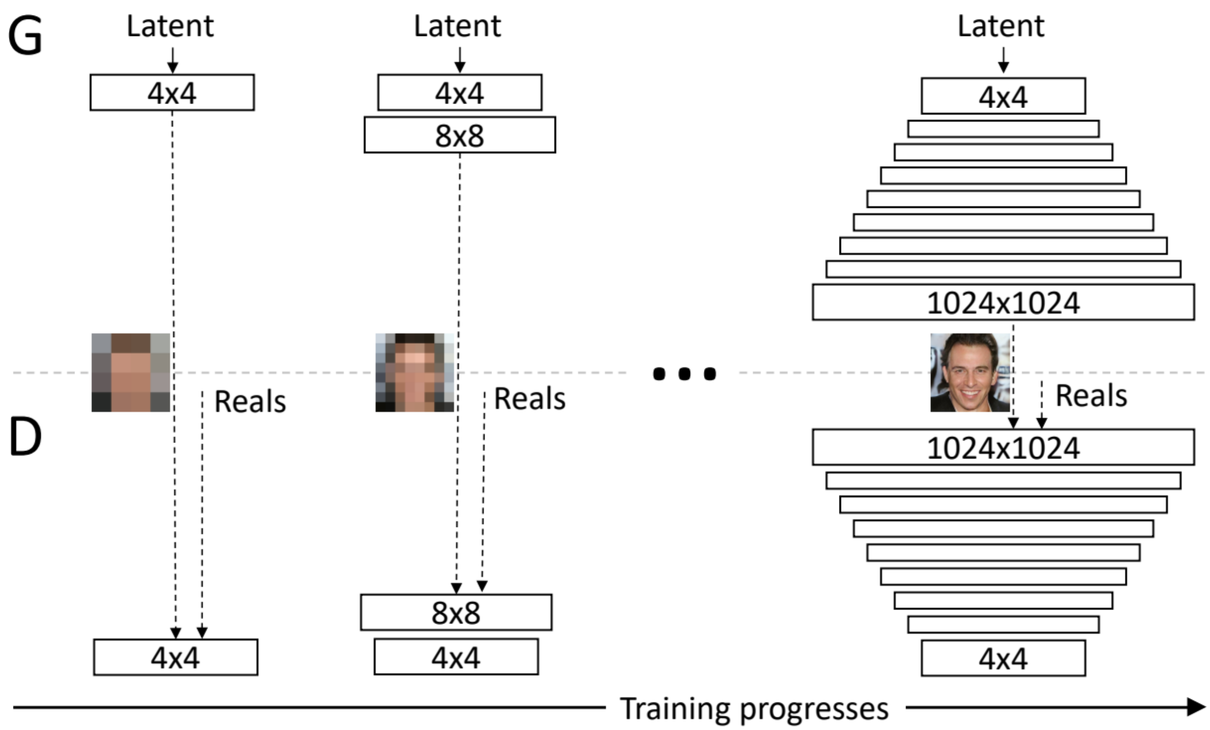


Fig4. Progressively adding layers in Generator and Discriminator[16]

The artist begins by sketching a rough outline inside the image region, then subsequently renders intricate textures based on the colors of the reference image, resulting in a cohesive and harmonious overall composition. The incremental generation method enhances the organization of the generation process and simplifies the current stage's task by leveraging the completion of previous abstract level generation[8]. The incremental generation approach(fig.4) may use prior learning from completed subtasks to speed up the process of learning new subtasks.

2.5 SUPER RESOLUTION(SR GAN)

Low-resolution (LR) images may be made to seem as good as high-resolution (HR) ones by a technique called Single Image Super-Resolution (SISR). Medical imaging, satellite imaging, security and surveillance, astronomy and image processing includes the critical Single Image Super-Resolution (SISR) issue. Due to the growing volume of high-resolution photographs and videos available online, there is a significant need for efficient storage, transmission, and sharing of this material while minimizing storage and bandwidth costs. Furthermore, the high-resolution photos are often resized to conveniently accommodate various display displays with varying resolutions.

The super-resolution issue involves the removal of noise and degradations caused by the use of a low-resolution acquisition method, ultimately leading to photographs of superior visual quality. A GAN-based technique inspired by the single image SR method(fig.5), is developed and evaluated to improve the resolution [9]. One potential option is to acquire low resolution (LR) pictures and use the Super Resolution GAN to generate a high resolution (HR) version [10]. In the majority of analysis pipelines, the data is resampled using interpolation or single image super-resolution methods. This is done after using a group of generative adversarial networks (GANs) to enhance the quality of the data and training them with various adversarial goals [11, 12]. An efficient Generative Adversarial Network (GAN) powered Super-Resolution (SR) model was used to produce undesirable content. The discriminator of the model was constructed using the Least Square Loss function [13]. Generator and VGG loss are calculated using eq.10, eq.11 and eq.12.

