International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 16, No.1 (Aug-24)

http://dx.doi.org/10.12785/ijcds/160150

Investigation of Different Generative Adversarial Networks Techniques for Image Restoration

Shanmukha Prasanthi[1], Madhuri Rayavarapu[2], Sasibhushana Rao[3] and Goswami[4]

[1,2,3] *Department of Electronics and Communication Engineering,Andhra University college of engineering, Visakhapatnam, India.*
[4]*Department of Electronics and Communication Engineering, Gayatri vidya parishad college of engineering for women, Visakhapatnam, India*

Received 28 Oct. 2023, Revised 24 Apr. 2024, Accepted 1 May. 2024, Published 1 Aug. 2024

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 they are 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. GAN has proven to be useful in various domains, such as object recognition, text synthesis, face ageing, image manipulation, image overpainting, image stitching, human pose synthesis, visual salience prediction, stenographic applications, and many more. Several researchers have investigated various types of GANs but comprehensive analysis and comparison of different types of recent GAN's like Deep Convolutional GAN, Wasserstein GAN,Auto Enocoder, Cycle GAN,Progressive GAN and Super Resoultion GAN have not been published so far in the literature. In this article, quantitative and qualitative analysis are carried on to evaluate their suitability of the GAN for a particular application.Examples of applications encompass texture production, facial reconstruction, facial recognition, high-resolution imaging, music composition, drawing creation, cosmetic enhancements, image transformation, voice synthesis, medical diagnostics, and video editing. No single GAN cannot fulfil the desired requirements for all the applications. The article concludes with a discussion of the possible uses of GANs 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.

1. INTRODUCTION

Unsupervised learning is a cornerstone of modern research in the field of artificial intelligence and complements the dominance of supervised machine learning methods. In this field, Generative Adversarial Networks (GANs) have proven to be a groundbreaking example that demonstrates the fusion of generative modelling and deep learning techniques, enabling advances in unsupervised learning. Wang et al. (2024) was a pivotal moment in demonstrating the effectiveness of GANs and led to their widespread application in supervised and unsupervised learning domains [1]. GANs work on the principle of adversarial training, in which two networks, the generator (G) and the discriminator (D), engage in strategic competition [2]. The discriminator learns to distinguish between real and synthetic data and thus acts as a binary classifier, while the generator attempts to generate data that can deceive the discriminator as shown in Fig:1. Since the pivotal work by Wang et al. in 2024, GANs have found widespread applications in various supervised and unsupervised learning domains. This paper aims to provide a comprehensive analysis and comparison of different types of recent GAN architectures, which has been lacking in the existing literature. The main objectives of this study are to investigate the mathematical foundations, advantages, and limitations of various GAN variants, including Variational Auto Encoder (VAE), Deep Convolutional GAN (DCGAN), Wasserstein GAN (WGAN), Progressive GAN (Pro GAN), Super Resolution GAN (SRGAN), and Cycle GAN providing insights into their potential applications and areas for future research and development. This research quantitatively evaluates and compares the performance of these GAN architectures in terms of losses, convergence, and computational time requirements and to qualitatively assess the suitability of each GAN variant for specific applications, such as image generation, image-to-image translation, super-resolution, and style transfer. The paper is structured as follows: Section 2 provides an overview of the different GAN architectures investigated

E-mail address: prashanthitammineni.rs@andhrauniversity.edu.in, madhurirayavarapu.rs@andhrauniversity.edu.in, sasigps@gmail.com, rajkumargoswami@gmail.com

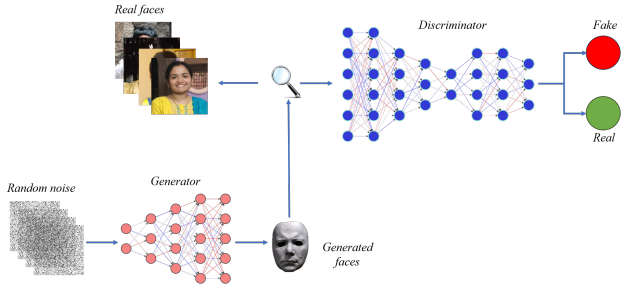

Figure 1. Structure of Generative Adversarial Network

in this study. Section 3 presents the experimental results, including quantitative analyses of losses and computational times, as well as qualitative evaluations of the generated outputs. Section 4 concludes the paper by summarizing the findings, highlighting the potential applications of GANs and discussing future research directions in this field.

2. VARIOUS TYPES OF GAN

Machine learning practitioners are increasingly using generative adversarial networks to enhance image processing. Applications that greatly benefit from the use of Generative Adversarial Networks 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.

