 Iraqi Journal of Science, 2026, Vol. 67, No. 7, pp: 4009 -4030
DOI: 10.24996/ijs.2026.67.7.32

ISSN: 0067-2904

Addressing Dataset Imbalance with Modified Generative Adversarial Networks

Roaa Razaq[1], Ebtesam N. AlShemmary[*2], Zhentai Lu[3]
[1]Department of Computer Science, Faculty of Computer Science and Mathematics, University of Kufa, Kufa, Iraq.
[2]IT Research and Development Center, University of Kufa, Kufa, Iraq.
[3]School of Biomedical Engineering, Southern Medical University, Guangzhou 510515, China.

Received: 15/ 11/2024 Accepted: 14/ 7/2025 Published: 30/ 7/2026

Abstract
Class imbalance in datasets remains a major problem in machine learning; most algorithms produce a biased model that generalizes poorly on the minority classes. This becomes an important factor in applications where minority Class samples are very important, such as medical diagnosis, accounting fraud detection, financial analysis, and so on. This paper proposes a new method to overcome the problem of class imbalance by using Generative Adversarial Networks (GANs). Specifically, the proposed method aims at obtaining a set of synthetic instances for the minority classes, which helps to balance the training set and improve classifiers' performance. It introduces a modified Generative Adversarial Network (MGAN), which generates synthetic images of the minority class (128 × 128 × 3 pixels), thereby enhancing classifier resilience. Empirical evaluation indicates that MGAN surpasses conventional data balancing methods in medical imaging contexts. The effectiveness of this image filtering approach regarding lesion detection is experimentally confirmed when using two medical datasets of endoscopic and pathological images. The outcomes represent a high performance of the variant classifiers, which outperforms the conventional approaches when using the altered MGAN.

Keywords: Class imbalance, Classifier performance, Generative Adversarial Networks (GANs), High-resolution images, Synthetic data generation.

معالجة اختلال التوازن في مجموعات البيانات باستخدام الشبكات التنافسية التوليدية

رؤى رزاق[1] , ابتسام نجم الشمري[*2] , زينتاي لو[3]
[1]قسم علوم الحاسوب، كلية علوم الحاسوب والرياضيات، جامعة الكوفة، الكوفة، العراق.
[*2]مركز البحث والتطوير لتكنولوجيا المعلومات، جامعة الكوفة، الكوفة، العراق.
[3]كلية الهندسة الطبية الحيوية، جامعة جنوب الصين الطبية، قوانغتشو 510515، الصين.

*Email: dr.alshemmary@uokufa.edu.iq

الخلاصة

لا يزال اختلال توازن الفئات في مجموعات البيانات يُمثل مشكلة رئيسية في التعلم الآلي، حيث تُنتج معظم الخوارزميات نموذجًا متحيزًا يُعمم بشكل سيئ على فئات الأقلية. ويُصبح هذا عاملًا مهمًا في التطبيقات التي تُعدّ فيها عينات فئات الأقلية بالغة الأهمية، مثل التشخيص الطبي، وكشف الاحتيال المحاسبي، والتحليل المالي، وما إلى ذلك. تقترح هذه الورقة البحثية طريقة جديدة للتغلب على مشكلة اختلال توازن الفئات باستخدام الشبكات التوليدية التنافسية (GANs). وتهدف الطريقة المقترحة تحديدًا إلى الحصول على مجموعة من الحالات الاصطناعية لفئات الأقلية، مما يُساعد على موازنة مجموعة التدريب وتحسين أداء المُصنِّفين. وتُقدِّم هذه الطريقة شبكة توليدية تنافسية مُعدّلة (MGAN)، تُولِّد صورًا اصطناعية لفئة الأقلية (128 × 128 × 3 بكسل)، مما يُعزز مرونة المُصنِّفين. ويُشير التقييم التجريبي إلى أن MGAN تتفوق على طرق موازنة البيانات التقليدية في سياقات التصوير الطبي. وقد تم تأكيد فعالية أسلوب تصفية الصور هذا فيما يتعلق بكشف الآفات تجريبيًا عند استخدام مجموعتي بيانات طبية من الصور التنظيرية والمرضية. تمثل النتائج أداءً عاليًا لمصنفات المتغيرات، والتي تتفوق على الأساليب التقليدية عند استخدام MGAN المعدل.

1. Introduction

One of the challenges associated with the development of machine learning (ML) models is the presence of an excessive number of samples in one class while another class has very few samples [1]. This classification imbalance can be described as the disproportionate allocation of a large number of samples to one class compared to another [2]. Such inequality can be observed in both binary and multiclass classification scenarios [3]. The class with a comparatively small number of samples is referred to as the minor class, while the class with the largest number of samples is known as the major class [4]. In binary classification problems, the term is specifically used to denote the class with fewer samples. This issue has become a significant concern across various fields, including biology, healthcare, finance, telecommunications, and disease diagnosis [3]. Consequently, it is regarded as one of the most critical problems in the data mining process [5]. Generally, ML algorithms are coded in such a way that they run perfectly well when a data set is balanced, while they do not perform well on an unbalanced data set [6]. Therefore, it is possible to indicate that there are certain conditions where, in general, it is impossible to develop a solution using conventional approaches to machine learning algorithms with the primary goal of identifying positive samples, or the so-called rare samples; for example, credit card fraud detection and identification of malignant tumor cells [7]. For instance, Caruana et al. (2015)'s study on the model aimed at identifying which pneumonia patients needed hospitalization, and which could be discharged [8]. However, their proposed approach produced inaccurate risk predictions for patients with asthma or chest pain for estimating a lower basic risk of death. A large number of ML algorithmic techniques have been put forward, and their accuracy still tends to favor the major class. For instance, assume there is an imbalanced dataset that includes 10924 non-cancerous (low density) and 260 malignant (high density) cell image data. When using traditional ML algorithms, there is a high probability of their testing data being fully classified into the major class and ranging between 0% and 10% of the minor class; therefore, 234 of the minor class will be classified into the major class [9]. Consequently, 234 people suffering from cancer would be diagnosed as having other diseases. It is even costlier in medical treatment, and a mistake in diagnosing a malignant cell has serious health implications and can lead to the death of the patient [10].

Imbalanced datasets present challenges such as ensuring the quality and diversity of generated samples, maintaining training stability, and avoiding issues like mode collapse. To address these challenges, a new data augmentation method utilizing Generative Adversarial Networks (GANs) is proposed to generate diverse and realistic samples for the minority class.

This approach aims to balance the dataset without altering the feature space learning, thereby enhancing the model's generalization ability for both in-domain and out-of-domain test data. There are several aspects of computer vision for which GANs have been widely applied. Here we wanted to demonstrate one of the big benefits of GANs, namely, the completion or generation of natural images from random noise. This dynamic characteristic of GAN makes them particularly attractive because they can be applied to almost any data representation type, including temporal data, audio signals, and image data.

It has been quite common to observe that traditional GANs produce images of a maximum resolution of 28 x 28 pixels or 64 x 64 pixels, depending on the dataset. The 28x28 resolution is more common for simpler data sets like MNIST, which is the data of the grayscale images of handwritten digits [11], whereas the 64 x 64 resolution is used by more complex data sets; for instance, CIFAR-10 [12]. It becomes very challenging to extend GANs and generate higher resolution images that can be 128 x 128 pixels and above; there is a need to employ complex model structures and methods. The traditional GAN architecture of generator and discriminator gives grid of images with size 8×8 images and default generate small images size like (64×64×3) pixels or (28×28×3) as documented in the PyTorch library (an open-source deep learning library developed primarily by Meta AI. It provides tools for building and training machine learning and deep learning models, especially neural networks.). The key contributions of this paper are changing the size of the generated images from this size, the structure of the generator and the discriminator must be changed to become synthetic images at (128×128×3) pixels, to be able to be used in the proposed diagnostic system. The proposed GAN architecture comprises generator and discriminator networks tailored to generate (128×128×3) pixel images. The architecture modifications are designed to ensure compatibility with applications requiring high-resolution images; this modification of GAN can be called MGAN.

The paper is organized as follows: Section 2 discusses previous work in the context of class imbalance and presents methods that are still in use and some of the novel methods based on AI techniques. In section 3, GANs and the latest research findings on GANs are described with reference to the theory of GANs. Section 4 describes the method, including the modified GAN architecture used in this research. Section 5 contains the implementation of the experiments. Section 6 is the conclusion and discussion of the research, while Section 7includes the recommendations for future research.

2. Related Works

Lee and Park [13] introduced a network-based intrusion detection system aimed at addressing class imbalance issues commonly encountered in intrusion datasets. To tackle this, they proposed the use of Generative Adversarial Networks (GANs) for both binary and multiclass classification tasks. GANs, which fall under the domain of unsupervised deep learning, are capable of learning the underlying distribution of input data and generating new samples that closely mimic this distribution. This characteristic makes GANs particularly effective in handling challenges such as class overlap and noisy data, as they can resample marginal classes to improve class representation.

Nie and Trullo [14] showed that the applied method using GAN architecture indeed creates realistic medical images that improve the dataset's variety and utility for multiple clinical tasks.

