




# A brief study of generative adversarial networks and their applications in image synthesis

Harshad Sharma<sup>1</sup> · Smita Das<sup>1</sup> 

Received: 2 March 2023 / Revised: 29 May 2023 / Accepted: 1 July 2023 /  
Published online: 24 July 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

## Abstract

Image Synthesis (IS), an expansion to Artificial Intelligence (AI) and Computer Vision, is the technique of artificially producing images that retains some specific required contents. An adequate procedure to handle IS problem is to tackle it using the Deep Generative Models. Generative Models are broadly utilized in numerous sub fields of AI and have empowered versatile demonstration of perplexing scenarios including image, text and music. In this paper, a particular class of Deep Generative model namely Generative Adversarial Networks (GAN) has been considered to provide a way to acquire deep illustrations derived from backpropagation signals and without the use of wide range of annotated training data. The design of GAN architecture plays a key role in image synthesis and the motive behind this paper is to analyse GAN architecture based on different variants of GANs with respect to Image Synthesis. Furthermore, a compact categorization of GANs along with their key features, pros and cons have been investigated to identify the research challenges in this field.

**Keywords** Deep generative models · Generative adversarial networks · Image synthesis · Computer vision

## 1 Introduction

Generative Adversarial Networks or simply GANs [14] are those class of generative models that appear under the category of unsupervised learning in machine learning framework but are trained in self-supervised fashion. It is a way of propagative modelling based on deep learning methods involving Neural networks. GAN involves into the learning of the intricate underlying data patterns in the input data distributions. Using this concept, one can generate near similar examples or data distributions that are possibly identical or represents the same semantic notion as that of the input data.

---

✉ Smita Das  
smitadas.nita@gmail.com

<sup>1</sup> Department of Computer Science & Engineering, NIT Agartala, Agartala, Tripura 799046, India

### 1.1 Why GAN is important?

Crudely the foremost clue behind GAN is to use two neural networks namely Discriminator(D) and Generator(G) and make them participate against each other in order to gain leverage over the quality of data generation. The purpose of the Discriminator is to distinguish the real and the fake data distribution, while the task of the generator is to create data distribution that are nearly the same used in training. The value function used in GAN [14] is defined as follows:

$$\min_G \max_D \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [1 - \log D(G(z))] \tag{1}$$

In the above equation

- $G(z)$  is the Generator’s Output for noise  $z$ .
- $D(x)$  is the Discriminator’s Output for data  $x$ .
- $D(G(z))$  is the Discriminator’s output for  $G(z)$ .
- $\mathbb{E}(x)$  is expected value over all data instances and  $\mathbb{E}(z)$  is expected value over the random noise.

The job of the Discriminator(D) is to assign value to  $D(x)$  as close as to 1, since by checking the original data it tries to maximize the score; while on seeing the data generated by the Generator(G) it tries to minimize it, so it gives the value close to 0. The equation is tweaked in order to maximize the overall value function. While the Generator(G) tries to minimize it, thereby contrasting the Discriminator (D). Looking at it from Game Theory Perspective, the setup of GAN is a contest amongst two neural networks Discriminator(D) and Generator(G), where each of them is trying to surpass the other. Looking at the equation (1) and Fig. 1, we may note that the objective of the Discriminator(D) is to maximize the above value function while that of the Generator(G) is to minimize it.

The importance of the GAN lies in the fact that it finds its application in various areas such as Computer Vision and Computer Graphics applications. Compared to other Generative models such Variational Auto Encoders(VAE), GAN is able to handle estimated density functions sharply, eliminating biases, the ability to generate required samples and congruity with the neural architecture [14].

### 1.2 Usage of GANs in image synthesis

Various interesting utilization of GANs which captured a wider landscape in Computer vision, are listed out in this subsection. Although enumerations of applications can be exhaustively large, the focus is kept on some important use cases only.

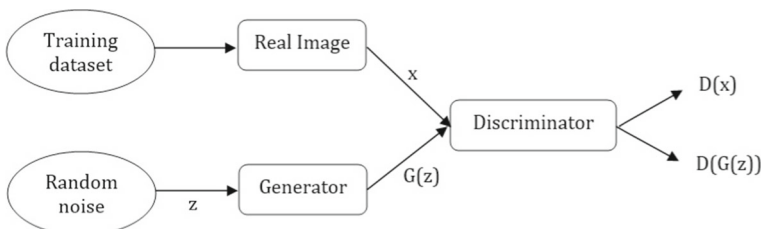


Fig. 1 Typical GAN architecture

- **Image to Image Translation:** This is a difficult approach to map with the given source image into the targeted image. This is done by the pix2pix GAN [20], which is based on conditional adversarial network.
- **Text to Image Translation:** This is served by StackGAN [74] that can generate 256x256 photo-realistic image from the simple text description.
- **Super Resolution:** Photo-Realistic Super Resolution GAN or SRGAN [31] aims at generating images of high resolution from low resolution image.
- **Image Blending:** Image Blending is a composition technique as discussed in Gaussian Poisson GAN or simply GP-GAN [70] that creates a blend between two images with the help of mask intended for making fine adjustments between the source images.
- **Cross Domain Image Generation:** It defines the problem of mapping or more specifically translating image of one domain to another. In [65], the authors have discussed an unsupervised approach for the problem aided with a GAN.
- **Generating Anime Characters:** Creating anime characters by making use of GAN [21] can literally cut down the expenses of Game development firms and Anime production houses.

### 1.3 Contributions

In this study, we have pursued to bring out applications of GAN in Image Synthesis from its inception to get an overall view of the research landscape in the field of generative models. Existing studies primarily focuses on problems faced by GAN during training and loss objective optimization, or study of various GANs with respect to a particular use case of application. This investigation provides a brief, yet a panoptic glance of research application and scope of improvement to the readers. It includes the necessary amount of exposure to the utility portion of GAN pertaining to Image Synthesis and possible related fields. Finally, this work targets to inspire the creativity aspect of the generative models for the researchers around the globe. The prime objectives of this paper are:

- To present a wide-ranging review of the very recent GAN techniques based on their categorization along with GAN variants and their key highlights in Image Synthesis.
- Identify various issues affecting GANs and the feasible solution to acquire a stable GAN environment.
- Raise a few research queries related to GAN for future direction of work in the field of image synthesis.

Therefore, in this paper, Section 2 depicts the background of GAN along with categorizations of GAN. The key factors affecting GANs and the probable solutions to them has been discussed in Section 3. Analysis of different variations of GANs specifically for Image Synthesis has been discussed in Section 4 followed by discussion in Section 5. Finally, the paper has been concluded with future direction of work in Section 6.

## 2 Background of GAN

For understanding the concepts of GAN, one has to go through the Deep Generative Models [57] which forms the basis for the GAN [13].

## 2.1 Overview of generative models

These models provide a way of formulating data distributions in an unsupervised fashion. Deep Generative Model has achieved a tremendous amount of momentum with the improvisation of deep neural architecture. The goal of any generative model is to gain an insight in the underlying structure or pattern in the data distribution and therefore the key choice for such goal turns out to be neural network for its capability of approximating any function [38]. In other words, using neural networks it becomes easy for the generating models to produce new samples that follows the same probabilistic distribution of the desired samples. A rough classification of Generative Models is shown in Fig. 2. An explicit probability can be maximized for Explicit Density Generative Models. On the other hand, Implicit Density Generative Models depict a probability distribution across the location space of the data. GAN comes under the Direct Implicit Density Generative Models where the capacity to get samples from the probability distribution is often the indirect method of dealing with it. The training procedure for GANs solely uses the model’s capacity to produce samples.

## 2.2 Structure of GAN

Generically, the Generative Adversarial Networks follow the principle of Two player Zero Sum Game, where their objective is to maximize its own gain while minimizing the same for the opponent, thereby justifying the adversarial approach for the learning. GAN was first formally proposed by Goodfellow et al. [14], it has two components namely Discriminator and Generator. The job of the discriminator is to classify the image data, whether it is real or fake. On the basis of the genuineness, it assigns some score to it; while the generator seeks to improve its performance by producing realistic images, in order to fool the discriminator. Both of which tries to maximize and minimize the value function described in Equation (1). The process continues until an equilibrium is reached. As described in [14], this is achieved when data distribution of the real samples becomes equal to that of generated samples i.e.

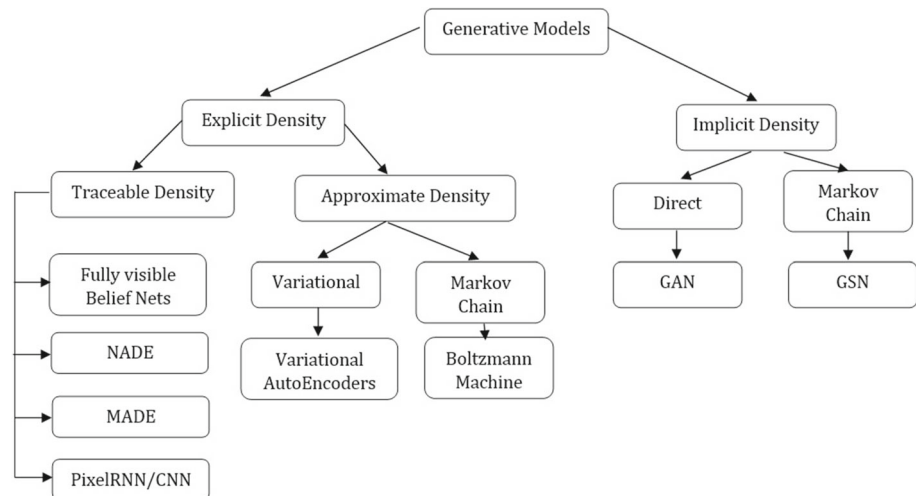


Fig. 2 Broader Generative Model Taxonomy [13]

$p_g = p_{data}$ . So ultimately forcing the discriminator to guess whether the image is real or fake i.e. 50% chances of being either of the two.

### 2.3 GAN classification based on discriminator and generator

Authors in [71], have classified various types of GANs based on their Discriminator and Generator. Although classification based on number of generators and discriminators may give an idea of nature of task it intends to do, however it may not be adequate that one architecture replicates the same behaviour in other problem description. Since our prime objective is to provide a GAN categorization used in Image Synthesis and art generation, GAN classification based only on Discriminator and Generator may not be sufficient to serve our purpose.

### 2.4 Evaluation of GANs

The main difficulty of evaluating the performance of GANs is that one have no way to know how good its performance is. The following evaluation methods are followed to check the GAN performance.

- **Layman’s Evaluation:** Checking Training curve, tuning hyper-parameters, and evaluating predictions after fixed iterations. This is Naive and tedious way to do the assessment.
- **FID based Evaluation:** Introduced by Martin et al. [17] the Frechet Inception Distance (FID) score is a statistic that determines how far apart feature vectors determined for actual and artificially created pictures are from one another. The score enumerates the statistics on computer vision aspects of the original pictures that were derived using the Inception v3 model for image classification, and it compares the two groups. A perfect score of 0.0 indicates that the two sets of photographs are identical, whereas lower values show that the two sets of images are more similar or have more in common statistically. *Lower FID scores* have been found to *correlate better* with higher quality photos when used to assess the effectiveness of images produced by generative adversarial networks.
- **IS based Evaluation:** Tim Salimans et al. [59] came up with an objective metric for assessing the calibre of generated pictures, especially artificial images produced by GAN models, is the Inception Score, or IS for short. The inception score entails classifying the produced pictures using a deep learning neural network model that has already been trained for image classification [64]. The model is used to classify lots of produced photos. In further detail, the likelihood of the image falling into each class is predicted. The inception score is then calculated by adding together these predictions. The IS Score checks for two important notions diversity and saliency of the produced results. Diversity focuses to make sure we get samples from all the classes. And Saliency make sure the produced images can be categorised with *high confidence* i.e., high scores corresponding to one class only.

