FMDB *Transactions on Sustainable Health Science Letters*

Innovative Methods for White Blood Cell Classification Using AOA-Based PWGAN-GP Model

B. Santosh Kumar[1,*], Rakesh Chandrashekar[2], Thirumalraj Karthikeyan[3], S. Venkatasubramanian[4], Sheila Agnes Vidot[5]

[1]Department of Computer Science and Engineering, New Horizon College of Engineering, Bengaluru, Karnataka, India.
[2]Department of Mechanical Engineering, New Horizon College of Engineering, Bengaluru, Karnataka, India.
[3]Department of Artificial Intelligence, Trichy Research Labs, Quest Technologies, Tiruchirappalli, Tamil Nadu, India.
[4]Department of Computer Science and Business Systems, Saranathan College of Engineering, Trichy, Tamil Nadu, India.
[5]Department of Medical Consultant, IPS Health, Mahe, Victoria, Seychelles.
skumars1803@gmail.com[1], rakesh2687@gmail.com[2], thirumalraj.k@gmail.com[3], veeyes@saranathan.ac.in[4], vidotsheila@gmail.com[5]

Abstract: White blood cells are needed to assess the immune system, but the pathologist's ability influences the blood smear's conclusion. Most machine learning methods classify white blood cells at a single level. White blood cells must be assessed to assess the human immune system, and this study provides a new method. This work improves classification accuracy by leveraging deep learning rather than pathologists' expertise. Preprocessing using Edge Image Enhancement reduces noise. Five datasets with four cell groupings are used. The innovative barnacles mating optimiser (BMO) selects features. The method's core is a progressive Wasserstein generative adversarial network with gradient penalisation (PWGAN-GP). Progressive training makes training the generator model easier. Built sample resolution is improved iteratively. A loss function is added to the discriminator to quantify sample similarity, thereby improving sample reliability and authenticity. The Aquila Optimiser Algorithm tunes the hyperparameters of the PWGAN-GP classification model. The proposed PWGAN-GP model outperformed others across all datasets, achieving 99.22% in Blood Cell Detection, 99.18% in Complete Blood Count, 98.7% in White Blood Cells, 98.82% in Kaggle Blood Cell Images, and 98.91% in Segmentation, as well as Classification.

Keywords: Feature Selection; Image Enhancement; Noise Reduction; Gradient Penalisation; Aquila Optimiser Algorithm; Hyperparameter Tuning; Barnacles Mating Optimiser; Cell Detection.

Received on: 02/01/2025, **Revised on:** 13/03/2025, **Accepted on:** 12/05/2025, **Published on:** 06/12/2025

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSHSL

DOI: https://doi.org/10.69888/FTSHSL.2025.000510

Cite as: B. S. Kumar, R. Chandrashekar, T. Karthikeyan, S. Venkatasubramanian, and S. A. Vidot, "Innovative Methods for White Blood Cell Classification Using AOA-Based PWGAN-GP Model," *FMDB Transactions on Sustainable Health Science Letters*, vol. 3, no. 4, pp. 202–217, 2025.

1. Introduction

[*]Corresponding author.

As a standard laboratory test, a peripheral blood smear can provide the doctor with valuable information about the patient's overall health. It measures the quantity and quality of blood components, namely platelets and cells. The two main types of cells are the WBC, also called leukocytes, and the red blood cells, also called RBCs [1]. In turn, five distinct types of nucleated cells are involved in disorders and chronic processes such as anaemia, leukaemia, and malnutrition; the total white blood cell count—defined as the percentage difference between the subtypes—provides crucial information [2]. An analyst can manually count white blood cells by examining a blood smear under a microscope. This allows them to distinguish between subtypes primarily by examining the morphological features of the cell nucleus and cytoplasm. But if the analyst isn't well-trained, this process could go awry because it is so dependent on their time and expertise [3]. Furthermore, clinical laboratories frequently experience significant demand for this haematological evaluation due to its routine nature, thereby increasing workload and impacting performance. This is why it's crucial to equip labs with computer-aided diagnosis (CAD) technologies to aid in diagnosis. For example, differential white blood cell count classification using image processing approaches has been implemented in CAD systems [4]; [5]. By reducing bias and inter-observer variability, this automated leukocyte classification enables faster, more repeatable results. However, researchers still face challenges when automating medical examination observation due to the inherent complexity of the process. Acquiring precision on par with that of an expert is of the utmost importance. When it comes to identifying and categorising these photographs, the automation isn't perfect.