Jiang et al. [15] employed GAN-based features to overcome issues of class imbalance in time series information. Sharma et al. [16] outlined a method to synthesize samples with an

enhanced Gaussian distribution by employing GANs—a problem area typically challenging when utilizing other imbalance techniques. Their study presents two GAN-based approaches that are 10% more accurate than other methods and generate data that visually resembles real samples.
In another study, Loey et al. [17] used a classical data augmentation approach together with conditional GANs on a COVID-19 CT dataset. They realized that COVID-19 categorization was greatly improved.

Similarly, Xu et al. [18] employed a conditional GAN for the super-resolution of chest X-ray images. Conditional GANs have also been applied in cross-modality image translation. A brief review of different GAN-based models with their respective limitations, advantages, findings, and methods is summarized in **Error! Reference source not found.** for medical image data enhancement, anomaly detection, and classification.

Additionally, when compared to state-of-the-art techniques, Sampath et al. [19] discussed various methods of GANs to handle disparities in computer vision. They also discussed how best to improve image quality and sample diversity with improvements to GANs that better address minority classification. Sauber-Cole and Khoshgoftaar [20] focused on the use of GANs in tabular data, addressing concerns such as stability and sample diversity. Chakraborty et al. [21] offered a decade-long overview of GANs, including architectural evolution, training techniques, and application domains.

Table 1: includes a summary of GAN-Based Models Utilized in Medical Image Data Applications.

Author/Year	Method	Findings	Advantage	Limitation
Joohwa Lee, et al. [13]	Used Random Forest for classification and resampled data.	GAN model improved classification performance on imbalanced data, achieving 99.88%.	GAN-RF model improves classification performance over a single RF model	Imbalanced data can lead to overfitting and data loss SMOTE can create overlapping classes and noise in the data
Yinqiu Feng et al. [14].	Sparse coding, patch-based, and atlas-based methods are used for feature matching	The proposed GAN method generates realistic medical images with commendable generalization prowess. Synthesized images maintain structural and textural features similar to real images.	Generates synthetic images from limited real data, expanding the dataset efficiently. Enhances the richness and diversity of medical image data through synthesis. Robust performance in generating realistic images closely resembling authentic ones	Limited medical image data availability due to cost and privacy concerns Traditional methods struggle with capturing complex features, authenticity, and diversity.
Wenqian Jiang, et al. [15]	Proposed architecture detects abnormal samples with an encoder-decoder-encoder generator	GAN-based approach achieves 100% accuracy in anomaly detection on datasets.	Outputs higher anomaly scores for fault detection	Insufficient abnormal samples impact the machine learning algorithm's accuracy.
Anuraganand Sharma et al. [16]	SMOTified-GAN combines SMOTE and GAN for improved minority class samples.	Improves F1 score by up to 9%. Reasonable time complexity of O (N 2d2T) for efficient training	Addresses class imbalance issues effectively with reasonable time complexity	GAN was not designed for oversampling imbalanced classes.
Mohamed Loey, et al. [17].	In Preprocessing using classical data augmentation and Conditional Generative Adversarial Nets. In AlexNet, VGGNet16, VGGNet19, GoogleNet, and ResNet50 were the classifiers used.	ResNet50 is best for COVID-19 detection with 82.91% testing accuracy.	Classical data augmentation and CGAN improve deep transfer learning model performance.	The limited size of the COVID-19 CT database hinders model performance. The overfitting issue addressed by classical data augmentation and CGAN
Liming Xu, et al. [18]	Downscaling images to obtain low-resolution counterparts .The classifier model uses supervised generative adversarial nets for high-resolution image recovery.	Proposed SNSR-GAN method outperformed state-of-the-art SR approaches by 13% Achieved 9.3% and 10.5% average increment on GAN-train and GAN-test	Evaluation showed improved pathological invariance and acceptability scores.	Unsuitable common criteria like PSNR and VIF for medical image evaluation. Inaccurate and unstable density ratio estimation by the discriminator during training.
Vignesh Sampath et al. [19]	Various GAN modifications.	Improved minority class representation.	Enhances image quality and sample diversity.	Challenges in balancing classes in computer vision tasks.
Rick Sauber-Cole and Taghi M. [20]	GAN-based techniques.	Improved stability and diversity of generated samples.	Effective balancing of tabular data.	Stability and diversity challenges in tabular data.
Tanujit Chakraborty et al. [21]	Comprehensive review of GAN developments.	Significant advancements in GAN	Addressing limitations in GAN training stability and sample quality.	Ongoing advancements in GAN architecture and applications

3. Generative Adversarial Networks and Recent Advances

As a new field in unsupervised machine learning, Generative Adversarial Networks (GANs) were introduced by Goodfellow et al. in 2014 [22]. GANs consist of two neural networks: a generator and a discriminator. The generator attempts to create synthetic (fake) data that closely resembles real data, while the discriminator's role is to identify whether the data is real or generated, as illustrated in Figure 1. The network learns from these examples, and the insights gained are used to fine-tune the generator, enabling it to produce data that is as realistic as possible, as judged by the discriminator [9].

In the context of a Generative Adversarial Network (GAN), the generator model aims to minimize the cost or loss function, which quantifies the difference between the generated data and the training data annotated with class labels. Conversely, the discriminator model is trained in a more nuanced manner, as the optimization of its loss function measures the similarity between the generated data and real data [5]. Algorithm 1 illustrates that the learning procedure of a GAN involves the simultaneous training of both a discriminator and a generator. The input to the generator network consists of a noise vector from a latent space, which is then processed through a differentiable function to transform the noise vector into a fake yet plausible image. Meanwhile, the discriminator network, functioning as a binary classifier, attempts to differentiate between real images (label 1) and those artificially generated by the generator network (label 0). Therefore, the objective function of GANs can be defined as [18]:

$$\min_G \max_D V(D,G) = E_{x\sim p_r(x)}[\log D(x)] + E_{z\sim p_z(z)}\left[\log\left(1 - D\big(G(z)\big)\right)\right] \tag{1}$$

Where:

- $V(D,G)$: is the cross-entropy loss of D and G.
- $G(z)$: is the real data learned from a random dataset.
- (x): represents the real data.
- $D(x)$: is the discrimination accuracy of real data.
- $D\big(G(z)\big)$: is the discrimination accuracy of $G(z)$ ·
- $P_x(x)$ and $P_z(z)$: are respectively the probability distributions of the real data and the real-like data.
- E: is the expectation function.

Given a random noise vector z and a real image x.The generator attempts to minimize the loss $[\log(1 - D\big(G(z)\big)]$ while the discriminator attempts to maximize it, as shown in Eq. (1). For a fixed G, the optimal D is given by [12]:

$$D^*(x) = \frac{p_r(x)}{p_g(x)+p_r(x)} \tag{2}$$

Theoretically, when G is trained to its optimal state, the generated data distribution $p_g(x)$ gets closer to the real data distribution $p_r(x)$. If $p_g(x) = p_r(x), D^*(x)$ in Eq. (2) becomes $1/2$ [23]. This means that the discriminator is maximally confused and cannot differentiate between fake images and real ones. When the discriminator D reaches its optimal, the loss function for the generator G can be represented by substituting in $\mathrm{D}^*(\mathrm{x})$.

The optimized function of the discriminator ($\boldsymbol{D}^*$) is determined by the expectation sum of $D(G(z))$ and $D(x)$, as follows [14]:

$$\boldsymbol{D}^* = \boldsymbol{arg}\,\boldsymbol{\max_D} \boldsymbol{E}_{\boldsymbol{P_x(x)}}[\boldsymbol{log\,D(x)}] + \boldsymbol{E}_{\boldsymbol{P_z(z)}}[\boldsymbol{log(1 - D(G(z)))}] \tag{3}$$

Where:

- $\mathbb{E}_{P_{\text{data}}(x)}[log\,D(x)]$ represents the discriminator's loss when classifying real data.

- $\mathbb{E}_{P_z(z)}[log(1 - D(G(z)))]$ represents the generator's loss, as it tries to make $G(z)$ indistinguishable from real data by minimizing the discriminator's ability to classify it as fake.

The optimized function of the generator (G^*) is given as equation:

$$G^* = arg\min_G E_{P_z(z)}\left[log\left(1 - D\big(G(z)\big)\right)\right] \quad (4)$$

Where

- $E_{P_z(z)}$ *is* Expectation over the prior noise distribution P_z

In mathematical terms, Generative Adversarial Networks (GANs) aim to replicate a specific probability distribution. Consequently, loss functions, such as adversarial loss, are employed to measure the distance between the distribution of generated fake data and that of real data, as illustrated in Figure 1.

