Journal of Robotics and Control (JRC)
Volume 6, Issue 1, 2025
ISSN: 2715-5072, DOI: 10.18196/jrc.v6i1.24160

A Review on Comparative Analysis of Generative Adversarial Networks' Architectures and Applications

Ranjith Bhat [1*], Raghu Nanjundegowda [2]
[1] Research Scholar, Faculty of Engineering and Technology, JAIN (Deemed to be University), Bengaluru, India
[1] Assistant Professor, Department of Robotics and AI Engineering, NMAM Institute of Technology, NITTE (Deemed to be University), Nitte, India
[2] Associate Professor, Department of Electrical and Electronics Engineering, JAIN (Deemed to be University), Bengaluru, India
Email: [1] ranjithbhat@gmail.com, [1] ranjith.bhat@nitte.edu.in, [2] raghu1987n@gmail.com
*Corresponding Author

***Abstract*—Generative Adversarial Networks (GANs) are a major advancement in generative modeling, surpassing traditional machine learning models in tasks such as image generation, super-resolution, and image-to-text translation. A GAN consists of two neural networks: a Generator (G), which creates data from noise or a latent vector, and a Discriminator (D), which determines whether the data is real or generated. These networks train competitively, improving each other iteratively to produce increasingly realistic outputs. However, GANs face challenges like mode collapse, unstable training, and convergence issues, leading to the adoption of strategies such as instance normalization and enhanced loss functions. Future research can focus on improving stability, developing novel loss functions, and applying GANs in unsupervised learning. Performance metrics like Inception Score, Fréchet Inception Distance (FID), and Structural Similarity Index (SSIM) are essential for evaluating and comparing GAN architectures. Additionally, ethical concerns, including the misuse of GANs for deepfakes and synthetic data, underscore the importance of transparency, accountability, and ethical standards in research and deployment. This review provides an accessible introduction to GANs for novice researchers, along with a detailed analysis of their limitations, applications, and future prospects, offering valuable insights and direction for advancing this field.**

Keywords*—Artificial Intelligence (AI); Computer Vision (CV); Deep Learning; Generative Adversarial Networks (GAN); Image Synthesis.

I. Introduction

Generative Adversarial Networks (GANs) have transformed the domain of Artificial Intelligence by facilitating the creation of remarkably realistic data, including images, text, and speech. These advancements are based on machine learning and deep learning technologies, enabling computers to emulate human cognitive processes and decision-making. Advances in machine learning (ML) have led to the development of strong generative models that can produce realistic data. Among these, GANs have drawn a lot of interest due to their capacity to use competitive learning between two neural networks to produce high-quality, lifelike data. Many algorithms have been built throughout the years for developing systems that can mimic the human brains. Also, these algorithms have been implemented using a number of programming languages. Convolutional Neural Networks (CNNs), in particular, are Deep Learning models that use numerous layers to learn how to extract features directly from raw input. Throughout the training process, these models automatically recognize hierarchical characteristics, ranging from low-level patterns (like edges) to high-level structures (like objects or faces). Use of high-level features to handle unstructured data [1] is perhaps Deep Learning's greatest asset. Since Deep Learning uses generative modelling to develop unstructured data, like new images or texts, it has a huge impact in the field of generative networks.

The domain of generative modelling has undergone substantial transformations, with preliminary models serving as a basis for the advancement of sophisticated methodologies such as Generative Adversarial Networks. Generative models like Boltzmann Machines, Gaussian Mixture Models, and Hidden Markov Models were commonly utilized to generate structured data before the advent of GANs [2]. However, these conventional models were not capable of producing high-dimensional, high-quality outputs, and they significantly depended on assumptions about the underlying data distributions. The first introduction to GAN was first proposed by the researcher Ian Goodfellow of Google Brain in 2014 [2]. Numerous algorithms have been proposed as a result of advancements in generative AI. A generative based modelling particularly in the image domain has made a significant progress since the mid-2018. Examples of language modelling like Google's "BERT (Bidirectional Encoder Representations from Transformers)", "GPT-3 (Generative Pretrained Transformer-3)" [3], synthesis of speech using Parallel WaveGAN, and musing composition [4]-[7] using MuseNet have dominated in their respective fields. These advancements are not limited to image generation but also extend to language modelling, speech synthesis, and even music composition. However, the image generation domain has a significant contribution from various GAN models like StyleGAN, BigGAN, (all explained further). NVIDIA's StyleGAN is able to generate images [8] of human faces that look very realistic [9]. A GAN consists of two neural networks a Generator (G) and a Discriminator (D). The random noise during the commencement of the training is converted to a real sample by the Generator [10], while the Discriminator decides if the sample is real or artificially produced by the Generator [11]. The important components of a GAN are shown in the Fig. 1.

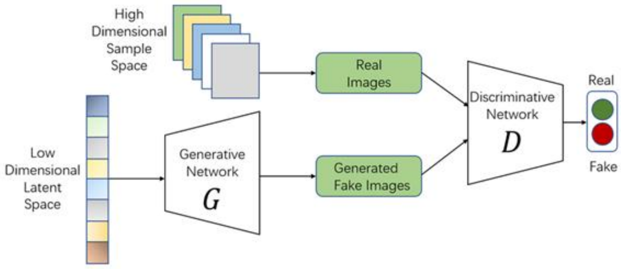

Fig. 1. Components of GAN's training algorithm [2]

Since Generative Adversarial Networks are trained on losses, taking losses into account is another crucial component of GAN training. Initially, a random selection is made from the training set to choose an actual sample. Before the Discriminator is trained, the generator output is collected into a training set. Discriminator's objectives for real images are "1," whereas the targets for generated images are "0." Real images offer values close to "1," whereas the fake offers "0".

Training GANs encompasses several fundamental concepts that enhance their performance and stability. Binary cross-entropy, a prevalent loss function, quantifies the divergence between expected and actual outputs, directing networks towards optimal performance. The sigmoid activation function assists the Discriminator by converting outputs into probabilities ranging from 0 to 1, enabling it to differentiate between genuine data and produced samples. To tackle prevalent issues, like as mode collapse and unstable convergence, sophisticated methods like instance normalization and minibatch discrimination are utilized. Instance normalization standardizes inputs across batches to facilitate the learning process, whereas minibatch discrimination promotes variation by ensuring the Generator yields various outputs throughout the rounds by considering only samples with a stipulated batch size. These strategies jointly augment the resilience and dependability of GAN training, facilitating the production of realistic and high-quality outputs.

The binary cross entropy between the output of the Discriminator and the goal "1" is the loss function. The target is a binary value, and the sigmoid activation function is applied to a single output unit [12]. The Discriminator's weight should be frozen during GAN training so that only the Generator's weight is modified. If not, it is changed to accept the image that was generated as real. It is highly recommended to use random noise when making the graphics. The letter z stands for random noise. The noise-generated images are in the $G(z)$. The most prevalent input is gaussian noise with a normal distribution. In order to train the GAN, both networks must be incrementally updated and tweaked recursively. The $D(X)$ for any input image X denotes a "unit probability for authenticity" and a "zero probability" for falsification. Fitting the real data distributions of $p_{data}(x)$ and $p_G(x)$ is the aim of generative modelling. Therefore, it's crucial to reduce differences between two distributions while training generative models.

A. Contribution and the Criteria of the Paper

The following are the paper's contributions:

- Examination of how various GAN architectures and their objective functions (like cross-entropy and Wasserstein loss) impact training stability, performance, and the quality of generated data.
- Challenges in GAN training with plausible counter measures (in section V).
- Providing a variety of GAN application instances to develop services, for better performances.