### 2.5 Extensive GAN categorization

In this subsection based on the fundamental concepts of Deep Generative Models, a broad categorization of GAN has been prepared. GAN has been broadly classified in four types namely: Loss Objective based, Regularization based, Architecture based and AdHoc Application based. Fig. 3 shows the categorization of GAN.

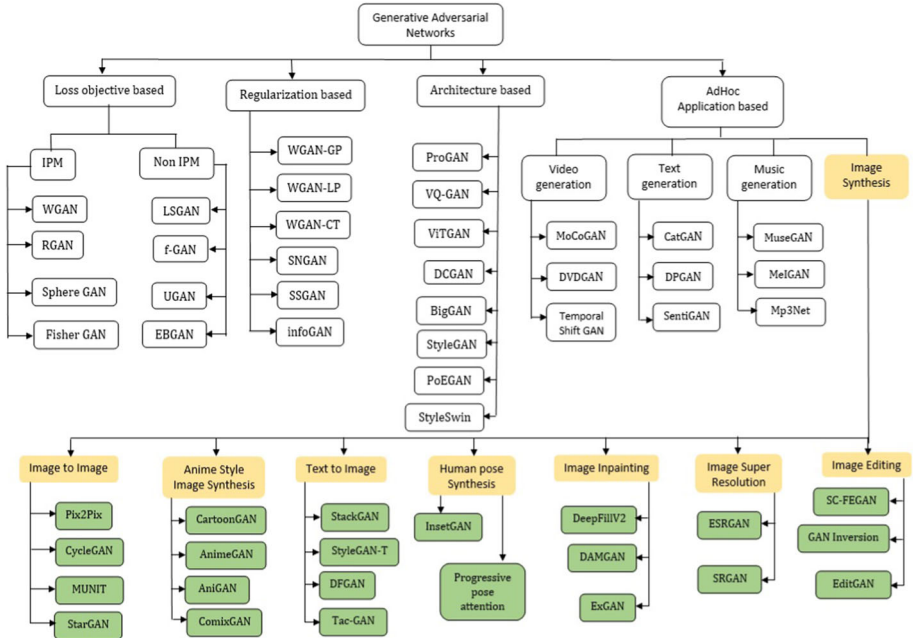


Fig. 3 Categorization of GAN

### 2.5.1 Classification based on loss objective

The Loss function are the key element in training of GANs, since they play a key role in the process of optimization of GAN. They are divided into two category namely IPM [44] and Non IPM based. In the former, the discriminator of the GAN, are subjected to certain constraint i.e. a certain group of Functions [22], as in Wasserstein GAN, it is restricted to 1-Lipschitz functions, or SphereGAN [49], where they uses hypersphere to bound the objective function. While in the latter i.e. non IPM based GANs, there are no such restrictions. Example of IPM based GAN would be WGAN [2], RGAN [22], fisherGAN [43] etc. while non IPM based GAN would be fGAN [46], UGAN [41], EBGAN [76].

### 2.5.2 Classification based on regularization

Regularization techniques play a key role in stabilizing the overall GAN training, they are incurred into the loss function of the GAN, to penalize a certain aspect of the stability problem like mode collapse problem or overfitting the real and synthesized sample as in LS-GAN [39].GANs based on use of regularization includes WGAN-GP [16], WGAN-LP [50], SNGAN [42], SSGAN [6] and many more.

### 2.5.3 Classification based on GAN architecture

In this classification, GANs are subjected to changes revolving around their architecture and how they are being used with other notable methods or models. For eg. a typical generator includes CNN based Up-Scaler [52](more specifically transposed convolutions), however

using transformer it is possible to model image generation more coherently, or dividing the generation task into multiple steps [24], wherein each steps, different aspect of the image is taken care of. This may also include fusing it with other modeling techniques like VAEs [28] or use the notion of generator in other Computer vision tasks like view synthesis [40].

### 2.5.4 Classification based on AdHoc applications

In this Sub-category, we classify GANs based on the minute custom based application as opposed to the generic task like image synthesis. For example, various tasks can be grouped together under the image synthesis part like image to image translation where we keep the semantic information of the source and try to re-draw it in the target style. Similarly image super resolution deals with blurred and noisy artifacts relating to the image. With improved quality, images tends to have good signal to noise ratio. Other tasks under these banner includes Human image pose synthesis, Anime style image synthesis, image inpainting, text to image generation i.e. based conditional image generation based on text prompt. Since, this study is mainly related with Image Synthesis, more details discussion on GAN applications in Image Synthesis (highlighted part of Fig. 3) is provided in Section 4.

## 3 Key factors affecting GAN

In this section we discuss the primary issues that researches face when they work with GANs and we also address how to fix it.

### 3.1 Issues with GANs

The following issues are primarily associated with GANs:

- **Mode Collapse** - Mode Collapse or The Helvetica Scenario [13] is a problem that appears when the generator realizes to map various input instances to the same output instance. Although unlike partial mode collapse, complete mode collapse is a phenomenon that occurs once in a blue moon.
- **Vanishing Gradients** - The problem of vanishing gradients [68] are as common in GANs as in deep neural models. If the Discriminator of the GAN gets really strong then it can inhibit the learning process of Generator, thereby reducing the gradient of Generator to almost zero. Therefore, GANs can be prevented from producing valuable score that might be beneficial for the learning process.
- **Convergence** - The issue of Convergence [29] takes place due to the way GANs are setup for the training. The phase of training of GANs are based on establishing the Nash equilibrium. GANs exhibit oscillatory behaviour and are subject to converge around local equilibrium rather than the global equilibrium.

### 3.2 Some fixes for stable GAN training

The possible remedies for avoiding instability in training GANs are as following:

- **Input Normalization:** Normalize the input images between -1 and 1, and use tanh as the last layer activation.

- **Modified Loss Function:** Using the Modified Loss function described in [14] can be fruitful. In practice, during the training of generator, flipping the label works well.
- **Modifying Sampling Technique:** Sampling techniques like spherical z in [69] can improve the performance.
- **BatchNorm, Noise and Decay:** For each set of real and fake images, construction of mini-batches for the training may be fruitful. Also, inclusion of noise to Discriminator as explained in [1] and addition of Gaussian noise to every layer of generator as discussed in [75] may be followed.
- **Avoiding Sparse Gradients:** The stability of GAN depends on the gradient where sparse gradient reduces performance score. So, a good choice would be to use LeakyReLU, for Down-sampling the image using Average pooling and for up-sampling pixelshuffle as discussed in [63]
- **Using Stability measures from Reinforcement Learning:** The technique of replay buffer and Deterministic Policy Gradient as in [51] can yield the desired result.
- **Choice of Optimizer and Loss Function:** As in [52], the author has laid emphasis on using ADAM optimizer for Generator and Stochastic Gradient Descent for Discriminator for obtaining a stable training result. However in [2], author has demonstrated that using Wasserstein distance (or earth-mover distance) solves the problem of vanishing gradients and mode collapse, along with RMSProp instead of ADAM optimizer.

Some of the helpful Loss Function for the stable training of GANs have been summarized as in Table 1.

### 4 Analysis of GAN variants in use of image synthesis

In this section we’ll take a look into the variants of the GANs that are designed with a focus on image synthesis, in order to generate various style, types, and to diversify the image generation task. Since in Computer Vision, the image synthesis is tricky to tackle, generative models like GANs comes into play. The various classifications of GANs in image synthesis are as follows:

**Table 1** GAN loss function

GAN Variant	Loss Function
GAN [13]	$L_D^{GAN} = E[\log(D(x))] + E[\log(1 - D(G(z)))]$ $L_G^{GAN} = E[\log(D(G(z)))]$
WGAN [2]	$L_D^{WGAN} = E[D(x)] - E[D(G(z))]$ $L_G^{WGAN} = E[D(G(z))]$
LSGAN [39]	$L_D^{LSGAN} = E \left[ (D(x) - 1)^2 \right] + E \left[ D(G(z))^2 \right]$ $L_G^{LSGAN} = E \left[ (D(G(z)) - 1)^2 \right]$
WGAN-GP [15]	$L_D^{WGAN-GP} = L_D^{WGAN} + \lambda E[( \nabla D(\alpha x + (1 - \alpha)G(z))  - 1)^2]$ $L_G^{WGAN-GP} = L_G^{WGAN}$
Multi-Hinge Loss [27]	$L_D = E[\text{relu}(1 - D(x))] + E[\text{relu}(1 + D(G(z)))]$ $L_G = -E[D(G(z))]$



### 4.1 Image to image

Image to Image Translation is a classical problem in Computer Vision, where the objective is to learn a mapping between a source and a target image. It has variety of application like style transfer, image enhancement, day to night translation or even seasonal manipulation etc. With the progress of GANs, this problem has been widely addressed if not completely resolved.

Following are some important variants of Image to Image classification types:

- **pix2pix** - This was introduced by Phillip et al. [20], prior to CycleGAN, with major difference being that here paired training examples have been used, so as to make it train in supervised fashion. Figure 4 shows pix2pix GAN architecture. This architecture of pix2pix follows a general U-Net [56] type architecture with having encoders and decoders with skip connections. The loss function for this variant is as follows

$$L_{GAN}(G, D) = \mathbb{E}_y[\log(D(y))] + \mathbb{E}_{x,z}[\log(1 - D(G(x, z)))] \tag{2}$$

and

$$L_{L1}(G) = \mathbb{E}_{x,y,z}[||y - G(x, z)||_1] \tag{3}$$

so the final expression becomes

$$G^* = \underset{G}{\operatorname{argmin}} \underset{D}{\operatorname{max}} L_{GAN}(G, D) + \lambda L_{L1}(G) \tag{4}$$

- **CycleGAN** - This variant of GAN was introduced by Zhu et al. [77] in 2017 and it was specifically meant for Image to Image Translation without paired image (Fig. 5). For this, the author(s) have introduced the Cycle Inconsistency loss, that indicates the quality of the generated image. The architecture of CycleGAN consist of two generators and two discriminators, implicating the intuition of cycle consistency loss. The loss function of the Cycle GAN is as follows:

$$L_{cyc}(G, F) = \mathbb{E}[||F(G(x)) - x||_1] + \mathbb{E}[||G(F(y)) - y||_1] \tag{5}$$

- **MUNIT** - Multimodal Unsupervised Image to Image Translation (MUNIT) was proposed by Huang et al. [18]. This Image Translation model presumes image representation can be divided into two parts - content code(domain invariant) and style code(domain specific property). From the source domains, which have a multimodal conditional distribution,

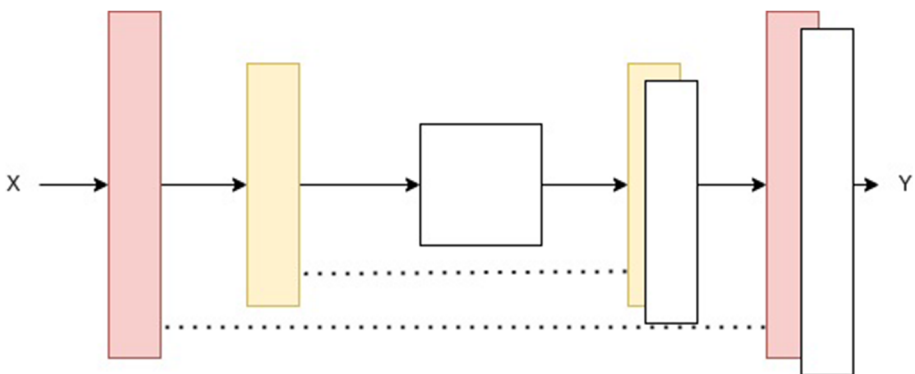
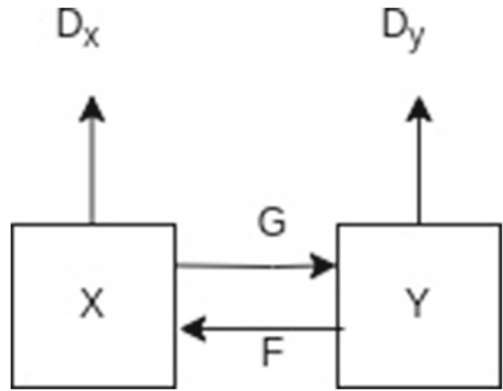


Fig. 4 Architecture of pix2pix

**Fig. 5** Architecture of CycleGAN [77]



the model can produce a variety of consequences. It trains a set of two auto-encoders, one of which encodes the image’s information and the other its style, allowing the creation of multimodal pictures. The architecture of MUNIT is in Fig. 6.