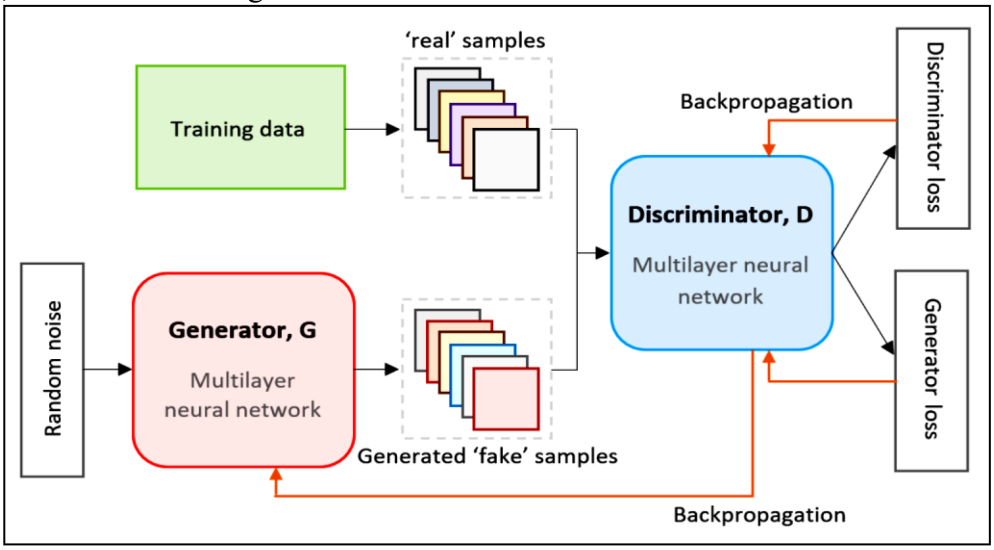


Figure 1: GAN Architecture [24].

The architecture of GAN can be summarized as follows [25]:

- **Generator (G):** The Generator network begins with random noise and aims to produce data that closely resembles the actual data distribution. This network is composed of layers that progressively transform the noise into coherent data, typically utilizing a series of convTransposed layers.
- **Discriminator (D):** The Discriminator network functions as a binary classifier that differentiates between real data and data generated by G. It is trained to maximize the probability of accurately identifying whether a sample is real or fake.
- **Adversarial Training:** The Generator and Discriminator are trained simultaneously, as outlined in Algorithm 1. This illustrates the training process of GAN, with the Generator seeking to minimize the classification error of the Discriminator while the Discriminator aims to maximize it. Mathematically, the objective is to solve the minimax game defined by Eq. (1).
- **Loss Functions:**

1. The Discriminator tries to maximize the probability of assigning the correct label to both real and generated data:

$$\mathcal{L}_D = -\frac{1}{m}\sum \left[\log D(X_{\text{real}}) + \log(1 - D(G(Z)))\right] \quad (5)$$

2. The Generator tries to fool the Discriminator by minimizing:

$$\mathcal{L}_G = -\frac{1}{m}\sum \left[\log D\big(G(Z')\big)\right] \quad (6)$$

This adversarial process allows GANs to generate highly realistic outputs, making them widely used in tasks such as image synthesis, style transfer, and data augmentation. As outlined in Algorithm 1, GAN training steps can be described as follows [26]:

1. **Initialize the generator and discriminator networks** with random weights.
2. **Input random noise (z)** into the generator, which generates a synthetic output (G (z)).
3. **Discriminator training:**

a. Feed real data (from the actual dataset) into the discriminator, and it classifies them as real.

b. Feed the synthetic data (G (z)) from the generator into the discriminator, and it classifies them as fake.

c. Calculate the discriminator loss based on how well it distinguishes real from fake data.

d. Update the discriminator's weights using backpropagation to minimize the loss.

4. **Generator training:**

a. Pass random noise (z) through the generator to create new synthetic data.

b. Feed the synthetic data to the discriminator, but this time, the goal is to fool the discriminator into classifying it as real.

c. Calculate the generator loss based on the discriminator's output (the difference between predicted real and fake labels).

d. Update the generator's weights to minimize the generator loss.

5. **Repeat steps 3 and 4** iteratively, alternating between updating the generator and the discriminator until the generator produces data indistinguishable from the real data.

The process continues until the generator generates realistic data that the discriminator can no longer distinguish from the real data.

Algorithm 1: The Training of GAN

Input: Initialization:

- Initialize the parameters of the Generator G with θ_G.
- Initialize the parameters of the Discriminator D with θ_D.

Output: The objective for the GAN is a minimax game:

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

- The Discriminator D tries to maximize $V(D,G)$ by correctly classifying real and fake data.
- The Generator G tries to minimize $V(D,G)$ by fooling the Discriminator into classifying fake data as real.

Training: Repeat for a number of iterations (until convergence):

1 Discriminator Training:

Step 1: Sample a minibatch of m real data examples from the training dataset $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$ of the distribution $p_{\text{data}}(x)$.

Step 2: Sample a minibatch of m random noise vectors $\{z^{(1)}, z^{(2)}, \ldots, z^{(m)}\}$ from the noise distribution $p_z(z)$.

Step 3: Feed the fake data $G(Z)$ into the discriminator, obtaining$D(G(Z))$.

Step 4: Compute the Discriminator's loss for the real data: $\text{Loss}_{\text{real}} = \frac{1}{m}\sum_{i=1}^{m} \log D(x^{(i)};\theta_D)$

Step 5: Compute the Discriminator's loss for the fake data: $\text{Loss}_{\text{fake}} = \frac{1}{m}\sum_{i=1}^{m} \log(1-D(G(Z)))$

Step 6: Compute the total Discriminator loss: $\text{Loss}_D = -(\text{Loss}_{\text{real}} + \text{Loss}_{\text{fake}})$

Step 7: Update the Discriminator parameters θ_D by minimizing Loss_D using a gradient descent optimizer:

$\theta_D \leftarrow \theta_D - \eta_D \nabla_{\theta_D}\text{Loss}_D$ Where, η_D is the learning rate for the Discriminator.

2 Generator Training:

Step 1: Sample a minibatch of m random noise vectors $\{z^{(1)}, z^{(2)}, \ldots, z^{(m)}\}$ from the noise distribution $p_z(z)$

Step 2: Generate a minibatch of fake data examples using the Generator: $\hat{x}^{(i)} = G(z^{(i)};\theta_G)$ for $i = 1,2,\ldots,m$

Step 3: Compute the Generator's loss based on the Discriminator's output: $\text{Loss}_G = -\frac{1}{m}\sum_{i=1}^{m} \log D(\hat{x}^{(i)};\theta_D)$

Step 4: Update the Generator parameters θ_G by minimizing Loss_G using a gradient descent optimizer:

$\theta_G \leftarrow \theta_G - \eta_G \nabla_{\theta_G}\text{Loss}_G$ Where, η_G is the learning rate for the Generator.

Repeat: Continue iterating between training the Discriminator and the Generator until the GAN reaches a convergence.

4. Materials and Methods

4.1. The Proposed Model

In this proposed model, MGAN, firstly enhances the generator structure by increasing the depth of the generator through additional layers, enabling it to capture more complex features and generate more realistic data. The generator starts with a latent noise vector and progressively up samples it using transposed convolution layers to create a high-resolution RGB image of size (128×128×3). The image size increases from (64×64×3) to (128×128×3), and the modification of architecture can be as follows [27]:

Generator: The generator is designed to take a latent vector and produce an image through a series of transposed convolutional layers, batch normalization, and activation functions. The generator consists of five blocks, and each block has three hidden layers, the first layer is convTransposed2D of (4 × 4) kernels and 512, 256, 128, 64, and 32 filters, the second layer is a batch-normalization layer, and a ReLU activation function is the third layer, only last block have two hidden layer, convolutional layer and a Tanh activation function. The modification of the generator by adding a new block with output (64 x 32 x 32), which is located in the fifth block, to become six blocks in the generator, as shown in Figure 2, and Algorithm 2. A new block based convolutional neural network (CNN) architecture specifically designed for the purpose of enhancing the accuracy of diagnosis of bladder cancer [28].

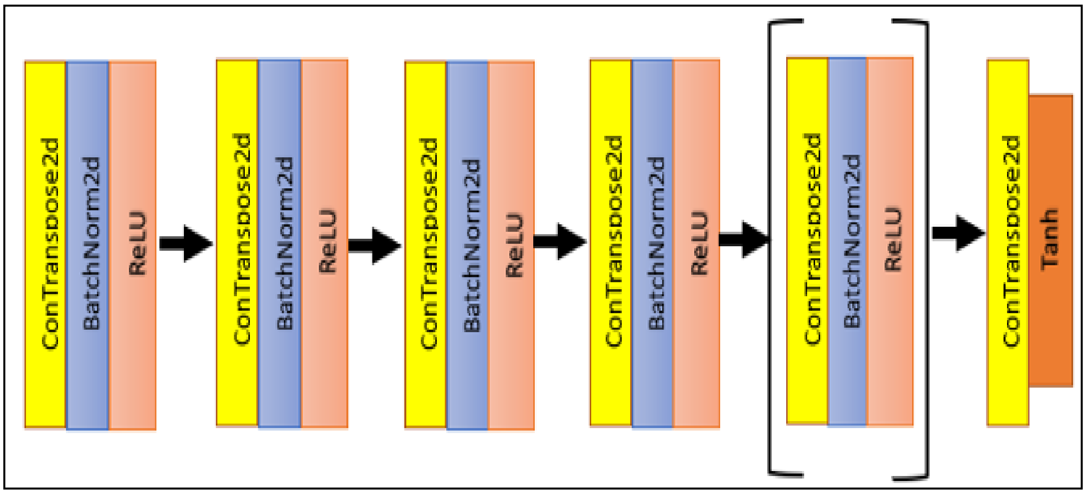


Figure 2: The new structure of the generator.