The articles which are referred for writing this paper are as per the following criteria:

- As per the chronology of appearance of a specific topic.
- Paper having good relevance and impact factor.

II. Perspective Study of Generative Adversarial Networks (GAN)

A GAN architecture competes two neural layers to create new data that closely resembles real data and is likely to be used as inputs. GANs excel at image processing, making them ideal for the field. GAN's training process should be closely monitored as Generator losses are used as a substantial feedback in generating the images of the next iteration. Among GAN's many challenges, its objective function is crucial. Using an unstable function may cause the GAN to lose control during training. Loss fluctuation is quite slow and even prevent the convergence of GAN [13]. This section covers GAN architecture principles, goal functions, latent space, and issues. GAN relies on the two-player least-max null-sum game for better optimization and training results.

A. WGAN (Wasserstein Generative Adversarial Networks)

The learning technique used by the Wasserstein GAN (WGAN) is unsupervised with a GAN Discriminator architecture. Wasserstein Loss, which correlates the sample quality to Generator convergence, was introduced by WGAN [14]. Earth Mover distance is the technique used for gradient update which provides meaningful gradients, improve training stability, reduce mode collapse, and better align generated and real data distributions. The WGAN loss is crucial because it links the Generator's convergence to the sample's quality. The output prediction p_i need not be restricted from 0 to 1, but could occupy a wide range from -∞ to ∞, and for binary cross-entropy loss used are y_i = *1*, y_i = -1 instead of y_i = 1, y_i = 0, as sigmoid activation was not used here [13]. WGAN's Discriminator was very critical in nature and the Wasserstein loss function of this GAN can be written as per (1).

$$-\frac{1}{n}\sum_{i=1}^{n}(\,y_i\,p_i) \tag{1}$$

Also, $p_i = D(G(z_i))$ for the synthesised image and the respective target y_i=*-1* is then compared for the calculation of the loss. The loss can be minimised using (2).

$$min_D - (E_{x\sim p_x}[D(x)] - E_{z\sim p_z}[D(G(z))]) \quad (2)$$

The differentiator will try to maximize the loss by making the real image score better. Comparing the prediction is a major and significant part of the Generator training by lowering the loss function for better results. By making sure that the Generator is updated properly and properly training the Discriminator, we can make the Wasserstein loss function converge better. In contrast to the original GAN, it is crucial to look out for a too strong Discriminator in this case. Training the Discriminators and Generators can be balanced using Wasserstein loss. Every time the Generator is updated, the Discriminator is typically updated five times. The reduction of reviewers' weights by WGAN has significantly slowed learning [15]. If the gradient is off, the weight update process becomes challenging. As a result, 'WGAN-GP (Gradient Penalty)' was developed [16].

B. *WGAN-GP (Wasserstein GAN-Gradient Penalty)*

The WGANGP uses unsupervised learning for Lipschitz constraint architecture. Instead of clipping weights, WGANGP uses input critic function [16]. Same critic function updates gradients. The WGANGP improved greatly over the WGAN. GAN training is reliable with WGAN-GP fixing mode collapse and unstable training. WGAN-GP's critic loss function penalizes gradients [17]. Criticism shouldn't batch normalize the process and maintain its weight. Loss like gradient penalty is negligible because "batch normalization" corelates images [18] also the weights enforce the Lipchitz constraint on this model, minimizing the gradient penalty term. Only part of the gradient is calculated by WGAN-GP. Gradients disappear or explode without careful clipping threshold adjustment due to cost function-weight restriction interactions, making WGAN optimization difficult. It is challenging to identify the gradient everywhere during the training phase. The gradient is only partially computed by WGAN-GP. We discover that interactions between the weight restriction and the cost function make the WGAN optimization process difficult because they might cause gradients to explode or disappear if the clipping threshold is not adjusted carefully. It was demonstrated that weight clipping in WGAN had problems, so a penalty for criteria loss was included as a substitute that did not have the same problems. With this strategy, stability was established across a range of designs and a good modelling performance was demonstrated. With a more dependable training approach for GANs, the work was anticipated to pave the door for enhanced modelling skills on large-scale image and language data.

C. *SAGAN (Self-Attention GANs)*

For sequence models like transformers, attention is an algorithm that is employed. In the SAGAN model, the attention method is applied to GAN [19]. The Fig. 2 shows architecture of the self-attention algorithm. Here the feature mapped image is further convoluted using a 1×1 kernel to further produce an attention map. The self-attestation is possible to be achieved in the last stage wherein the same kernel is used again.

GAN convolution feature maps without attention can only process local data. Accurate location data replaces high-level features in this approach. Hence the model cannot learn the relationship between distant pixels. SAGAN focus slightly resembles human perception as shown in Fig. 3.

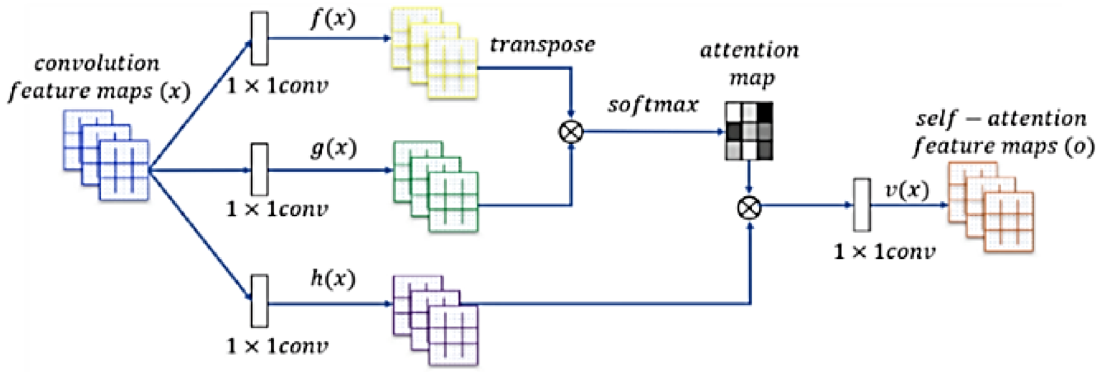

Fig. 2. Algorithm of Self-attention, SAGAN ('Self-Attention Generative Adversarial Networks') [19]

Fig. 3. Image generated by SAGAN and its attention map [19]

D. DCGAN (Deep Convolutional GANs)

DCGAN also uses a fully unsupervised learning method which in fact hasn't been receiving much attention in the current days having an architecture with constraints that learn the representation hierarchy from the section of object to the scenes in both Generator and the Discriminator [20]. Both neural networks update their gradients using stochastic gradient descent (SGD) and Adam optimizers. Convolutional layers comprise the whole DCGAN's interior [21]. DCGAN uses the same fundamental algorithm as the conventional GAN. The convolutional layer maps and transforms the noise from a Generator which produces 100-dimensional noise. The construction of DCGAN's Generator is shown in Fig. 4.

By using CNN as the two adversarial networks of GAN, DCGAN improves performance for the first time. For the VAE (Variational Autoencoder), the Generator performs the same function as the decoder [21]. The Generator's input is a vector drawn from the multivariate standard normal distribution. The Generator freezes and refines the Discriminator for thousands of epochs. The output image's dimensions correspond to those of the source image. For DCGAN, the optimal results are obtained using the Adam (Adaptive Moment Estimation) optimizer with a learning rate of "0.002" [22]. A wide range of representations are learnt by DCGAN, from objects to scenes. Furthermore, the acquired properties can be utilized as a universal image representation for many applications. Raymond applied DCGAN [23] to fill in sections of the image that were missing or empty. For the first time, DCGAN uses Convolutional Neural Network (CNN) as a Generator and Discriminator of GAN, improving performance. Presently, each GAN structure has a convolutional layer. Therefore, GAN already implies what "DC" means.