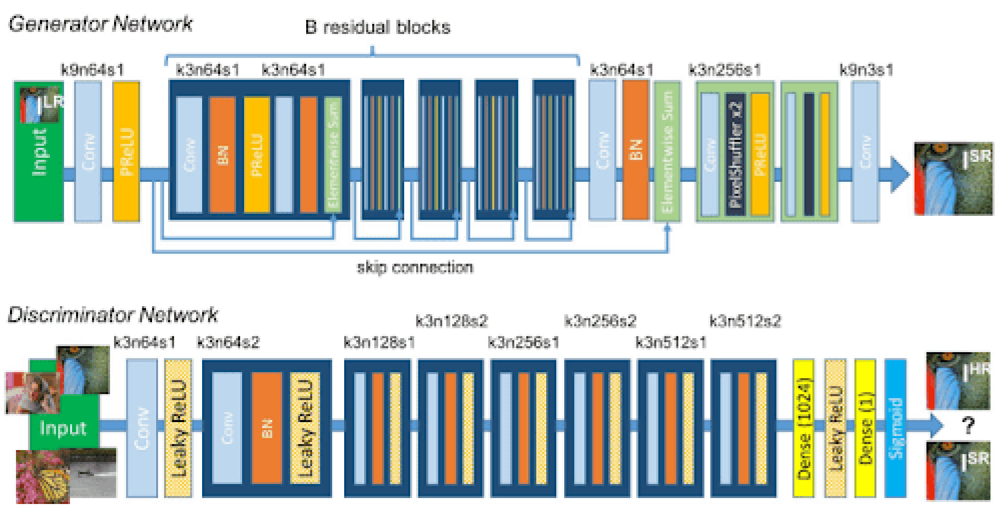


Fig5. Architecture of Super Resolution(Generator and Discriminator)[15]

$$l^{SR} = l_X^{SR} + 10^{-3}\, l_{Gen}^{SR} \tag{9}$$

$$l_{Gen}^{SR} = \sum_{n=1}^{N} - log\ D_{\theta_D}(G_{\theta_G}(I^{LR})) \tag{10}$$

$$\mathrm{l}_{\mathrm{VGG(i,j)}}^{\mathrm{SR}} = \frac{1}{\mathrm{W_{i,j}H_{i,j}}} \sum_{\mathrm{x=1}}^{\mathrm{W_{i,j}}} \sum_{\mathrm{y=1}}^{\mathrm{H_{i,j}}} (\emptyset_{\mathrm{i,j}}(\mathrm{I^{HR}})_{\mathrm{x,y}} - \emptyset_{\mathrm{i,j}}(\mathrm{I^{SR}})_{\mathrm{x,y}})^2 \tag{11}$$

$$l_{MSE}^{SR} = \frac{1}{r^2 WH} \sum_{x=1}^{rW} \sum_{y=1}^{rH} \left(I_{x,y}^{HR} - G_{\theta_G}(I^{LR})_{x,y}\right)^2 \tag{12}$$

2.6 CYCLE GAN

In order to construct a mapping between the image distributions or to transfer the features of one picture to another, cycle GAN is utilized. In Cycle GAN, the issue is approached as an image reconstruction task. Initially, we get a picture (p) as input and transform it into a reconstructed image using the generator G_0. Subsequently, we use a generator F to invert this procedure, transforming the rebuilt picture back into the original image. Next, we compute the mean square error loss by comparing the actual picture with the reconstructed image. The key characteristic of this cycle GAN is its ability to carry out image conversion for unpaired images, where there is no inherent correlation between the input and output images[14].

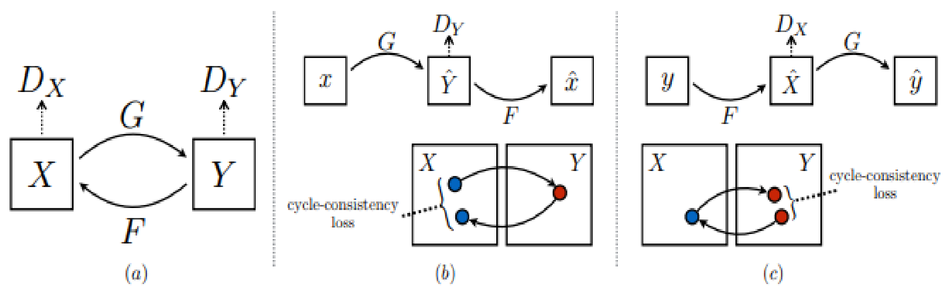


Fig.6Architecture of Cycle GAN[18]