Algorithm 2: The Generator Network

Input: Random noise vector sampled from a latent space (128 x 1 x 1).
Output: Generated image of size (3 x 128 x 128).

Begin:
Initialize latent vector of shape (128 x 1 x 1)
First Transpose Convolution Layer:
- Perform transpose convolution on latent_vector with:
  - Input: 128 channels, Output: 512 channels
  - kernel_size = 4, stride = 1, padding = 0
- Apply batch normalization to the output
- Apply ReLU activation function
- Output shape: (512 x 4 x 4)

3. Second Transpose Convolution Layer:
- Perform transpose convolution on the output from Step 2 with:

- Input: 512 channels, Output: 256 channels
- kernel_size = 4, stride = 2, padding = 1
- Apply batch normalization to the output
- Apply ReLU activation function
- Output shape: (256 x 8 x 8)

4. Third Transpose Convolution Layer:
    - Perform transpose convolution on the output from Step 3 with:
        - Input: 256 channels, Output: 128 channels
        - kernel_size = 4, stride = 2, padding = 1
    - Apply batch normalization to the output
    - Apply ReLU activation function
    - Output shape: (128 x 16 x 16)
5. Fourth Transpose Convolution Layer:
    - Perform transpose convolution on the output from Step 4 with:
        - Input: 128 channels, Output: 64 channels
        - kernel_size = 4, stride = 2, padding = 1
    - Apply batch normalization to the output
    - Apply ReLU activation function
    - Output shape: (64 x 32 x 32)
6. Fifth Transpose Convolution Layer:
    - Perform transpose convolution on the output from Step 5 with:
        - Input: 64 channels, Output: 32 channels
        - kernel_size = 4, stride = 2, padding = 1
    - Apply batch normalization to the output
    - Apply ReLU activation function
    - Output shape: (32 x 64 x 64)
7. Final Transpose Convolution Layer:
    - Perform transpose convolution on the output from Step 6 with:
        - Input: 32 channels, Output: 3 channels
        - kernel_size = 4, stride = 2, padding = 1
    - Apply Tanh activation function
    - Output shape: (3 x 128 x 128)
8. Return the generated image of size (3 x 128 x 128)

End

Discriminator: The network of discriminator is trained and prepared to solve the maximization problem presented in Eq. (1), which consists of a series of conv2D layers, batch normalization, and activation functions, according to the new structure, adding a new block with dimenution (3 x 32 x 32) of the beginning discriminator to match the output of generator, and become the number of blocks are six as well, as shown in Figure 3, each block consists of three hidden layers, convolutional layers with 4 × 4 kernels and 32, 64, 128, 256, and 512 filters followed by a batch-normalization layer and LeakyReLU activation function, and only in the last block using sigmoid activation function. Algorithm 3 explains the new architecture.

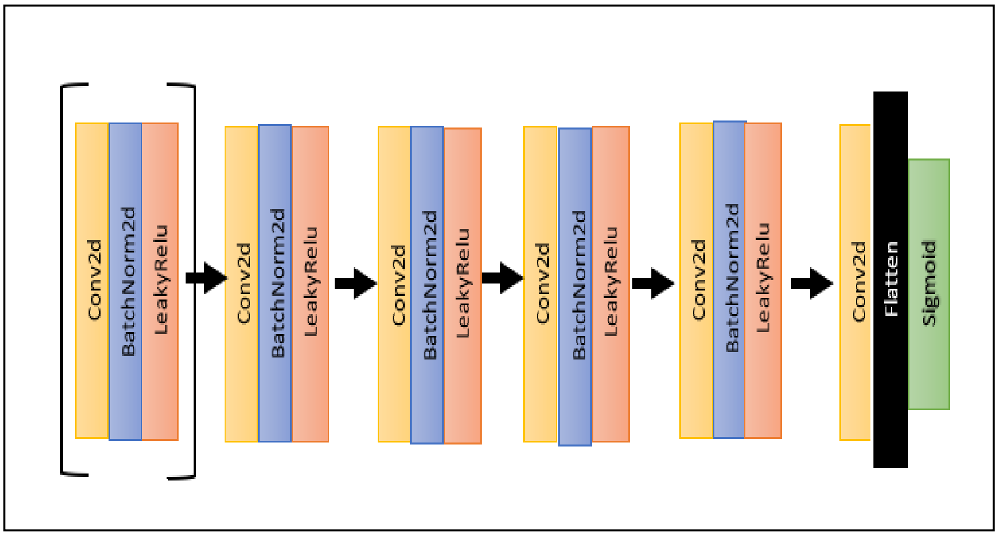


Figure 3: The new structure of the discriminator.

Algorithm 3: The Discriminator Network

Input: Image with size (3, 128, 128).

Output: A scalar value between 0 and 1, representing the probability that the input image is real or fake.

Begin:

1. Initialize input image of shape (3, 128, 128)

2. First Convolution Layer:
   - Perform convolution on input image with:
     - Input channels: 3, Output channels: 32
     - kernel_size = 4, stride = 2, padding = 1
   - Apply batch normalization to the output
   - Apply LeakyReLU activation function with negative slope = 0.2
   - Output shape: (32, 64, 64)

3. Second Convolution Layer:
   - Perform convolution on the output from Step 2 with:
     - Input channels: 32, Output channels: 64
     - kernel_size = 4, stride = 2, padding = 1
   - Apply batch normalization to the output
   - Apply LeakyReLU activation function with negative slope = 0.2
   - Output shape: (64, 32, 32)

4. Third Convolution Layer:
   - Perform convolution on the output from Step 3 with:
     - Input channels: 64, Output channels: 128
     - kernel_size = 4, stride = 2, padding = 1
   - Apply batch normalization to the output
   - Apply LeakyReLU activation function with negative slope = 0.2
   - Output shape: (128, 16, 16)

5. Fourth Convolution Layer:
   - Perform convolution on the output from Step 4 with:
     - Input channels: 128, Output channels: 256
     - kernel_size = 4, stride = 2, padding = 1
   - Apply batch normalization to the output
   - Apply LeakyReLU activation function with negative slope = 0.2
   - Output shape: (256, 8, 8)

6. Fifth Convolution Layer:
   - Perform convolution on the output from Step 5 with:
     - Input channels: 256, Output channels: 512
     - kernel_size = 4, stride = 2, padding = 1
   - Apply batch normalization to the output
   - Apply LeakyReLU activation function with negative slope = 0.2
   - Output shape: (512, 4, 4)

7. Final Convolution Layer:
   - Perform convolution on the output from Step 6 with:
     - Input channels: 512, Output channels: 1
     - kernel_size = 4, stride = 1, padding = 0
   - Output shape: (1, 1, 1)

8. Flatten the output to convert it into a single scalar value (1x1x1).

9. Apply the Sigmoid activation function to scale the output to the range [0, 1], representing the probability that the input image is real or fake.

10. Return the final scalar value representing the probability that the input image is real.

End

4.2 Dataset Description

The presented case studies demonstrate the effectiveness of the proposed GAN-based approach in various medical imaging applications, highlighting improvements in classifier performance and diagnostic accuracy. In this study, two distinct datasets, as shown in Figure 4, were used to generate images for the dataset. These datasets include a proprietary collection developed at Zagazig University in Egypt and a set of endoscopic images from patients undergoing clinical procedures. The comprehensive details of these datasets are as follows:

A. Dataset1 (Pathological):

The first dataset used in this research is a proprietary collection created by the team at Zagazig University in Egypt, authorized under the Institutional Review Board (IRB) number

11044-22-8-2023. This dataset consists of 2,629 pathological images categorized into three classes: non-invasive malignant, invasive malignant, and normal bladder mucosa, which serves as a standard for deep learning measurement [29].

B. Dataset 2 (Endoscopic):

The second dataset comprises 1,754 endoscopic images from 23 patients undergoing Trans-Urethral Resection of Bladder Tumor (TURBT). These images are labelled based on histopathology analysis from the resected tissue. The endoscopic procedures utilize White Light Imaging (WLI) and, when available, Narrow Band Imaging (NBI). This dataset is categorized into four classes following the World Health Organization WHO and the International Society of Urological Pathology guidelines: Low-Grade Cancer (LGC), High-Grade Cancer (HGC), and No Tumor Lesion (NTL), which includes cystitis and other inflammatory conditions, and Non-Suspicious Tissue (NST) [30].

Table 2 categorizes a dataset of cases into different classes with a corresponding number of cases in each class. Specifically, it lists two distinct data sources identified as Dataset 1 and Dataset 2. Each dataset contains multiple classes, each with a certain number of cases. For instance, dataset 1 includes classes like "Noninvasive", "Invasive", and "Normal", with the respective counts of 660, 1178, and 454 cases, respectively. Similarly, Dataset 2 enumerates cases under 'LGC', 'HGC', 'NTL', and 'NST' with counts of 647, 469, 134, and 504, respectively. These slides include different types of tissue sections stained with various dyes, such as hematoxylin and eosin (H&E) and other staining methods. Histological images are crucial for examining the microscopic structure of tissues, aiding in the diagnosis and study of diseases, particularly cancer. On the right, under the title "B: examples of Data set2," endoscopic images are shown. These images are captured using an endoscope, a specialized medical instrument equipped with a camera allowing internal visualization of an organ or tissue. Endoscopic imaging is commonly used for diagnosing and monitoring conditions within the gastrointestinal tract, respiratory system, and other internal structures. Both types of data sets are vital in medical research and diagnostics, providing comprehensive views from the cellular to the organ level. There is a state of imbalance in the blocks with both bases, as shown in Figure 5.

