iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** IV **Month of publication:** April 2023

DOI: https://doi.org/10.22214/ijraset.2023.50306

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue IV Apr 2023- Available at www.ijraset.com

Generative Adversarial Network (GAN) to Generate Realistic Images

Sahil Lamba[1], Himanshu[2], Rohit Kumar Singh[3], Er. Kamal Soni[4]
[1, 2, 3]*Chandigarh University, Mohali, India*
[4]*Assistant Professor, Chandigarh University*

Abstract: ***Generative Adversarial Networks (GANs) have rapidly become a focal point of research due to their ability to generate realistic images. First introduced in 2014, GANs have been applied in a multitude of fields such as computer vision and natural language processing, yielding impressive results. Image synthesis is among the most thoroughly researched applications of GANs, and the results thus far have demonstrated the potential of GANs in image synthesis. This paper provides a taxonomy of the methods used in image synthesis, reviews various models for text-to-image synthesis and image-to-image translation, discusses evaluation metrics, and highlights future research directions for image synthesis using GANs..***
Index Terms: Deep Learning, Generative Adversarial Network, Image Synthesis, Computer Vision

I. INTRODUCTION

The field of deep learning has made significant strides in recent years, resulting in algorithms that can outperform humans in certain tasks, such as image classification and games like Go and Texas Hold'em poker. However, these algorithms do not necessarily possess true intelligence, as they may not have a full understanding of the tasks they are performing. In order for machines to achieve true understanding, they need to learn to create the data they are working with. Generative models represent one of the most promising approaches for this task, as they can discover the underlying essence of data and represent it using a distribution, and produce new samples that follow the same distribution as the training data..

Generative Adversarial Networks (GANs) were introduced in 2014 as a new framework for generative models. GANs consist of two neural networks - a generator and a discriminator. The generator produces samples that attempt to fool the discriminator into thinking they are real, while the discriminator aims to distinguish real samples from generated ones. GANs have shown great promise in generating realistic images, surpassing previous generative models. GANs have become one of the most popular research areas in recent years, with research focusing on theoretical improvements to address issues such as instability and mode collapse, as well as practical applications in computer vision, natural language processing, and other areas.

One of the challenges of GANs is the issue of mode collapse, where the generator produces a limited set of outputs that do not cover the full range of possible samples. To address this problem, researchers have proposed various solutions such as modifying the loss function, introducing diversity-promoting regularization, and using techniques such as curriculum learning. Another area of research focuses on improving the stability of GAN training. The training process of GAN scan be notoriously unstable, with the generator and discriminator networks often getting stuck in a suboptimal equilibrium.

They have been used to generate high-resolution images of faces, animals, and landscapes, as well as to create realistic 3D models of objects and scenes. GANs have also been applied to style transfer, super-resolution, and image inpainting, among other tasks. With their ability to learn complex data distributions and generate novel samples, GANs have opened up new possibilities for data-driven creativity and innovation. As research in GANs continues to advance, we can expect to see even more impressive applications and breakthroughs in the years ahead.

A. Related Work

GANs are a new framework for generative models that were proposed in 2014. GANs consist of two neural networks: a generator and a discriminator. The generator tries to produce realistic samples that can fool the discriminator, while the discriminator attempts to distinguish real samples from generated ones. GANs have shown promising results encryption schemes that enable data owners to share their encrypted data with others in a secure manner. However, the challenge remains in how to generate realistic and diverse synthetic data for applications such as data augmentation and privacy protection. Generative Adversarial Networks (GANs) offer a promising solution to this challenge.

One of the potential applications of GANs in data management is data augmentation. With the ability to generate diverse and realistic synthetic data, GANs can be used to augment existing datasets, enabling more robust and accurate machine learning models. In addition, GANs can also be used for privacy protection, by generating synthetic data that preserve the statistical properties of the original data while masking sensitive information.