The overall Loss objective of MUNIT is as follows:

$$\begin{aligned} & \min_{E1, E2, G1, G2, D1, D2} \max L(E1, E2, G1, G2, D1, D2) \\ & = L_{GAN}^{x1} + L_{GAN}^{x2} + \lambda_x (L_{recon}^{x1} + L_{recon}^{x2}) + \lambda_c (L_{recon}^{c1} + L_{recon}^{c2}) \\ & \quad + \lambda_s (L_{recon}^{s1} + L_{recon}^{s2}) \end{aligned} \tag{6}$$

– **StarGAN**

Using just one model, the cutting-edge and scalable method StarGAN [7] can translate images between different domains. StarGAN’s unified model design enables the concurrent training of several datasets from various domains inside a single network. This results in both StarGAN’s new ability to dynamically translate an input picture to any chosen target domain. It employs an additional classifier that gains knowledge about the mapping between the original image and its associated domain. Discriminator may forecast the generated picture’s domain when Generator creates a new image conditioned on a target domain “C” (for example, brunette hair), therefore Generator will continue to generate new images until Discriminator predicts it as the target domain “C” (brunette hair). The Overall Loss Objective of StarGAN is as follows

$$\begin{aligned} L_D & = -L_{adv} + \lambda_{cls} L_{cls}^r, \\ L_G & = L_{adv} + \lambda_{cls} L_{cls}^f + \lambda_{rec} L_{rec} \end{aligned} \tag{7}$$

The architecture of StarGAN is in Fig. 7.

**4.2 Anime style image synthesis**

This subsection deals with one of the most versatile application of GANs in Anime industry. Generating animated landscape requires substantial amount of physical resource and manpower, and it tends to be laborious at time. And this is where GANs finds it usage, since it has the capability to generate images of anime characters, scenery, and even diverse art style suited as per the requirement(s).

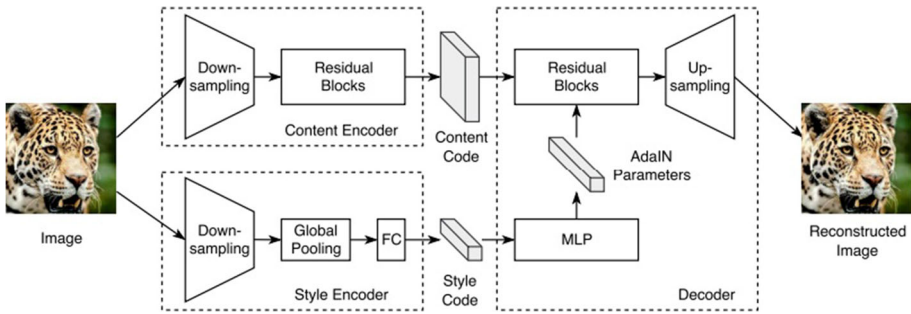


Fig. 6 MUNIT Architecture [18]

Following are some important variants of Anime Style Image Synthesis classification types:

- **CartoonGAN** - Chen et al. [5] proposed a method that takes a real world image and converts it into cartoon or anime style image. Earlier methods struggle in making animation since cartoon has rich features when it comes to textures, shading, and edges. The authors emphasised on the improvement of the image quality and also usage of novel losses for the similarity and the semantic retention of the image. The overall architecture of this GAN is in the Fig. 8.

The Loss Function for the CartoonGAN is as follows

$$L(G, D) = L_{adv}(G, D) + L_{content}(G, D) \tag{8}$$

where

$$L_{adv}(G, D) = \mathbb{E}_{c_i \sim S_{data}(c)} [\log(D(c_i))] + \mathbb{E}_{e_j \sim S_{data}(e)} [\log(1 - D(e_j))] + \mathbb{E}_{p_k \sim S_{data}(p)} [\log(1 - D(G(p_k)))] \tag{9}$$

and

$$L_{con}(G, D) = \mathbb{E}_{p_i \sim S_{data}(p)} [\|VGG_I(G(p_i)) - VGG_I(p_i)\|_1] \tag{10}$$

The Adversarial Loss takes care of issues related to the sharp edges, while the content loss ensures the smoothness in shading and textures.

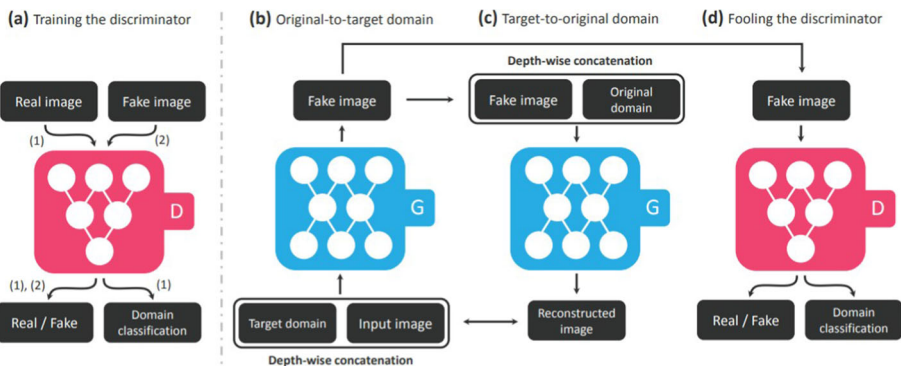


Fig. 7 StarGAN Architecture [7]

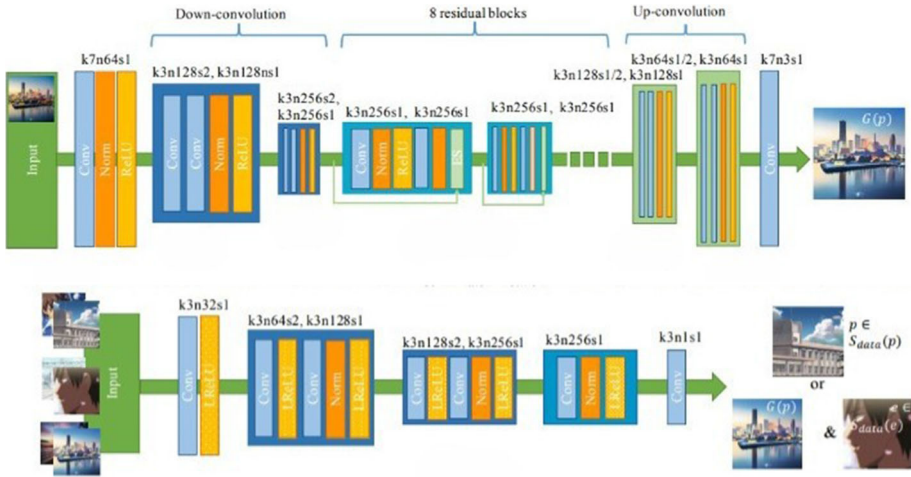


Fig. 8 The architecture of the CartoonGAN from [5]

- **AnimeGAN** - Jie Chen et al. [4] proposed a lightweight implementation of the GAN that turns a natural realistic image into an Anime rich style image. This approach fuses neural style transfer technique with GAN in order to create an Anime like texture. As discussed in the paper, the parameters of the proposed GAN requires lower memory capacity, thus making it lightweight for the end to end application. The architecture of the AnimeGAN is in the Fig. 9.

The Loss function for this GAN variant is as follows

$$L(G, D) = \omega_{adv}L_{adv}(G, D) + \omega_{con}L_{con}(G, D) + \omega_{gra}L_{gra}(G, D) + \omega_{col}L_{col}(G, D) \tag{11}$$

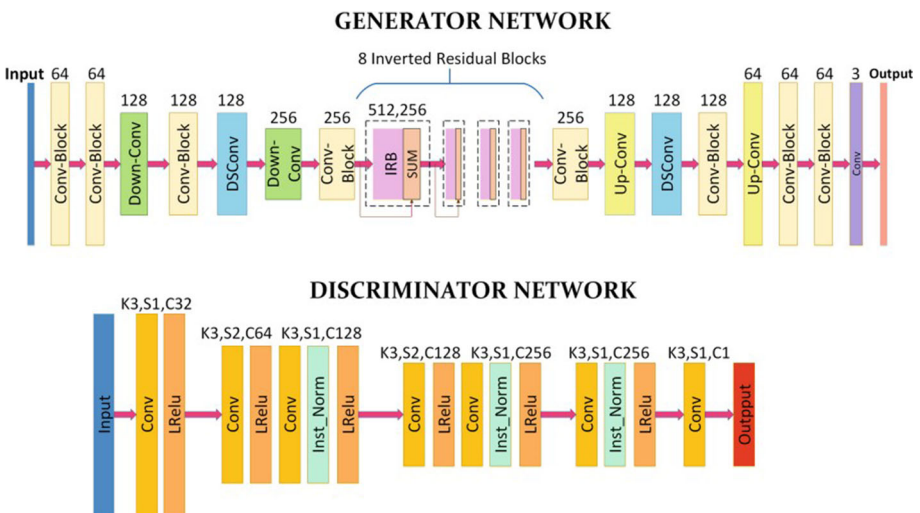


Fig. 9 AnimeGAN Architecture from [4]

The principal loss function of this variant consists of four Loss terms,  $L_{adv}$  or adversarial loss that affects animation transformation process,  $L_{con}$  is the content loss which helps to determine the overall content retention,  $L_{gra}$  is the grayscale style loss that helps images so generated to have clear texture and lines in the image, and finally  $L_{col}$  is the color reconstruction loss, that signifies how much the generated image has retained the original color of the input image.