Cycle GAN's design distinguishes itself from previous GANs by using two mapping functions (G_0 and G_1) as generators, along with their respective discriminators (D_p and D_q): The functions for mapping the generator are as follows:

$$\mathbf{G: P \rightarrow Q}$$
$$\mathbf{F : Q \rightarrow P}$$

P, Q represent the distribution of the input picture and the distribution of the intended output respectively. The related discriminators are:

D_p: differentiate between G_0 (Generated Output) and Q (actual Output)

D_q: Differentiate G_1(Q) (Generated Inverse Output) from P (Input distribution)

Cycle GAN is a computational framework used to transfer the unique characteristics of one picture to another, or to map the image distribution to a new distribution(fig.6). Within the framework of Cycle GAN, the issue is addressed as a job of reconstructing images. The first stage involves taking an input picture (p) and transforming it into the reconstructed image using the generator G_1. Afterwards, the previously described process is reversed by using a generator G_1 to convert the reconstructed picture back to its original state. Subsequently, the mean squared error loss is calculated by juxtaposing the actual picture with the reconstructed image. The remarkable characteristic of this cycle-GAN is its ability to perform image translation on unpaired images, where there is no inherent association among the input and as well as output images.

The distinguishing feature of Cycle GAN's design is in its incorporation of two mapping functions, labelled as G_0 and F, which operate as generators. Furthermore, there are also D_p and D_q discriminators that correlate to these. The generator is mapped by the P,Q functions. The discriminator is responsible for distinguishing between the produced output (G_0(P)) and the genuine output (Q). Cyclic and adversarial losses are calculated using eq.13 and eq.15

$$\mathrm{Loss}_{\mathrm{cyc}}\,(\mathrm{G},\mathrm{F},\mathrm{P},\mathrm{Q}) = \frac{1}{m}\sum_{i=1}^{m}[\mathrm{F}(\mathrm{G}(\mathrm{p_i})) - \mathrm{p_i}] + [\mathrm{G}(\mathrm{F}(\mathrm{q_i})) - \mathrm{q_i}] \tag{13}$$

$$\mathrm{Loss}_{\mathrm{full}} = \mathrm{Loss}_{\mathrm{adv}} + \lambda\,\mathrm{Loss}_{\mathrm{cyc}} \tag{14}$$

$$\mathrm{Loss}_{\mathrm{adv}}(\mathrm{G},\mathrm{D_q},\mathrm{P}) = \frac{1}{m}\sum_{i=1}^{m}\left(1 - \mathrm{D_q}(\mathrm{G}(\mathrm{p_i}))\right)^2 \tag{15}$$

$$\mathrm{Loss}_{\mathrm{adv}}(\mathrm{F},\mathrm{D_p},\mathrm{Q}) = \frac{1}{m}\sum_{i=1}^{m}\left(1 - \mathrm{D_p}(\mathrm{F}(q))\right)^2 \tag{16}$$

In this context, it is important to differentiate between the generated inverse output of the function G_0(Q) and the input distribution P. Even though the training loop looks complicated, it consists of four basic steps(fig.7):

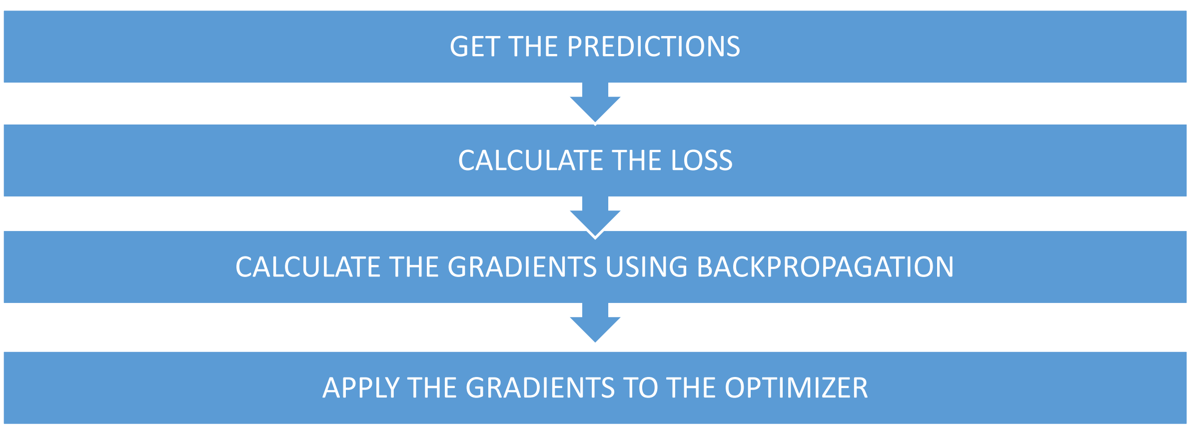


Fig.7 Flowchart of Cycle GAN training process

III. RESULTS AND DISCUSSION

The Python language and PyTorch framework were utilized to implement all the different variations. The models underwent online training in the Google Colab environment with GPU acceleration enabled. All three datasets photos are scaled to dimensions of 64×64×3. The batch size for each variant is set to 128 for all datasets. For all datasets, the value of Striding is consistently 2 in every experiment. The dimension of the latent vector is set to 100. The different versions were trained for a total of 20 epochs using all of the datasets.