E. BEGAN (Boundary Equilibrium GANs)

BEGAN uses both supervised and unsupervised learning technique with a multilayer DNN ("Deep Neural Network") architecture. It learns characteristics for appropriate semantic tasks and adopt then in a testing situation. It also gained interest due to the fact that its Discriminator was a Convolutional Autoencoder (CAE) and it possessed convergence judgement qualities that DCGAN lacked [24]-[25]. Fig. 5 shows the network architecture for the Generator termed here as Decoder and the Discriminator termed as Encoder.

Image's latent space is learnt while keeping and modifying the balance between the two neural networks by BEGAN. The space-constrained BEGAN-CS ("BEGAN with Constrained Space") was introduced, however it failed to address the mode collapse [26]. To address the issue like mode collapse [26], modified the encoder and decoder as well as the BEGAN-CS Discriminator's structure from AE to VAE [27]. LReLU (Leaky Rectified Linear Unit) replaced ELU (Exponential Linear Unit) as the activation function [28]-[29].

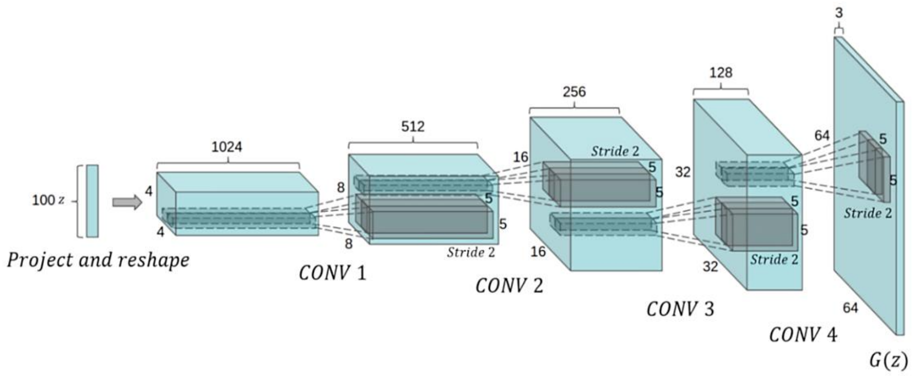

Fig. 4. Structure of the Generator in DCGAN (Deep Convolutional GANs) [21]

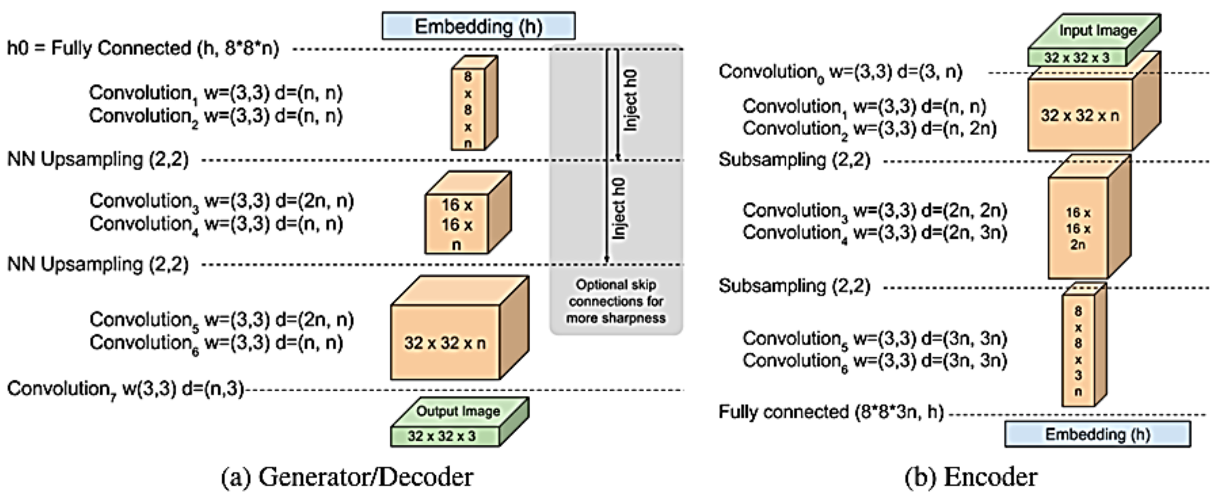

Fig. 5. Network architecture for the a) Generator and b) Discriminator [24]

F. PGGAN (Progressive Growing of Generative Adversarial Networks)

Progressive Growing of GANs was able to present a novel method for training Generative Adversarial Networks [8]. The plan was to progressively expand the Generator and Discriminator's sizes. As the training progressed, further layers simulating even finer details were added, starting at a low resolution. This, not only greatly stabilizes the training process but also speeds it up, allowing us to produce pictorial data of never-before-seen quality, like CELEBA images at 10242. Additionally, it was suggested that an easy way to increase the variety of generated images be used. In the unsupervised CIFAR10 competition, a record inception score of 8.80 was set. In addition, several implementation details were suggested that are essential to avoid unhealthful competition between the Discriminator and Generator. PGGAN trains Generators and Discriminators using low-resolution 4×4 pixel pictures. The process of PGGAN training is shown in Fig. 6.

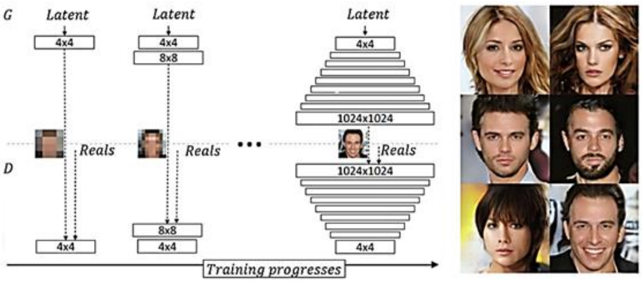

Fig. 6. Network architecture for the Generator and Discriminator [24]

High output resolutions can be handled reliably and efficiently thanks to the implementation. A selection of the 1024×1024 images produced by the previously stated network are shown in Fig. 7.

Fig. 7. 1024×1024 images generated using the CELEBA-HQ dataset [24]

G. CGAN (Conditional GAN)

The adversarial training approach offers significant flexibility in the composition of this hidden representation as the Generator mixes the prior input noise $p_z(z)$, and input y into a single hidden representation. Here x and y are inputs into the Discriminator, and here, a multilayer perceptron (MLP) [30] acts as the discriminative function. The objective function of 2-player minimax game (as mentioned before as well) is represented in (3). The simplified structure of the CGAN is depicted in Fig. 8.

$$min_G max_D V(G,D) = E_{x\sim p_{data}(x)}[log\ D(x|y)] + E_{z\sim p_z(z)}[log\ (1 - D(G((z|y)))] \quad (3)$$

- $E_{x\sim p_{data}(x)}$ represents the expected value of a function over the real data distribution $p_{data}(x)$.
- E represents the expectation operator, that calculates the average or expected value.
- $x\sim p_{data}(x)$ represents the variable x is sampled from the real data distribution $p_{data}(x)$, i.e., the probability distribution of the actual data that the GAN is trying to model.
- $p_{data}(x)$ represents the probability distribution of the real data.