- **AniGAN** - Bing et al. [37] proposed styled guided GAN for the Generation of the anime version of the photo of the face, given a reference or source image. The style transfer task is often deemed as difficult due to the high degree of variance amongst the anime faces giving rise to the complexities, also to preserve the generated faces from distortion and other anomalies. The paper implements new generator architecture and two normalization functions that retains semantic information from the source image and wither away the artifacts. The Architecture of AniGAN is in the Fig. 10. The Loss Function for the AniGAN is

$$L_{adv} = \mathbb{E}_x[\log(D_X(x))] + \mathbb{E}_{x,y}[\log(1 - D_X(G(y, x)))] + \mathbb{E}_y[\log(D_Y(y))] + \mathbb{E}_{y,x}[\log(1 - D_Y(G(x, y)))] \tag{12}$$

$$L_{fm} = \mathbb{E}_h[\sum_{k \in K_1} \|\overline{D}_U^k(h) - \overline{D}_U^k(G(h, h))\|_1] \tag{13}$$

$$L_{rec} = \|G(x, x) - x\|_1 \tag{14}$$

and the overall principal loss is as follows

$$L_G = L_{adv} + \lambda_{rec} \cdot L_{rec} + \lambda_{fm} \cdot (L_{fm} + L_{dfm}) \tag{15}$$

$$L_D = -L_{adv} \tag{16}$$

- **MontageGAN** - It is yet another variant that trains multiple parts or layers of the image and then place it together like a puzzle in order to form a larger and meaningful image. The authors of the paper [62] proposes a method that comprises of two step. The first step trains GAN to generate different part of image, followed by a global GAN that learns to put various parts generated in previous step, so as to form a complete image. The architecture of the MontageGAN is given in the Fig. 11. One thing to note here is that the

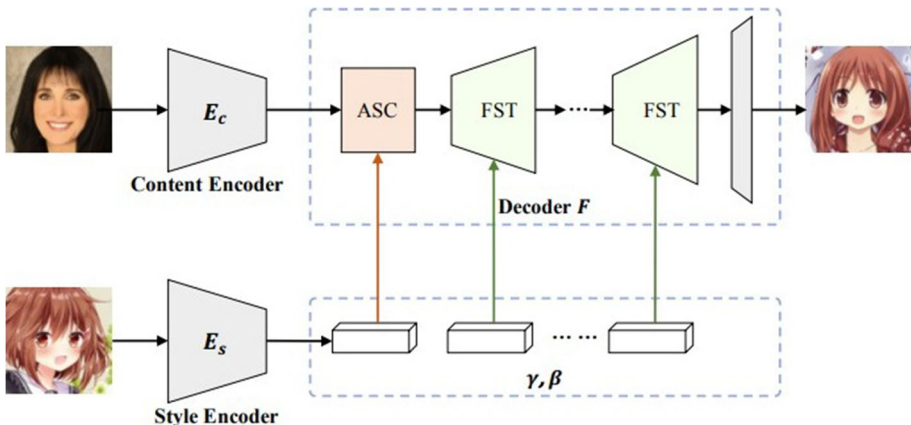


Fig. 10 Architecture of AniGAN [37]

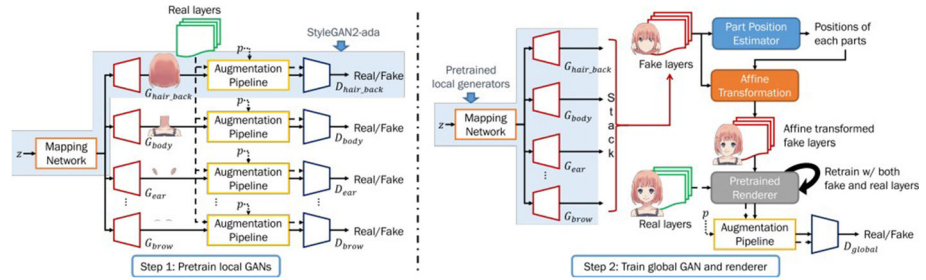


Fig. 11 The Architecture of MontageGAN [62]

authors used the StyleGAN2-ada [26] for both local and global GAN, by modifying its parameters for the purpose here. For training puposes, the pixelwise Mean-Squared-Error is calculated for the target image and the rendered image.

### 4.3 Text to image

Text-to-Image Creation using Generative Adversarial Networks (GAN) can generate pictures from text descriptions. Instead of giving the generator only noise as input, the textual description is first converted into a text embedding, combined with the noise vector, and then given as input. This formulation will help the generator produce images that are in line with the input description rather than randomly producing images. Following are some important variants of Text to Image classification types:

- **StackGAN** - Reed et al. [54] proposed a text guided method for image synthesis using DCGAN in 2016. In that work the authors to bridge the gap between text and image synthesis by using a novel architecture that formulated a way to control synthesis of visuals with the text. The architecture for their model is shown in Fig. 12. The loss function for their model is

$$L_{style} = \mathbb{E}_{t, z \sim N(0,1)} \|z - S(G(z, \varphi(t)))\|_2^2 \tag{17}$$

where  $\varphi(t)$  is the text encoding, G is the generator and z is the noise sampled from Normal Distribution.

- **StyleGAN-T** - It has been proposed by Axel et al. [60] is one of the fastest Text to Image GAN based on StyleGAN-XL [61]. StyleGAN-T takes into account the unique needs of large-scale text-to-image synthesis, including enormous capacity, robust training on datasets like MS-COCO [33], strong text alignment, and a balance between controlled

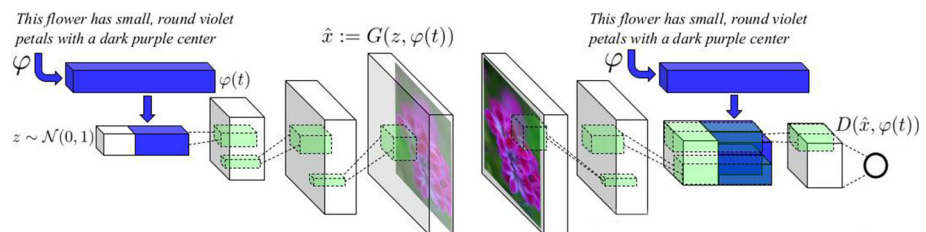


Fig. 12 StackGAN architecture [54]

variation and text alignment. With respect to sample quality and speed, StyleGAN-T greatly exceeds earlier GANs and distilled diffusion models [45, 55, 58], the earlier state-of-the-art models in quick text-to-image synthesis. The Architecture of StyleGAN-T is shown in Fig. 13. The Loss objective of StyleGAN-T is based on CLIP Encoder [53], hence it has been formulated as

$$L_{CLIP} = \arccos^2(c_{image} \cdot c_{text}) \tag{18}$$

– **DFGAN**

Proposed by Ming et al. [66], Deep Fusion GAN(DF-GAN) addresses the issue of complexity of generating vivid images from textual description. In their work they formulate a *novel approach* wherein a one-stage text-to-image backbone that can immediately create high-resolution images without becoming entangled with several generators. And a custom Discriminator with Machine Aware-Gradient Penalty(MA-GP), which dramatically improves a text-image semantic consistency with minimum overheads. Finally, a new Deep text-image fusion block (DFBlock), which more thoroughly integrates text and visual elements. The architecture of DF-GAN is in Fig. 14.

The overall loss function of this approach turns out to be as follows:

$$L_D = -\mathbb{E}_{x \sim P_r} [\min(0, -1 + D(x, e))] - (1/2)\mathbb{E}_{G(z) \sim P_g} [\min(0, -1 - D(G(z), e))] - (1/2)\mathbb{E}_{x \sim P_{mis}} [\min(0, -1 - D(x, e))] + k\mathbb{E}_{x \sim P_r} [(\|\nabla_x D(x, e)\| + \|\nabla_e D(x, e)\|)^p] \tag{19}$$

and

$$L_G = -\mathbb{E}_{G(z) \sim P_g} [D(G(z), e)] \tag{20}$$

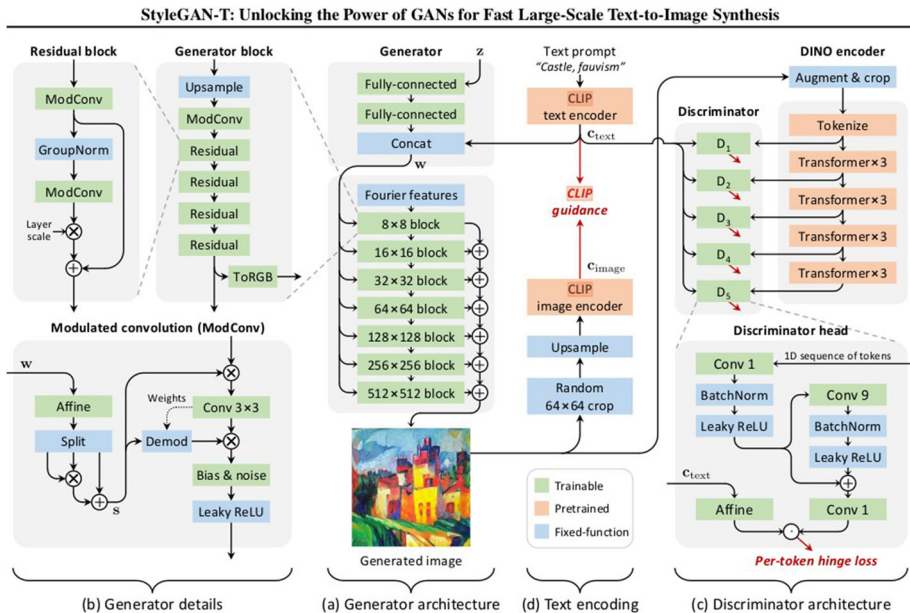


Fig. 13 StyleGAN-T Architecture [60]



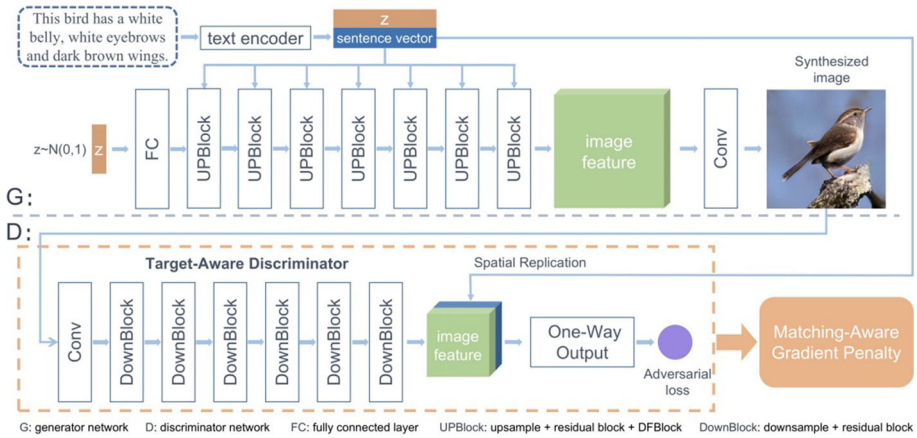


Fig. 14 Architecture of DFGAN [66]

- **TAC-GAN** Text Conditioned Auxiliary Classifier GAN or shortly TACGAN [8] is a GAN variant that converts a textual description to corresponding picture. By configuring the produced pictures on a text description rather than a class label, TAC-GAN improves upon AC-GAN [47]. A noise vector and another vector with an embedded representation of the textual description serve as the foundation for the input vector of the generative network in the TAC-GAN model. The Discriminator shares similarities with the AC-GAN, however it has been enhanced to accept text data as input before accomplishing classification. The Final loss objective is as follows

$$L_{D_y} = H(D_y(I_r, l_r), Q_r) + H(D_y(I_f, l_r), Q_r) + H(D_y(I_w, l_r), Q_w) \quad (21)$$

and

$$L_{G_y} = H(D_y(I_f, l_r), Q_f) \quad (22)$$

### 4.4 Human pose synthesis

Human Pose Synthesis is a challenging problem in computer vision. It involves the modification of image with respect to the pose of the subject, to a desired target orientation or pose. The majority of current techniques use a decoder to create the image texture for the target posture after synthesising the texture of the entire reference human picture into a latent space. Yet, it is challenging to reconstruct the entire human image’s precise texture. Following are some important variants of Human Pose Synthesis classification types:

- **InsetGAN** - Full body image generation is also a trickier problem in computer vision, prior techniques have taken pose generation into account. The author of InsetGAN [12] has devised a novel way to deal with this problem. This proposed method works in a multi-GAN setting, the prime contribution of this work makes it produce 1024x1024 resolution image by taking latent encodings of multiple generators and combining them, followed by the optimization. The architecture of this GAN is in the Fig. 15. The Loss function for this variant is combination of various other losses as follows:

$$\min_{(w_a, w_b)} (L_{coarse} + L_{border} + L_{reg} + L_{face} + L_{body}) \quad (23)$$