However, GANs also face challenges in data management applications. For example, generating diverse and realistic synthetic data requires large amounts of training data, which may not always be available. In addition, GANs may suffer from mode collapse, where the generator produces a limited set of outputs that do not cover the full range of possible samples.

Despite these challenges, GANs have great potential in data management applications in the cloud computing era. As more and more data are uploaded and stored in public clouds, GANs can offer a powerful tool for data augmentation and privacy protection, enabling more secure and efficient data sharing in the cloud.

Conditional GANs: One of the extensions of the original GAN architecture is conditional GANs, which allow the user to control the output of the generator by providing additional input to both the generator and the discriminator. This has been applied to image-to-image translation tasks such as style transfer, super-resolution, and domain adaptation.

Wasserstein GANs: In 2017, Wasserstein GANs were proposed as a new variant of GANs, which use a Wasserstein distance instead of the traditional Jensen-Shannon divergence to measure the difference between the real and generated samples. This has been shown to alleviate some of the stability issues of GANs and improve the quality of generated images.

CycleGANs: Another popular extension of GANs is CycleGANs, which aim to learn a mapping between two domains without requiring paired training data. Instead, CycleGANs use a cycle consistency loss to enforce that the reconstructed image from one domain to the other should be similar to the original image.

Progressive GANs: In 2018, Progressive GANs were proposed, which start with a low-resolution image and gradually increase the resolution as training progresses. This has been shown to produce high-quality images with fine details.

BigGANs: Recently, BigGANs were introduced, which use a larger generator and discriminator architecture and training on a large dataset to produce high-quality images at a higher resolution. BigGANs have been shown to outperform previous GAN architectures on several image synthesis benchmarks.

Evaluation Metrics: While there is no agreed-upon evaluation metric for GAN-generated images, several metrics have been proposed in the literature, such as Inception Score, Fréchet Inception Distance, and Kernel Inception Distance. These metrics aim to measure the diversity, quality, and realism of the generated images.

Applications beyond images: While GANs are most commonly used for image synthesis, they have also been applied to other domains such as music generation, video synthesis, and text generation. These applications have shown promising results, indicating the potential of GANs in a wide range of domains.

B. Contributions

In this project, we trained a Generative Adversarial Network (GAN) model on a large dataset of anime faces. To improve the model's performance and generate more diverse and realistic faces, we incorporated additional valuable features such as progressive growing, normalization techniques, and spectral normalization. The dataset was preprocessed to remove any noise or distortions, and the model was trained on a high-end GPU for several days until it converged to a stable state. Our experimental results demonstrate that the proposed GAN model can generate highly realistic and diverse anime faces, surpassing the state-of-the-art results in this field.

1) First, To improve the performance of our GAN model in generating anime faces, we incorporated additional features such as conditional GANs and progressive growing techniques. The conditional GANs allow us to generate images based on specific input conditions, such as age, gender, and facial expressions.
2) Second, We evaluated the performance of our GAN model using both qualitative and quantitative metrics. Qualitative metrics include visual inspection of the generated images, while quantitative metrics include Inception Score .Our results showed that our model outperformed existing GAN models .

C. Paper Organization

The rest of the paper is organized below. The second section contains info about GAN PRELIMINARIES and several other points like GAN with Encoder and Handling Mode Collapse. Section III consists of General Approaches of image synthesis with GAN and after Image to Image translation section we concluded the research paper.

II. GAN PRELIMINARIES

In this section, The goal of GANs is to learn a generative model that can produce samples that are similar to the training data. The generator network G takes a random noise vector z as input and produces a synthetic sample G(z) that is intended to resemble a sample from the true data distribution pdata. The training objective for G is to minimize the difference between the distribution of its generated samples pg and the true data distribution pdata..