These will have a diagnosis, which, in turn, will raise treatment costs and have a detrimental effect on patients' chances of recovery and survival [6]. The algorithm that detects and segments cells is the backbone of the computer-aided system that classifies white blood cells. By analysing images, it will determine what parts of the cell are important based on morphological factors [7]. Cell segmentation is often a challenging process when working with tissue samples. Leukocyte staining reveals a dark nucleus, making this task easy to accomplish in cell smears. Cell border delineation, overlapping cell separation, and acquisition-related noise and artefact removal provide the greatest obstacles [8]. Numerous approaches to leukocyte classification and segmentation have been investigated, given AI's benefits for image processing. Complex algorithms, such as deep learning (DL) models, are among these methods, which extend from simpler ones such as support vector machines [9]; [10]. When it comes to medical picture categorisation, convolutional neural networks (CNNs) from DL models have been the most effective [11]. Deep classification has two primary components: feed-forward neural networks. This model uses several convolutional and pooling layers, including linear layers that highlight the input's characteristics, to extract features [12]. In extracted features, a fully linked dense layer performs the classification phase. According to this architecture, numerous CNN models have been suggested for particular tasks, including classification (e.g., AlexNet, ResNet, VGG, and GoogLeNet) (e.g., fully connected, which have been successfully used to process images of blood smears for differential WBC counting [13].

1.1. Motivation

The paramount importance of precise white blood cell analysis in assessing the state of the human immune system is what drives this effort. The study aims to improve classification accuracy by utilising deep learning techniques, acknowledging the limits of standard methods that rely on pathologists' expertise [27]. The research aims to contribute to the field by presenting a new methodology that addresses the limitations of one-level classification approaches. Contributing to the improvement of immune system assessment techniques, the study aims to provide a comprehensive and effective solution for improving the precision of white blood cell classification [28]. This will be achieved through the integration of innovative preprocessing techniques, feature selection methods, and an enhancement generator model.

1.2. Main Contributions

- A thorough groundwork for analysis was laid by utilising five datasets that comprised four cell groups: lymphocytes, monocytes, segmented neutrophils, and eosinophils.
- Using the barnacles mating optimiser (BMO) to choose unique features and use preprocessing techniques like Edge Image Enhancement to reduce noise, both of which improve data quality and model performance.
- Third, researchers stabilised the generated samples and made it easier to iteratively improve resolution by building an enhancement generator model based on a PWGAN-GP.
- Improving the efficacy and resilience of the PWGAN-GP classification model through the incorporation of the Aquila Optimiser Algorithm (AOA) for hyperparameter tuning.

2. Related Works

Ahmad et al. [14] introduced a more effective hybrid method for WBC subtype classification. The first step was to use deep neural networks, such as Darknet53, to extract the best deep features from the improved, segmented WBC images. Next, a predator algorithm (ECMPA) was utilised to filter the serially fused feature vector. Based on principles found in nature, this meta-heuristic optimisation method kept the strongest traits and eliminated the weaker ones. Several kernel values were used to classify the summary feature vector. A publicly available dataset containing five distinct WBC subtypes and 5,000 synthetic

photos was used to validate the methods. With a feature vector size reduction of nearly 95%, the system attained an average accuracy of 99.9%. The feature selection method outperformed traditional meta-heuristic algorithms in terms of convergence performance. In an effort to overcome the limitations of manual diagnosis, Almurayziq et al. [15] used automated methods to categorise datasets of microscopic blood samples to detect illnesses in white blood cells (WBCs) at an early stage. Several suggested systems were implemented. Initially, a hybrid approach combining the grey-level co-occurrence matrix (GLCM) was used to analyse the dataset, with features extracted using neural network algorithms such as feed-forward neural networks (FFNNs) and artificial neural networks (ANNs). All algorithms improved the accuracy of WBC diagnosis. The second step was to use pre-trained convolutional neural network (CNN) models for early detection of WBC illness [29]. These models included AlexNet, ResNet-50, and ResNet-18. When it came to early diagnosis of WBC illnesses, all the models achieved remarkable results [30]. Third, a hybrid approach was used, combining two blocks: one for efficient, accurate deep feature extraction with CNNs, and another for deep feature classification with SVMs. Using support vector machines (SVMs), four hybrid approaches were developed: ResNet-50, GoogLeNet, ResNet-18, and AlexNet.