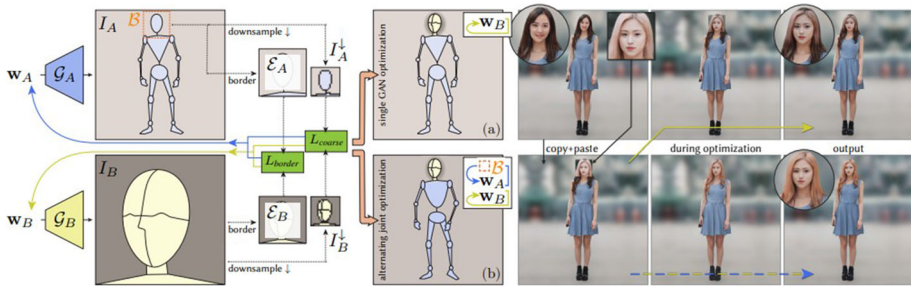


Fig. 15 InsetGAN Architecture from [12]

where

$$L_{coarse} := \lambda_1 L_1(I_A^\downarrow, I_B^\downarrow) + \lambda_2 L_{lips}(I_A^\downarrow, I_B^\downarrow) \tag{24}$$

$$L_{border} := \lambda_3 L_1(\varepsilon_8(B(I_A)), \varepsilon_8(I_B)) + \lambda_4 L_{lips}(\varepsilon_8(B(I_A)), \varepsilon_8(I_B)) \tag{25}$$

$$L_{border} := \lambda_{r1} \| \mathbf{w}^* - \mathbf{w}_{avg} \| + \lambda_{r2} \sum_i \| \delta_i \| \tag{26}$$

and

$$L_{face} := \lambda_7 L_1(R^I(I_B), R^I(I_{ref})) + \lambda_8 L_{lips}(R^I(I_B), R^I(I_{ref})) \tag{27}$$

– **Progressive Pose Attention Transfer for Person Image Generation** - Proposed by Zhen et al. [78], this GAN is made for Pose transfer i.e. transferring pose of source image to the target. The network’s generator is made up of a series of Pose-Attentional Transfer Blocks, each of which transmits particular regions it attends to in order to gradually create the human picture. In comparison to those in earlier studies, produced human images are much more realistic-looking because they have superior appearance and form coherence with the input images. On datasets like Market-1501 and DeepFashion [36], the effectiveness and efficiency of the suggested network are validated numerically as well as qualitatively. The generator architecture is in Fig. 16. The overall loss function for this variant of GAN can be summarised as

$$L_{full} = \underset{G}{\operatorname{argmin}} \underset{D}{\operatorname{max}} \{ \alpha * L_{GAN} + L_{combL1} \} \tag{28}$$

### 4.5 Image inpainting

The art of rebuilding damaged or missing portions of a picture is known as image inpainting, and it is easily adaptable to films. The usage of image inpainting has opened up a wide range of possibilities. An artificial image inpainter’s primary goal is to create pictures in which the blank spaces have been filled with appeal that is both aesthetically and linguistically realistic. We may safely acknowledge that it is a difficult assignment. Following are some important variants of Image Inpainting classification types:

- **DeepFillV2** - Yu et al. [72] introduced a GAN that uses gated convolution and attention for the image in-painting job. This technique is useful in lot of scenarios since it uses free-form mask that can be used anywhere in the image. The Architecture of DeepFillV2 is shown in Fig. 17.

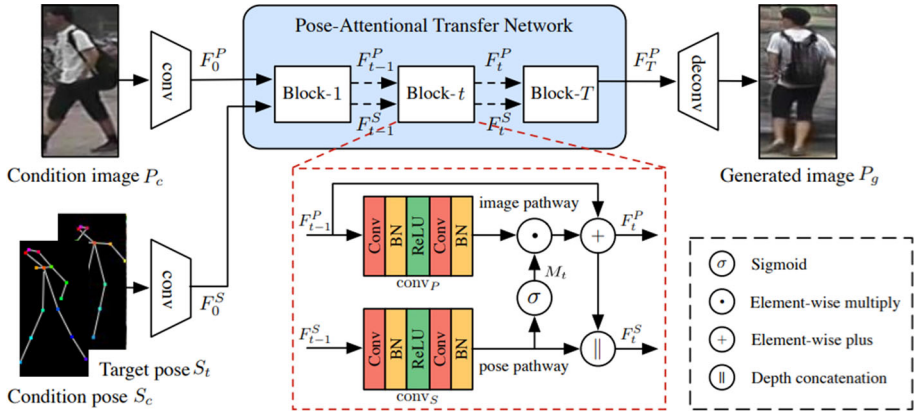


Fig. 16 Architecture of progressive pose transfer network [78]

They introduced a patch based GAN loss function, that uses the following loss function

$$L_G = \mathbb{E}_{z \sim P_z(z)} [D^{sn} G(z)] \tag{29}$$

for the Generator and for the discriminator

$$L_{D^{sn}} = \mathbb{E}_{x \sim P_{data}(x)} [\max(0, 1 - D^{sn}(x))] + \mathbb{E}_{z \sim P_z(z)} [\max(0, 1 + D^{sn} G(z))] \tag{30}$$

• DAMGAN

Although several prominent GAN-based networks have already been suggested for picture inpainting, synthesised images still have pixel errors or colour inconsistencies during the production process, which are typically referred to as false textures. Cha et al. [3] proposes a GAN-based model employing dynamic attention map which focuses on identifying false texture and generates dynamic attention maps to lessen pixel inconsistency from the feature maps in the generator. This is done to decrease pixel inconsistency disorder caused by fake textures. The Architecture of the DAMGAN is in Fig. 18.

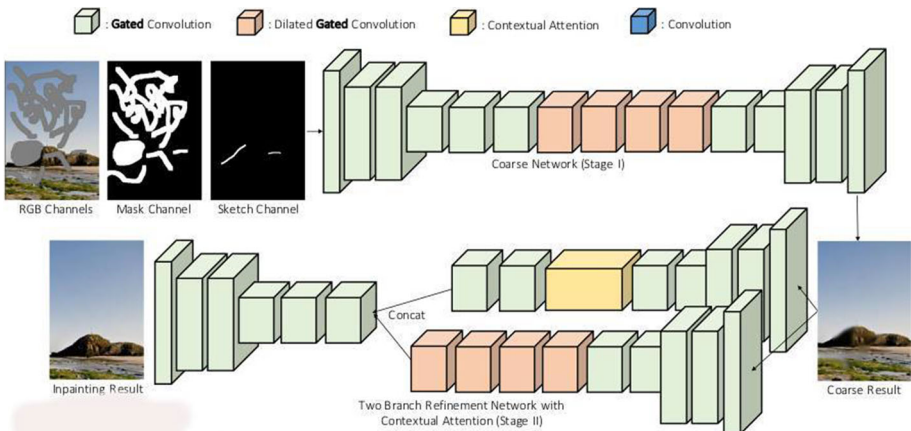


Fig. 17 DeepFillV2 architecture [72]

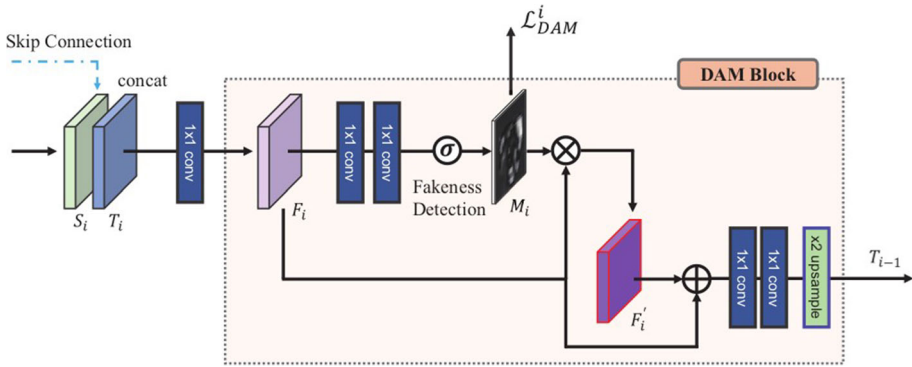


Fig. 18 Dynamic Attention Map of DAMGAN [3]

The full loss objective of DAMGAN is as follows

$$L_{total} = \lambda_{re} \cdot L_{re} + \lambda_{adv} \cdot L_{adv} + \lambda_{DAM} \cdot L_{DAM} \tag{31}$$

• **ExGAN**

Dolhansky et al. [10] proposed a **novel approach** for in-painting. The Authors proposes a conditional GAN that generates tailored, high-quality in-painting results from exemplar data. The authors suggest utilising either a reference image of the area to be in-painted or a perceptual code defining that object as example information. This additional information may be introduced into the adversarial network several times, boosting its descriptive power, in contrast to earlier conditional GAN formulations. The loss objective for ExGAN is as follows

$$\begin{aligned} \min_G \max_D V(D, G) = & \mathbb{E}_{\mathbf{x}_i, \mathbf{c}_i \sim p_{data}(\mathbf{x}, \mathbf{c})} [\log D(\mathbf{x}_i, \mathbf{c}_i)] \\ & + \mathbb{E}_{\mathbf{c}_i \sim p_c, G(\cdot) \sim p_z} [\log(1 - D(G(\mathbf{x}_i, \mathbf{c}_i)))] + \|G(\mathbf{x}_i, \mathbf{c}_i) - \mathbf{x}_i\|_1 \\ & + \|C(G(\mathbf{x}_i, \mathbf{c}_i) - \mathbf{c}_i)\|_2 \end{aligned} \tag{32}$$

**4.6 Image super-resolution**

Images can be enhanced and have their resolution increased using both traditional and cutting-edge super-resolution techniques. It is frequently used in the following applications, such as surveillance, to locate, recognise, and apply facial recognition to low-resolution pictures captured by security cameras. Following are some important variants of Image Super-resolution classification types:

- **ESRGAN** - Image Super Resolution is one of the sophisticated task in computer vision, prior to Deep Learning method, classical statistical method has been employed to solve the problem, however those methods didn't yielded the desired result. SRGAN [31] made significant progress but still lacking in problems dealing with fine grain details and artifacts. ESRGAN by Wang et al. [67] has deal with the problem by working on network architecture of SRGAN and modifying the loss function. The Architecture of ESRGAN is in Fig. 19. The loss function for the Generator of ESRGAN is

$$L_G = L_{percep} + \lambda L_G^R + \eta L_1 \tag{33}$$

### Residual in Residual Dense Block (RRDB)

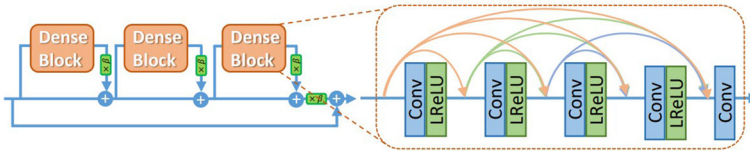


Fig. 19 ESRGAN architecture

where

$$L_1 = \mathbb{E}_{x_1} \|G(x_i - y)\|_1 \tag{34}$$

and

$$L_G^{Ra} = -\mathbb{E}_{x_r} [\log(1 - D_{Ra}(x_r, x_f))] - \mathbb{E}_{x_f} [\log(D_{Ra}(x_f, x_r))] \tag{35}$$

while the loss for the Discriminator of the ESRGAN is as follows

$$L_D^{Ra} = -\mathbb{E}_{x_r} [\log(D_{Ra}(x_r, x_f))] - \mathbb{E}_{x_f} [\log(1 - D_{Ra}(x_f, x_r))] \tag{36}$$

- **SRGAN** - Introduced by Ledig et al. [31], It's the first framework capable of predicting natural images that are four times more photorealistic. It makes use of a perceptual loss function with adversarial loss and content loss components. Using a discriminator network that has been trained to distinguish between the super-resolved pictures and the original photo-realistic images, the adversarial loss drives the output to the natural image manifold. The authors also employ a content loss that is driven by perceptual similarity rather than resemblance in pixel space. The goal of this design is to preserve the image's finer textures when we upscale it, guaranteeing that its quality is unaffected. Other techniques, such as bilinear interpolation, can be used to accomplish this task, but they suffer from smoothing and picture information loss. The authors of this research suggested two designs, one without GAN (SRResNet), and the other with GAN (SRGAN). It is determined that SRGAN generates images that are more aesthetically beautiful and has greater accuracy than SRGAN. The architecture of SRGAN is in Fig. 20.