The MNIST dataset is utilized for training the variational autoencoder. The MNIST dataset comprises handwritten digits and their corresponding labels. The dataset is divided into two segments: the training segment, which contains 60,000 samples, and the test segment, which has 10,000 cases. The images in the MNIST dataset are shrunk to dimensions of 28×28×1. The activation function used is RELU with a filter size of 3×3 and a buffer size of 1024. The use of Variational Autoencoders (VAE) enables the creation of generative models and the representation of latent space (as shown in Figure 9). VAEs are also less prone to overfitting. In epoch 20, the VAE has produced a variety of distinct digits. Figure 8 demonstrates that superior outcomes have been achieved after epoch 15. Put simply, this implies that the generator has acquired knowledge of the distribution and can be understood as the model reaching a state of convergence. The value of the loss function at epoch 1 is 217.7, while at epoch 5 it is 165.4. One drawback of VAEs is that, due to the introduced noise and incomplete reconstruction, the generated samples using the conventional decoder are significantly more fuzzy compared to other GANs

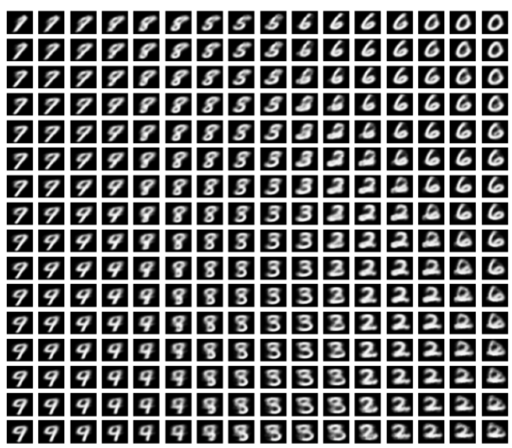


Fig8. Output image generated by Variational Auto Encoder

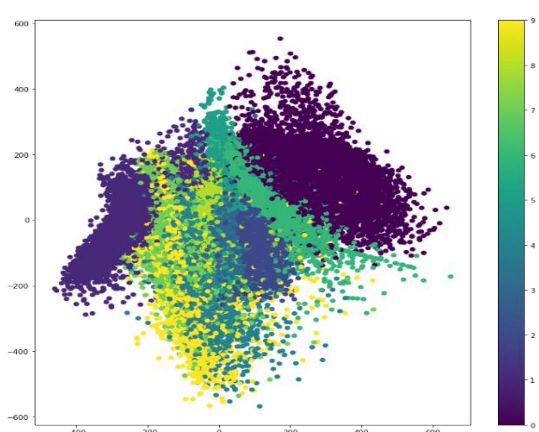


Fig.9 Latent space plot of Variational Auto Encoder trained for 20 epochs on MNIST dataset.

The Celeba dataset, specifically figure 10, is utilized for training the DCGAN. The dataset has 200,000 photos that are resized to dimensions of 64×64×3. The discriminator employs Leaky ReLU as the activation function in all layers, except for the last layer which utilizes the sigmoid function. The generator employs the ReLU activation function in all layers, except for the final layer which utilizes the Tangent Hyperbolic activation function. During the sixth epoch, the discriminator loss was 1.782 and the generator loss was 0.65. At epoch 55, the discriminator loss is 0.7 and the generator loss is 1.05, as depicted in Figure 11. DCGAN has demonstrated superior stability, ease of training, and, notably, the ability to generate high-quality outputs. Nevertheless, it continued to encounter the issues of mode collapse and non-convergence.