Table 1: Classification of cases in Dataset 1 and Dataset 2 across multiple classes.

Material	Class	No. of Cases
Dataset 1	Noninvasive	660
	Invasive	1178
	Normal	454
Dataset 2	LGC	647
	HGC	469
	NTL	134
	NST	504

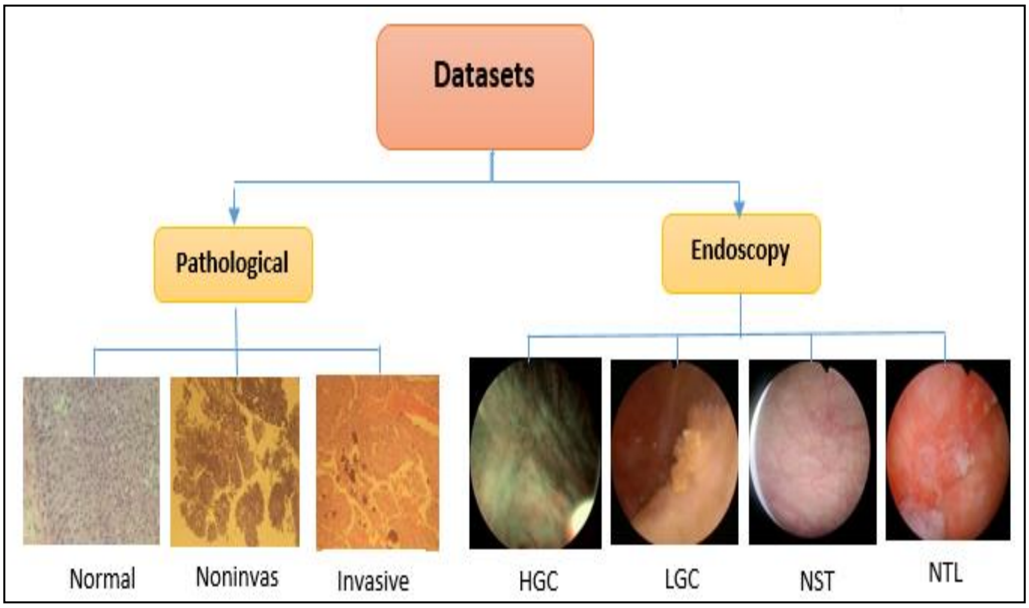


Figure 4: Examples of two datasets.

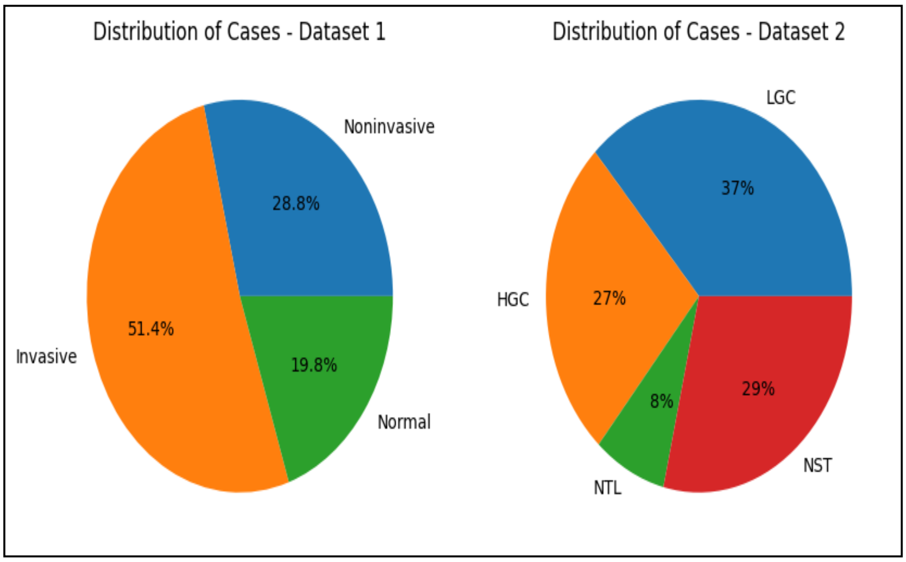


Figure 5: Imbalance distribution of histological and endoscopic data sets.

5. Results and Discussion

This section presents the results of the experiments conducted using different Convolutional Neural Network (CNN) models to address the class imbalance problem using the proposed Modified Generative Adversarial Network (MGAN) framework.

5.1 CNN Models for Comparison

To evaluate the effectiveness of the MGAN-based approach in handling class imbalance, three widely recognized CNN architectures were used for comparison: One of the most well-known deep convolutional neural networks is the VGG16 [31], then there is the MobileNetV2 [32], and finally the InceptionV3 [33]. These models were chosen because of their classification performance in image recognition tasks, their applicability to transfer learning methods, and their effectiveness in enhancing the precision of bladder cancer image analysis [34].

5.2 Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a system of deep learning algorithms that are used in image classification and especially object detection. They include convoluted, pooling, and fully connected layers, which basically take images and find image patterns. In the present research, CNNs were used for the purpose of evaluating the quality and usefulness of the synthetic images for addressing matters related to class imbalance problems.

5.3 Experimental Implementation

The experiments were performed by applying the CNN models to the imbalanced datasets and the equalized datasets acquired with the help of the modified GAN structure. To assess the extent to which MGAN generated data enhanced the classification performance of every one of the CNN models, four measures of performance were used, including accuracy, precision, recall, and F1-score.

5.4 Results and Analysis

The comparative analysis of the CNN models revealed that using MGAN-generated synthetic data led to significant improvements in the classification of minority classes. The VGG19 model, when trained with MGAN-balanced data, showed a notable increase in accuracy, demonstrating the potential of MGANs to enhance model performance in scenarios with limited training samples. MobileNetV2 and Inception v3 also exhibited improved performance metrics, validating the effectiveness of the proposed MGAN-based approach across different CNN architectures.

There is a state of imbalance in the blocks with both datasets; therefore, using GAN to solve this problem and create a balanced dataset.

The result of using the standard GAN of Dataset1 and the resulting images generated with resolution 64*64, as shown in Figure 6. This is not feasible for images that need to be generated for use in training and diagnosis of bladder cancer; therefore, a proposed modification of the structure of the GAN, and the resulting structure after modification, as shown in Figure 7, and Figure 8, that some samples from the generated images of Datasets and 2. The resulting images are of size 128*128, to use in the training.

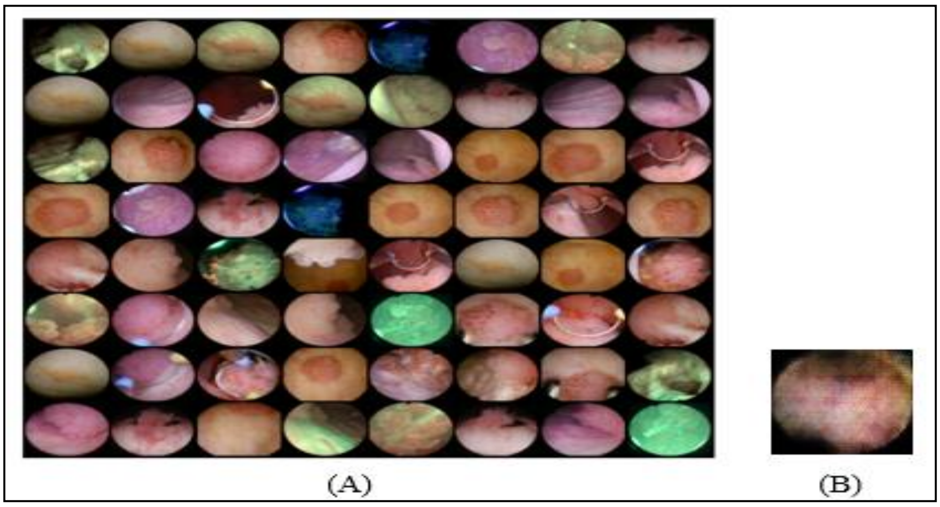


Figure 6: (A): The traditional GAN to generate a grid with size (8*8) samples, and each image with size 64*64, (B): example of one sample in the grid.