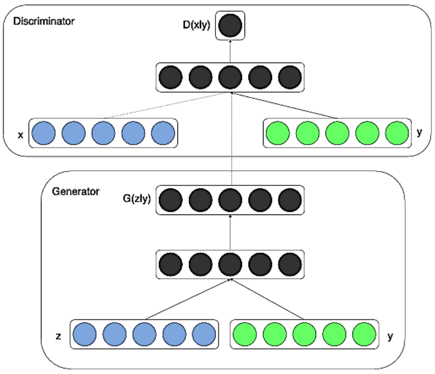

Fig. 8. Conditional adversarial network [10]

H. AAE (Adversarial Autoencoder)

An adversarial autoencoder (AAE) has been used in this instance to create supervised, semi-supervised, and unsupervised learning strategies. To update the gradients, SGD with reconstruction and regularization phases has also been utilized. The adversarial autoencoder is a probabilistic autoencoder that uses the recently proposed Generative Adversarial Networks (GAN) to do variational inference by comparing the aggregated posterior of the autoencoder's hidden code vector to any arbitrary prior distribution. Since the aggregated posterior and the prior are matched, generating from any part of the prior space yields relevant samples. Thus, a deep generative model mapping the enforced prior data distribution is created by the adversarial autoencoder's decoder [31].

Applications for the adversarial autoencoder include dimensionality reduction, data visualization, semi-supervised classification, unsupervised clustering, separating the style and content of images.

I. BigGAN

Problems like producing high-resolution, diverse samples from challenging datasets like ImageNet by training GAN networks at their greatest size and investigating the instabilities unique to it can be handled. By adding orthogonal regularization to the Generator, we find that it may be made vulnerable to a simple "truncation trick," allowing for precise control over the trade-off between sample fidelity and variety by reducing the input variation. BigGAN was developed by DeepMind extension of SAGAN, currently produces the top results [32] in imagine generation. Eight times bigger than SAGAN, its deployment size is 2048. Each layer's channel size was likewise increased by 50% [10]. The results are best depicted in Fig. 9 where the learning rate of 0.0002 and a batch size of 128 the images using batch normalization were obtained.

Fig. 9. Resulted images of BigGAN's truncation technique [10]

J. StyleGAN

The G and D training which is capable of creating very large, high-quality images by gradually extending them from small to big images during training, is the forerunner of the StyleGAN as shown in Fig. 10. Instead of using a point from the latent space as input, the StyleGAN Generator now uses noise layers and a separate mapping network as sources of randomness to create an artificial image.

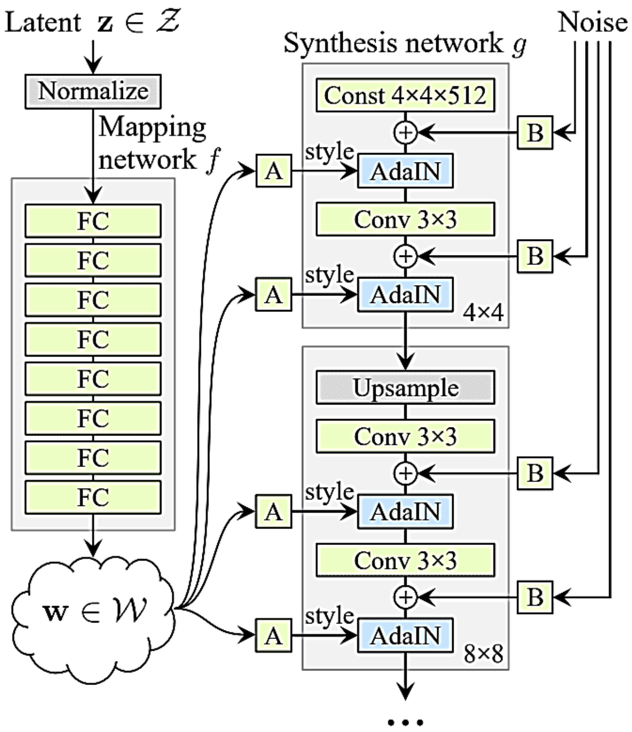

Fig. 10. StyleGAN generator structure [11]

StyleGAN in [12] proposed a technique called AdaIN -a style transfer method to address the issue of latent space entanglement. Feature map x_i's mean μ and variance δ can be modified as AdaIN employs reference style bias $y_{b,i}$ and scale $y_{s,i}$. $y_{b,i}$ produced by the layers of the synthesis network shown as (4).

StyleGAN in [12] proposed a technique called AdaIN to address the issue of latent space entanglement. Feature map x_i's mean μ and variance δ can be modified as AdaIN employs reference style bias $y_{b,i}$ and scale $y_{s,i}$. $y_{b,i}$ produced by the layers of the synthesis network shown as (4).

$$AdaIN(xi, y) = y_{s,i} \frac{x_{i-}\mu(x_i)}{\delta(x_i)} + y_{b,i} \quad (4)$$

The *AdaIN* layer stops the transmission of style information between these layers. Fig. 11 shows the output images of StyleGAN. With a learning rate of 0.0001 and a batch size of 64 the images using batch normalization were obtained.

Fig. 11. High-quality images generated faces using the StyleGAN [12]

III. MATH BEHIND THE GAN

A. Mathematical Notations

The generative model may generate images from scratch by comprehending and gaining knowledge about the training data's statistical distribution. In both cases, the network weight is learnt via backpropagation [13]. The key source of reference for the GAN literature on multidimensional vectors in [33], which highlights vectors in the probability space. Here, 'z' is a common representation for latent vectors. In the field of signal processing, vectors are represented using lowercase symbols to highlight their multidimensionality.

Here the $P_{data}(x)$ is the probability density function for a random vector of real data x of $R^{|x|}$. Input noise distribution being $P_z(z)$ and the probability density is a mathematical expression for the likelihood that a continuous random variable will fall within a specified interval. The distribution of vectors generated in G is termed as $P_g(x)$. The weights which are learned from the Generator G is θ_g and from that of the Discriminator D is θ_d. To maximize this $D(G(z))$ function is the Generator's goal. Stated differently, its goal is to have the Discriminator produce more false instances. Among these roles following below:

- $D(x)$ is the Discriminator's output for a true case.
- Given the noise z, the Generator's output is $G(z)$.
- For an artificial instance, the Discriminator produces $D(G(z))$.
- The output of Discriminator D, need not be in the range of 1 to 0.

B. Discriminator Loss

The Discriminators job is to correctly specify the sample as either fake or real. Hence the loss function can be given by (5).

$$L_{(D)} = max\ [log(D(x)) + log(1 - D(G(z)))] \tag{5}$$

C. Generator Loss

The Discriminator and Generator in the network are competing with each other. As a result, it will attempt to minimize its loss function as mentioned in (6).

$$L_{(G)} = min\ [log(\ D(x)) + log(1 - D(G(z)))] \tag{6}$$

D. Combined Loss Function

Hence the combined loss function can be given by (7).

$$L = min_G max_D[log(\ D(x)) + log(1 - D(G(z)))] \tag{7}$$

This loss function is only applicable to a single data point, so in order to examine the full dataset, we must take the aforementioned equation's expectation as in (8).

$$\begin{aligned} &min_G max_D V(G, D) \\ &= min_G max_D (E_{x\sim p_{data}(x)}[\log D(\mathrm{x})] \\ &+ E_{z\sim p_z(z)}\left[\log\left(1 - D(G(\mathrm{z}))\right)\right]) \end{aligned} \tag{8}$$

IV. LIMITATIONS OF GANS

The loss function derived has some limitations which are described in this section.

A. Vanishing Gradient

Vanishing gradients are a substantial obstacle in GAN training, especially when the Discriminator becomes excessively powerful, resulting in inadequate feedback for the Generator to enhance its performance. This problem occurs during backpropagation, in which the gradient-employed to update the network's weights-propagates from the output layer to the starting layers. As it propagates, the gradient may attenuate, resulting in the previous layers either ceasing to learn or updating at an exceedingly sluggish rate, a phenomenon referred to as the vanishing gradient problem.