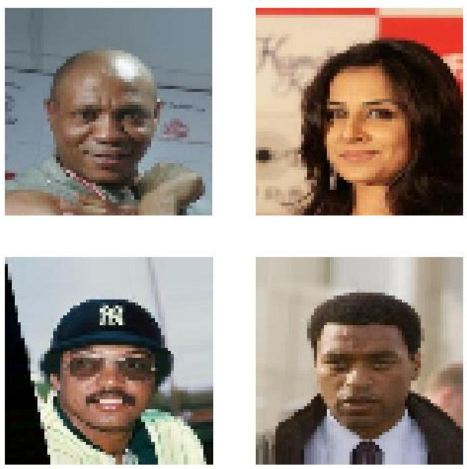


Fig10. Sample images of Celeba dataset to train the DCGAN for face generation

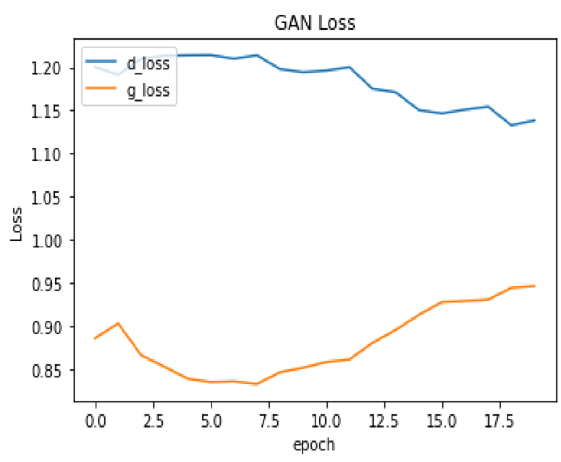


Fig11. Generator and discriminator loss plot of Deep Convolutional GAN

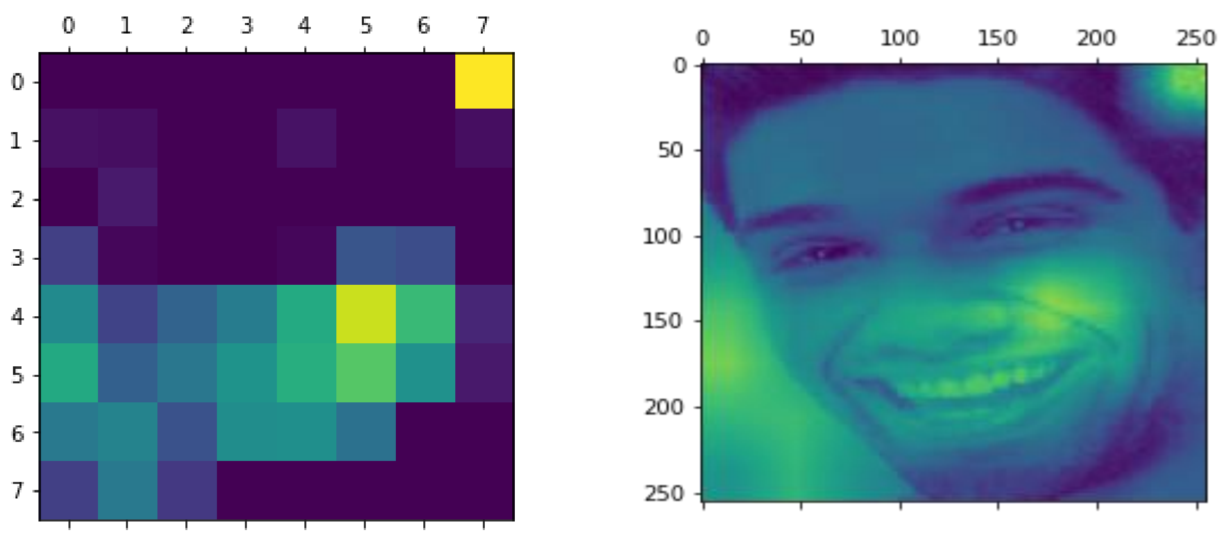
The WGAN is trained using the Celeba dataset, which is known for its high quality (see Figure 12). The dataset has 25,056 photos that have been resized to dimensions of 64×64×3. Within the discriminator, the loss curve of the WGAN (Fig.13) demonstrates convergence in contrast to the DCGAN loss curve. The discriminator loss for Epoch 6 is -9.59, whereas the generator loss is -2.93. At epoch 50, the discriminator loss is -6.4 and the generator loss is 10.56. It addresses the issues of the vanishing gradient and mode collapse. Exhibits a higher rate of convergence compared to DCGAN. WGAN was determined to exhibit much slower performance in comparison to the other GANs. The performance improvement of the model compared to the DCGAN in terms of generated outcomes is not evident. Weight clipping is employed to restrict the capacity of the models to acquire intricate distributions.