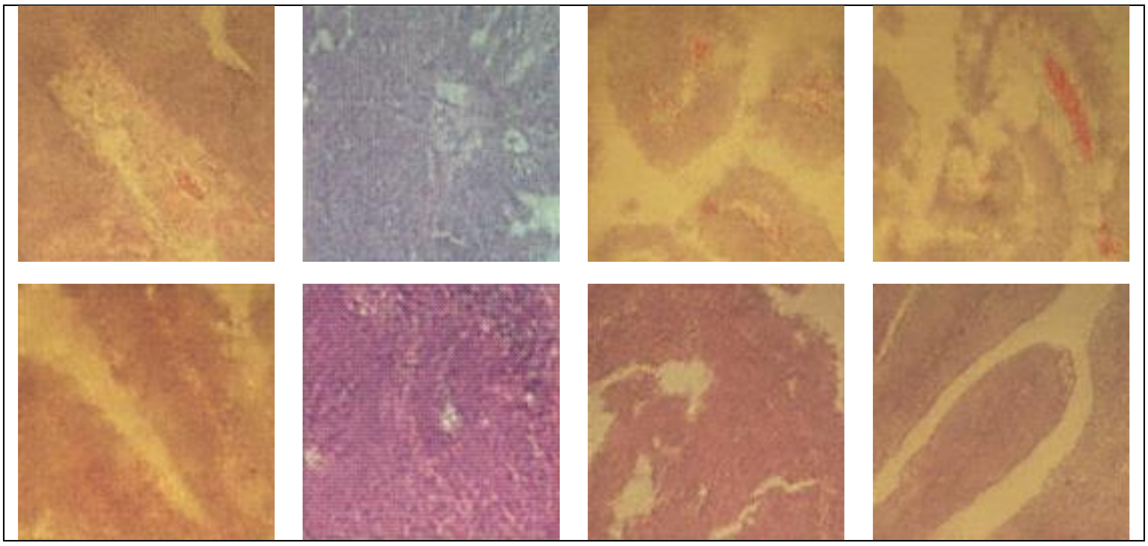


Figure 7: Generated 128x128 images from Dataset 1 using the proposed method.

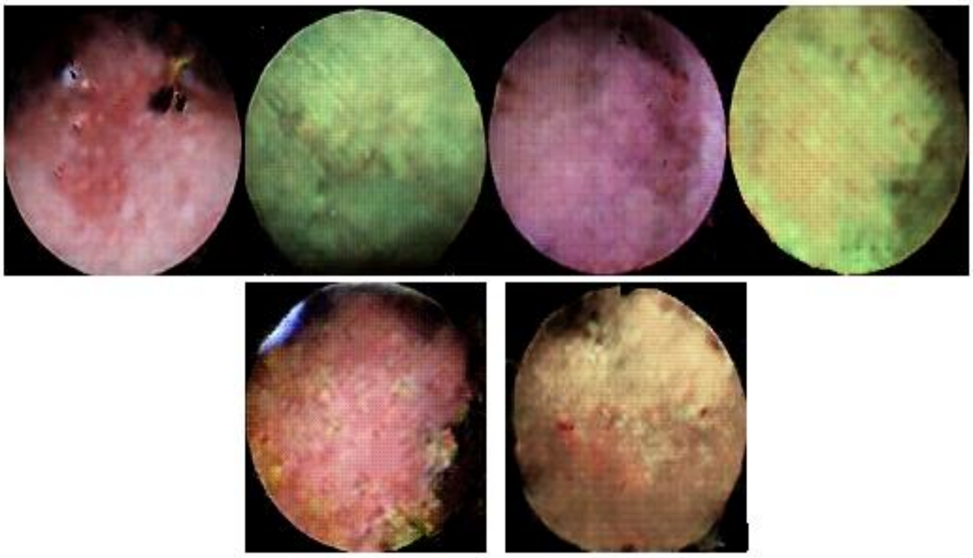


Figure 8: Generated 128*128*3 images from Dataset2 using the proposed method.

Using the MGAN, 240 Non-invasive images and 550 Normal images were generated for Dataset 1, without generating any Invasive images, thereby balancing Dataset 1. For Dataset 2, 350 LGC images, 560 HGC images, 850 NTL images, and 500 NST images were generated, as shown in Table 2 and Figure 9. In the proposed model, every 100 images are generated in 3200 seconds over 20 epochs. Figure 10 depicts the loss curves for the generator and discriminator, along with the score between real and fake images. The substantial gap in Figure 10 indicates the effectiveness of the proposed model, as there is a significant difference between the generator and discriminator losses, and between the scores of real and fake images.

Table 2: Classification of cases in Dataset 1 and Dataset 2 across multiple classes after MGAN.

Material	Class	No. of Cases
Dataset 1	Noninvasive	1000
	Invasive	1178
	Normal	1000

Dataset 2	LGC	1000
	HGC	1000
	NTL	1000
	NST	1000

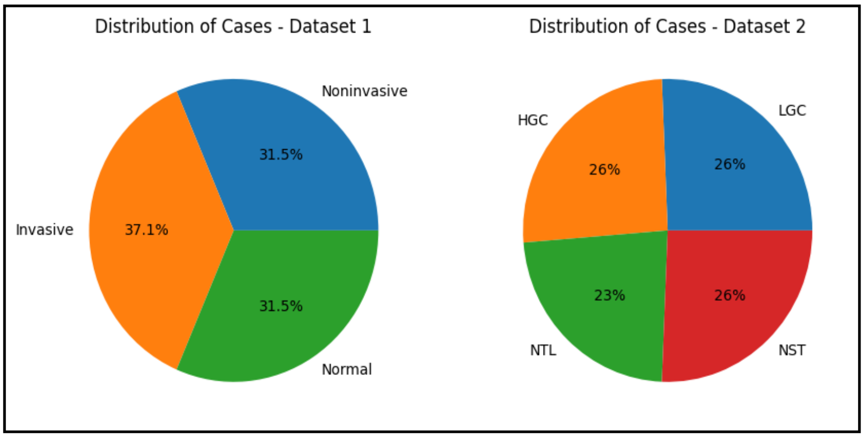
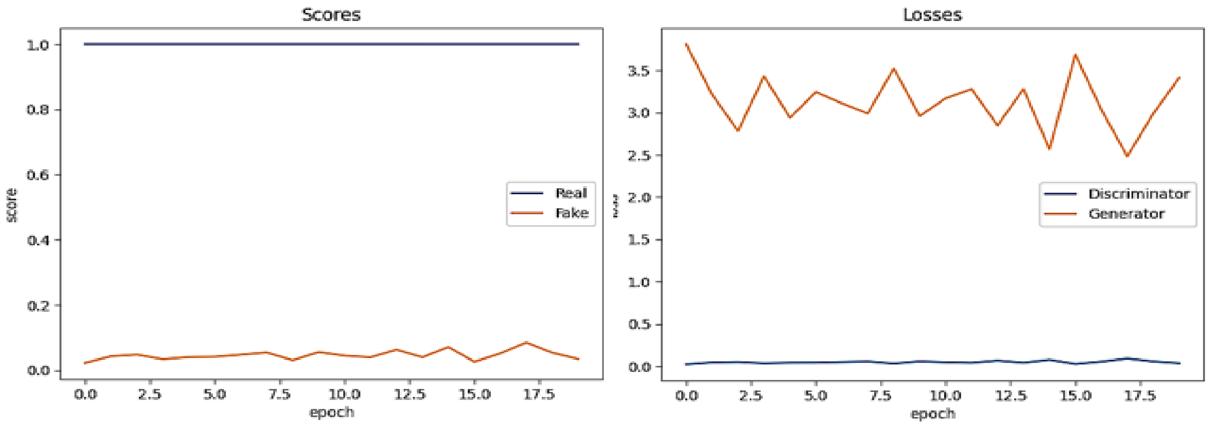


Figure 9: Balance distribution of Histological and Endoscopic Data Sets after using MGAN.



(A) (B)

Figure 10: The performance after using MGAN, (A) the scores between real and fake, and (B) the losses between generator and discriminator.

When performing the training and testing on datasets by using multi-CNN models such as (InceptionV3, VGG16, CNN), for Dataset1 and 2, and measuring the accuracy before and after using GAN and MGAN for 50 epochs, the results will be as shown in Table 3,5, 6, and 7. From these comparisons in these Tables, it was found that using MGAN with datasets and performing the training, the high accuracy of CNN, VGG19, and InceptionV3 achieved 98.8%, 98.5%, and 99.91% accuracy, respectively.

Table 3: The accuracy of the training and test Dataset 1 before MGAN.

Model	Train-ACC	Test-ACC	Precision	Sensitivity	F1 Score	Time in seconds
CNN	93.89%	91.95%	0.9042	0.9239	0.9140	14257
MobileNet-V2	91.67%	89.15%	0.9000	0.9319	0.9398	13422
Inception-V3	92.02%	91.25%	0.9142	0.9119	0.9540	14963
VGG16	93.67%	88.95%	0.9033	0.9067	0.8980	11209

Table 4: The accuracy of the training and test Dataset 2 before MGAN.

Model	Train-ACC	Test-ACC	Precision	Sensitivity	F1 Score	Time in seconds
CNN	91.89%	90.91%	0.9123	0.9001	0.9089	11117
MobileNet-V2	91.37%	89.00%	0.9100	0.9229	0.9120	13402
Inception-V3	90.55%	89.25%	0.9111	0.9129	0.9230	12003
VGG16	93.07%	88.95%	0.9133	0.9007	0.8900	11309

Table 5: The accuracy of the training and test Dataset 1 after MGAN.