Activation functions such as ReLU (Rectified Linear Unit) and Leaky ReLU address this problem by averting saturation, a phenomenon that arises when activation values diminish excessively, as seen with Sigmoid or Tanh functions. In contrast to Sigmoid, which compresses outputs within the range of 0 to 1 and results in nearly negligible gradients for extreme positive or negative inputs, ReLU produces zero for negative inputs and the corresponding input value for positive inputs. This simplicity guarantees that gradients do not attenuate across layers. Leaky ReLU enhances this by permitting a little, non-zero gradient for negative inputs, so preserving the gradient flow across the network and enabling deeper layers to learn effectively. These non-saturating functions are essential for stabilizing GAN training, particularly in deep architectures.

Attempts to remedy vanishing gradient are:

- We may use activation functions like ReLU or LeakyReLU, instead of Sigmoid or tanh which cause the gradient to diminish exponentially.
- Even after training the Discriminator to its maximum efficiency, the Wasserstein loss is intended to prevent vanishing gradients was suggested in the original GAN study.

B. Mode Collapse

As G's primary goal was to deceive D into producing monotonous output, the Generator may accidentally lock itself into a mode during training that only ever generates the same output. This is known as mode collapse. Mode collapse first occurs when if lesser number of samples are found by the Generator. No additional sample can be created other than this small sample. A single sample or a mode that deceives the Discriminator is frequently identified by Generators. Then, they can connect this sample to any latent space location. Due to this loss function gradient decreases to a value that is nearly equal to zero [33]-[35]. They also applied minimax to the total loss values in multiple steps to update the model. By anticipating the Discriminator's update, Generator is able to produce superior images.

Attempts to remedy for Mode Collapse are:

- Use Wasserstein loss function, for Discriminator training to its best performance.
- GAN training with a variety of data samples.

C. Failure to Converge

GAN convergence problems are common. By presuming that two neural networks are competing against one another with the hope that both networks would eventually attain equilibrium, adversarial training settings may appear to be unstable.

Attempts to remedy for Failure to Converge are:

- This can be achieved by either using a small number of features to discriminate between the training data and the Generator's output, or by adding noise to the Discriminator's inputs (both the real and synthetic data) to prevent it from becoming overconfident in its classification.
- Punishing Discriminator weights, by regularizing the training of Generative Adversarial Networks.

V. APPLICATIONS OF GANS

GAN produces realistic samples, making it a useful generative model [10][12][24]. Additional statistical judgements or knowledge of real-world data distribution are unnecessary [36][37]. We are reviewing a few published and improved computer vision applications from the available recent research [38]-[40].

A. Better Image Generation Quality and Super Resolution of Images:

SAGAN was developed [41] to describe long-range dependency through attention for image production concerns. SAGAN outperformed the highest inception score from 36.8

to 52.52 and minimized Fréchet Inception difference from 27.62 to 18.65 on the difficult ImageNet dataset. Research works [42]-[43], say GANs use intermediate representations, not lower-resolution images [44]. Also, the trained model leverages real image data during sampling to build a high-quality image from a low-resolution image. The research in [45] suggests integrating structural network design, antagonistic loss, and perceptual loss to build an improved model. Other researches also mainly focus on building a better and efficient model to generate better quality images and videos that look realistic and can be used for futuristic applications in spite of an unbalanced dataset [46]-[49]. GANs also have been used in image domain conversion like from sketch to realistic photos or conversion of maps to satellite images as a practical application.

B. *Video Generation and Prediction*

For moving objects and scenic dynamics, computer vision is an issue. The scenes transform model incorporates video generation (e.g., future prediction) and recognition (e.g., action classification). The article [50] trained the CNN with an input sequence to construct realistic frameworks [51].

C. *Image to Image & Text to Image Transformation*

For image-image creation, conditional opponent networks are ideal using the Pix2pix model [52]. CycleGAN, which utilizes paired images from two domains (e.g., sketch-to-photo or painting-to-photo conversion), enhances the transformation process by maintaining the integrity of the original image. It achieves this through a forward and reverse transformation cycle, ensuring that the original image can be accurately reconstructed, even though some challenges with cycle consistency remain [53]. This speeds data processing and expands the method's uses. Springer uses an unmatched record of painters and natural image graphs to create images like Picasso or Monet [54]. However, there are research works that suggest improving the scores by increasing the unbalanced dataset using evasion techniques [55]-[57] as well.

D. *Identifying Objects*

The process of identifying actual objects in pictures or videos, such as faces, bikes, and buildings, is called object detection. Item identification algorithms often use extracted features and learning methodologies to identify individual examples of a certain object type. Advanced driver assistance, security, and monitoring are used by all Advanced Driving Assistance Systems (ADAS). Small sized items are typically hard to see due to their great contrast and low resolution. A modern Perceptual Generative Adversarial Network (Perceptual GAN) has been developed by in an effort to close the representational gap between small and large objects.

VI. COMPARISON OF GAN VARIANTS

The Table I provides a clear comparing of the overview of the strengths, weaknesses, and advancements of GAN variants. All the selections of hyperparameters in the discussed variants are specific to the needs to the model to be built and can be customized accordingly.

TABLE I. THE COMPARATIVE ANALYSIS OF GAN VARIANTS

GAN Variant	Strengths	Weaknesses
DCGAN	• Uses convolutional layers for better feature learning • Improves stability with batch normalization • Improves over a base line GAN	• Mode collapse still occurs • Limited to fixed-size inputs
WGAN	• Wasserstein loss ensures meaningful gradients • Reduces mode collapse • More stable convergence • Improves over a base line GAN	• Weight clipping causes slow convergence • Requires careful tuning of Critic updates
WGAN-GP	• Gradient penalty replaces weight clipping • Ensures smooth gradients and better convergence • Improves over WGAN	• Training is still slower due to Critic updates Computationally expensive
SAGAN	• Introduces self-attention to capture long-range dependencies • Improves generation of high-resolution images • Improves over DCGAN	• Increased computational cost • Slower training due to attention layers
StyleGAN	• Provides fine control over style and features • Reduces entanglement in latent space • Improves over SAGAN	• Computationally intensive • Can still exhibit minor feature entanglement
BigGAN	• Generates high-quality, large-scale images (e.g., ImageNet) • Supports truncation trick for balancing quality and diversity • Improves over SAGAN	• Requires massive computational resources • Instability at large scale

VII. QUANTITATIVE PERFORMANCE COMPARISON

Particularly in the context of image creation, two measures that are frequently used to assess the quality of generative models are Fréchet Inception Distance (FID) and Inception Score (IS). FID score measures the distance between feature vectors from real and generated pictures. The score compares the two groups' computer vision statistics from the Inception-v3 model for image categorization. When assessing generative adversarial network images, lower FID scores correspond better with higher quality images [58]-[61]. Based on how effectively generated images are classified by a trained Inception model, IS assesses their quality. The performance improves as the Inception Score (IS) value increases and the FID value decreases. The four ImageNet-based models' scores are displayed in Table II. BigGAN-deep performed best of the four models with an FID of 5.7, while BigGAN performed poorest with 8.7.

BigGAN-deep scored the highest in IS with 124.5 points, while BigGAN performed the poor with 98.8 points, a difference in performance of 25.7 points. And the Table III. Shows the various GAN performances on Canadian Institute for Advanced Research-10 (CIFAR-10) dataset.