When analysing the information for early WBC illness identification, all methods showed encouraging results. The ResNet-50 model achieved an area under the curve of 99.99%, sensitivity of 99.25%, specificity of 99.75%, precision of 99.5%, and accuracy of 99.3%. Kannan et al. [16] employed a fully connected network to segment and categorise white blood cell (WBC) images obtained from a microscope, both in healthy and ill states. An analysis of the classifier's performance was conducted. In that order, 96.84% for accuracy, 96.26% for sensitivity, 97.35% for specificity, and 96.39% for precision. The study conducted by Bairaboina and Battula [17] utilised a robust deep learning model to classify white blood cells in peripheral blood smear images. To perform leukocyte segmentation, a network was constructed using W-Net, a convolutional neural network (CNN) for white blood cell (WBC) classification. The next step was to use a GhostNet to retrieve important feature maps. After that, a ResNeXt-based Wildebeest Herd Optimisation (WHO) approach was used to classify them. A data augmentation strategy based on Deep Convolutional Generative Adversarial Networks (DCGANs) was also active to address the issue of imbalanced data. Compared with existing methods, the proposed approach achieved 98.61% accuracy on the single-cell morphological dataset, 99.24% on the Blood Cell Count and Detection (BCCD) dataset, and 99.16% on the Classification (LISC) dataset, validating the model's performance. The suggested method is useful and flexible enough to be applied in clinical settings for microscopic investigation of blood cells. Dong et al. [18] presented a method for classifying white blood cells using ensemble CNNs. The diversity of base learners was initially enhanced by enlisting three high-performing CNNs with different structures: VGG16, ResNet50, and Inception V3.

After that, based on the base learner, the ensemble learning combination technique was designed using the Gompertz function. Finally, to classify white blood cells, an ensemble CNN model was created that combined the results of multiple individual models' learning processes and used diverse data. This study's ensemble learning technique showed promise for clinical use, enabling precise and reliable white blood cell classification, according to empirical results. Almurayziq et al. [15] set out to develop automated systems that could identify datasets of microscopic blood samples to detect disorders in white blood cells (WBCs) early, thereby eliminating the need for manual diagnosis. Several suggested systems were implemented. Initially, a hybrid approach combining the grey-level co-occurrence matrix (GLCM) was used to analyse the dataset, with features extracted using neural network algorithms such as feed-forward neural networks (FFNNs) and artificial neural networks (ANNs). All algorithms improved the accuracy of WBC diagnosis.

The second step was to use WBC illness early detection. These models included AlexNet, ResNet-50, and ResNet-18. When it came to early diagnosis of WBC illnesses, all the models achieved remarkable results. Third, a hybrid approach was used, combining two blocks: one for efficient, accurate deep feature extraction with CNNs, and another for deep feature classification with SVMs. Using support vector machines (SVMs), four hybrid approaches were developed: ResNet-50, GoogLeNet, ResNet-18, and AlexNet. When analysing the information for early WBC illness identification, all methods showed encouraging results. A method for white blood cell (WBC) categorisation was described by Saikia and Devi [19]. This method used a Grey Level Co-occurrence Matrix (GLCM), an ANOVA test, and Zero Phase whitening. In addition, the K-Nearest Neighbour (K-NN) classifier was used to assess the performance. On the dataset, the suggested method successfully classified four kinds of (WBCs)— with an accuracy of 94.25%.

2.1. Research Gaps

Automated systems for classifying white blood cell (WBC) images have come a long way; however, many questions remain unanswered. One issue is that there hasn't been enough research on how well deep learning models perform across different datasets and in actual clinical settings, despite extensive research on these methods. Secondly, while many methods aim to improve classification accuracy, most ignore other crucial aspects such as interpretability, computational efficiency, and scalability. To further aid in identifying the most effective strategies for WBC classification, more extensive benchmarking and comparative studies are required to assess the performance of different procedures across varied situations and datasets [20].