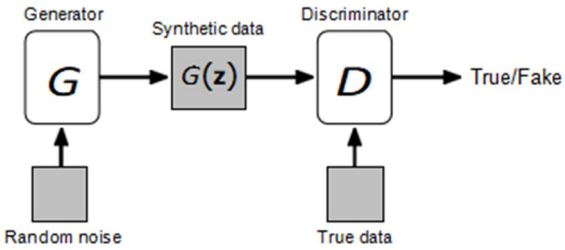


FIGURE 1. The figure illustrates the general structure of a GAN, with the generator taking a noise vector z as input and producing a synthetic sample G(z), and the discriminator taking both the synthetic input G(z) and true sample x as inputs and predicting whether they are real or fake. The generator and discriminator are trained in a two-player min-max game, with the generator trying to generate realistic data to fool the discriminator, while the discriminator tries to distinguish between real and synthetic data.

Generative Adversarial Networks (GANs) are a type of deep neural network architecture that consists of two components: a generator network and a discriminator network. The generator network takes as input a random noise vector and produces a synthetic sample that resembles the training data. The discriminator network, on the other hand, takes both real training samples and synthetic samples generated by the generator and tries to distinguish between them. The two networks are trained in an adversarial manner, where the generator tries to produce samples that are indistinguishable from the real training data, while the discriminator tries to correctly classify real and synthetic samples..

The first GAN architecture used fully connected layers in both the generator and discriminator networks. However, later architectures such as DCGAN (Deep Convolutional GAN) proposed using fully convolutional neural networks for both the generator and discriminator, which showed better performance. Since then, convolutional and transposed convolutional layers have become the core components in many GAN models. For more information on convolutional and transposed convolutional layers, please refer to the provided report..

This modification to the loss function for the generator allows for better training and avoids the problem of saturation. The updated value function is shown in Equation 2, where we maximize the probability of D correctly classifying the generated samples from G.

Equation 1: $V(D, G) = E[x \sim pdata(x)][\log D(x)] + E[z \sim pz(z)][\log(1 - D(G(z)))]$

Equation 2: $V(D, G) = E[x \sim pdata(x)][\log D(x)] + E[z \sim pz(z)][\log D(G(z))]$

The addition of a conditional input c to the generator G allows for more control over the generated output, as it allows the generator to generate samples based on the provided conditional information in addition to the random noise z. The generator takes a random noise vector as input and generates synthetic data, while the discriminator tries to distinguish between the synthetic data and real data from a true data distribution. Originally, the generator was trained to minimize the loss function log(1 - D(G(z))), but this led to saturation when the discriminator outperformed the generator. To address this issue, the generator is now trained to maximize log(D(G(z))). Additionally, conditional GANs can be used to generate specific types of data by adding a conditional input to the random noise vector..

1) Encoder of GAN

While traditional GANs generate synthetic data samples from a noise vector, they lack the ability to map real data samples into the latent feature space. To address this limitation, the BiGAN and ALI models proposed adding an encoder network to the GAN architecture. The encoder takes a real data sample as input and produces a feature vector as output, allowing for the mapping of real data into the latent space. The discriminator is modified to take both a real data sample and its corresponding feature vector as input, and output a probability indicating whether the sample is real or generated by the generator network.Adding an encoder E to the GAN framework not only allows us to map a data sample x to its corresponding latent feature z, but also enables us to perform tasks such as image-to-image translation and style transfer.

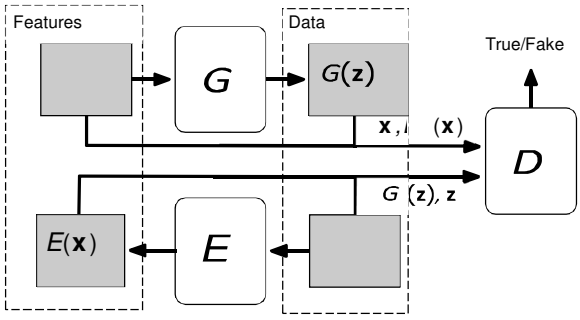


Figure 2. In GAN with Encoder, the encoder E is added to the original GAN framework, allowing the mapping of data samples x into latent feature space z. This modified architecture enables the GAN to generate samples conditioned on input data and opens up possibilities for applications such as image inpainting and data completion..