Fig12. Samples images of Celebahq dataset to train WGAN

Fig13. Discriminator and Generator loss plot of WGAN

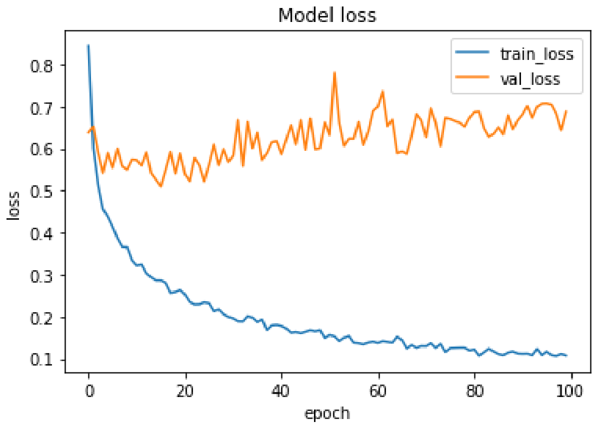
The emotion dataset is utilised for training the Pro GAN model(fig.14). The dataset has 6800 photos that have been resized to dimensions of 256×256. By employing this progressive training approach(fig.15), the model is able to attain unparalleled outcomes while maintaining superior stability throughout the whole training procedure. Progressive GAN training necessitates substantial memory capacity and abundant resources. The duration of the initial epoch was 27 seconds and the loss was 0.1770 (fig.16(a)). The accuracy of the Pro GAN model was 0.9260 and the top-k accuracy was 0.9858(fig.16(b)).

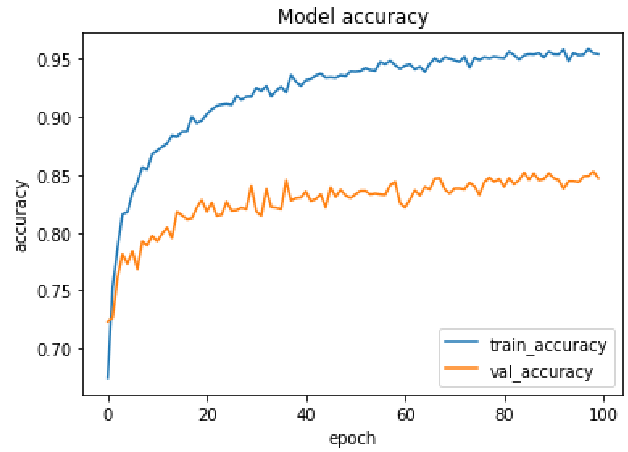
Fig.14 Sample images of emotion data set



15.(a) 15.(b)

Fig.15 Progressive growing layers of Pro GAN output





16.(a) 16.(b)

Fig.16 Model loss and model accuracy plots of Pro GAN

The SRGAN model is trained using the Celebahq dataset, which consists of 30,000 images. The photos in the dataset collection are downsized to dimensions of 256 by 256. The number of epochs used is 20, and the kernel size is 3. The activation layer employed is the hyperbolic tangent (tanh), the batch size is set to 128, and the image has a form of 64*64*3. The duration of the first epoch is 110 seconds, with a discriminator loss of 0.5162 and a generator loss of 0.3425. The duration of the 5th epoch is 422 milliseconds, with a discriminator loss of 0.0393 and a generator loss of 0.164. The output of super resolution GAN is shown in fig.17 along with low and high resolution images.

Fig.17 Low and high resolution output of SRGAN

The Cycle GAN model is trained using a face mask dataset(fig.18) consisting of 5000 images. The number of epochs required is 200. The image can be resized to 64x64x3 dimensions. The chosen activation function is leaky Rectified Linear Unit (ReLU), and the batch size is set to 128. During the initial epoch, the time required is 47 seconds. The generator loss for masked data is 10.1713, and for unmasked data it is 10.1601. The discriminator loss for masked data is 0.7018, and for unmasked data it is 0.6790. During the 5th epoch, the time required is 25 seconds. The generator loss for masked data is 7.5927, and the generator loss for unmasked data is 7.6371. On the other hand, the discriminator loss for masked data is 0.5121, and the discriminator loss for unmasked data is 0.5203. Cycle GAN transition for face mask removal application is shown in fig.19

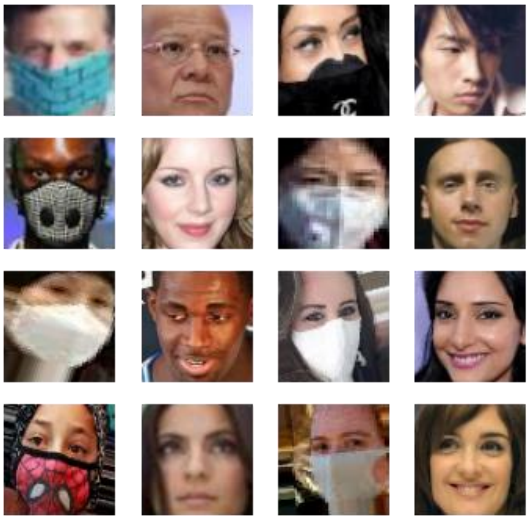


Fig.18 sample inputs of face mask dataset

Fig.19 Output of cycle GAN for face mask Removal application

Table 1 presents a quantitative examination of several types of GANs in terms of their losses and the time required. A qualitative study of several types of GANs, focusing on their applications, advantages, and limitations presented in table 2.