TABLE II. THE IS AND FID OF SEVERAL MODELS DEVELOPED USING THE IMAGENET DATABASE

Dataset	Model	References	IS ↑	FID ↓
ImageNet (128x128)	BigGAN-deep	[11]	124.5	5.7
	CR-BigGAN	[42]	-	6.7
	BigGAN+DiffAugment	[58]	100.8	6.8
	BigGAN	[11]	98.8	8.7

TABLE III. THE MODELS' FID FOR CIFAR-10 DATASET (PERFORMANCE IMPROVES WHEN THE FID VALUE LOWERS)

Dataset	Model	References	FID ↓
CIFAR-10 (64x64)	StyleGAN2+ADA+Tuning	[8]	2.92
	CR BigGAN+DiffAugment	[61]	4.30
	BigGAN+DiffAugment	[58]	4.61
	StyleGAN2+DiffAugment	[13]	5.79
	BigGAN+MIX	[59]	8.17
	BigGAN+CR+LT	[61]	9.80
	WGAN-ALP	[17]	12.96
	BigGAN	[10]	14.73
	WGAN-GP	[18]	29.30

Numerous Generative Adversarial Network (GAN) models have surfaced in recent years, each with unique benefits and drawbacks. The development of Wasserstein GAN (WGAN) and its enhanced variant WGAN-GP aimed to tackle several significant obstacles of conventional GANs, specifically the problems related to mode collapse and vanishing gradients [62]-[65]. The Wasserstein distance is used in WGAN to help give a more stable loss function that more closely matches the caliber of the samples that are generated. This leads to more seamless training dynamics and enhances the model's convergence [66]-[68]. WGANs do have certain restrictions, though. The Discriminator and Generator must be carefully balanced when training WGAN, and although the gradient penalty added in WGAN-GP is useful [69], it can slow down training and increase computing complexity [70], which further can be adapted for numerous applications [71].

By adding self-attention mechanisms, Self-Attention GAN (SAGAN) expands on the conventional GAN framework and enables the model to generate samples by focusing on distinct areas of the image [72]. Especially for complicated datasets like ImageNet, this attention method helps the model catch long-range connections between pixels [73], improving generation quality. SAGAN is a well-liked option for assignments requiring fine-grained details because of its capacity to produce high-quality images. However, compared to more straightforward models like Deep Convolutional GAN (DCGAN), the additional complexity of self-attention layers raises computing demands and may slow down training [74]. Despite being fundamental and successful, DCGAN has limitations when it comes to producing high-resolution images and handling mode collapse, which restricts its use in more complex applications [75]. Significant advances in GAN technology have been made with BigGAN and StyleGAN, especially in the area of high-resolution image production [76]. BigGAN has emerged as the standard model for high-resolution tasks due to its reputation for producing large-scale, high-quality images [77]. However, without substantial system capacity, it is difficult to implement due to its high computational resource requirements and large-scale instability [78]. StyleGAN provides unrivalled quality in object and facial synthesis thanks to its innovative design, which gives control over several components of the picture generating process [79]-[81]. For tasks demanding high fidelity, its capacity to separate features and offer precise control over image properties makes it perfect. Even with its improvements, StyleGAN is still computationally costly, and problems like feature entanglement are not totally eliminated even though they have decreased [82]. Each of these models has trade-offs in terms of complexity, resource costs, and application but offers distinctive answers to particular issues within GAN training.

VIII. CONCLUSION

This review provides a comprehensive analysis of various GAN architectures, including DCGAN, WGAN, WGAN-GP, SAGAN, StyleGAN, focusing on their specific contributions, applications, and inherent limitations. Each variant introduces unique solutions to challenges, such as WGAN's use of Wasserstein loss to stabilize convergence and WGAN-GP's gradient penalty to mitigate mode collapse. SAGAN incorporates self-attention mechanisms to enhance high-resolution image generation, while StyleGAN addresses latent space disentanglement. Despite these advancements, persistent issues like unstable training and mode collapse remain significant obstacles, especially for large-scale implementations like BigGAN. Solutions such as fine-tuning critic updates and the introduction of adaptive optimizers have shown promise but are often computationally intensive. Additionally, GAN applications extend beyond image generation, covering areas like video prediction and text-to-image transformation, yet their adoption in domains such as natural language processing remains limited due to data structure differences. The review emphasizes the importance of further research in developing more stable training methods, improving model interpretability, and expanding GAN usage to underexplored fields. Practical implications suggest that practitioners must carefully balance computational resources with performance needs, particularly when deploying GANs in real-world scenarios. Furthermore, ethical concerns—such as the misuse of GANs for deepfakes creation—highlight the need for responsible research practices and regulatory measures. Future research should prioritize developing scalable architectures with reduced computational overhead and investigate novel loss functions that improve stability across diverse applications. This synthesis underscores GANs' transformative potential but also stresses that addressing their current limitations through targeted innovations is essential for unlocking new opportunities in both academic research and industrial applications.

REFERENCES

[1] F. PirahanSiah, "Computer Vision, Deep Learning, Deep Reinforcement Learning," *Scienceopen*, 2019, doi: 10.14293/s2199-1006.1.sor-uncat.clzwyuz.v1.

[2] I. Goodfellow, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139-144, 2020, doi: 10.1145/3422622.

[3] R. Viñas, T. Azevedo, E. R. Gamazon, and P. Liò, "Gene Expression Imputation with Generative Adversarial Imputation Nets," *BioRxiv*, 2020, doi: 10.1101/2020.06.09.141689.

[4] U. Kamath, K. L. Graham, and W. Emara, "Bidirectional Encoder Representations from Transformers (BERT)," *Transformers for Machine Learning*, pp. 43-70, 2022, doi: 10.1201/9781003170082-3.

[5] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, and D. Amodei, "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 1877-1901, 2020.

[6] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv preprint arXiv:2005.00341*, 2020.

[7] R. Yamamoto, E. Song, and J. -M. Kim, "Parallel Wavegan: A Fast Waveform Generation Model Based on Generative Adversarial Networks with Multi-Resolution Spectrogram," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6199-6203, 2020, doi: 10.1109/ICASSP40776.2020.9053795.

[8] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training Generative Adversarial Networks with limited data," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 12104-12114, 2020.

[9] R. Cakaj, J. Mehnert, and B. Yang, "Spectral Batch Normalization: Normalization in the Frequency Domain," *2023 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-10, 2023, doi: 10.1109/ijcnn54540.2023.10191931.

[10] W. Kłos, P. Coronica, K. Seeliger, and M. N. Hebart, "Training BigGAN on an ecologically motivated image dataset," *2022 Conference on Cognitive Computational Neuroscience*, 2022, doi: 10.32470/ccn.2022.1144-0.

[11] B. Brock, J. Donahue, and K. Simonyan, "Large Scale GAN Training for High Fidelity Natural Image Synthesis," *arXiv preprint arXiv:1809.11096*, 2020.

[12] T. Karras, S. Laine, and T. Aila, "A style-based Generator architecture for Generative Adversarial Networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4401-4410, 2019.

[13] N. Meira, M. Silva, A. Bianchi, and R. Oliveira, "Generating Synthetic Faces for Data Augmentation with StyleGAN2-ADA," *Proceedings of the 25th International Conference on Enterprise Information Systems*, pp. 649-655, 2023, doi: 10.5220/0011994600003467.

[14] S. Kumar and S. Dhawan, "A detailed study on Generative Adversarial Networks," in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, pp. 641-645, 2020.

[15] V. Raj, R. Kumar, and N. Kumar, "A Scrupulous Framework to Forecast the Weather using CNN with Back Propagation Method," *2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, vol. 8, pp. 177-181, 2022, doi: 10.1109/icac3n56670.2022.10074346.

[16] Y. Wang, N. Polson, and V. O. Sokolov, "Data Augmentation for Bayesian Deep Learning," *Bayesian Analysis*, vol. 18, no. 4, 2023, doi: 10.1214/22-ba1331.