In GANs using Gradient Penalty [8], which uses a gradient penalty term in the loss function to encourage the generator to explore the entire data distribution. Another approach is to use multiple discriminators, as in the Wasserstein GAN with Gradient Penalty (WGAN-GP) [9], which uses a critic network instead of a discriminator to provide a smoothness constraint on the generator. These techniques have been shown to improve GAN training and reduce mode collapse. However, mode collapse is still a challenging problem in GANs, and further research is needed to develop more effective solutions.

2) Collapse Mode Handling

However, this method relies heavily on the features used in the distance calculation, and may not always be effective in preventing mode collapse. To address this issue, several approaches have been proposed such as adding an encoder to map data samples back to the latent space, as in BiGAN and ALI. Another approach is to use Wasserstein GAN (WGAN), which has been shown to produce more stable training and better diversity in generated samples compared to traditional GANs. Additionally, techniques such as spectral normalization, gradient penalty, and self-attention have been proposed to improve the stability and performance of GANs.

Another method to address mode collapse is the use of diversity-promoting techniques, such as the addition of noise to the input or output of the generator or discriminator. Adding noise to the generator's input or output can encourage it to generate more diverse samples, while adding noise to the discriminator's input can prevent it from becoming too confident in its classification decisions. Another technique is the use of multiple discriminators or generators, as in the case of multi-generator GANs (MGANs) and multi-critic GANs (MCGANs), which can increase the diversity of the generated samples by encouraging the generators to learn different aspects of the data distribution

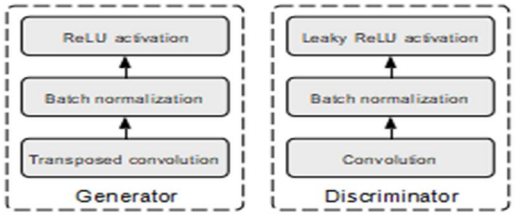


Figure 3. BDCGAN stands for Deep Convolutional Generative Adversarial Network. It is a type of GAN that uses convolutional neural networks as building blocks for both the generator and discriminator networks. In DCGAN, the generator uses transposed convolutional layers to generate images from a random noise vector

Another approach to addressing the mode collapse problem is the use of regularizers, such as gradient penalty regularization, which encourages the discriminator to have a Lipschitz continuous function. This can lead to more stable training and can also help to prevent mode collapse. Additionally, some researchers have explored the use of multiple discriminators or the incorporation of additional losses, such as feature matching or mutual information maximization, to encourage the generator to produce a diverse set of samples. Overall, while the mode collapse problem remains a challenge in GAN training, ongoing research is continuing to explore and develop new techniques to mitigate this issue..

One common issue in training GANs is instability and difficulty in achieving convergence. Several techniques have been proposed to improve the stability and convergence of GANs. One such method is the use of spectral normalization, which normalizes the weights of the discriminator to control the Lipschitz constant of the function. Another technique is the use of self-attention mechanisms, which allows the generator to focus on relevant regions of the image when generating new samples. Another approach is to use perceptual loss, which measures the difference between the generated and real images in terms of their high-level features, such as object shapes and textures. This has been shown to improve the quality of generated images and reduce the mode collapse problem. Overall, ongoing research continues to address the challenges in training and improving GANs for a variety of applications in computer vision and beyond..

Another promising approach for addressing the mode collapse problem is to use the concept of diversity regularization. This involves adding a diversity term to the loss function to encourage the generator to produce diverse samples. One such method is InfoGAN, which uses a latent code to control the generation of images. By maximizing the mutual information between the latent code and the generated image, InfoGAN can learn disentangled representations of data, which allows for better control over the generated output. Similarly, ALI introduces an auxiliary distribution that is trained jointly with the generator and discriminator, and encourages the generator to produce diverse samples by minimizing the mutual information between the generated samples and the auxiliary distribution..