Table.1 Quantitative analysis of different types GAN's in terms of losses and time taken for epoch

	LOSS(for epoch 1)		Time taken for first epoch(seconds)
Type of GAN	Discriminator loss	Generator loss	
Variational auto encoder	217.77		5
DCGAN	1.9	0.4	20
WGAN	-10.8	-4	239
Pro GAN	0.177		27
SR GAN	0.34	0.51	110
CYCLE GAN	10.17	0.7	47

Table2: Qualitative analysis of different types of GANs

DIFFERENT GAN's	ADVANTAGES	DISADVANTAGES	APPLICATIONS
Pro GAN	It enables the generation of pictures with different resolutions while preserving image quality.	Training Pro GAN necessitates the use of high-performance GPUs or TPUs, significant memory capacity, and extensive computing resources.	Applicable to a range of activities involving the creation of images, including as generating art, altering images, and even synthesizing medical images.
WGAN	Convergent, More Stable, Easy to train	It requires long training time and may not faithfully compute the Wasserstein-1 distance in high dimensions	WGANs can be used for artistic style transfer, allowing users to apply the style of one image to another while preserving the content, Data augmentation, Anomaly Detection, Domain Adaptation.
SR GAN	It generates very authentic, high-definition pictures from low-resolution photographs.	Due to the model's need of a lower quality picture input, it is not suitable for the sample generation job.	Applications such as improving the visual quality of photographs by increasing their resolution or raising the resolution of medical imaging.
DC GAN	The capacity to produce superior pictures via the use of convolution layers.	A potential stability issue that may arise is known as mode collapse.	The system uses deep convolutional neural networks.
VARIATIONAL AUTO ENCODER	The primary objective is to acquire latent representations of data, reconstruct data distributions, and compute loss functions.	VAEs may provide challenges in training and need more computing resources. Furthermore, the acquired latent space representation may provide challenges in terms of interpretation, and the produced data's quality might be constrained by the model's architecture and training data.	It is used for tasks such as data reduction, picture production, and representation learning.
CYCLE GAN	It has the ability to transform photos from one domain to another while maintaining the original content.	It only acquires knowledge of one-to-one correspondences, although the majority of interactions across other fields tend to be more complex.	A proficient Generative Adversarial Network (GAN) driven Super-Resolution (SR) model was used to generate undesired material.

IV. CONCLUSION AND FUTURE WORK

The appeal of GANs lies not just in their capacity to produce lifelike patterns, but also in their promise for future progress in the discipline. The progress in deep learning methods may be attributed to subtle alterations made to the architectures of different GAN's. Generative Adversarial Networks have significant promise in both theoretical and computational domains, particularly when integrated with Deep Learning algorithms.

In order to achieve training stability and address these issues, numerous derivative models of GANs with enhanced loss functions have continuously arisen. This study primarily examines the mathematical foundations of the two kinds of loss functions employed by the original GAN. It also provides a comprehensive analysis of their respective advantages and disadvantages. The issue with GANs primarily stems from the choice of loss function employed. New variations of loss functions have been discovered to enhance the training process in GANs, surpassing the performance of traditional architectural GANs. Consequently, numerous alternative objective function GANs have been suggested. This work offers a comprehensive analysis of the different variations of loss functions in GANs, providing a mathematical understanding of each. Additionally, it offers a concise overview of their benefits, uses, and drawbacks. The paper conducted a comprehensive examination of various variants of Variational Auto Encoder, DC GAN, WGAN, Pro GAN, and SR GAN. These models were tested using MNIST, celeba, celebahq, Emotion, and face mask datasets. The analysis included both qualitative and quantitative evaluations of the acquired outcomes. An extensive analysis of the losses in different GANs has been carried out to identify the most appropriate method for a specific application.

REFERENCES

[1] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014) Generative Adversarial Nets. Proceedings of the 27th International Conference on Neural Information Processing Systems, 2, 2672-2680.

[2] Q. Xu, Z. Wu, Y. Yang and L. Zhang, "The difference learning of hidden layer between autoencoder and variational autoencoder," 2017 29th Chinese Control And Decision Conference (CCDC), Chongqing, China, 2017, pp. 4801-4804, doi: 10.1109/CCDC.2017.7979344.

[3] H. Hu, M. Liao, W. Mao, W. Liu, C. Zhang and Y. Jing, "Variational Auto-Encoder for text generation," 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 2020, pp. 595-598, doi: 10.1109/ITOEC49072.2020.9141571.