The Loss function for this GAN is as follows

$$l^{SR} = l_X^{SR} + 10^{-3} \cdot l_{gen}^{SR} \tag{37}$$

#### 4.7 Image editing

Recently, interest in deep learning-based image editing has grown, especially with the introduction of GANs. The development of GAN-based picture editing has led to its widespread use in computer vision. There are several methods for tweaking photos. Following are some important variants of Image Editing classification types:

- **SC-FEGAN**

Youngjoo et al. [23] proposed a novel method for image editing., by making use of an end-to-end trainable CNN, that synthesizes image, when user(s) gives input in form of Sketch, color, free-form mask etc. Moreover the authors formulates an additional style loss, the strategy can also provide outcomes that are realistic. Using the suggested network design and loss algorithms, it can provide results of excellent quality and realism. The architecture of SC-FEGAN is in Fig. 21.

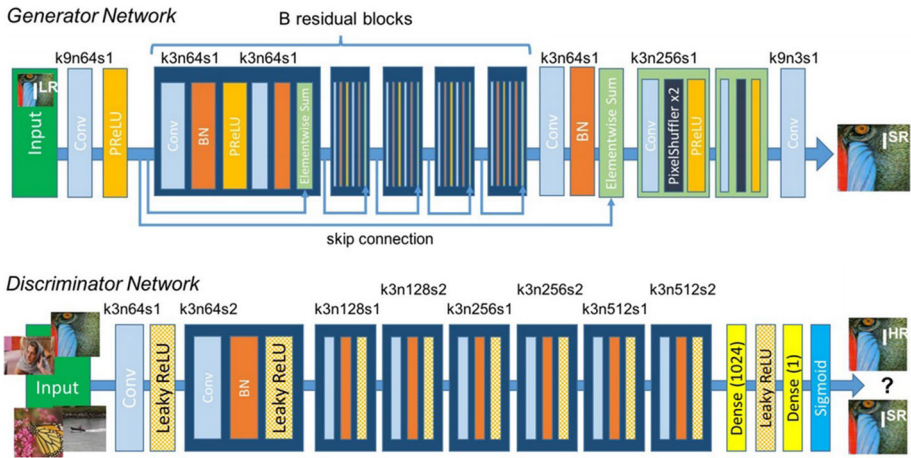


Fig. 20 SRGAN Architecture [31]

The Loss Functions of SC-FEGAN are given below:

$$L_{G_{SN}} = -\mathbb{E} [D (I_{comp})] \tag{38}$$

and

$$L_G = L_{per-pixel} + \sigma L_{percept} + \beta L_{G_{SN}} + \gamma (L_{style} (I_{gen}) + L_{style} (I_{comp})) + \nu L_{tv} + \epsilon \mathbb{E} [D (I_{gt})^2] \tag{39}$$

and

$$L_D = \mathbb{E} [1 - D (I_{gt})] + \mathbb{E} [1 + D (I_{comp})] + \theta L_{GP} \tag{40}$$

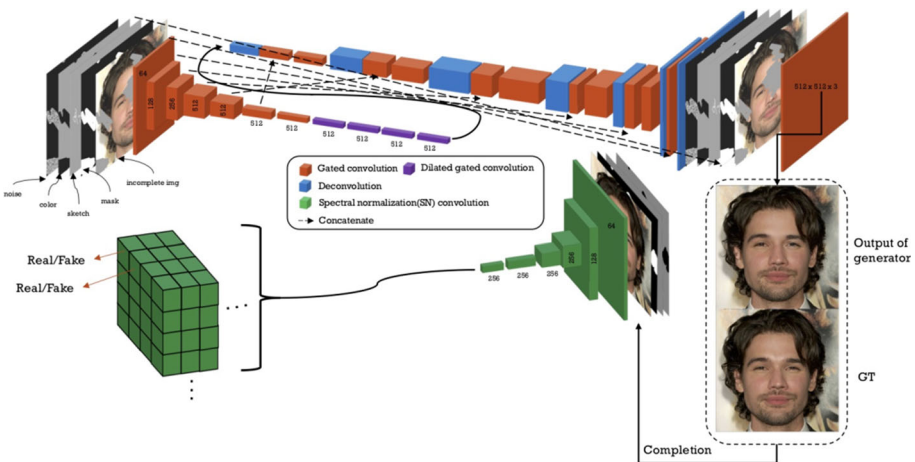


Fig. 21 SC-FEGAN Architecture [23]

– **In Domain GAN Inversion**

Zhu et al. [79], proposed a image editing method using GAN inversion method. Typically, Generator in GANs learns a mapping from latent space to image distribution, however this way it won't yield great result since there's little to control the feature of image in synthesis process. GAN inversion is based on inverse mapping from image distribution to latent space. And this way most accurate latent space can be figure out that brings out the desired featured in edited image.

In addition to accurately reconstructing the input picture, author's in-domain GAN inversion technique also makes sure that the inverted code is semantically intelligible for editing. In order to project a given picture into the native latent space of GANs, the model must first train a brand-new domain-guided encoder. Then, in order to improve the recovery of the target picture and fine-tune the code generated by the encoder, we suggest domain-regularized optimization. Many trials indicate that this inversion approach greatly outperforms state-of-the-arts in real picture reconstruction while also making other image altering chores easier. The Architecture of this GAN variant is in the Fig. 22. The Loss objective is summarised as follows

$$z^{inv} = \arg \min_z \|x - G(z)\|_2 + \lambda_{vgg} \|F(x) - F(G(z))\|_2 + \lambda_{dom} \|z - E(G(z))\|_2 \tag{41}$$

– **EditGAN**

EditGAN, proposed by Huan et al. [34], is a first of its kind GAN driven **novel** image editing framework, that allows high level and high precision image editing and that too in semantic level, by making use of segmentation mask. In order to get the intended output image with only the required features modified, manipulating particular features typically requires enormous datasets and specialists to know which characteristics to change inside the model. Instead, EditGAN learns to match segmentations to photos using just a small number of labelled instances, enabling one to alter the images using segmentation, or to put it another way, with rapid drawings. It maintains the complete image quality while enabling a degree of flexibility and detail that has never been possible before.

By Adjusting the segmentation mask to the intended edit and optimising the latent code to be compatible with the new segmentation mask, the RGB picture is essentially changed throughout the image modification or editing process. EditGAN learns the editing vectors in latent space that carry out the changes in order to increase editing efficiency. These editing vectors may be used to modify other pictures directly, with no further optimization

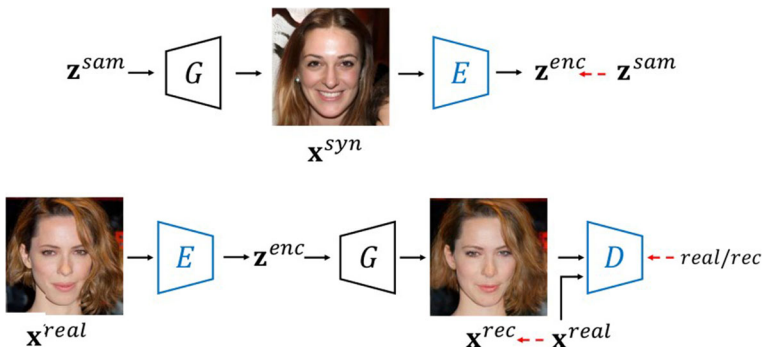


Fig. 22 In-Domain GAN inversion architecture [79]



procedures or very little. The loss objective for EditGAN is as follows:

$$L_{editing}(\delta w^+) = \lambda_1^{editing} L_{RGB}(\delta w^+) + \lambda_2^{editing} L_{CE}(\delta w^+) + \lambda_3^{editing} L_{ID}(\delta w^+) \tag{42}$$

The EditGAN pipeline is in the Fig. 23.

### 4.8 Other prominent models for image synthesis

In this section, a discussion has been made based on some of the important GAN variants those have shown significant results in the past decade. A few of the variants of GANs discussed in this subsection focuses on view synthesis, which is another interesting research area in Computer Graphics; while we also bring out some of the State-Of-The-Art GAN models, which has gained a widespread popularity based on its performance.

- **VQ-GAN** - Vector Quantized GAN aka VQ-GAN [11] is a type of GAN variant that takes and combine the advantages of CNN and Transformer architecture. Specifically it is an amalgamation of VQ-VAE [48] and GAN [14]. Traditional CNN architecture fails to encompass overall global context w.r.t. semantic relations in the image, thereby with the help of transformer it facilitate the modeling of long range dependency in pixel of image by representing it discretely.

Further the scaling issue of the transformer is resolved with the help of the codebook, which serves as the intermediary between the learned part of the visual aspect of the image and sequence processing of transformer. It is achieved by the process of vector quantization, which is a signal processing trick to encode an information, here its the image being represented as vectors in consideration.

The principal loss function for this GAN is as follows

$$L_{VQ}(E, G, Z) = \|x - \tilde{x}\|^2 + \|sg[E(x)] - z_q\|_2^2 + \|sg[z_q] - E(x)\|_2^2 \tag{43}$$

$$L_{GAN}(\{E, G, Z\}, D) = [\log(D(x)) + \log(1 - D(\tilde{x}))] \tag{44}$$

And

$$L_{transformer} = \mathbb{E}_{x \sim p(x)} [-\log(p(s))] \tag{45}$$

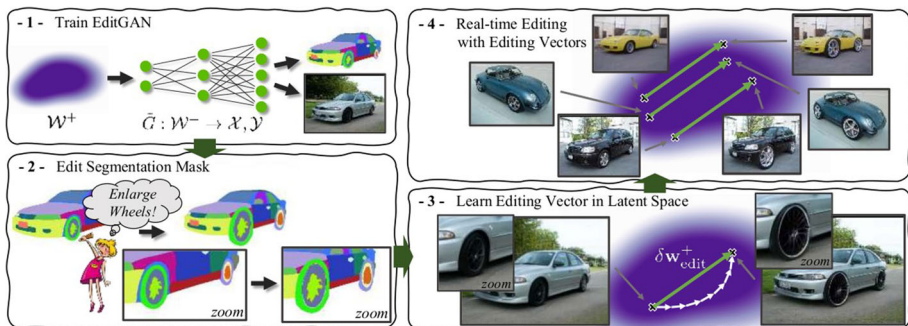


Fig. 23 EditGAN Pipeline [34]

- **ViTGAN** - This variant of GAN explores the idea of using the Vision transformer in the generic GAN architecture. ViTGAN [32] is a GAN with self-attention and novel regularisation strategies that overcomes the unstable training of Vision Transformers. It is created by combining a prominent vision transformer approach based on patch tokens, which is commonly employed in classification applications, with the GAN framework. On some datasets, ViTGAN outperforms StyleGAN2 despite only having a 64x64 resolution. The main idea behind this approach is
  - **Better Regularization of the Discriminator:** Replacing the dot product with Euclidean distance in self attention mechanism and tying the weight of projection matrix of query and key.
  - **Better Spectral Normalization:** Given their sensitivity to the size of the Lipschitz constant, standard spectral normalisation in addition to R1 gradient penalty slow down the training of GANs using Transformer blocks. A quick approach is to enhance the spectral norm of the projection matrix by multiplying the normalised weight matrix of each layer by the spectral norm at startup.
  - **Overlapped Image Patches:** The authors let a little amount of patch overlap to avoid the discriminator over-fitting and memorization of the local cues from the normal non-overlapping patches. They assert that it might give the Transformer a stronger sense of place and make it less sensitive to predetermined grids.
- **StyleSwin** - Inspired by Swin Transformer [35], Bowen et al. introduces StyleSwin [73], a GAN variant that utilizes Swin Transformer, in its Generator with little tweaks in order to gain performance edge compared to other State-Of-The-Art models. As described in the paper, authors emphasizes Swin transformer is used in the proposed generator's style-based design. The authors suggest twofold attention, which concurrently uses the context of the local and the shifted windows, to produce a bigger receptive field and better generation quality.
 