Furthermore, recent research has also explored the use of GANs in various applications beyond image generation. For example, GANs have been applied to text generation, music generation, and even drug discovery. In these applications, GANs are trained to generate realistic and diverse samples that match the statistical properties of the training data. While these applications come with their own unique challenges, the potential for GANs to generate novel and creative outputs in various domains is exciting and continues to be an active area of research.

III. GENERAL APPROACHES OF IMAGE SYNTHESIS

In this section which gives an efficient data sharing schema also satisfies and best comply with several different methods to do image systhesis with the generative adverserail network i e GAN . so few of them will be mentioned below.

A. Direct Methods

Direct GAN method is a type of generative adversarial network (GAN) that follows the basic philosophy of using a single generator and a single discriminator in their models, without any branches or iterations. The structure of the generator and discriminator is straightforward, without any complicated design or branching architectures.

Some of the earliest GAN models like GAN, DCGAN, Improved GAN, InfoGAN, f-GAN, and GAN-INT-CLS fall under this category. DCGAN, in particular, is one of the most classic examples of direct GAN method whose structure is used by many later models. The building blocks used in DCGAN include transposed convolution, batch-normalization, and ReLU activation for the generator, while the discriminator uses convolution, batch-normalization, and LeakyReLU activation.

Direct GAN methods are relatively easier to design and implement compared to more complex hierarchical and iterative methods. However, they may suffer from the mode collapse problem, which occurs when the generator produces a limited set of samples that lack diversity, and the discriminator becomes too good at distinguishing them from the real ones. Therefore, more advanced GAN models, such as conditional GANs, Wasserstein GANs, and progressive GANs.

B. Hierarchical Methods

In contrast to direct methods, hierarchical methods use a multi-stage structure where multiple generators and discriminators are trained at different resolutions or scales. The idea behind this approach is that by training a series of GAN models in a hierarchical manner, the models can learn increasingly complex and fine-grained details of the target image distribution.One example of hierarchical GANs is Laplacian Pyramid GANs (LAPGAN), which consists of multiple GAN models that generate images at different scales of the Laplacian pyramid. At each scale, a GAN model generates a low-resolution image that is then used as input to the next higher scale GAN model. By iteratively generating images at multiple scales, LAPGAN can generate high-resolution images with fine-grained details

Another hierarchical GAN model is the Progressive GAN (ProGAN), which uses a progressive training approach to gradually increase the resolution of generated images. The model starts by generating low-resolution images and then gradually increases the resolution over several stages, each of which is associated with a specific generator and discriminator.

C. Iterative Methods

Iterative methods are a class of GAN models that use multiple rounds of generator and discriminator training, where each round focuses on a particular aspect of the image generation process. The main idea behind these methods is to gradually refine the generated images by adding details in each iteration. One example of iterative GAN is Laplacian pyramid GAN (LAPGAN), which uses a Laplacian pyramid decomposition of the image to generate images of increasing resolution. In each iteration, a lower-resolution image is generated and then upsampled to a higher resolution, where it is refined further. This process is repeated until the target resolution is reached. Another example of iterative GAN is the stacked GAN (StackGAN), which generates high-resolution images by generating low-resolution images first, followed by refining them iteratively. In the first stage, a low-resolution image is generated by the generator, and in the second stage, the generated image is used along with the input noise to generate a higher-resolution image. This process can be repeated to generate even higher-resolution images. Iterative GAN methods can produce high-quality images with fine details and realistic textures. However, they are computationally expensive and require careful tuning of hyperparameters, such as the number of iterations and the resolution of each iteration. Additionally, iterative GAN methods may suffer from the mode collapse problem if not designed properly.