Model	Train-ACC	Test-ACC	Precision	Sensitivity	F1 Score	Time in seconds
CNN	98.81%	93.91%	0.9503	0.9501	0.9889	23347
MobileNet-V2	94.30%	92.00%	0.9400	0.9329	0.9720	24562
Inception-V3	98.35%	93.25%	0.9711	0.9829	0.9830	24683
VGG16	95.01%	90.91%	0.9433	0.9507	0.9504	25229

Table 6: The accuracy of the training and test Dataset 2 after MGAN.

Model	Train-ACC	Test-ACC	Precision	Sensitivity	F1 Score	Time in seconds
CNN	99.09%	94.91%	0.9833	0.9781	0.9800	23247
MobileNet-V2	98.07%	95.10%	0.9870	0.9899	0.9711	24962
Inception-V3	98.22%	93.35%	0.9744	0.9759	0.9780	24883
VGG16	98.44%	94.91%	0.9633	0.9647	0.9874	25129

The experimental results indicate that MGANs are highly effective in addressing class imbalance. Unlike traditional methods, GANs generate diverse and realistic samples, enhancing the classifier's ability to learn from minority class instances. However, the quality of generated samples and MGAN training stability are critical factors influencing performance. Generating images at a higher resolution (128x128 pixels) ensures better quality and more detailed synthetic samples, which is especially important for medical applications. However, the quality of generated samples and the stability of MGAN training are critical factors that influence performance. After applying MGAN-based preprocessing, execution time increased by approximately 85–115%, depending on the specific model and dataset employed. This increase is primarily due to the additional computational overhead introduced during data augmentation and class balancing procedures. Despite the added computational cost, the performance improvements across various evaluation metrics justify this trade-off. Significant enhancements were observed in accuracy, precision, and particularly the F1 Score, indicating more balanced and effective classification, especially for minority classes. These improvements are particularly valuable in critical application domains such as medical imaging and intrusion detection, where accurate identification of rare events or conditions is essential. Therefore, although the preprocessing introduces a computational burden, the resulting performance gains—especially in high-stakes and sensitive contexts—render the trade-off both acceptable and beneficial.

An additional analysis was conducted to assess the statistical significance of differences in test accuracies under various training-testing configurations and across multiple epochs. Specifically, a paired t-test was applied to compare the test accuracies of the CNN model with and without the MGAN enhancement over multiple epochs (10–50), as shown in Table 8. The paired t-test was chosen due to its suitability for comparing two sets of related accuracy scores obtained under similar conditions (i.e., identical training/validation/testing splits, with and without the MGAN component). The null hypothesis posits that no significant difference exists between the two sets of test accuracies. A summary of the test results is presented in Table 9.

The paired t-test yielded a p-value of $p < 0.01$, indicating a statistically significant improvement in test accuracy for the MGAN-enhanced CNN model compared to the baseline CNN model. This result confirms that the observed improvement is unlikely to have occurred by chance and is statistically meaningful. Furthermore, the consistently high performance across different epochs suggests that the proposed approach exhibits robustness and generalization capability.

Table 7: The performance of CNN with pathological diagnosis dataset.

Model	Epoch	Training		Validation		Testing Acc.%
		Loss	Acc. %	Loss	Acc.%	
Without MGAN	10	0.1011	88.90	0.2011	80.88	70.99
	20	0.1007	89.10	0.1066	89.90	70.90
	30	0.1067	90.01	0.1056	86.01	87.01
	40	0.1028	89.98	0.1034	88.00	85.00
	50	0.1039	91.89	0.1016	90.00	90.91
With MGAN	10	0.0131	99.60	0.0518	98.83	97.81
	20	0.0000	100.00	0.0027	99.48	98
	30	0.0254	98.85	0.0999	98.18	98
	40	0.0000	100.00	0.0012	99.39	98.90
	50	0.0032	99.85	0.0124	99.43	99.21

Table 8: Test results summary.

Epoch	Accuracy (Without MGAN)	Accuracy (With MGAN)
10	70.99%	97.81%
20	70.90%	98.00%
30	87.01%	98.00%
40	85.00%	98.90%
50	90.91%	99.21%

To evaluate the model's performance on the testing dataset, we used the following PyTorch-based code snippet. The evaluation was conducted in inference mode using torch.no_grad() to prevent gradient calculation and reduce memory usage. The outputs of the model were compared with the true labels to compute the loss and classification accuracy, as shown in Algorithm 4:

Algorithm 4: PyTorch code used for model evaluation over the test set.

```
def evaluate_model(model, dataloader, criterion, class_names):
with torch.no_grad():
    for inputs, labels in dataloader:
        inputs, labels = inputs.to(device), labels.to(device)

        outputs = model(inputs)
        loss = criterion(outputs, labels)
        total_loss += loss.item()
```

```
_, preds = torch.max(outputs, 1)
correct += (preds == labels).sum().item()
total += labels.size(0)

all_preds.extend(preds.cpu().numpy())
all_labels.extend(labels.cpu().numpy())
```

5.5 Comparison with State-of-the-Art Techniques

The proposed study builds on recent advancements in using GANs to tackle class imbalance in various domains. Sampath et al. [19] explored GAN modifications to improve minority class representation in computer vision, while Sauber-Cole and Khoshgoftaar [20] focused on balancing tabular data. The proposed approach, however, is specifically tailored to medical imaging, with unique modifications in the GAN architecture that enhance the generation of high-resolution (128x128 pixels) synthetic images, crucial for accurate medical diagnoses. Furthermore, Chakraborty et al. [21] highlighted key advancements in GANs over the past decade, reinforcing the relevance of the proposed MGAN-based method in addressing current limitations in GAN training stability and sample quality. The comparative analysis of the results with traditional techniques demonstrates the significant improvements achieved through the proposed modifications, particularly in the accurate classification of minority classes in medical datasets. Table 9 compares state-of-the-art papers with the proposed work, highlighting the most important points of comparison and a summary of each.

Table 9: Comparison of state-of-the-art papers and proposed work.

Paper	Summary	Comparison with the Proposed Work
A survey on generative adversarial networks for imbalance problems in computer vision tasks by Vignesh Sampath et al. [19].	Reviews various GAN-based techniques to address imbalance issues in computer vision. Discusses methods for enhancing image quality and sample diversity.	The proposed approach uses modified GANs to generate high-resolution images for minority classes. Our modifications, like increased generator depth, improve performance in medical imaging, which might differ from the general computer vision applications discussed by Sampath et al.
The use of generative adversarial networks to alleviate class imbalance in tabular data: a survey by Rick Sauber-Cole and Taghi M. Khoshgoftaar [20].	Explores GAN applications to balance tabular data, addressing stability and diversity challenges.	The proposed work focuses on image data, demonstrating the flexibility and robustness of GAN architecture modifications. Emphasize the precision of our synthetic data for medical imaging, which requires more precise synthetic data than tabular forms.
Ten Years of Generative Adversarial Nets (GANs): a survey of the state-of-the-art by Tanujit Chakraborty et al. [21].	Covers a decade of GAN developments, noting advancements in architecture, training stability, and applications.	The proposed work aligns with these advancements by generating high-quality, balanced medical images. Our MGANs generate 128x128 pixel images, crucial for medical diagnosis that may not be covered extensively in the general GAN literature.

6. Conclusion

This paper presents an MGAN approach to solving the class imbalance problem in datasets. The experiments demonstrate that MGANs can generate realistic synthetic samples for the minority class, thereby balancing the dataset and improving classifier performance. Traditional data augmentation methods, such as rotation, flipping, and resizing, are less effective for medical images, emphasizing the need for advanced techniques like GANs. Generating high-resolution images (128x128 pixels) further improves the quality of synthetic samples and classifier performance.

Comparisons with state-of-the-art techniques reveal that the approach aligns with recent advancements in GAN modifications aimed at improving minority class representation (Sampath et al.), balancing tabular data (Sauber-Cole and Khoshgoftaar), and overall developments in GAN architecture and training stability (Chakraborty et al.). However, this work is specifically tailored to medical imaging, with unique modifications that enhance the generation of high-quality synthetic images, crucial for accurate medical diagnoses.
The main conclusions are summarized as follows:
1. The MGAN-based data augmentation model employs adversarial learning to produce realistic samples for the minority class. These realistic samples, when added to the original dataset, significantly improve the homogeneity of the overall dataset, thereby augmenting the training samples for diagnostic models.
2. Post data augmentation, the recognition accuracy of minority samples shows a remarkable improvement. The proposed MGAN model demonstrates a stronger performance in addressing imbalanced samples compared to traditional methods.
Future work will focus on optimizing GAN architectures and exploring hybrid approaches that combine GANs with traditional methods. This ongoing optimization aims to further enhance the generation of synthetic samples and improve the performance of classifiers in various applications.
The proposed MGAN-based approach significantly advances the state-of-the-art in addressing class imbalance in medical datasets, offering a robust solution that enhances both data quality and classifier performance.

Conflict of Interest: The authors declare that they have no conflicts of interest.

Acknowledgements
The authors would like to extend their sincere gratitude to everyone who contributed to the completion of this research paper.