A. 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 and the Variational Autoencoder 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. [3]. The Variational Autoencoder tackles the issue of uncontrolled latent space in Autoencoders, 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.

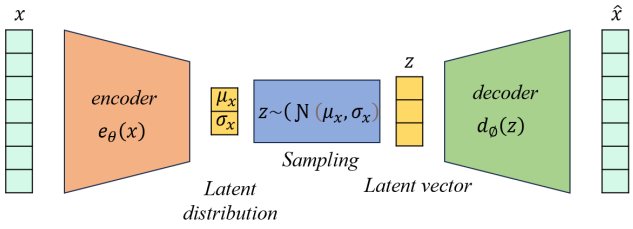

Figure 2. Architecture of variational auto encoder

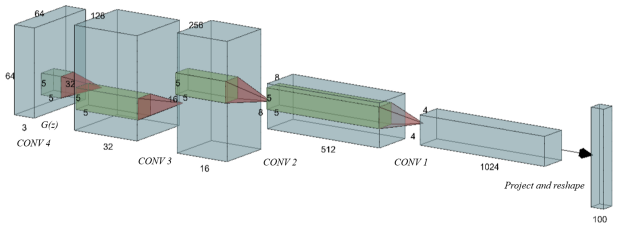

Figure 3. Architecture of DCGAN (Deep Convolutional Generative Adversarial Networks)

This limitation helps to guarantee that the latent space is properly controlled [4]. The encoder processes a picture and generates two vectors, which respectively reflect the average (μx) and variability (x) shown in Fig:2. The mean vector and the standard deviation vector are added together, with the standard deviation vector first being multiplied by a tiny random value () as noise. This results in a modified vector (Z) (eq.1) that has the same size. The loss function for a Variational Autoencoder 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 [5]. The KL divergence loss (Eq. 4) quantifies the discrepancy between the learned probability distribution and the predefined reference distribution.

$$z = \mu_x + \sigma_x \varepsilon, \varepsilon \sim \eta(0,1) \quad (1)$$

$$Reconstructionloss = \|x - y\|_2 = |\ x = d_\emptyset(\mu_x + \sigma_x \varepsilon)\|_2 \quad (2)$$

Where x is input and y‘ is reconstructed output

$$\mu_x, \sigma_x = e_\theta(x), \varepsilon \sim \eta(0, I) \quad (3)$$

$$SimilarityLoss = KLDivergence = D_K L((_{x,x})\|(O, I)) \quad (4)$$

$$Loss = ReconstructionLoss + SimilarityLoss \quad (5)$$

B. Deep Convolutional Generative Adversarial Networks (DCGAN)

DCGAN is a network design for Generative Adversarial Networks that has received a lot of attention for its efficiency shown in Fig:3. The architecture mostly comprises convolutional layers, excluding max-pooling or fully linked

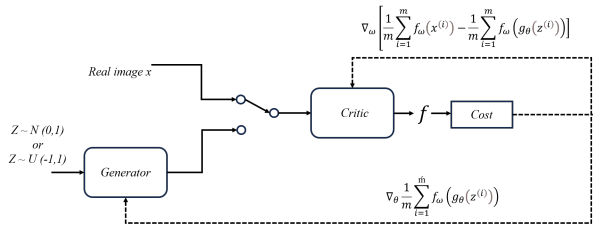

Figure 4. Architecture of wassertain GAN

layers [6]. The method employs convolutional stride and transposed convolution to reduce and increase the resolution, respectively. Generator and Discriminator loss are calculated using equation 6 and equation 7. The architecture of DCGAN involves the following steps: i)Replacing every occurrence of max-pooling with convolution stride. ii)Utilize transposed convolution for the purpose of up sampling. iii) Remove the layers that are fully interconnected. iv)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. v)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. vi)Use the Leaky Rectified Linear Unit activation function in the discriminator.