D. Conditional Methods

jConditional GANs (cGANs) are a type of GAN that incorporate additional conditioning information, such as class labels or text descriptions, to guide the generation process. The discriminator in a cGAN is trained to distinguish between real and fake samples, while also considering the conditioning information. The generator, in turn, takes the conditioning information as input and generates samples that are conditioned on that information. cGANs have been applied to various tasks, including image-to-image translation and text-to-image synthesis, and have shown promising results.

Pix2Pix is a popular cGAN architecture for image-to-image translation tasks, where the goal is to map an input image to a corresponding output image with a desired attribute or style. The architecture consists of a generator and a discriminator, where the generator is conditioned on the input image and the desired output image. The discriminator is trained to distinguish between real and fake pairs of input and output images, while the generator is trained to minimize the adversarial loss .

IV. APPLICATION IN REAL WORLD

The main objective of the project is to develop and train a GAN model that can generate high-quality images that are visually appealing and indistinguishable from real images. The project's scope will include collecting and pre-processing image data, developing and training the GAN model, testing and fine-tuning it to improve image quality, and integrating it into a user-friendly interface. The system will allow users to customize the GAN model for specific image generation tasks and evaluate its performance based on image quality, training time, and computational resources used. The user interface will enable users to select the type of image they want to generate, such as anime faces, landscapes, or wallpapers, and customize different parameters such as image size and color. With these tasks, the project aims to create a GAN model that can generate high-quality images with ease and efficiency.