[4] H. Meng and F. Guo, "Image Classification and Generation Based on GAN Model," 2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), Taiyuan, China, 2021, pp. 180-183, doi: 10.1109/MLBDBI54094.2021.00042.

[5] M. Arjovsky, S. Chintala and L. Bottou, "Wasserstein GAN", *Proc. Int. Conf. Mach. Learn.*, vol. 70, pp. 214-223, Aug. 2017, [online] Available: http://proceedings.mlr.press/v70/arjovsky17a.html.

[6] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin and A. C. Courville, "Improved training of wasserstein GANs", *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, pp. 5767-5777, 2017, [online] Available: http://papers.nips.cc/paper/7159-improved-training-of-wassersteingans.pdf

[7] T. Karras, T. Aila, S. Laine and J. Lehtinen, "Progressive growing of GANs for improved quality stability and variation" in arXiv:1710.10196, 2017, [online] Available: https://arxiv.org/abs/1710.10196

[8] D. Kim, M. Kim, G. Kwon and D.-S. Kim, "Progressive face super-resolution via attention to facial landmark" in arXiv:1908.08239, 2019, [online] Available: https://arxiv.org/abs/1908.08239.

[9] Demiray, Bekir Z., Muhammed Sit, and Ibrahim Demir. "D-SRGAN: DEM super resolution with generative adversarial networks." SN Computer Science 2, no. 1 (2021): 1-11.

[10] Sood, Rewa, BinitTopiwala, KarthikChoutagunta, RohitSood, and MirabelaRusu. "An application of generative adversarial networks for super resolution medical imaging."In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 326-331.IEEE, 2018.

[11] Delannoy, Quentin, Chi-Hieu Pham, Clément Cazorla, Carlos Tor-Díez, Guillaume Dollé, Hélène Meunier, Nathalie Bednarek, Ronan Fablet, Nicolas Passat, and François Rousseau. "SegSRGAN: Super-resolution and segmentation using generative adversarial networks—Application to neonatal brain MRI." Computers in Biology and Medicine 120 (2020): 103755.

[12] Ren, Haoyu, Amin Kheradmand, Mostafa El-Khamy, Shuangquan Wang, DongwoonBai, and Ju ngwon Lee. "Real-world super-resolution using generative adversarial networks. "In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 436-437. 2020.

[13] Shamsolmoali, Pourya, MasoumehZareapoor, Ruili Wang, Deepak Kumar Jain, and Jie Yang. "G-GANISR: Gradual generative adversarial network for image super resolution." Neuro computing 366 (2019): 140-153.

[14] V. Sandfort, K. Yan, P. J. Pickhardt and R. M. Summers, "Data augmentation using generative adversarial networks (CycleGAN) to improve generalizability in CT segmentation tasks", *Sci. Rep.*, vol. 9, no. 1, Dec. 2019.

[15] Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, ´ Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi Twitter.

[16] Sharma, Parth. (2022). Brief History of Generative Adversarial Networks. 10.31224/2637.

[17] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. In arXiv [cs.LG]. http://arxiv.org/abs/1511.06434

[18] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv preprint arXiv:1703.10593, 2017.

[19]https://towardsdatascience.com/difference-between-autoencoder-ae-and-variational-autoencoder-vae-ed7be1c038f2

BIOGRAPHIES OF AUTHORS

	**Tammineni Shanmukhaprasanthi** obtained Bachelor's degree in Electronics and Communication Engineering from Jawaharlal Nehru Technological University Kakinada university college of Engineering Vizianagaram, Andhra Pradesh in 2017.She received M.Tech degree in Electronics and Communication Engineering from Jawaharlal Nehru Technological University Kakinada, Andhra Pradesh in 2021. She is currently pursuing Ph.D. degree with Andhra University Visakhapatnam, India. Her research interests include microstrip patch antenna design, VLSI circuit design, image inpainting in image processing. She can be contacted at email: prashanthitammineni.rs@andhrauniversity.edu.in
	**Gottapu Sasibhushana Rao** is Senior Professor in the Department of Electronics & Communication Engineering, Andhra University College of Engineering, Visakhapatnam, India. He is a senior member of IEEE, fellow of IETE, member of IEEE communication Society, Indian Geophysical Union (IGU) and International Global Navigation Satellite System (IGNSS), Australia. Prof. Rao was also the Indian member in the International Civil Aviation organization (ICAO), Canada working group for developing SARPS. He has published more than 250 Technical and research papers in different National/International conferences and Journals. His current research areas include Cellular and Mobile Communication, GPS, Biomedical and Signal Processing, Under Water Image Processing and Microwave Engineering. He can be contacted at email: sasigps@gmail.com.