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

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

C. Wasserstein GAN(WGAN)

WGAN provides increased stability in training the model as compared to basic GAN designs [7]. The use of the loss function in WGAN(shown in Fig:4) provides a termination condition for assessing the model. Despite the 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 [8]. The mathematical formula is shown in equation 8.

$$I_critic(\omega) = maxE_(X \sim P_r)[f_\omega(x)] - E_(z \sim Z)[f_\omega(g_\theta(Z))] \tag{8}$$

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

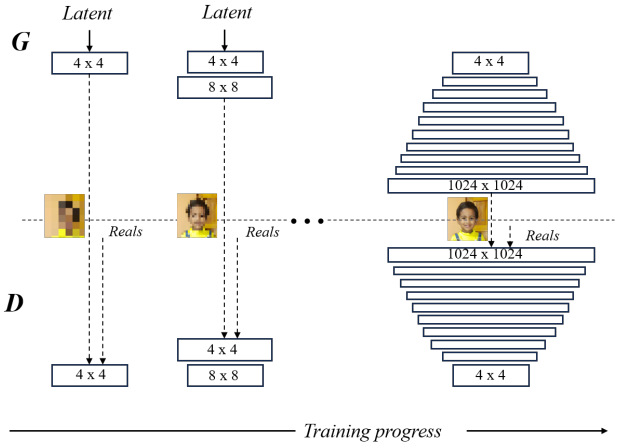

Figure 5. 4. Progressively adding layers in Generator and Discriminator

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.

D. 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 [9].

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. 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 [10]. The incremental generation approach (Fig:5) may use prior learning from completed subtasks to speed up the process of learning new subtasks.

E. 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,

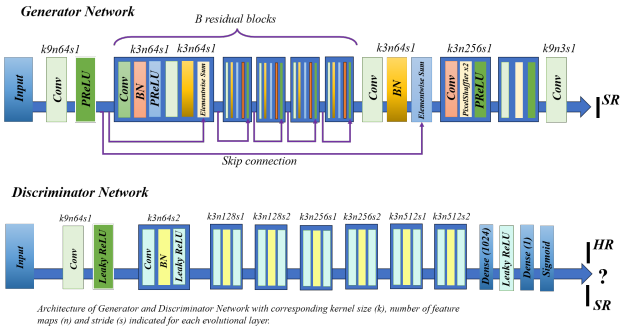

Figure 6. Architecture of Super Resolution (Generator and Discriminator

satellite imaging, security and surveillance, astronomy and image processing includes the critical Single Image Super-Resolution 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.

$$l^{SR} = l_x^{SR} + 10^{-3}.l_{gen}^{SR} \tag{9}$$

$$l_{gen}^{SR} = -\sum_{n=1}^{N-1} logD_{\theta D}(G_{\theta G}(I^{LR}) \tag{10}$$

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

$$l_{MSE}^{SR} = \frac{1}{r^2 W.H} \sum_{x=1}^{rW} \sum_{y=1}^{rH} .(I_{x,y}^{HR} - G_{\theta G}(I^{SR})_{x,y})^2 \tag{12}$$

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:6), is developed and evaluated to improve the resolution [11]. One potential option is to acquire low resolution pictures and use the Super Resolution GAN to generate a high-resolution version [12],[13]. This is done after using a group of generative adversarial networks to enhance the quality of the data and training them with various adversarial goals [14]. An efficient Generative Adversarial Network powered Super Resolution model was used to produce undesirable content. The discriminator of the model was constructed using the Least Square Loss function [15]. Generator and VGG loss are calculated using equations 10, 11 and 12.

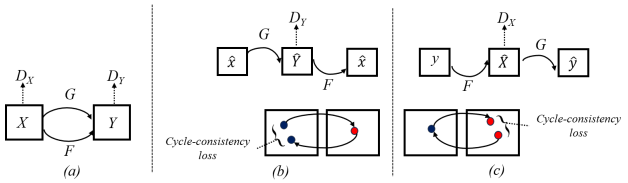

Figure 7. Architecture of Cycle GAN

F. 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 G0. Subsequently, a generator F to invert this procedure are used to transform the rebuilt picture back into the original image.

Next, computing 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 [16]. Cycle GAN's design distinguishes itself from previous GANs by using two mapping functions (G0 and G1) as generators, along with their respective discriminators (Dp and Dq). The functions for mapping the generator are G: P $\rightarrow Q, F : Q \rightarrow P$.

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

P,Q represent the distribution of the input picture and the distribution of the intended output respectively. The related discriminators are: Dp: differentiate between G0 (Generated Output) and Q (actual Output). Dq: Differentiate G1(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:7). 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 G1. Afterwards, the previously described process is reversed by using a generator G1 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 G0 and F, which operate as generators.

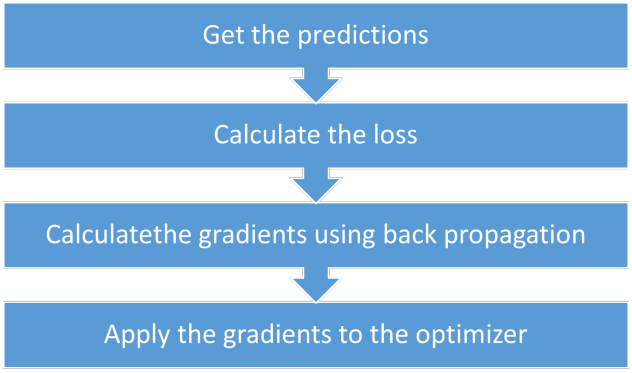

Figure 8. Flowchart of Cycle GAN training process

Furthermore, there are also Dp and Dq discriminators that correlate to these. The generator is mapped by the P, Q functions. The discriminator is responsible for distinguishing between the produced output (G0(P)) and the genuine output (Q). Cyclic and adversarial losses are calculated using equations 15 and 16.

$$Loss_{cyc}(G,F,P,Q) = \frac{1}{m}\sum_{i=1}^{m}[F(G(p_i)) - p_i] + [G(F(q_i)) - q_i] \quad (14)$$

$$Loss_{full} = Loss_{adv} + \lambda Loss_{cyc} \quad (15)$$

$$Loss_{adv}(G,D_q,P) = \frac{1}{m}\sum_{i=1}^{m}(1 - D_q(G(p_i)))^2 \quad (16)$$

$$Loss_{adv}(F,D_p,Q) = \frac{1}{m}\sum_{i=1}^{m}(1 - D_p(F(q)))^2 \quad (17)$$

In this context, it is important to differentiate between the generated inverse output of the function G0(Q) and the input distribution P. Even though the training loop looks complicated, it consists of four basic steps shown in fig:8.

3. 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 Fig:10). VAEs are also less prone to overfitting. In epoch 20, the VAE has produced a variety of distinct digits. Fig:9 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

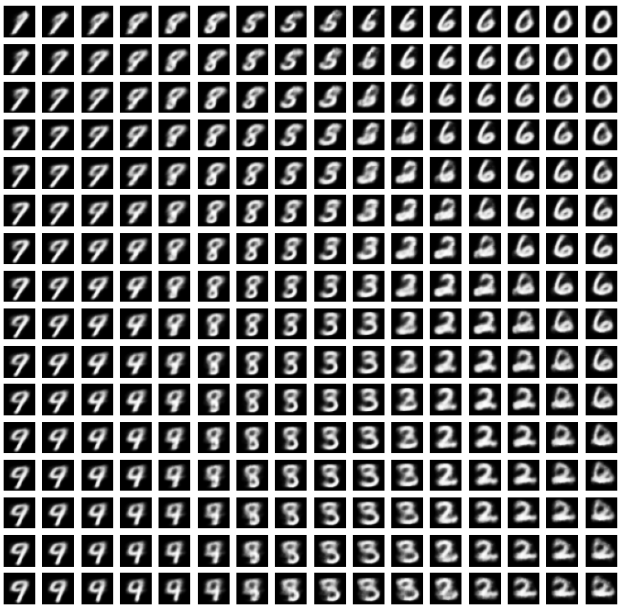

Figure 9. Output image generated by Variational Auto Encoder

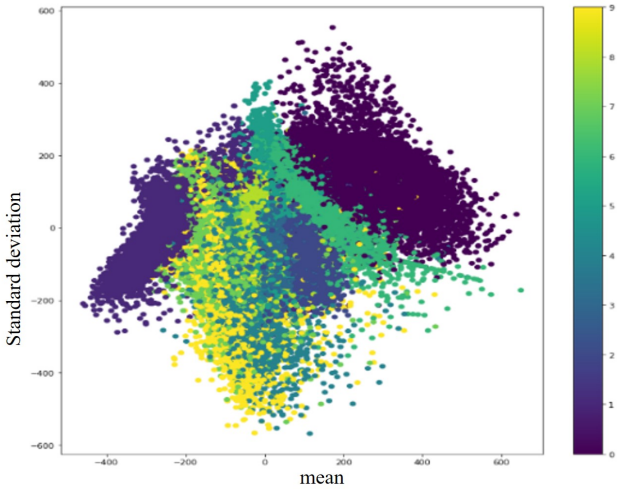

Figure 10. Latent space plot of Variational Auto Encoder trained for 20 epochs on MNIST dataset.

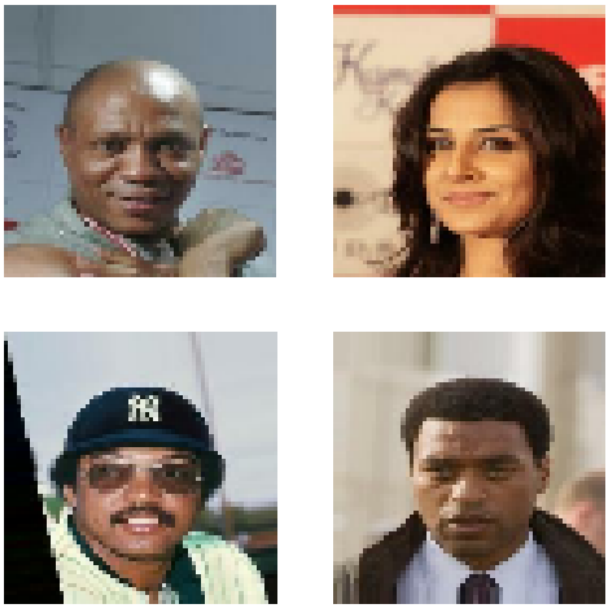

Figure 11. Sample images of Celeba dataset to train the DCGAN for face generation

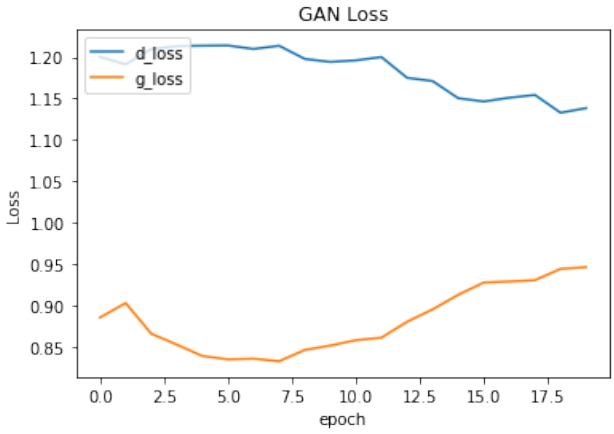

Figure 12. Generator and discriminator loss plot of Deep Convolutional GAN

Figure 13. Samples images of Celebahq dataset to train WGAN

Figure 14. Discriminator and Generator loss plot of WGAN

noise and incomplete reconstruction, the generated samples using the conventional decoder are significantly fuzzier compared to other GANs.The Celeba dataset, specifically Fig:11 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 Fig:12. 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. The WGAN is trained using the Celeba dataset, which is known for its high quality (see Fig:13). 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:14) 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

Figure 15. Sample images of emotion data set

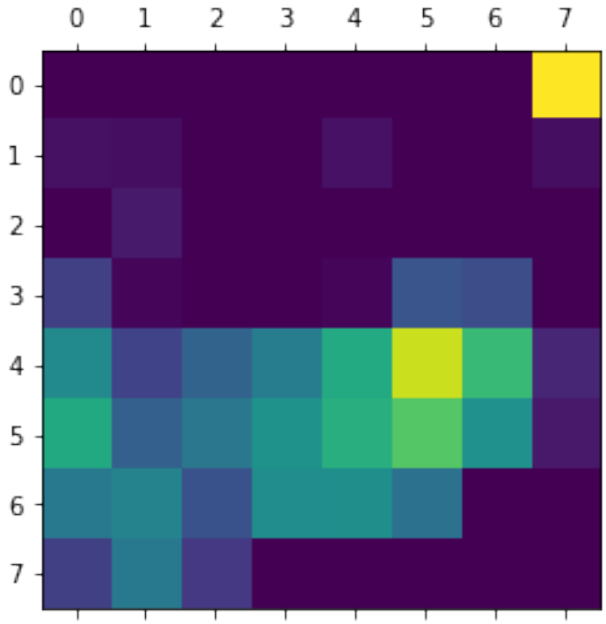

Figure 16. Progressive growing layers of ProGAN

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. The emotion dataset is utilized for training the Pro GAN model (Fig:15). The dataset has 6800 photos that have been resized to dimensions of 256×256. By employing this progressive training approach (Fig:16), 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. Output of proGAN is shown in Fig:17 The duration of the initial epoch was 27 seconds and the loss was 0.1770 as per Fig:18. The accuracy of the Pro GAN model was 0.9260 and the top-k accuracy was 0.9858 as shown in Fig:19.

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 64x64x3. 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:20 along with low and high resolution images.

The Cycle GAN model is trained using a face mask dataset(Fig:21) consisting of 5000 images. The number of epochs required is 200. The image can be resized to

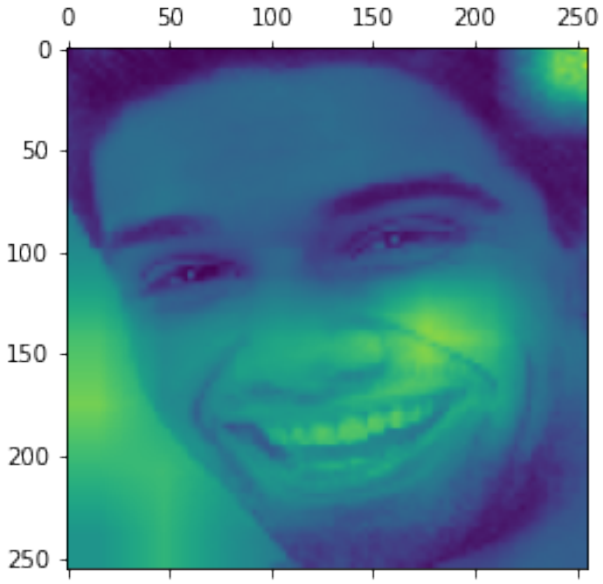

Figure 17. ProGAN output

64x64x3 dimensions. The chosen activation function is 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

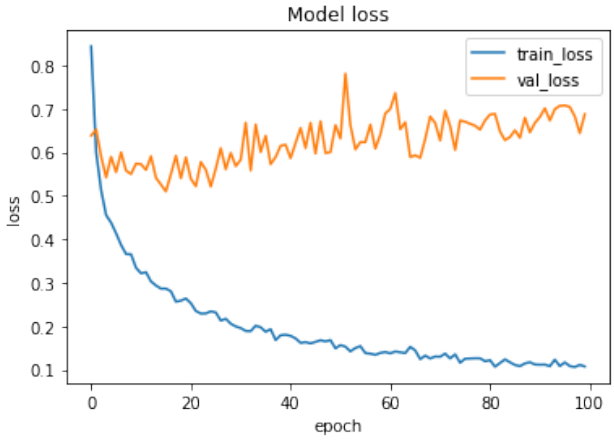

Figure 18. Model loss plot of ProGAN

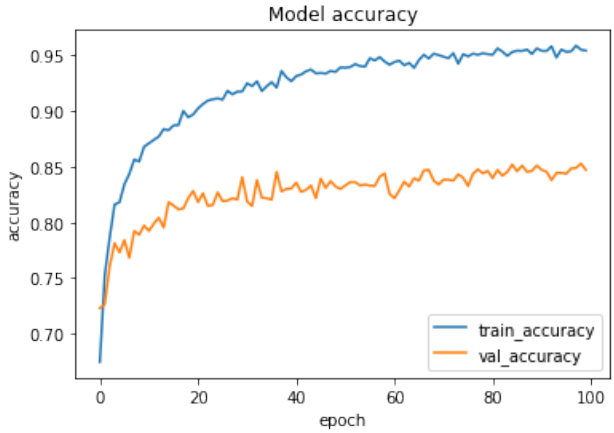

Figure 19. Model accuracy plot of ProGAN

Figure 20. Low and high resolution output of SRGAN

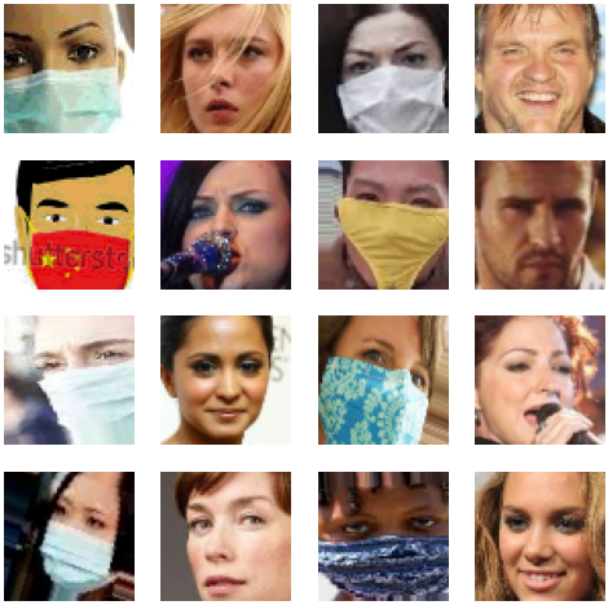

Figure 21. sample inputs of face mask dataset

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:22.

Tab: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 Tab:2.

4. 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. New variations of loss functions

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

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

TABLE II. Qualitative analysis of different types GAN's

DIFFERENT GAN's	ADVANTAGES	DISADVANTAGES	APPLICATIONS
Variational auto encoder	The main goal is to obtain hidden representations of data, rebuild the distributions of data, and calculate loss functions.	Training Variational Autoencoders (VAEs) might present difficulties and require more computational resources. Moreover, the obtained latent space representation may pose difficulties in terms of interpretation and the quality of the generated data could be limited by the architecture of the model and the training data.	It is utilized for activities such as data compression, image generation, and feature extraction.
DCGAN	The ability to generate high-quality images through the utilization of convolution layers.	An issue that could potentially affect stability is referred to as mode collapse.	The system employs deep convolutional neural networks.
WGAN	Convergent, more stable, and easily trainable.	The computation of the Wasserstein-1 distance in large dimensions may not be accurate and needs a significant amount of training time.	Wasserstein Generative Adversarial Networks can be utilized for artistic style transfer, enabling users to apply the style of one image to another while maintaining the original content. Additionally, WGANs can be employed for data augmentation, anomaly detection, and domain adaptation.
Pro GAN	It allows for the creation of images with varying resolutions while maintaining image fidelity.	Training Pro GAN requires the utilization of powerful GPUs or TPUs, substantial memory capacity, and adequate processing resources.	Applicable to various tasks encompassing picture creation, such as art generation, image manipulation, and even medical image synthesis.
SR GAN	It produces highly realistic, high-resolution images from low-quality photographs.	Because the model requires a lower resolution image as input, it is unsuitable for the task of generating samples.	Applications include enhancing the visual clarity of images by augmenting their resolution or elevating the resolution of medical imaging.
CYCLE GAN	It possesses the capability to convert images from one domain to another while preserving the original content.	It solely obtains information about one-to-one relationships, while most interactions in other domains tend to be more complicated.	A sophisticated Generative Adversarial Network (GAN) based Super-Resolution (SR) model was utilized to produce inappropriate material.

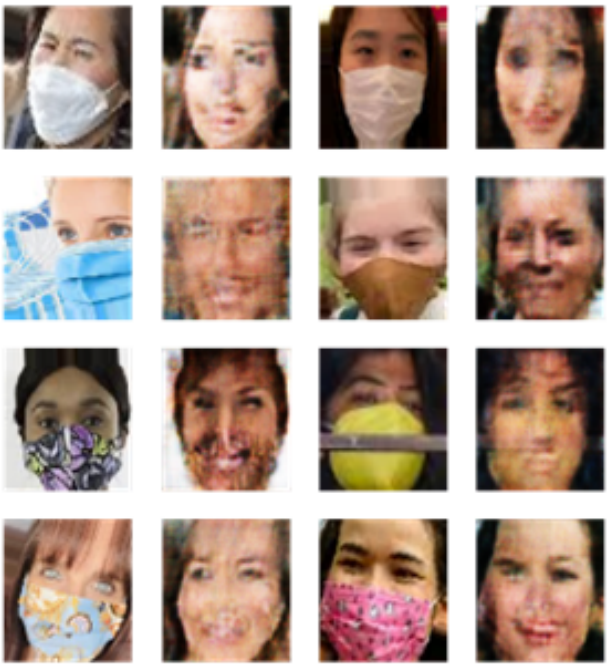

Figure 22. Output of cycle GAN for face mask Removal application

have been discovered to enhance the training process in GANs, surpassing the performance of traditional architectural GANs. This work presents a comprehensive analysis of the different variations of loss functions in GANs, providing a mathematical understanding of each. Additionally, it gives a concise overview of their benefits, uses, and drawbacks. Different GANs have various advantages and disadvantages. Pro GAN allows for image creation with varying resolutions while maintaining image fidelity, but requires powerful GPUs or TPUs. WGANs are convergent, stable, and easily trainable, but may not be accurate in large dimensions. SR GAN produces high-resolution images from low-quality photographs, but is unsuitable for sample generation. DC GAN generates high-quality images through convolution layers, but mode collapse can affect stability. Variational Autoencoders aim to obtain hidden representations of data, but may present difficulties and require more computational resources. CYCLE GAN converts images between domains while preserving original content, but only obtains information about one-to-one relationships. The research 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] S. M. K. Sistla, S. Jangoan, and I. A. Mohamed, "A state-of-the-art review on image synthesis with generative adversarial networks," *American Journal of Computing and Engineering*, vol. 7, no. 1, pp. 46–60, 2024.

[2] Y. Wang, M. Pang, S. Chen, and H. Rao, "Consistency-gan: Training gans with consistency model," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 14, 2024, pp. 15 743–15 751.

[3] Z. Wu, L. Cao, and L. Qi, "evae: Evolutionary variational autoencoder," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[4] K. Berahmand, F. Daneshfar, E. S. Salehi, Y. Li, and Y. Xu, "Autoencoders and their applications in machine learning: a survey," *Artificial Intelligence Review*, vol. 57, no. 2, p. 28, 2024.

[5] M.-S. Yu, T.-W. Jung, D.-Y. Yun, C.-G. Hwang, S.-Y. Park, S.-C. Kwon, and K.-D. Jung, "A variational autoencoder cascade generative adversarial network for scalable 3d object generation and reconstruction," *Sensors*, vol. 24, no. 3, p. 751, 2024.

[6] V. Teh, J. C. Y. Loong, E. D. H. Wai, L. J. Yang, C. J. Cheng, and Z. A. bin Abdul Salam, "Enhancing neural network models for mnist digit recognition," *Journal of Applied Technology and Innovation (e-ISSN: 2600-7304)*, vol. 8, no. 1, p. 15, 2024.

[7] S. Sabnam and S. Rajagopal, "Application of generative adversarial networks in image, face reconstruction and medical imaging: challenges and the current progress," *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, vol. 12, no. 1, p. 2330524, 2024.

[8] A. Roy and D. Dasgupta, "A distributed conditional wasserstein deep convolutional relativistic loss generative adversarial network with improved convergence," *IEEE Transactions on Artificial Intelligence*, 2024.

[9] G. Sun, G. Peng, X. Tian, L. Li, Y. Zhao, and Y. Wang, "A data retrieval method based on agcn-wgan," in *2024 IEEE 4th International Conference on Power, Electronics and Computer Applications (ICPECA)*, 2024, pp. 13–17.

[10] S. Raj, J. Mathew, and A. Mondal, "Generalized and robust model for gan-generated image detection," *Pattern Recognition Letters*, 2024.

[11] S. R. Gani, H. Spoorthi, and M. Rafi, "A review on super resolution algorithms and applications."

[12] W. Wu, S. Wang, W. Chen, Z. Qi, Y. Zhao, C. Zhong, and Y. Chen, "Computational integral imaging reconstruction based on generative adversarial network super-resolution," *Applied Sciences*, vol. 14, no. 2, p. 656, 2024.

[13] X. Lu, X. Xie, C. Ye, H. Xing, Z. Liu, and C. Cai, "A lightweight generative adversarial network for single image super-resolution," *The Visual Computer*, vol. 40, no. 1, pp. 41–52, 2024.

[14] J. Li, Z. Pei, W. Li, G. Gao, L. Wang, Y. Wang, and T. Zeng, "A systematic survey of deep learning-based single-image super-resolution," *ACM Computing Surveys*, 2024.

[15] H. Xiao, X. Wang, J. Wang, J.-Y. Cai, J.-H. Deng, J.-K. Yan, and Y.-D. Tang, "Single image super-resolution with denoising diffusion gans," *Scientific Reports*, vol. 14, no. 1, p. 4272, 2024.

[16] X. Ma, "A comparison of art style transfer in cycle-gan based on

different generators," in *Journal of Physics: Conference Series*, vol. 2711, no. 1. IOP Publishing, 2024, p. 012006.

Tammineni Shanmukha Prasanthi She is a Ph.D. research scholar in the Department of Electronics and Communication, Andhra University. Visakhapatnam, India. She received M.Tech. degree in Electronics and Communication Engineering from JNTU Kakinada, Andhra Pradesh in 2021. Her research interests include microstrip patch antenna design, VLSI circuit design, image inpainting in image processing. ORCID ID: 0009-0000-5352-2265.

Swarajya Madhuri Rayavarapu She is a Ph.D. research scholar in the Department of Electronics and Communication, Andhra University. Her Research interests include Deep Learning, GAN (Semi-supervised Machine Learning) in medical Image Processing, Applying deep learning techniques to 5G-Mobile Communication (Layer 2 of RAN). ORCID ID: 0000-0001-6346-8274

Gottapu Sasibhushana Rao He is the Principal of Andhra University, College of Engineering (A) and also a Senior Professor in the ECE Department of 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. His current research areas include Cellular and Mobile Communication, GPS, Biomedical and Signal Processing, Under Water Image Processing and Microwave Engineering. ORCID ID: 0000-0001-6346-8274

Rajkumar Goswami He completed his M.Tech in Radar and Communications from IIT Delhi. Dr. RK Goswami subsequently, completed his PhD in Electronic and Communication Engineering from Andhra University during which, he designed the Forward error Correction Schemes in respect of multipath channel His areas of interests include Computer Networks, Signal Processing, Image Processing, Artificial Intelligence and Software Engineering. ORCID ID: 0000-0002-0651-6783