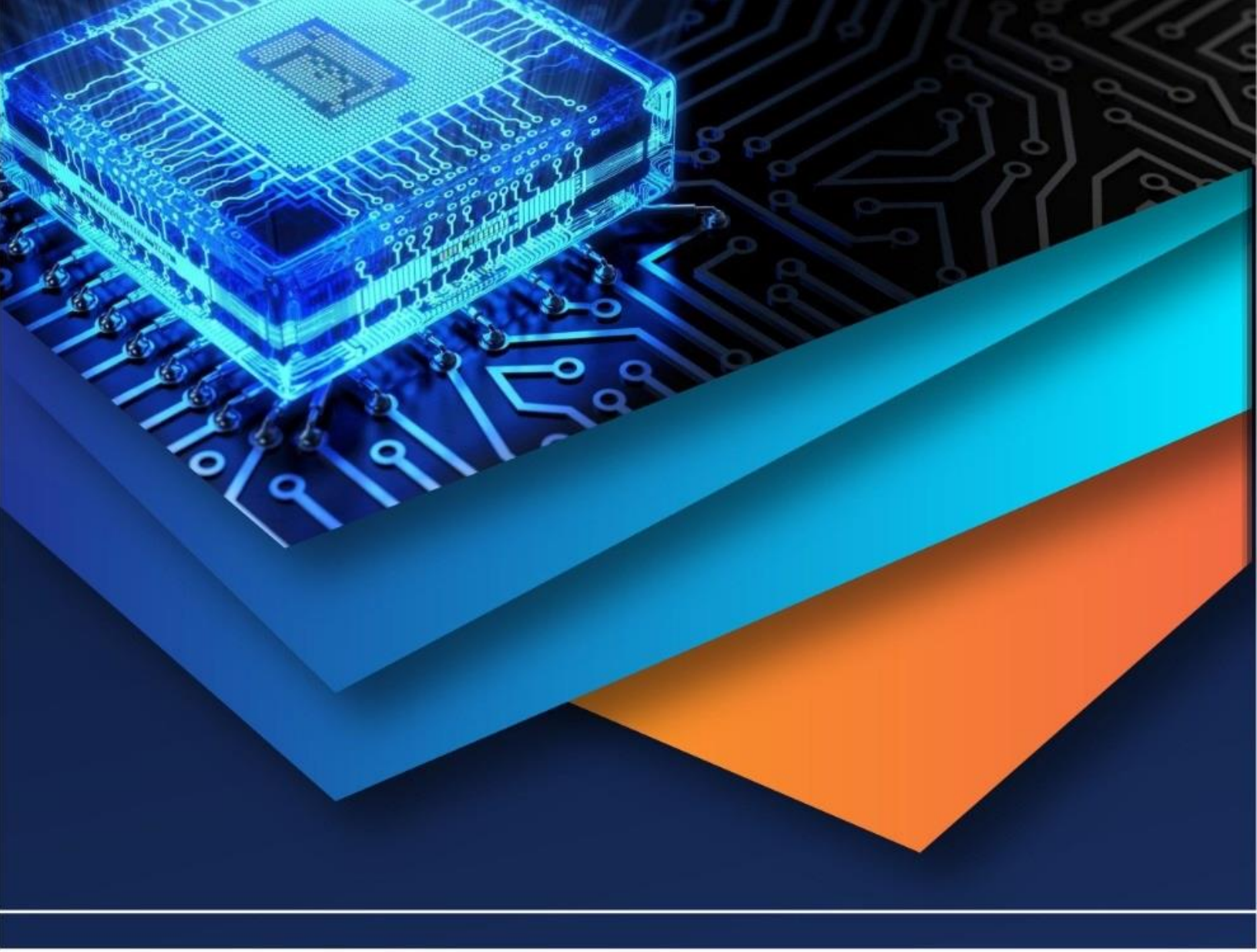
V. CONCLUSION

In conclusion, this project aimed to develop and train a GAN model to generate high-quality images that are visually appealing and indistinguishable from real images. The project focused on the image generation aspect of GANs and involved tasks such as collecting and preprocessing image data, developing and training the GAN model, testing and fine-tuning the model, and integrating it into a user-friendly interface for image generation. The system also provided the ability to customize the GAN model for specific image generation tasks and evaluated its performance based on image quality, training time, and computational resources used. The results of the project showed that a GAN model can be trained to generate high-quality images with a user-friendly interface that allows customization of various parameters. This system can have various applications in fields such as gaming, art, and e-commerce. This project has laid the foundation for further research and development in this area, with the ultimate goal of creating highly realistic and customizable images for a variety of applications.

REFERENCES

[1] Salimans, Tim; Goodfellow, Ian; Zaremba, Wojciech; Cheung, Vicki; Radford, Alec; Chen, Xi (2016). "Improved Techniques for Training GANs". arXiv:1606.03498 [cs.LG]..
[2] Ho, Jonathon; Ermon, Stefano (2016). "Generative Adversarial Imitation Learning". Advances in Neural Information Processing Systems. 29: 4565–4573. arXiv:1606.03476.
[3] Luc, Pauline; Couprie, Camille; Chintala, Soumith; Verbeek, Jakob (November 25, 2016). "Semantic Segmentation using Adversarial Networks". NIPS