REFERENCES
[1] Y. Yang and Z. Xu, "Rethinking the value of labels for improving class-imbalanced learning," Advances in Neural Information Processing Systems, vol. 33, pp. 19290-19301, 2020. https://github.com/YyzHarry/imbalanced-semi-self.
[2] A. Fernández, S. García, and F. Herrera, "Addressing the classification with imbalanced data: Open problems and new challenges on class distribution," in Proc. Int. Conf. Hybrid Artif. Intell. Syst., Springer, 2011, pp. 1-10. https://doi.org/10.1007/978-3-642-21219-2_1
[3] S. Wang, Y. Dai, J. Shen, and J. Xuan, "Research on expansion and classification of imbalanced data based on SMOTE algorithm," Scientific Reports, vol. 11, no. 1, pp. 1-11, 2021. https://doi.org/10.1038/s41598-021-03430-5
[4] M. M. Ahsan, S. A. Luna, and Z. Siddique, "Machine-learning-based disease diagnosis: A comprehensive review," in Healthcare, vol. 10, p. 541, MDPI, 2022. https://doi.org/10.3390/healthcare10030541
[5] Q. Yang and X. Wu, "10 challenging problems in data mining research," Int. J. Inf. Technol. Decis. Making, vol. 5, no. 4, pp. 597-604, 2006. https://doi.org/10.1142/S0219622006002258
[6] S. García and F. Herrera, "Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy," Evol. Comput, vol. 17, no. 3, pp. 275-306, 2009. https://doi.org/10.1162/evco.2009.17.3.275
[7] B. Das, N. C. Krishnan, and D. J. Cook, "Handling class overlap and imbalance to detect prompt situations in smart homes," in Proc. 2013 IEEE 13th Int. Conf. Data Mining Workshops, IEEE, 2013, pp. 266-273. https://doi.org/10.1109/ICDMW.2013.18
[8] R. Caruana, Y. Lou, J. Gehrke, P. Koch, M. Sturm, and N. Elhadad, "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission," in Proc. 21st ACM

SIGKDD Int. Conf. Knowl. Discovery Data Mining, 2015, pp. 1721-1730. https://doi.org/10.1145/2783258.2788613

[9] R. R. Fletcher, A. Nakeshimana, and O. Olubeko, "Addressing fairness, bias, and appropriate use of artificial intelligence and machine learning in global health," Front. Artif. Intell., vol. 3, p. 116, 2021. https://doi.org/10.3389/frai.2020.561802

[10] R. Longadge and S. Dongre, "Class imbalance problem in data mining review," arXiv preprint, arXiv:1305.1707, 2013. [Online]. Available: https://arxiv.org/abs/1305.1707

[11] A. Radford, "Unsupervised representation learning with deep convolutional generative adversarial networks," arXiv preprint, arXiv:1511.06434, 2015. [Online]. Available: https://arxiv.org/abs/1511.06434

[12] A. Krizhevsky and G. Hinton, *Learning Multiple Layers of Features from Tiny Images*, 2009, p. 7.

[13] J. Lee and K. Park, "GAN-based imbalanced data intrusion detection system," Pers. Ubiquitous Comput., vol. 25, pp. 121-128, 2019. https://doi.org/10.1007/s00779-019-01332-y

[14] Dong, C. C. Loy, et al., "Image super-resolution using deep convolutional networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 2, pp. 295-307, 2015. https://doi.org/10.1109/TPAMI.2015.2439281

[15] W. Jiang, Y. Hong, B. Zhou, X. He, and C. Cheng, "A GAN-based anomaly detection approach for imbalanced industrial time series," IEEE Access, vol. 7, pp. 143608-143619, 2019. https://doi.org/10.1109/ACCESS.2019.2944689

[16] A. Sharma, P. K. Singh, and R. Chandra, "Smotified-GAN for class imbalanced pattern classification problems," IEEE Access, 2022. https://doi.org/10.1109/ACCESS.2022.3158977

[17] M. Loey, G. Manogaran, and N. E. M. Khalifa, "A deep transfer learning model with classical data augmentation and CGAN to detect COVID-19 from chest CT radiography digital images," Neural Comput. Appl., pp. 1-13, 2020, doi: 10.1007/s00521-020-05437-x. https://doi.org/10.1007/s00521-020-05437-x

[18] L. M. Xu, X. H. Zeng, Z. W. Huang, W. S. Li, and H. Zhang, "Low-dose chest X-ray image super-resolution using generative adversarial nets with spectral normalization," Biomed. Signal Process. Control, vol. 55, p. 101600, 2020, doi: 10.1016/j.bspc.2019.101600. https://doi.org/10.1016/j.bspc.2019.101600

[19] V. Sampath, I. Maurtua, J. J. Aguilar Martin, and A. Gutierrez, "A survey on generative adversarial networks for imbalance problems in computer vision tasks," J. Big Data, vol. 8, pp. 1-59, Dec. 2021. https://doi.org/10.1186/s40537-021-00414-0

[20] R. Sauber-Cole and T. M. Khoshgoftaar, "The use of generative adversarial networks to alleviate class imbalance in tabular data: A survey," J. Big Data, vol. 9, no. 1, p. 98, Aug. 2022. https://doi.org/10.1186/s40537-022-00648-6

[21] T. Chakraborty, U. R. KS, S. M. Naik, M. Panja, and B. Manvitha, "Ten years of generative adversarial nets (GANs): A survey of the state-of-the-art," Mach. Learn. Sci. Technol., vol. 5, no. 1, p. 011001, Jan. 2024. https://doi.org/10.1088/2632-2153/ad1f77

[22] Goodfellow, et al., "Generative adversarial nets," Advances in Neural Information Processing Systems, pp. 2672-2680, 2014.

[23] V. Sampath, I. Maurtua, J. J. Aguilar Martin, and A. Gutierrez, "A survey on generative adversarial networks for imbalance problems in computer vision tasks," J. Big Data, vol. 8, pp. 1-59, 2021. https://doi.org/10.1186/s40537-021-00414-0

[24] M. Little, R. Allmendinger, and S. S. Samani, "Generative adversarial networks for synthetic data generation: A comparative study," arXiv preprint, arXiv:2112.01925, 2021.

[25] Y. Song, J. Hu, Q. Wu, F. Xu, S. Nie, Y. Zhao, and Z. Yi, "Automatic delineation of the clinical target volume and organs at risk by deep learning for rectal cancer postoperative radiotherapy," Radiother. Oncol., vol. 145, pp. 186-192, 2020. https://doi.org/10.1016/j.radonc.2020.01.020

[26] Y. Du, R. Yang, Z. Chen, L. Wang, X. Weng, and X. Liu, "A deep learning network-assisted bladder tumour recognition under cystoscopy based on Caffe deep learning framework and EasyDL platform," Int. J. Med. Robot. Comput. Assist. Surg., vol. 17, no. 1, pp. 1-8, 2021. https://doi.org/10.1002/rcs.2169

[27] E. N. AlShemmary, R. Razaq, and Z. Lu, "MGAN Algorithm," *Kaggle*, 2025. [Online]. Available: https://www.kaggle.com/m/279233. DOI: 10.34740/KAGGLE/M/279233.
[28] B. K. Hilal and E. N. AlShemmary, "Detecting Hypertrophic Cardiomyopathy: A Deep Learning Approach with CNNs and Swin Transformers," *Int. J. Intell. Eng. Syst.*, vol. 18, no. 4, Apr. 2025. DOI: 10.22266/ijies2025.0531.04
[29] O. S. Khedr, M. E. Wahed, A. S. R. Al-Attar, and E. A. Abdel-Rehim, "The classification of the bladder cancer based on Vision Transformers (ViT)," Scientific Reports, vol. 13, no. 1, p. 20639, 2023. https://doi.org/10.1038/s41598-023-47992-y
[30] J. F. Lazo, B. Rosa, M. Catellani, M. Fontana, F. A. Mistretta, G. Musi, et al., "Semi-supervised bladder tissue classification in multi-domain endoscopic images," IEEE Trans. Biomed. Eng., vol. 70, no. 10, pp. 2822-2833, 2023, doi: 10.1109/TBME.2023.2828390. https://doi.org/10.1109/TBME.2023.3265679
[31] K. R. Rangarajan and R. Purushothaman, "Disease classification in eggplant using pre-trained VGG16 and MSVM," Scientific Reports, vol. 10, no. 1, p. 2322, 2020. https://doi.org/10.1038/s41598-020-59108-x
[32] W. Wang, Y. Li, T. Zou, X. Wang, J. You, and Y. Luo, "A novel image classification approach via dense-MobileNet models," Mobile Inf. Syst., 2020. https://doi.org/10.1155/2020/7602384
[33] D. Chen, Wang, L. Hao, X. Liu, Y. Zeng, J. Chen, and G. Zhang, "Pulmonary image classification based on inception-v3 transfer learning model," IEEE Access, vol. 7, pp. 146533-146541, 2019. https://doi.org/10.1109/ACCESS.2019.2946000
[34] R. Alkhalidy and E. AlShemmary, "Accurate bladder cancer classification and prognosis with a hybrid vision transformer," Int. J. Intell. Eng. Syst., vol. 18, no. 1, Jan. 2025. DOI: 10.22266/ijies2025.0229.43