[17] J. Jam, C. Kendrick, V. Drouard, K. Walker, G.-S. Hsu, and M. Yap, "Symmetric Skip Connection Wasserstein GAN for High-resolution Facial Image Inpainting," *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pp. 35-44, 2021, doi: 10.5220/0010188700350044.

[18] Y. Chen and X. Hou, "An Improvement based on Wasserstein GAN for Alleviating Mode Collapsing," *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-7, 2020, doi: 10.1109/ijcnn48605.2020.9207717.

[19] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention Generative Adversarial Networks," in *Proceedings of the International Conference on Machine Learning*, pp. 7354-7363, 2019.

[20] H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree, "Regularisation of neural networks by enforcing Lipschitz continuity," *Machine Learning*, vol. 110, pp. 393-416, 2021.

[21] M. Ghayoumi, "Deep Convolutional Generative Adversarial Networks (DCGANs)," *Generative Adversarial Networks in Practice*, pp. 220-257, 2023, doi: 10.1201/9781003281344-8.

[22] D. Pham and T. Le, "Auto-Encoding Variational Bayes for Inferring Topics and Visualization," *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 5223-5234, 2020, doi: 10.18653/v1/2020.coling-main.458.

[23] V. K. Chauhan, "Stochastic Trust Region Inexact Newton Method," *Stochastic Optimization for Large-scale Machine Learning*, pp. 123-138, 2021, doi: 10.1201/9781003240167-10.

[24] C. Shi, W. Liu, J. Meng, Z. Li, and J. Liu, "Semantic-wise hybrid attention generative adversarial network for image inpainting," *Research Square*, 2023, doi: 10.21203/rs.3.rs-3382745/v1.

[25] Z. Luo, H. Yu, and Y. Zhang, "Pine Cone Detection Using Boundary Equilibrium Generative Adversarial Networks and Improved YOLOv3 Model," *Sensors*, vol. 20, no. 16, p. 4430, 2020, doi: 10.3390/s20164430.

[26] F. Ros and R. Riad, "Deep clustering techniques based on autoencoders," *Unsupervised and Semi-Supervised Learning*, pp. 203-220, 2023, doi: 10.1007/978-3-031-48743-9_11.

[27] Z. Dai, L. Zhao, K. Wang, and Y. Zhou, "Mode Standardization: A Practical Countermeasure Against Mode Collapse of Gan-Based Signal Synthesis," *Applied Soft Computing*, vol. 150, p. 111089, 2023, doi: 10.2139/ssrn.4423104.

[28] S. W. Park, J. H. Huh, and J. C. Kim, "BEGAN v3: avoiding mode collapse in GANs using variational inference," *Electronics*, vol. 9, no. 4, p. 688, 2020.

[29] Y. Bodyanskiy, A. Deineko, V. Skorik, and F. Brodetskyi, "Deep Neural Network with Adaptive Parametric Rectified Linear Units and its Fast Learning," *International Journal of Computing*, pp. 11-18, 2022, doi: 10.47839/ijc.21.1.2512.

[30] R. Alkhatib, "Artificial Neural Network Activation Functions in Exact Analytical Form (Heaviside, ReLU, PReLU, ELU, SELU, ELiSH)," *Authorea Preprints*, 2021, doi: 10.36227/techrxiv.15096888.v1.

[31] X. Li and Z. Zhang, "The comparison between Conditional Generative Adversarial Nets and Deep Convolutional Generative Adversarial Network, and its GUI-related application," *2021 2nd International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE)*, pp. 601-609, 2021, doi: 10.1109/icbase53849.2021.00119.

[32] L. Nguyen, "Adversarial Variational Autoencoders to Extend and Improve Generative Model," *Computer Science and Mathematics*, 2023, doi: 10.20944/preprints202308.0131.v1.

[33] P. N. Deelaka, "Neural Artistic Style Transfer with Conditional Adversarial Network," *Available at SSRN 4358610*, 2023, doi: 10.2139/ssrn.4358610.

[34] M. Ghayoumi, "Generative Adversarial Networks (GANs) for Images," *Generative Adversarial Networks in Practice*, pp. 436-477, 2023, doi: 10.1201/9781003281344-14.

[35] M. Xu, "Towards generalized implementation of Wasserstein distance in GANs," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 12, pp. 10514-10522, May 2021.

[36] J. Hwang and D. Suh, "CC-GAIN: Clustering and Classification-Based Generative Adversarial Imputation Network for Missing Electricity Consumption Data Imputation," *Expert Systems with Applications*, vol. 255, p. 124507 2023.

[37] H. Alqahtani, M. Kavakli-Thorne, and G. Kumar, "Applications of generative adversarial networks (GANs): An updated review,"

Archives of Computational Methods in Engineering, vol. 28, pp. 525-552, 2021.

[38] A. Karnewar and O. Wang, "MSG-GAN: Multi-scale gradients for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7799-7808, 2020.

[39] A. Zhavoronkov *et al*., "Potential non-covalent SARS-CoV-2 3C-like protease inhibitors designed using generative Deep Learning approaches and reviewed by human medicinal chemist in virtual reality," *ChemRxiv*, 2020.

[40] M. Yurt, S. U. Dar, A. Erdem, E. Erdem, K. K. Oguz, and T. Çukur, "mustGAN: multi-stream generative adversarial networks for MR image synthesis," *Medical Image Analysis*, vol. 70, p. 101944, 2021.

[41] A. You, J. K. Kim, I. H. Ryu, and T. K. Yoo, "Application of generative adversarial networks (GAN) for ophthalmology image domains: a survey," *Eye and Vision*, vol. 9, no. 1, p. 6, 2022.

[42] M. N. Minaidi, C. Papaioannou, and A. Potamianos, “Self-Attention Based Generative Adversarial Networks For Unsupervised Video Summarization,” *2023 31st European Signal Processing Conference (EUSIPCO)*, pp. 571-575, 2023, doi: 10.23919/eusipco58844.2023.10289808.

[43] L. Zhang, Z. Xu, Y. Liu, T. Qiao, H. Su, and Y. Luo, “Transformer Fault Diagnosis Based on Adversarial Generative Networks and Deep Stacked Autoencoder,” *Heliyon*, vol. 10, no. 9, 2024.

[44] H. Wang, W. Wu, Y. Su, Y. Duan, and P. Wang, "Image super-resolution using an improved generative adversarial network," in *2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC)*, pp. 312-315, 2019.

[45] X. Wang *et al*., "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018.

[46] S. W. Park, J. S. Ko, J. H. Huh, and J. C. Kim, "Review on generative adversarial networks: focusing on computer vision and its applications," *Electronics*, vol. 10, no. 10, p. 1216, 2021.

[47] W. Zhang, P. Zhang, Y. Yu, X. Li, S. A. Biancardo, and J. Zhang, "Missing data repairs for traffic flow with self-attention generative adversarial imputation net," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7919-7930, 2021.

[48] D. Mahapatra and Z. Ge, "Training data independent image registration using generative adversarial networks and domain adaptation," *Pattern Recognition*, vol. 100, p. 107109, 2020.

[49] K. K. Babu and S. R. Dubey, "CSGAN: Cyclic-synthesized generative adversarial networks for image-to-image transformation," *Expert Systems with Applications*, vol. 169, p. 114431, 2021.

[50] M. Torabi, “SumcaVer1: Mean Square Prediction Error Estimation in Small Area Estimation,” *CRAN: Contributed Packages*, 2024, doi: 10.32614/cran.package.sumcaver1.

[51] S. Talafha, B. Rekabdar, C. P. Ekenna, and C. Mousas, “Attentional Adversarial Variational Video Generation via Decomposing Motion and Content,” *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, pp. 45-52, 2020, doi: 10.1109/icsc.2020.00014.