Moreover, it demonstrates how supplying the absolute position knowledge that window-based transformers have lost is extremely beneficial to the generation quality. The rich expressivity of transformers benefits both the coarse geometry and the fine structures in the proposed StyleSwin, which is adaptable to high resolutions. Blocking artefacts do, however, poses problem during high-resolution synthesis because executing the local attention block-by-block might cause the spatial coherency to be broken. The authors utilise a wavelet discriminator to investigate the spectrum disparity and successfully suppress the artefacts as a solution to this problem. Many tests demonstrate their superiority to earlier transformer-based GANs, particularly at high resolutions like 1024x1024.
- **PoEGAN** - Introduced by Xun Huang et al. PoEGAN [19] is an unique variant of GAN, that uses multi modality of user's input. Put it differently, it has the capability of synthesizing multifarious images using multiple modality like segmentation mask, sketches, style references and text etc. The gist behind this approach is **Product of Expert**, as described by authors in the paper, as an approach that involves multiplying the probabilities together, then re-normalizing, to combine different probabilistic models of the same data. This is a particularly effective technique for modelling high-dimensional data, which concurrently satisfies a wide variety of low-dimensional requirements. In other words, the goal is to learn a generative model that can captures the multiple modality aspect tied to the image, with vivid features satisfying the imposed constraints. The training objective for PoEGAN can be formulated as

$$L^G = L_{GAN}^G + L_{KL} + \lambda_1 L_{cx} + \lambda_2 L_{cy}^G \quad (46)$$



and

$$L^D = L_{GAN}^D + \lambda_2 L_D^{cy} + \lambda_3 L_{GP} \quad (47)$$

- **StyleGAN-XL** - This is the first GAN variant which was capable of generating images at high scale resolution 1024x1024, proposed by Axel et al. [61], StyleGAN-XL is capable of inversion operation and image manipulation tasks. However, there are some limitation to StyleGAN-XL, since it is approximately three times bigger than StyleGAN3 [25] therefore incurring a high cost for fine-tuning, and as noted by the authors, it is still under performing when compared to diffusion based models [45].
- **SURF-GAN** - Kwak et al. [30] came up with a **novel 3D aware** GAN, namely SURF-GAN, an unique 3D-aware GAN that in unsupervised fashion finds controlled semantic characteristics. They created a 3D controllable generator that is capable of explicit control over posture and 3D consistent editing by injecting editing instructions from the low-resolution 3D-aware GAN into the high-resolution 2D StyleGAN. Their approach is seamlessly compatible with a number of well-researched 2D StyleGAN-based methods, including inversion, editing, and stylization. It also finds application in view synthesis. The loss objective for canonical view generation is given below

$$L^c = \lambda_1 L_W^c + \lambda_2 L_I^c + \lambda_3 L_{LPIPS}^c \quad (48)$$

and for target view generation it is

$$L^t = \lambda_4 L_W^t + \lambda_5 L_I^t + \lambda_6 L_{LPIPS}^t + \lambda_7 L_{reg} \quad (49)$$

A brief summary of the various GAN variants is represented in Table 2.

## 5 Discussion

In this section, a brief discussion about the article selection criteria has been carried out. Also key research challenges are identified for the future researchers in this field.

### 5.1 Various criteria for selection of the research articles

- **Sources and the Database(s):** Authors of this paper extensively relied on various available repository sources like arxiv repository, Springer, ACM Digital Library, IEEE Xplore, Semantic Scholars, OpenReview etc. In this flow of searching methodology Connected papers, Web of Science and PaperwithCode have been used significantly. Visuals tools available in Connected papers have helped to grasp the overall flow of dependency of related works, viz prior and derivative works related to the subject in focus.
- **Selection Criteria:** This work comprises of selected GAN models from 2014-2023 based on their relevance, peer impression, application to the wide domain of image synthesis. Use of keywords such as: GAN, image synthesis, text2image synthesis, GAN and NeRF etc have been used frequently. Based on this, preliminary screening has been carried out to select relevant papers pertaining to defined task. To diversify the search result, application aspect of the domain has been put through more emphasis that is reflected in categorization of GAN in this work.

**Table 2** Summarization of GAN Variants

Task	Year	Variants	Highlight(s)	Performance
Image to Image Translation	2017	pix2pix	Based on Conditional GAN. Utilises paired training approach.	Class IOU 0.18
	2017	CycleGAN	Unsupervised approach. Uses novel cycle consistency loss to improve the quality of image conversion.	Class IOU 0.11
	2018	MUNIT	Unsupervised approach but with multi modal conditional distribution. It may regulate the translation's style with the illustration style.	CIS 1.039
	2018	StarGAN	Capable of learning mapping across multiple domain with single generator and discriminator.	Error 2.12%
Anime Style Image Synthesis	2018	CartoonGAN	Transforms Photo-realistic images to HQ cartoon style images. Learning of different cartoon style theme is possible.	-
	2020	AnimeGAN	Amalgamation of GAN and NST technique. Lightweight implementation requires low memory without compromising the image quality.	-
	2021	AniGAN	Creates style guided anime images of portrait coherent to source images. Preserve the semantic information of the source.	FID 38.45
	2022	MontageGAN	Multi layer image generation using Local GAN and Global GAN.	FID 56.65
Text to Image	2016	StackGAN	Generates semantically closed images based on text description by two stage process.	IS 3.7
	2017	TAC-GAN	Based on AC-GAN, images are generated based on text description instead of class labels.	IS 3.45
	2022	DF-GAN	One stage text to image backbone for high fidelity image generation. Deep fusion block for proper association of visual and textual features.	FID 19.32
	2023	StyleGAN-T	Probably one of the Best GAN for the designated task. Utilises StyleGAN-XL with CLIP.	FID 7.3
Human Image/Pose Synthesis	2019	Progressive Attention Pose Transfer	Uses sequences of Pose transfer attention for various region. Better consistency.	IS 3.209
	2022	InsetGAN	Human body image synthesis is done by pretrained GAN models. Generates high resolution image under a generic canvas.	FID 25.33
Image Inpainting	2017	ExGAN	Conditional GAN that uses exemplar knowledge viz. referenced image with particular region of interest for high quality image generation.	FID 11.27

**Table 2** continued

Task	Year	Variants	Highlight(s)	Performance
Image Super Resolution	2018	DeepFillV2	Uses Gated Convolution. Able to handle image with inconsistent missing regions.	FID 13.5
	2022	DAMGAN	Utilises a Dynamic Attention Map. Diminishes pixel level inconsistency to some extent.	SSIM 0.960
	2017	SRGAN	New Perceptual Loss Function. High image up-scaling Factor.	SSIM 0.6688
	2018	ESRGAN	Addresses issues with SRGAN by elevating perceptual quality by improving the architecture of SRGAN.	SSIM 0.741
Image Editing	2019	SC-FEGAN	Uses free form mask, sketch and colour as a means to edit images. Capable of generating high fidelity images using CNN.	SSIM 0.9618
	2020	In-Domain GAN Inversion for Image Editing	Based on GAN inversion principle, enables editing at pixel level with preserving semantic information.	FID 42.64
	2021	EditGAN	Higher precision and requires lesser annotated data. Model learns an arbitrary number of editing vector that can later be used.	FID 41.74
View Synthesis	2022	SURF-GAN	GAN based on NeRF, combines style with StyleGAN,	FID 4.72
General Purpose Image Generation	2021	ViT-GAN	Incorporated vision transformer in generator architecture. Overlapping regions for better training.	FID 6.66
	2021	VQ-GAN	Utilises the local spatial feature of CNN for obtaining high quality of image synthesis. Introduces the notion of Codebook.	FID 10.7
	2022	PoEGAN	Uses multi modal image synthesis. Enables better control over image features and contents.	FID 8.3
	2022	StyleGAN-XL	Capable of generating image of size 1024x1024. Editing and Inversion is possible.	FID 1.85
	2022	StyleSwin	Incorporates Swin Transformer in its Generator. Uses double attention mechanism and shifted windows method.	FID 3.25

*FID* Fretchet Inception Distance, *IS* Inception Score, *SSIM* Structural Similarity, *Class IOU* Intersection over Union

- **Inclusion and Exclusion Criteria:** For inclusion, authors of this paper relied heavily on application of GANs pertaining to image synthesis and computer vision. For Exclusion, especially biomedical imaging application of GAN has been excluded for the fact that it requires a separate review to cover intricate details from medical point of view. For the rest of the matter, to declutter the large number of publication, works which have less impact on real life use case and have minimal use case, have been left out for the simplicity of the review process.

## 5.2 Research challenges and opportunities

GAN continues to remain a central theme in Image Synthesis, as a result of which various incremental and ground breaking improvements are taking the Computer vision community by storm. The efficacy of the approach revolves around the fact that it incorporates various changes in its architecture, optimization objective and loss function for domain specific tasks. Fusion of concepts like attention mechanism, 3D vision and controllable scene generation may help to improve results. The following are some of the important research queries available in the recent literature:

1. **Need for Proper Dataset for Evaluating GAN:** As far as the matter of image generation is concerned, the authors in their work may tend to use dataset that may have class imbalance problem, i.e. dataset having an unequal numbers of sample data pertaining to a particular class or label, thereby making dataset skewed. And if such a dataset is to be used, then the GAN may suffer from biasness toward one particular class of image, which in turn affects the diversity and fidelity of the generated images.
2. **Need for Proper Evaluation Metrics for GANs:** A most commonly used metric for evaluating GAN are Inception Score(IS) and Fretchet Inception Distance(FID). However they have their limitations as FIDs score could give inconsistent result with respect to CelebA Dataset, from a judgement of human point of view. To sidestep this pitfalls we emphasis on the use of precision and recall based techniques to aid FID and IS score.
3. **Scope of Transfer Learning in GAN:** In our works, so far we investigated, we found lack of use of transfer learning, in majority of the work we reviewed so far, as a means to improve the performance of GAN. Since Transfer learning is an active point of research, it have substantial amount of impact multifarious fields. For generative models, computer vision and other related fields it addresses the the issue of training a model under limited amount of data, fine tuning the model reduces not only the training time but also saves intensive computations thus saving us from increase in carbon footprint.
4. **Customization of GAN model subjected to Limited Hardware Resource:** GANs discussed in this works are trained on high end GPUs(NVIDIA primarily), however not everyone has the access to this GPUs, thereby making them reliant on GPU rental services or turning to Transfer Learning paradigm. Therefore there is sufficient room for research for making generative models like GANs to run in such hardware constrained environment and how could model be improved or trained under such circumstances.
5. **GANs vs Diffusion models:** With the advent of diffusion based model [9], the trend in image synthesis has taken an extreme turn. It is reflected by the drop in the number of publications w.r.t GANs, unless the fallacies associated with the GANs are addressed, it would be cumbersome to maintain a steady pace in research direction of GANs, since diffusion based models are highly user driven guided synthesis process that has taken over the AI space by storm. And Focus has shifted to diffusion based models.