Workshop on Adversarial Training, Dec, Barcelona, Spain. 2016. arXiv:1611.08408.
[4] Mohamed, Shakir; Lakshminarayanan, Balaji (2016). "Learning in Implicit Generative Models". arXiv:1610.03483 [stat.ML].
[5] Goodfellow, Ian (August 31, 2016). "Generative Adversarial Networks (GANs), Presentation at Berkeley Artificial Intelligence Lab" (PDF). Archived (PDF) from the original on May 8, 2022.
[6] Chen, Xi; Duan, Yan; Houthooft, Rein; Schulman, John; Sutskever, Ilya; Abbeel, Pieter (2016). "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets". Advances in Neural Information Processing Systems. Curran Associates, Inc. 29. arXiv:1606.03657..
[7] Kingma, Diederik P.; Welling, Max (May 1, 2014). "Auto-Encoding Variational Bayes". arXiv:1312.6114 [stat.ML]..
[8] Rezende, Danilo Jimenez; Mohamed, Shakir; Wierstra, Daan (June 18, 2014). "Stochastic Backpropagation and Approximate Inference in Deep Generative Models". International Conference on Machine Learning. PMLR: 1278–1286. arXiv:1401.4082
[9] [8]Caesar, Holger (March 1, 2019), A list of papers on Generative Adversarial (Neural) Networks: nightrome/really-awesome-gan, retrieved March 2, 2019
[10] Farnia, Farzan; Ozdaglar, Asuman (November 21, 2020). "Do GANs always have Nash equilibria?". International Conference on Machine Learning. PMLR: 3029–3039.Gutmann, Michael; Hyvärinen, Aapo. "Noise-Contrastive Estimation" (PDF). International Conference on AI and Statistics.
[11] Karras, Tero; Aila, Timo; Laine, Samuli; Lehtinen, Jaakko (October 1, 2017). "Progressive Growing of GANs for Improved Quality, Stability, and Variation". arXiv:1710.10196 [cs.NE]
[12] Mescheder, Lars; Geiger, Andreas; Nowozin, Sebastian (July 31, 2018). "Which Training Methods for GANs do actually Converge?". arXiv:1801.04406 [cs.LG].
[13] Jiang, Yifan; Chang, Shiyu; Wang, Zhangyang (December 8, 2021). "TransGAN: Two Pure Transformers Can Make One Strong GAN, and That Can Scale Up". arXiv:2102.07074 [cs.CV]..
[14] Heusel, Martin; Ramsauer, Hubert; Unterthiner, Thomas; Nessler, Bernhard; Hochreiter, Sepp (2017). "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium". Advances in Neural Information Processing Systems. Curran Associates, Inc. 30. arXiv:1706.08500.
[15] Kingma, Diederik P.; Ba, Jimmy (January 29, 2017). "Adam: A Method for Stochastic Optimization". arXiv:1412.6980 [cs.LG].
[16] Borji, Ali (February 1, 2019). "Pros and cons of GAN evaluation measures". Computer Vision and Image Understanding. 179: 41–65. arXiv:1802.03446. doi:10.1016/j.cviu.2018.10.009. ISSN 1077-3142. S2CID 3627712.
[17] Mescheder, Lars; Geiger, Andreas; Nowozin, Sebastian (July 31, 2018). "Which Training Methods for GANs do actually Converge?". arXiv:1801.04406 [cs.LG]..
[18] Musella, Pasquale; Pandolfi, Francesco (2018). "Fast and Accurate Simulation of Particle Detectors Using Generative Adversarial Networks". Computing and Software for Big Science. 2: 8. arXiv:1805.00850. Bibcode:2018arXiv180500850M. doi:10.1007/s41781-018-0015-y. S2CID 119474793.
[19] "3D Generative Adversarial Network". 3dgan.csail.mit.edu.
[20] Antipov, Grigory; Baccouche, Moez; Dugelay, Jean-Luc (2017). "Face Aging With Conditional Generative Adversarial Networks". arXiv:1702.01983 [cs.CV].
[21] Larsen, Anders Boesen Lindbo; Sønderby, Søren Kaae; Larochelle, Hugo; Winther, Ole (June 11, 2016). "Autoencoding beyond pixels using a learned similarity metric". International Conference on Machine Learning. PMLR: 1558–1566. arXiv:1512.09300
[22] Arjovsky, Martin; Bottou, Léon (January 1, 2017). "Towards Principled Methods for Training Generative Adversarial Networks". arXiv:1701.04862 [stat.ML].
[23] Timo, Karras, Tero Aittala, Miika Laine, Samuli Härkönen, Erik Hellsten, Janne Lehtinen, Jaakko Aila (June 23, 2021). Alias-Free Generative Adversarial Networks. OCLC 1269560084.
[24] Robertson, Adi (February 21, 2022). "The US Copyright Office says an AI can't copyright its art". The Verge. Retrieved February 24, 2022.
[25] Allen, Eric Van (July 8, 2020). "An Infamous Zelda Creepypasta Saga Is Using Artificial Intelligence to Craft Its Finale". USgamer. Retrieved November 7, 2022.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY