

Enhancing the Performance of Generative Adversarial Networks with Identity Blocks and Revised Loss Function to Improve Training Stability

Mohamed Fathallah^a, Mohamed Sakr^b and Sherif Eletriby^b

^aDepartment of Computer Science, Faculty of Computers and Information, Kafra El-Sheikh University, Kafra El-Sheikh 33511, Egypt

^bDepartment of Computer Science, Faculty of Computers and Information, Menoufia University, Menoufia 32511, Egypt

Mohammed.fci_0146@fci.kfs.edu.eg, mohamed.sakr@ci.menofia.edu.eg, sherif.eletriby@ci.menofia.edu.eg

Abstract

Generative adversarial networks (GANs) are a powerful deep learning model for synthesizing realistic images; however, they can be difficult to train and are prone to instability and mode collapse. This paper presents a modified deep learning model called Identity Generative Adversarial Network (IGAN) to address the challenges of training and instability faced by generative adversarial models in synthesizing realistic images. The IGAN model includes three modifications to improve the performance of DCGAN: a non-linear identity block to ease complex data fitting and reduce training time; a modified loss function with label smoothing to smooth the standard GAN loss function; and minibatch training to use other examples from the same minibatch as side information for better quality and variety of generated images. The effectiveness of IGAN was evaluated and compared with other state-of-the-art generative models using the inception score (IS) and Fréchet inception distance (FID) on CelebA and stacked MNIST datasets. The experiments demonstrated that IGAN outperformed the other models in terms of convergence speed, stability, and diversity of results. Specifically, in 200 epochs, IGAN achieved an IS of 13.6 and an FID of 46.2. Furthermore, the IGAN collapsed modes were compared with other generative models using a stacked MNIST dataset, showing the superiority of IGAN in producing all the modes while the other models failed to do so. These results demonstrate that the modifications implemented in IGAN can significantly enhance the performance of GANs in synthesizing realistic images, providing a more stable, high-quality, and diverse output.

Keywords: Generative adversarial network; deep learning; mode collapse; label smoothing; identity block.

1. Introduction

Generative adversarial networks, which were introduced by Goodfellow [1], are at the vanguard of efforts to generate high-fidelity and diversified images. In recent years, models learned directly from data have significantly advanced the state of generative image modeling, including in biomedical imaging [2, 3] and robotics [4, 5]. The GAN network contains two parts: generator (G) and discriminator (D) networks. The first network, G, maps a random noise vector to a data distribution to generate fake data. The second network, D, is used to differentiate between the real and the originated data by the generator network.

The zero-sum non-cooperative game is the foundation of GAN. The other loses if one wins, to put it briefly. Another name for a zero-sum game is minimax. Your actions aim to reduce those of your opponent, who strives to increase them. The GAN model converges in game theory when the Nash equilibrium is reached by the discriminator and the generator. Nash equilibrium happens when one player will not change its action regardless of what the opponent may do [6-9].

An explicit use of convolutional and convolutional-transpose layers in the discriminator and generator, respectively, makes a DCGAN [10] a straightforward extension of the GAN discussed above. The DCGAN training is dynamic and sensitive to almost all setup factors, including optimization, hyperparameters, and the architecture you choose for the model. Depending on the details of the application, the interaction between the generator and discriminator may be seen as either a cooperative or competitive game. For example, where D and G are collaborative to help improve performance, it is in data augmentation [11]. In this scenario, the generator's objective is to sample from a given data distribution by extracting important characteristics that can then be utilized to generate new data samples. The discriminator's objective is to make the generator match the real data distribution by providing useful feedback. In the context of healthcare applications, the GANs are employed to develop novel medications. In this scenario, the generator's objective is to come up with the ingredients for new medicines, either to improve existing treatments or to treat diseases that can't be cured. The discriminator's objective is to assist the generator in designing more effective drugs by analyzing their efficacy. Nevertheless, most of those architectures suffered in some way from model collapse while training [12, 13]. In another context, like generating passwords to crack bank account passwords, D and G are adversarial, as G plays the role of a hacker and D plays the role of a system firewall.

The mode collapse problem has been observed in GAN training. The mode collapse problem causes the generator to stick to some distributions' modes of the real data. These modes are the samples of data that the discriminator keeps recognizing as real distributions. The collapsed discriminator in this case sends back to the generator completely pointless weights. Because of these weights, the generator will keep generating the same data distributions (modes). As a result, only a small portion of the real data is sampled by the generator, and the GAN collapses to those distributions. The mode collapse is frequently sacrificed for more realistic individual samples. This trade-off of mode collapse for high-quality, realistic samples can lead to a biased model that produces a racial or gender-biased image. Another problem with giving up on preventing mode collapse to get a small, high-quality sample is that the model can be unstable and take a long time to train [14-17].

The contributions of the paper are as follows:

- 1) The paper introduces a promising version of the GAN architecture that can be expanded to generate a high-resolution image with high accuracy compared with other available versions of GANs.
- 2) The modified model can improve the fidelity and diversity of GAN's generated images.
- 3) The model is the first to use label smoothing in the loss function and minibatch training to reduce mode collapse and stabilize the training.
- 4) The model reduces the number of trainable parameters and therefore reduces the training time of GAN using identity blocks.

This paper is organized as follows: Section 2 provides an overview of related works, Section 3 introduces the proposed model of the IGAN, Section 4 introduces the experimental results and Section 5 introduces the conclusion and future work.

2. Related work

This section provides the latest research about solving the mode collapse problem during the GAN training process and also discusses the problem of improving the quality of the images with regard to the stability of the model. This section also discusses the limitations of the previous work. A major challenge in training generative adversarial networks is mode collapse. Several recent researches have incorporated novel target functions, network designs, or alternate training methods to relieve mode collapse. However, they frequently sacrifice image quality in order to attain their goals.

Duhyeon et al. [18] introduced Manifold-guided generative adversarial network (MGGAN) encourages the generator to learn the general modes of a data distribution by using a guidance network on an existing GAN architecture. A learnt manifold space, which is an effective representation of the coverage of the overall modes, is created from a picture by the guiding network. This guidance network's properties aid in penalizing mode imbalance. The experimental comparisons utilizing different baseline GANs revealed that MGGAN can be simply extended to existing GANs and address mode collapse without degrading the quality of the images.

Jinzheng et al. [19] presented a model instead of a single scalar, the idea of realness was expressed as a realness distribution. In order to give the generator more detailed instructions, the associated discriminator analyses the realness of an input sample from several perspectives. To modify the min-max game, they specifically included the realness measure objectives as mutation procedures that develop many persons. After that, they used a fresh least-squares fitness method to gauge how well the produced people performed. The measurement findings act as a crucial guide for us to keep only the efficient generators and get rid of the others [20].

StyleGAN [37] and ProGAN [38] are two popular GAN architectures for generating high-quality images. StyleGAN excels at producing highly realistic and diverse images with fine-grained control over the generated output, but has a larger computational complexity. ProGAN's progressive training approach enables faster convergence and more stable training but may produce higher FID scores, indicating that the generated images are less similar to real images. The IGAN, on the other hand, reduces the computational complexity using an identity block and a fully convolutional network design; it also solves the FID score problem in ProGAN uses a modified loss function to reduce the mode collapse, stabilize the training, and give fine control over diversity and fidelity.

3. Proposed work

This section presents the architecture of the IGAN as well as the improvements made to enhance and stabilize the GAN's training while boosting training speed. Firstly, the generator takes random noise, reshapes it into a 100-dimensional vector, and feeds this noise vector forward into the generator. Blocks 1 and 2 in the generator sample the noise vector to match the shape of the real image. Identity blocks in the generator help the activation bypass one or more layers and be added to the final activation. This helps mitigate the vanishing gradient problem, reduce the number of parameters, increase the training speed, and improve the accuracy of the model. The fake image from the generator is then passed to the discriminator. Secondly, the model trains the discriminator to maximize LD with both real and generated (fake) data, labeled 0.9 and 0.1, respectively, to reduce mode collapses. The discriminator uses a mini-batch layer in block 3, which enables it to use the other mini-batch examples as extra information. Lastly, the model trains the generator to minimize LG. The generator's loss LG is different from the standard generator's loss function. The IGAN generator loss LG produces non-saturating gradients, making training easier by using a decent rather than ascending optimizer. Fig. 1 shows the block diagram of the proposed model.

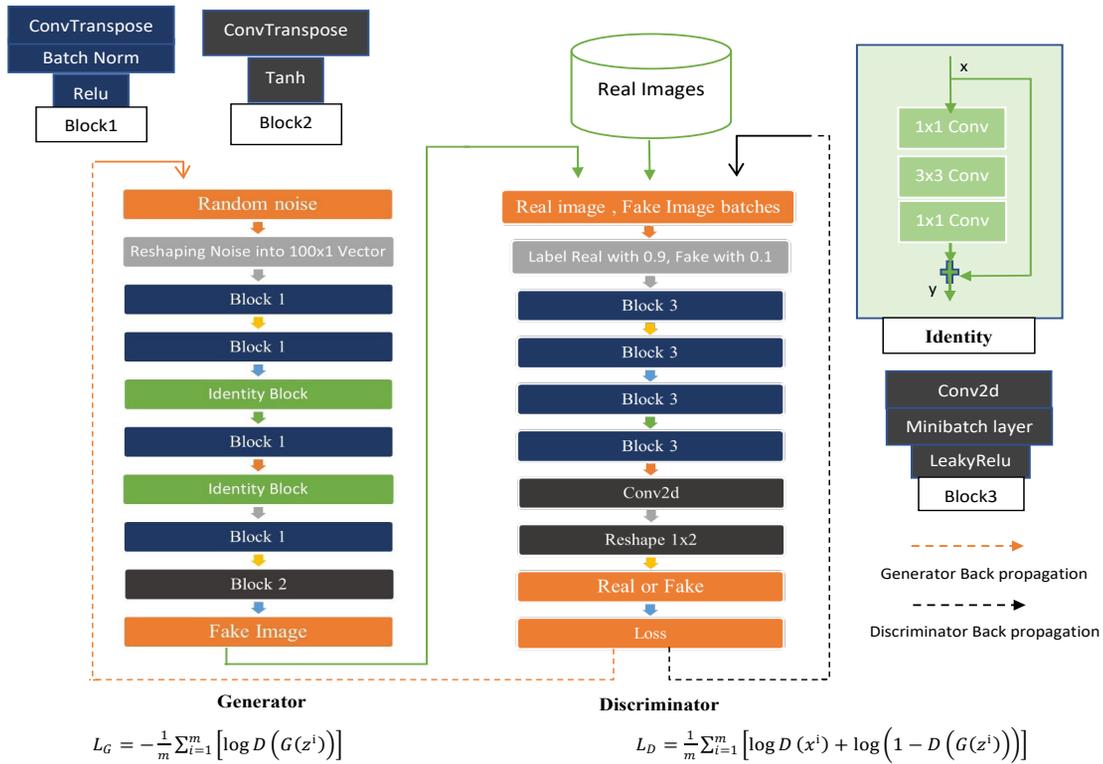


Fig. 1 Block diagram of IGAN architecture

3.1 The Proposed Modifications

The GANs models are designed to reach Nash equilibrium as shown in Formula (1). Where $z \in R^{d_z}$ is a latent variable sampled from distribution $p(z) \sim \mathcal{N}(0, I)$. The zero indicates that the mean of the normal distribution is zero and I represents the identity matrix, indicating that the variance is one in all dimensions.

$$\min_G \max_D E_{x \sim q_{\text{data}}(x)} [\log D(x)] + E_{z \sim p(z)} [\log(1 - D(G(z)))] \tag{1}$$

The modified model estimates the proximity of every pair of samples in a single minibatch. Then, the overall summary of a single data point is computed by adding its proximity to other samples in the same batch $o(x_i) = \sum_j c(x_i, y_j)$. Finally, $o(x_i)$ is explicitly added to the model. The discriminator is still required to output a single number for each example. This number indicates the likelihood of the example originating from the training data. The single output of the discriminator with minibatch training is allowed to use other examples in the minibatch as side information.

The mode collapse occurs when the discriminator returns 1 or 0 as the classification result for the real and fake images. The gradients at both ends will be close to 0 and 1, and the discriminator will be unable to provide useful feedback, which will lead to a vanishing gradient problem. Therefore, this paper modifies the loss function of GAN shown in formula (1) by adding label smoothing, where instead of providing 1 and 0 labels for real and fake data while training the discriminator, we used softened values of 0.9 and 0.1, respectively. The loss function used in this paper is expressed in Equations. (2), (3), (4), and (5).

$$L(G, D) = \min(L_G) + \max(L_D) \quad (2)$$

$$L_G = -\frac{1}{m} \sum_{i=1}^m \left[\log D \left(G(z^i) \right) \right] \quad (3)$$

$$L_D = \frac{1}{m} \sum_{i=1}^m \left[\log D(x^i) + \log \left(1 - D \left(G(z^i) \right) \right) \right] \quad (4)$$

$$D(x) = \begin{cases} 0.9 & \text{if } x \geq 0.5 \\ 0.1 & \text{if } x < 0.5 \end{cases} \quad (5)$$

The third modification includes using learning rate decay, which we found to be very effective with the Adam optimizer and improves the speed of training when used with minibatch learning. This modification improves the model's stability (oscillation in losses) as well as the FID and Inception Score when compared to DCGAN without this modification. Also, using identity block [22] in our architecture shows an increasing ability of the IGAN to model much more complex distributions, even in the early stages of learning, which led to an increase in the fidelity of images compared to the DCGAN.

Because GAN training is data-intensive, using datasets with fewer images per class, such as CIFAR100 or CIFAR10, makes it more difficult for the model to produce high-quality images. This has an impact on measuring mode collapse. As a result, this paper uses the CelebA [23] dataset. The properties and hyperparameters of the neural networks used to model the generator and discriminator are detailed in Table 1, Fig. 2, and Table 2, Fig. 3, respectively. Although the generator and discriminator designs can be tweaked through hyperparameter optimization, this is outside the scope of this work. The objective of this paper is to boost the performance of GANs with a fixed architecture. Thus, this paper introduces promising modifications to DCGAN and compares the new architecture modifications with DCGAN on the CelebA dataset.

Table 1. IGAN Generator architecture

Generator						
Layer	Conv.Type	In. Dim	Out. Dim	Kernel size	Strid	Padding
1	ConvTranspose2d	100	1024	4	1	0
			BatchNorm2d (1024, eps=1e-5 momentum=0.1)			
			Relu (Inplace=True)			
2	ConvTranspose2d	1024	512	4	2	1
			BatchNorm2d (512, eps=1e-5 momentum=0.1)			
			Relu (Inplace=True)			
3	ConvTranspose2d	512	256	4	2	1
			BatchNorm2d (256, eps=1e-5 momentum=0.1)			
			Relu (Inplace=True)			
4	ConvTranspose2d	256	256	1	1	0
			BatchNorm2d (256, eps=1e-5 momentum=0.1)			
			Relu (Inplace=True)			
5	ConvTranspose2d	256	128	4	2	1
			BatchNorm2d (128, eps=1e-5 momentum=0.1)			
			Relu (Inplace=True)			
6	ConvTranspose2d	128	3	4	2	1
			Tanh ()			

Fig. 2. IGAN Generator Architecture.

Table. 2. IGAN Discriminator architecture

Discriminator						
Layer	Conv.Type	In. Dim	Out. Dim	Kernel size	Strid	padding
1	Conv2d	3	64	4	2	1
BatchNorm2d (64, eps=1e-5 momentum=0.1) LeakyReLU(negative_slope=0.2)						
2	Conv2d	64	128	4	2	1
BatchNorm2d (128, eps=1e-5 momentum=0.1) LeakyReLU(negative_slope=0.2)						
3	Conv2d	128	256	4	2	1
BatchNorm2d (256, eps=1e-5 momentum=0.1) LeakyReLU(negative_slope=0.2)						
4	Conv2d	256	512	4	2	1
BatchNorm2d (512, eps=1e-5 momentum=0.1) LeakyReLU(negative_slope=0.2)						
5	Conv2d	512	1	4	2	0
Sigmoid Activation function with Label smoothing $0 \cong 0.1$ and $1 \cong 0.9$						

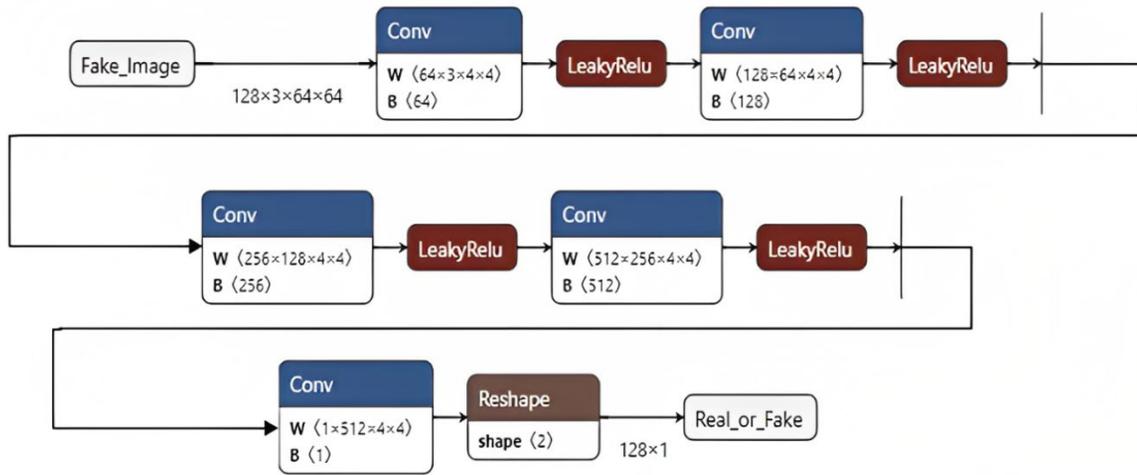


Fig. 3. IGAN Discriminator Architecture.

We named this new architecture IGAN since we used identity blocks to reduce the time needed to train it, as shown in layers 3 and 4 in Table 1. Using the identity block, we were able to reduce training time by reducing the number of trainable parameters without affecting performance. The number of parameters was reduced by 200K with each identity block we added. The improved architecture can then be utilized as a foundation for any cutting-edge GAN models that need to overcome the mode collapse problem. This is possible despite limited computing power.

The Adam optimizer is used for IGAN training in both the generator and discriminator networks, with a total of 200 epochs, a learning rate initialization of 2E-3, and learning rate decay. The comparison between DCGAN and IGAN is performed on a quantitative level by measuring the IS and FID. The IS takes a set of images and

outputs a floating-point value. The value indicates how realistic the output of a GAN is. The inception score is an automated alternative to having individuals rate image quality.

3.2 Dataset

This paper uses the CelebA dataset [23]. CelebA contains different face characteristics collections with over 200K celebrity photos, each with 40 attribute annotations. This dataset contains images with a wide range of pose variants and background clutter, including 10,177 identities, 202,599 face images, and 5 landmark locations, with 40 binary attribute annotations per image. Additionally, this paper measured the mode collapses of multiple models on the Stacked MNIST dataset following [24].

3.3 Hardware and Software Specifications

The numerical experiments are carried out with an Intel® Core i7-10750H CPU running at 2.60 GHz on the 10th and a NVIDIA GeForce RTX 3050Ti laptop. Table 3 shows the hardware and software specifications that have been used during the training process.

Table 3: Hardware and software Specification

Device	Description
Processors	Intel(R) Core (TM) i7-10750H CPU @ 2.60GHz
Random Access Memory (RAM)	16.0 GB
Graphical Processing Unit (GPU)	NVIDIA GeForce RTX 3050Ti
Space	Samsung SSD 970 EVO Plus 2TB
Programming language	Python

3.4 Evaluation metric

It is challenging to assess the effectiveness of generative models (e.g., GAN). The most effective method for assessing the visual quality of samples is to ask humans to determine the quality of the samples intuitively and reliably. This requires sufficient participants, and we did not have that. Also, we opt to use the inception score [25], a numerical assessment method, for quantitative evaluation calculated by Equation 6. x stands for one generated sample sampled from the generator, and y is the label predicted by the inception model [26].

$$I = \exp(E_{x \sim p_g} D_{KL}(p(y|x) || p(y))). \quad (6)$$

The idea behind this metric is that good models should generate high-diversity and high-fidelity images. Therefore, the divergence between the marginal distribution (real images) and the conditional distribution (generated images) should be large. This means the entropy of the conditional probability distribution is low, finding relevant objects and features, while the marginal probability distribution is high, finding a diverse set of features. We also adopt the widely used Fréchet Inception Distance (FID) [27].

$$FID = \|\mu_r - \mu_g\|^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}). \quad (7)$$

FID is based on measuring the image’s feature distance using a pre-trained Inception v3 network. This model is a pre-trained classifier that was trained on the ImageNet dataset. We used this model as a feature extractor. Those features are used to compare samples that are generated from the generator with the real data sample distributions, a feature-wise comparison. A lower FID shows that the generated images are closer to a realistic image distribution.

5. Experimental results and discussion

This section presents the results of the IGAN and also provides a comparison between the proposed model and other baseline GAN architectures like DCGAN, ProGAN, and StyleGAN.

5.1 Results after one and three epochs

Fig.4 provides a visual comparison of the images produced by DCGAN and IGAN in the first epoch. IGAN images show a more accurate representation of the facial structure with no mode collapse, as shown in Fig.4a when compared with Fig.4b. In terms of facial features, IGAN images are significantly easier to identify than those the DCGAN tries to sample. In the first stage of training, IGAN outputs outperform DCGAN outputs. Because the generator can now mimic more complex non-linear functions thanks to the identity block upgrade.

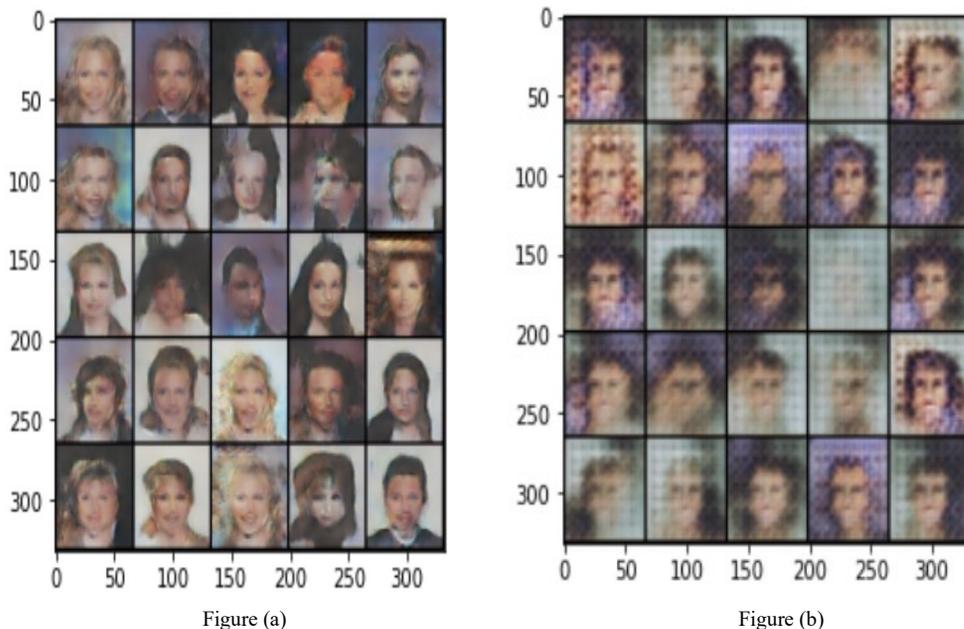


Fig.4. First epoch output Results. Figure (a) Proposed Model IGAN and Figure (b) DCGAN Model.

Fig.5 and Fig.6 show the visual output of the IGAN and DCGAN for the first three epochs, respectively. The mode collapse is obvious in the DCGAN output. IGAN output shows no sign of mode collapse at all. The output images from the IGAN show high diversity and high fidelity, even in the early stages of the training. Because mode collapse is difficult to detect in the advanced stages of training. We present results at an early stage of training.

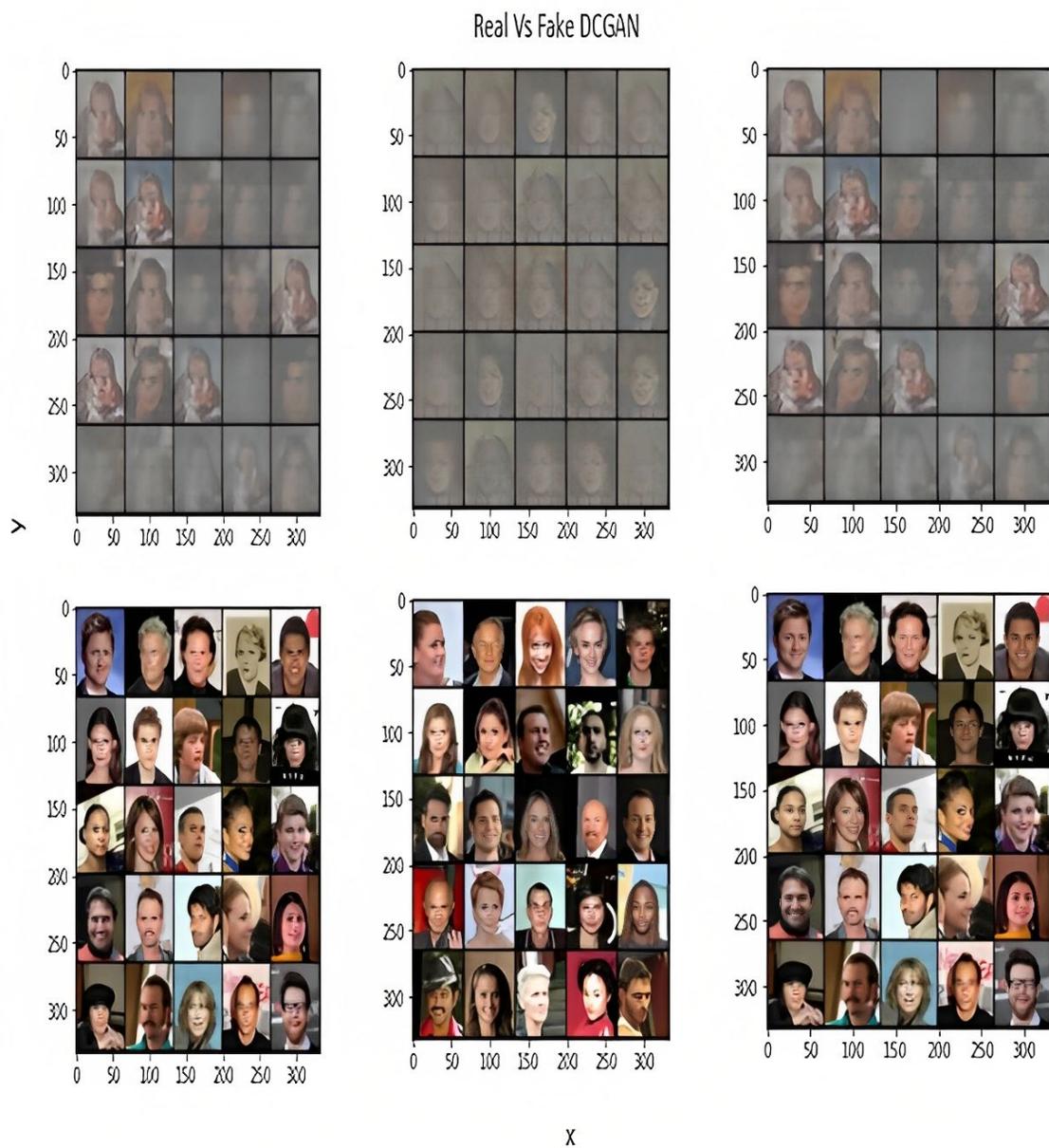


Fig. 5. visual output of the DCGAN for the first 3 epochs.



Fig. 6. visual output of the IGAN for the first 3 epochs.

5.2 Nash equilibrium results

Figure 7 (a) shows improved equilibrium between G and D during training in IGAN. Compared to DCGAN in Figure 7 (b), the G and D losses each have a different range of values. If the loss of the G is way higher than the loss of the D, this means the G will not be able to deceive the D. It will be hard to converge by then. In other words, if the discriminator loss is way lower than the generator loss, the discriminator will return 0 for all generated images, causing the generator to become stuck with low-quality images. Also, if the G loss is way lower than the discriminator, this means the D will return 1 for all the generated images. Hence, the generator will not improve. As a result, better GAN architecture must keep the D and G losses.

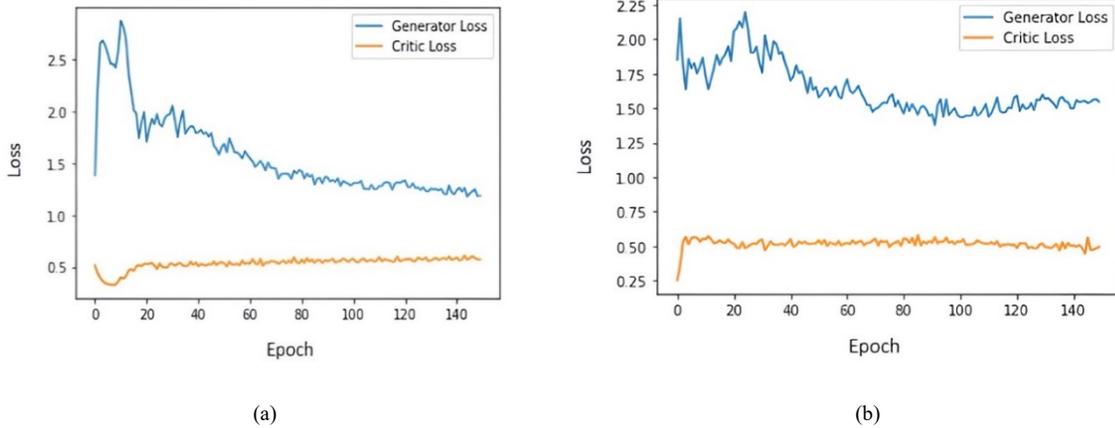


Fig. 7. Figures (a) and (b) show the generator and discriminator (critical) losses for the IGAN and DCGAN models, respectively.

5.3 Results of generated images

Table 4 presents the quantitative comparison of fidelity and diversity between the DCGAN, ProGAN, StyleGAN, and IGAN models during training by displaying both the FID and the IS score. The results show that IGAN architecture outperformed the other architectures after 200 epochs. The IGAN shows better capture of contours and distribution of the training dataset, maintaining stable training, reducing the time of training, and therefore reducing mode collapse. Figure 8 shows a graphical representation of the FID and IS scores of the IGAN model and the other state-of-the-art GAN models.

Table 4: Inception score and Fréchet results

Model	Inception Score \uparrow	FID Score \downarrow
DC-GANs	11.70	47.91
ProGAN	13.65	45.69
StyleGAN	13.71	44.23
IGAN	13.69	43.71

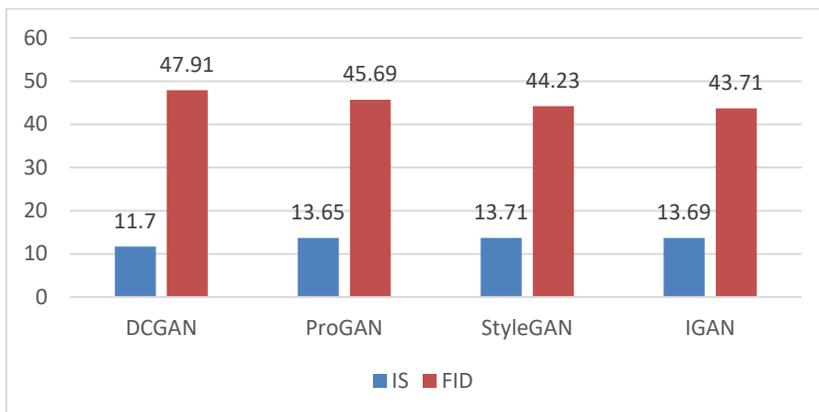


Fig. 8. Comparison chart of generated images between GAN architectures

5.4 Result of mitigation mode collapse

The result of comparing IGAN with other relative studies using the MNIST dataset is shown in Table 5. The MNIST dataset is consider simple, so this paper uses a set of transformation to create datasets with different level of complexity. The Transformation applied to MNIST dataset provide flexibility in generating datasets with varying degree of distribution complexity. Selecting $g(z)$, with progressively more complex transformation, can generate more demanding and complex synthetic datasets. The first five datasets are classified as levels 1-5, depending on their complexity. The levels are produced using simple transforms such as identity constant mapping (1 and z), small multi-layer perceptron (MLP), and well-conditioned linear transforms (A). This benchmark investigates mode collapse on several GANs models using two optimizers, SGD and ADAM, as shown in table 5. The results show that most of GAN model were robust to mode collapse until level 4 except for DCGAN. Only the IGAN and ProGAN made it to level 5 complexity.

Table 5: **T** means that all of the data modes is learned by the generator, whereas **F** indicates the generator suffer from mode collapse. The results are shown with the SGD (left) and ADAM (right) optimizers. MNIST results using the ADAM optimizer are given as a reference. The MNIST is a relatively simple dataset, with a complexity level between Levels 1 and 5.

$g(z)=$	1		A392x2		z		MLP		MLP, A392x2	
Model Name	Level 1		Level 2		Level 3		Level 4		Level 5	
DCGAN	F	T	F	T	F	F	F	F	F	F
ProGAN	T	T	T	T	T	T	T	T	T	T
StyleGAN	T	T	T	T	T	T	T	T	F	F
IGAN	T	T								

5.5 Distribution of the generated data vs real data

Fig. 8 illustrates the visualization of the pairwise multivariate distributions of the inception features for three different images. The images are a random selection of real vs. fake IGAN output. The features, which are nearly identical, are shown along the diagonal. This shows how the generated image feature distribution from IGAN is almost identical to the real data distribution.

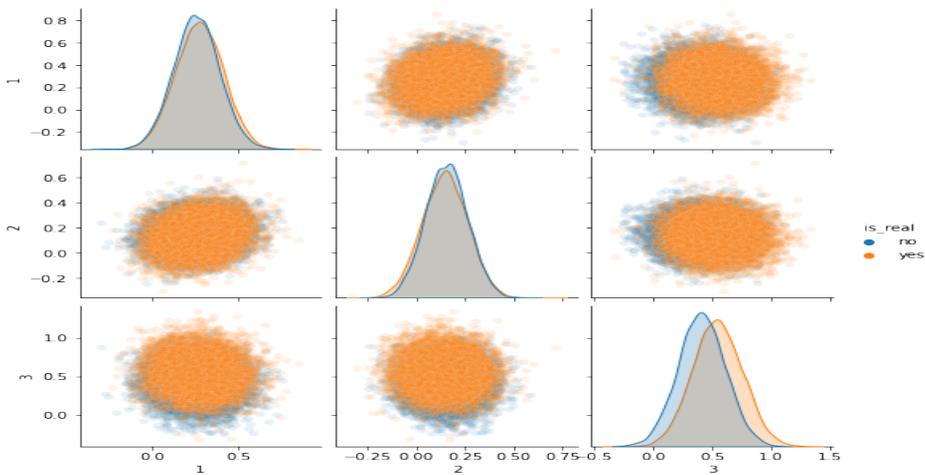


Fig. 9. pairwise multivariate distributions of the inception features generated from IGAN

6. Conclusion

This paper presents a novel, modified version of the GAN training architecture that includes an identity block. The model has been modified in three different ways: Modifications to the loss function and hyperparameters by using label smoothing and mini-batch training are employed to stabilize the model. The model has achieved better results in stability, generated image quality, and diversity compared to other state-of-the-art GAN models. A higher IS score and a lower FID score validate our findings. Although the model has performed well in terms of avoiding mode collapse, it still has some limitations when dealing with higher-resolution images. In the future, we plan to use IGAN to reduce mode collapse in settings for high-resolution image generation.

References

- [1] I. Goodfellow, J. Pouget, M. Mirza, B. Xu, D. Warde *et al.*, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, no. 1, pp. 2672-2680, 2014.
- [2] H. Meng and F. Guo, "Image classification and generation based on GAN model," *in.proc. International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI 2021)*, Taiyuan, China, pp. 180-183, 2021.
- [3] M. Abedi, L. Hempel, S. Sadeghi and T. Kirsten, "GAN-Based approaches for generating structured data in the medical domain," *Applied Sciences*, vol. 12, no. 14, pp. 1–16, 2022.
- [4] A. Seddik, M. Salah, G. Behery, A. El-harby, A. I. Ebada *et al.*, "A new generative mathematical model for coverless steganography system based on image generation," *Computers, Materials & Continua*, vol. 74, no.3, pp. 5087–5103, 2023.
- [5] J. Bian, X. Hui, S. Sun, X. Zhao and M. Tan, "A novel and efficient CVAE-GAN-based approach with informative manifold for semi-supervised anomaly detection," *IEEE Access*, vol. 7, pp. 88903–88916, 2019.
- [6] M. Ye and G. Hu, "Game design and analysis for price based demand response: an aggregate game approach", *IEEE Transactions on Cybernetics*, vol. 47, no.1, pp. 720-730, 2017.
- [7] P. Frihauf, M. Krstic and T. Basar, "Nash equilibrium seeking in non-cooperative games", *IEEE Transactions on Automatic Control*, vol. 57, no.2, pp. 1192-1207, 2012.
- [8] F. Salehisadaghiani and L. Pavel, "Distributed nash equilibrium seeking: a gossip-based algorithm", *Automatica*, vol. 72, no.2, pp. 209-216, 2016.
- [9] J. Koshal, A. Nedic and U. Shanbhag, "Distributed algorithms for aggregative games on graphs", *Operations Research*, vol. 64, no. 6, pp. 680-704, 2016.
- [10] A. Patil and A. Venkatesh, "DCGAN: deep convolutional GAN with attention module for remote view Classification," *in.proc. International Conference on Forensics, Analytics, Big Data, Security (FABS2022)*, Bengaluru, India, pp. 1-10, 2022.
- [11] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning", *Journal of Big Data*, vol. 6, no. 1, 2019.
- [12] Y. Du, W. Zhang, J. Wang and H. Wu, "DCGAN based data generation for process monitoring," *in.proc. IEEE 8th Data Driven Control and Learning Systems Conference (DDCLS 2019)*, Dali, China, pp. 410-415, 2019.
- [13] X. Ding and Q. He, "Energy fluctuated multiscale feature learning with deep ConvNet for intelligent spindle bearing fault diagnosis", *IEEE Transactions on Instrumentation & Measurement*, vol. 66, no. 8, pp. 1926-1935, 2022.
- [14] Y. Kossale, M. Airaj and A. Darouichi, "Mode collapse in generative adversarial networks: an overview," *in.proc. 8th International Conference on Optimization and Applications (ICOA 2022)*, Genoa, Italy, pp. 1-6, 2022.
- [15] T. Li, S. Zhang and J. Xia, "Quantum generative adversarial network: a survey," *Computers, Materials & Continua*, vol. 64, no.1, pp. 401–438, 2020.
- [16] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin and A. Courville, "Improved training of wasserstein gans", *Advances in Neural Information Processing Systems*, vol. 30, no. 1, pp. 5767-5777, 2017.

- [17] A. Van Den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, A. Graves et al., "Conditional image generation with pixelcnn decoders", *Advances in neural information processing systems*, vol. 29, no. 1, pp. 195-208, 2022.
- [18] D. Bang and H. Shim, "MGGAN: solving mode collapse using manifold-guided training," *in. proc. International Conference on Computer Vision Workshops (ICCVW2021)*, Montreal, BC, Canada, pp. 2347-2356, 2021.
- [19] J. Mu, Y. Zhou, S. Cao, Y. Zhang and Z. Liu, "Enhanced evolutionary generative adversarial networks," *Journal of Soft Computing*, vol.22, no.2, pp. 7534-7539, 2021.
- [20] H. Lee, M. Ra, and W.-Y. Kim, "Nighttime data augmentation using GAN for improving blind-spot detection," *IEEE Access*, vol. 8, pp. 48049–48059, 2020.
- [21] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401 – 4410, 2019.
- [22] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.
- [23] M. Lin, Q. Chen, and S. Yan, "Network in network," 2014.
- [24] Q. Hoang, T. D. Nguyen, T. Le, and D. Phung, "Mgan: Training generative adversarial nets with multiple generators," 2018.
- [25] A. Srivastava, L. Valkov, C. Russell, M. U. Gutmann, and C. Sutton, "Veegan: Reducing mode collapse in gans using implicit variational learning," vol. 2017-December, 2017.
- [26] S. T. Barratt and R. Sharma, "A note on the inception score," *ArXiv*, vol. abs/1801.01973, 2018.
- [27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," vol. 2016-December, 2016.