[52] D. Torbunov et al., "UVCGAN: Unet vision transformer cycle-consistent GAN for unpaired image-to-image translation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 702-712, 2023.

[53] Y. J. Yeo, M. C. Sagong, S. Park, S. J. Ko, and Y. G. Shin, "Image generation with self-pixel-wise normalization," *Applied Intelligence*, vol. 53, no. 8, pp. 9409-9423, 2023.

[54] C. F. Foo and S. Winkler, “Image Data Augmentation with Unpaired Image-to-Image Camera Model Translation,” *2022 IEEE International Conference on Image Processing (ICIP)*, pp. 3246-3250, 2022, doi: 10.1109/icip46576.2022.9897671.

[55] R. H. Randhawa, N. Aslam, M. Alauthman, and H. Rafiq, "Evasion generative adversarial network for low data regimes," *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 5, pp. 1076-1088, 2022.

[56] V. Chinbat and S. H. Bae, "Ga3n: Generative adversarial autoaugment network," *Pattern Recognition*, vol. 127, p. 108637, 2022.

[57] R. H. Randhawa, N. Aslam, M. Alauthman, M. Khalid, and H. Rafiq, "Deep reinforcement learning based Evasion Generative Adversarial Network for botnet detection," *Future Generation Computer Systems*, vol. 150, pp. 294-302, 2024.

[58] S. Zhao, Z. Liu, J. Lin, J. Y. Zhu, and S. Han, "Differentiable augmentation for data-efficient GAN training," in *Advances in Neural Information Processing Systems*, vol. 33, pp. 7559-7570, 2020.

[59] L. Cai, “Comparative Analysis the Super-Resolution Image Generation Performance Based on BigGAN and VQ-VAE-2,” *Highlights in Science, Engineering and Technology*, vol. 41, pp. 202-210, 2023, doi: 10.54097/hset.v41i.6812.

[60] P. Patel, N. Kumari, M. Singh, and B. Krishnamurthy, "LT-GAN: Self-supervised GAN with latent transformation detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 3189-3198, 2021.

[61] R. Quaicoo, R. Acheampong, P. Gyamenah, A. A. Dodoo, M. A. T. Soli, and J. K. Appati, “Adapting Triple-BigGAN for Image Detection Tasks: Challenges and Opportunities,” *Research Square*, 2024, doi: 10.21203/rs.3.rs-4262097/v1.

[62] S. A. Gebereselassie and B. K. Roy, “Secure Image Encryption Algorithm based on Two-Level Diffusion and Hybrid Chaotic Maps,” *2023 IEEE Silchar Subsection Conference (SILCON)*, pp. 1-6, 2023, doi: 10.1109/silcon59133.2023.10404972.

[63] B. Fathi-Vajargah, “Image Encryption Based on Permutation and Substitution Using Clifford Chaotic System and Logistic Map,” *Journal of Computers*, pp. 309-326, 2018, doi: 10.17706/jcp.13.3.309-326.

[64] S. Farwa, N. Muhammad, N. Bibi, S. A. Haider, S. R. Naqvi, and S. Anjum, “RETRACTED: Fresnelet approach for image encryption in the algebraic frame,” *Applied Mathematics and Computation*, vol. 334, pp. 343-355, 2018, doi: 10.1016/j.amc.2018.03.105.

[65] H. Yang, K.-W. Wong, X. Liao, W. Zhang, and P. Wei, “A fast image encryption and authentication scheme based on chaotic maps,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 11, pp. 3507-3517, 2010, doi: 10.1016/j.cnsns.2010.01.004.

[66] X. Zhang, “Application of Knowledge Distillation in Generative Adversarial Networks,” *2023 3rd Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS)*, pp. 65-71, 2023, doi: 10.1109/acctcs58815.2023.00014.

[67] A. A. Abd El-Latif and X. Niu, “A hybrid chaotic system and cyclic elliptic curve for image encryption,” *AEU - International Journal of Electronics and Communications*, vol. 67, no. 2, pp. 136-143, 2013, doi: 10.1016/j.aeue.2012.07.004.

[68] T. Miyato and M. Koyama, “Generative Adversarial Network (GAN),” *Computer Vision*, pp. 508-513, 2021, doi: 10.1007/978-3-030-63416-2_860.

[69] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric,” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 586-595, 2018, doi: 10.1109/cvpr.2018.00068.

[70] W. Sirichotedumrong, T. Maekawa, Y. Kinoshita, and H. Kiya, “Privacy-Preserving Deep Neural Networks with Pixel-Based Image Encryption Considering Data Augmentation in the Encrypted Domain,” *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 674-678, 2019, doi: 10.1109/icip.2019.8804201.

[71] M. Li, D. Lu, Y. Xiang, Y. Zhang, and H. Ren, “Cryptanalysis and improvement in a chaotic image cipher using two-round permutation and diffusion,” *Nonlinear Dynamics*, vol. 96, no. 1, pp. 31-47, 2019, doi: 10.1007/s11071-019-04771-7.

[72] X. Chen, H. Ma, P. Ji, H. Liu, and Y. Liu, “Based on GAN Generating Chaotic Sequence,” *Communications in Computer and Information Science*, pp. 37-49, 2020, doi: 10.1007/978-981-33-4922-3_4.

[73] M. Singh, N. Baranwal, K. N. Singh, and A. K. Singh, “Using GAN-Based Encryption to Secure Digital Images with Reconstruction through Customized Super Resolution Network,” *IEEE Transactions on Consumer Electronics*, vol. 70, no. 1, pp. 3977-3984, 2024, doi: 10.1109/tce.2023.3285626.

[74] D. Zhao, “High-security and low-complexity OCDM transmission scheme based on GAN enhanced chaotic encryption,” *Optics Express*, vol. 30, no. 19, p. 34898, 2022, doi: 10.1364/oe.465522.

[75] R. A. Hallman, “Poster EveGAN: Using Generative Deep Learning for Cryptanalysis,” *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pp. 3355-3357, 2022, doi: 10.1145/3548606.3563493.

[76] D. R. I. M. Setiadi, “PSNR vs SSIM: imperceptibility quality assessment for image steganography,” *Multimedia Tools and Applications*, vol. 80, no. 6, pp. 8423-8444, 2020, doi: 10.1007/s11042-020-10035-z.

[77] U. Sara, M. Akter, and M. S. Uddin, “Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study,” *Journal of Computer and Communications*, vol. 7, no. 3, pp. 8-18, 2019, doi: 10.4236/jcc.2019.73002.

[78] A. Orman, “Image Retrieval Using Pixel Similarity,” *Research Square*, 2023, doi: 10.21203/rs.3.rs-3311259/v1.

[79] P. Udayakumar and N. Rajagopalan, “(Retracted) Blockchain enabled secure image transmission and diagnosis scheme in medical cyber-physical systems,” *Journal of Electronic Imaging*, vol. 31, no. 6, 2022, doi: 10.1117/1.jei.31.6.062002.

[80] R. Bhat and R. Nanjundegowda, "Comparative Analysis of CryptoGAN: Evaluating Quality Metrics and Security in GAN-based Image Encryption," *Journal of Robotics and Control (JRC)*, vol. 5, no. 5, pp. 1557-1569, 2024.

[81] V. Ilichev and I. Chukhraev, “Data processing using Deep Learning of the generative-adversarial neural network (GAN),” *Neurocomputers*, 2021, doi: 10.18127/j19998554-202105-04.

[82] W. Serrano, “The Deep Learning Generative Adversarial Random Neural Network in data marketplaces: The digital creative,” *Neural Networks*, vol. 165, pp. 420-434, 2023, doi: 10.1016/j.neunet.2023.05.028.