## 6 Conclusion and future works

In this paper, we have analysed the basic structures and working principle of GANs in the field of image generation. In addition, we have proposed a categorization of GAN based on different variants and their prime features. The research in this direction is getting advanced day by day because the diverse concepts are fused with GAN to intensify the new variants. It is not that hard to notice the impeccable elevation in the quality of image synthesis. Although

the future in this direction seems bright, we expect researchers around the world are tackling the issues faced by GAN such as mode collapse, proper customization of hardware resources etc. With this, we believe the anomalies tied with GANs are minimised, if not completely fixed.

**Data Availability** Data sharing is not applicable to this article as no datasets were generated or analysed during the current study.

## References

1. Arjovsky M, Bottou L (2017) Towards principled methods for training generative adversarial
2. Arjovsky M, Chintala S, Bottou L (2017) Wasserstein gan
3. Cha D, Kim D (2022) Dam-gan: Image inpainting using dynamic attention map based on fake texture detection
4. Chen J, Liu G, Chen X (2020) Animegan: A novel lightweight gan for photo animation. In: Li W, Wang H, Liu Y (eds) Li K. Artificial Intelligence Algorithms and Applications, Singapore, Springer Singapore, pp 242–256
5. Chen Y, Lai YK, Liu YJ (2018) Cartoongan: Generative adversarial networks for photo cartoonization. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition 9465–9474
6. Chen T, Zhai X, Ritter M, Lucic M, Houthby N (2019) Self-supervised gans via auxiliary rotation loss. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition pp 12154–12163
7. Choi Y, Choi M, Kim M, Ha JW, Kim S, Choo J (2017) Stargan: Unified generative adversarial networks for multi-domain image-to-image translation
8. Dash A, Gamboa JCB, Ahmed S, Liwicki M, Afzal MZ (2017) Tac-gan - text conditioned auxiliary classifier generative adversarial network
9. Dhariwal P, Nichol A (2021) Diffusion models beat gans on image synthesis
10. Dolhansky B, Ferrer CC (2017) Eye in-painting with exemplar generative adversarial networks
11. Esser P, Rombach R, Ommer B (2020) Taming transformers for high-resolution image synthesis
12. Frühstück A, Singh KK, Shechtman E, Mitra NJ, Wonka P, Lu, J (2022) Insetgan for full-body image generation
13. Goodfellow I (2017) Nips 2016 tutorial: Generative adversarial networks
14. Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A, Bengio, Y (2014) Generative adversarial networks
15. Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville A (2017) Improved training of wasserstein gans
16. Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville AC (2017) Improved training of wasserstein gans. Advances in Neural Information Processing Systems 30
17. Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S (2017) Gans trained by a two time-scale update rule converge to a local nash equilibrium
18. Huang X, Liu MY, Belongie S, Kautz J (2018) Multimodal unsupervised image-to-image translation
19. Huang X, Mallya A, Wang TC, Liu MY (2021) Multimodal conditional image synthesis with product-of-experts gans
20. Isola P, Zhu JY, Zhou T, Efros AA (2016) Image-to-image translation with conditional adversarial networks
21. Jin Y, Zhang J, Li M, Tian Y, Zhu H, Fang Z (2017) Towards the automatic anime characters creation with generative adversarial networks
22. Jolicœur-Martineau A (2018) The relativistic discriminator: a key element missing from standard gan
23. Jo Y, Park J (2019) Sc-fegan: Face editing generative adversarial network with user's sketch and color
24. Karras T, Aila T, Laine S, Lehtinen J (2017) Progressive growing of gans for improved quality, stability, and
25. Karras T, Aittala M, Laine S, Härkönen E, Hellsten J, Lehtinen J, Aila T (2021) Alias-free generative adversarial networks
26. Karras T, Laine S, Aittala M, Hellsten J, Lehtinen J, Aila T (2020) Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition pp 8110–8119
27. Kvalerov I, Czaja W, Chellappa R (2019) cgans with multi-hinge loss
28. Kingma DP, Welling M (2013) Auto-encoding variational bayes

29. Kodali N, Abernethy J, Hays J, Kira Z (2017) On convergence and stability of gans. arXiv preprint [arXiv:1705.07215](https://arxiv.org/abs/1705.07215)
30. Kwak JG, Li Y, Yoon D, Kim D, Han D, Ko H (2022) Injecting 3d perception of controllable nerf-gan into stylegan for editable portrait image synthesis. In: European Conference on Computer Vision, Springer 236–253
31. Ledig C, Theis L, Huszar F, Caballero J, Cunningham A, Acosta A, Aitken A, Tejani A, Totz J, Wang Z, Shi W (2016) Photo-realistic single image super-resolution using a generative adversarial network
32. Lee K, Chang H, Jiang L, Zhang H, Tu Z, Liu C (2021) Vitgan: Training gans with vision transformers
33. Lin TY, Maire M, Belongie S, Bourdev L, Girshick R, Hays J, Perona P, Ramanan D, Zitnick CL, Dollár P (2014) Microsoft coco: Common objects in context
34. Ling H, Kreis K, Li D, Kim SW, Torralba A, Fidler S (2021) Editgan: High-precision semantic image editing
35. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin transformer: Hierarchical vision transformer using shifted windows
36. Liu Z, Luo P, Qiu S, Wang X, Tang X (2016) Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
37. Li B, Zhu Y, Wang Y, Lin CW, Ghanem B, Shen L (2021) Anigan: Style-guided generative adversarial networks for unsupervised anime face generation
38. Lu Y, Lu J (2020) A universal approximation theorem of deep neural networks for expressing probability distributions
39. Mao X, Li Q, Xie H, Lau RYK, Wang Z, Smolley SP (2016) Least squares generative adversarial networks
40. Meng Q, Chen A, Luo H, Wu M, Su H, Xu L, He X, Yu J (2021) Gnerf: Gan-based neural radiance field without posed camera
41. Metz L, Poole B, Pfau D, Sohl-Dickstein J (2016) Unrolled generative adversarial networks. arXiv preprint [arXiv:1611.02163](https://arxiv.org/abs/1611.02163)
42. Miyato T, Kataoka T, Koyama M, Yoshida Y (2018) Spectral normalization for generative adversarial networks. arXiv preprint [arXiv:1802.05957](https://arxiv.org/abs/1802.05957)
43. Mroueh Y, Sercu T (2017) Fisher gan. Advances in Neural Information Processing Systems 30
44. Müller A (1997) Integral probability metrics and their generating classes of functions. Adv Appl Probab 29(2):429–443
45. Nichol A, Dhariwal P, Ramesh A, Shyam P, Mishkin P, McGrew B, Sutskever I, Chen M (2021) Glide: Towards photorealistic image generation and editing with text-guided diffusion models
46. Nowozin S, Cseke B, Tomioka R (2016) f-gan: Training generative neural samplers using variational divergence minimization. Advances in Neural Information Processing Systems 29
47. Odena A, Olah C, Shlens J (2016) Conditional image synthesis with auxiliary classifier gans
48. Oord AVD, Vinyals O, Kavukcuoglu K (2017) Neural discrete representation learning
49. Park SW, Kwon J (2019) Sphere generative adversarial network based on geometric moment matching. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 4287–4296
50. Petzka H, Fischer A, Lukovnicov D (2017) On the regularization of wasserstein gans. arXiv preprint [arXiv:1709.08894](https://arxiv.org/abs/1709.08894)
51. Pfau D, Vinyals O (2016) Connecting generative adversarial networks and actor-critic methods
52. Radford A, Metz L, Chintala S (2015) Unsupervised representation learning with deep convolutional generative adversarial networks
53. Ramesh A, Dhariwal P, Nichol A, Chu C, Chen M (2022) Hierarchical text-conditional image generation with clip latents
54. Reed S, Akata Z, Yan X, Logeswaran L, Schiele B, Lee H (2016) Generative adversarial text to image synthesis
55. Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B (2021) High-resolution image synthesis with latent diffusion models
56. Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation
57. Ruthotto L, Haber E (2021) An introduction to deep generative modeling
58. Saharia C, Chan W, Saxena S, Li L, Whang J, Denton E, Ghasemipour SKS, Ayan BK, Mahdavi SS, Lopes RG, Salimans T, Ho J, Fleet DJ, Norouzi M (2022) Photorealistic text-to-image diffusion models with deep language understanding
59. Salimans T, Goodfellow I, Zaremba W, Cheung V, Radford A, Chen X (2016) Improved techniques for training gans
60. Sauer A, Karras T, Laine S, Geiger A, Aila T (2023) Stylegan-t: Unlocking the power of gans for fast large-scale text-to-image synthesis

61. Sauer A, Schwarz K, Geiger A (2022) Stylegan-xl: Scaling stylegan to large diverse datasets
62. Shee CF, Uchida S (2022) Montagegan: Generation and assembly of multiple components by gans. In: 2022 26th International Conference on Pattern Recognition (ICPR), IEEE. pp 1478–1484
63. Shi W, Caballero J, Huszár F, Totz J, Aitken AP, Bishop R, Rueckert D, Wang Z (2016) Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network
64. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2015) Rethinking the inception architecture for computer vision
65. Taigman Y, Polyak A, Wolf L (2016) Unsupervised cross-domain image generation
66. Tao M, Tang H, Wu F, Jing XY, Bao BK, Xu C (2020) Df-gan: A simple and effective baseline for text-to-image synthesis
67. Wang X, Yu K, Wu S, Gu J, Liu Y, Dong C, Loy CC, Qiao Y, Tang X (2018) Esrgan: Enhanced super-resolution generative adversarial networks
68. Weng L (2019) From gan to wgan. arXiv preprint [arXiv:1904.08994](https://arxiv.org/abs/1904.08994)
69. White T (2016) Sampling generative networks
70. Wu H, Zheng S, Zhang J, Huang K (2017) Gp-gan: Towards realistic high-resolution image blending
71. Yuan Y, Guo Y (2020) A review on generative adversarial networks. In: 2020 5th International Conference on Information Science, Computer Technology and Transportation (ISCTT) 392–401
72. Yu J, Lin Z, Yang J, Shen X, Lu X, Huang TS (2018) Free-form image inpainting with gated convolution. arXiv preprint [arXiv:1806.03589](https://arxiv.org/abs/1806.03589)
73. Zhang B, Gu S, Zhang B, Bao J, Chen D, Wen F, Wang Y, Guo B (2021) Styleswin: Transformer-based gan for high-resolution image generation
74. Zhang H, Xu, T, Li H, Zhang S, Wang X, Huang X, Metaxas D (2016) Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks
75. Zhao J, Mathieu M, LeCun Y (2016) Energy-based generative adversarial network
76. Zhao J, Mathieu M, LeCun Y (2016) Energy-based generative adversarial network. arXiv preprint [arXiv:1609.03126](https://arxiv.org/abs/1609.03126)
77. Zhu JY, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks
78. Zhu Z, Huang T, Shi B, Yu M, Wang B, Bai X (2019) Progressive pose attention transfer for person image generation
79. Zhu J, Shen Y, Zhao D, Zhou B (2020) In-domain gan inversion for real image editing

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.