3. Proposed Procedure

Figure 1 shows the proposed workflow of the WBC classification models.

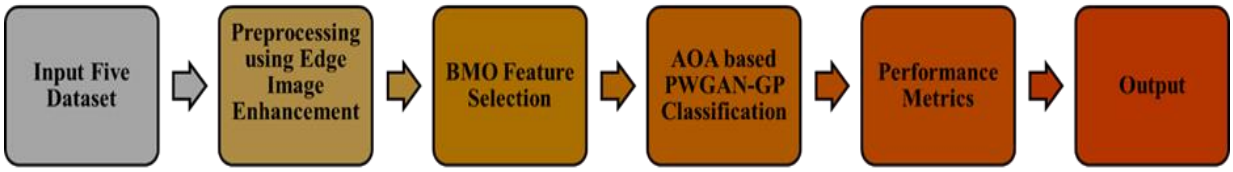

Figure 1: Block figure

3.1. Dataset Description

In this work, five diverse datasets were used [21]. The description of these sources shadows: Among the 100 annotated images in the Blood Cell Detection (BCD) dataset, 2237 have been labelled with cells. The images are in PNG format. Every picture has 256 RGB pixels across the width and height [31]. Annotation files and 360 ° images of blood smears comprise the Complete Blood Count (CBC) dataset [32]. The White Blood Cells (WBC) dataset includes 300 monochrome images and 100 colour images, with dimensions of 120×120 and 300×300, respectively [33]. Includes 12,500 enhanced JPEG images of blood cells with corresponding cell type labels (CSV) from the Kaggle Blood Cell Images (KBC) dataset. There are approximately 3,000 images of the four distinct cell types [34]. A collection of 250 BMP images of blood smears is known as the Leukocyte Images for Segmentation dataset. It has twenty-five pictures of basophils [35].

The amount of basophils was inadequate in the earlier datasets. Consequently, pictures of monocytes, lymphocytes, segmented neutrophils, and eosinophils were used to build the algorithm. To begin, researchers selected a subset of the KBC data set's images, used bounding boxes to identify and classify cells as either polymorphonuclear, or trained on 365 labelled images containing cell bounding boxes. A human expert classified the cell images as lymphocytes, monocytes, segmented neutrophils, or eosinophils using the Labelling tool. One approach involved training a CNN model on mononuclear cells to distinguish between lymphocytes and monocytes; 2282 images were used for this, while 2134 were used for the other approach. In contrast, a CNN model was trained on 2416 images for neutrophil segmentation and 2477 images for eosinophil segmentation.

3.2. Preprocessing Using Edge Image Enhancement

Edge detection filters applied in the spatial or frequency domain produce edge images E(x,y) with pixel intensities spanning a wide range, necessitating scaling [22]. The pixel intensities of the edges of an image, denoted as E(x,y), are transformed into linearly scaled intensities by applying the scaling Equation (1). $E_{sc}(x, y)$ such that $0 \leq I_{sc}(x, y) \leq 1$:

$$E_{sc}(x, y) = [E(x, y) - \min(E)] \left[\frac{1}{\max(E) - \min(E)}\right] \quad (1)$$

The scaled edge image $E_{sc}(x, y)$ needs contrast alteration to recover edge clearness. Equation (2) alters the scaled edge copy $E_{sc}(x, y)$ into copy $E_e(x, y)$, where $\mu_{E_{sc}}$ and $\sigma_{E_{sc}}$ are the means besides the standard nonconformity of, singly:

$$E_e(x, y) = [E_{sc}(x, y) - \min(E_{sc})] \left[\frac{2\sigma_{E_{sc}}}{\max(E_{sc}) - \min(E_{sc})}\right] + \mu_{E_{sc}} \quad (2)$$

3.3. Feature Selection Using BMO Algorithm

A feature selection algorithm called BMO is used in this paper. To improve accuracy and reduce computational complexity, the BMO procedure is used to extract a set of relevant features from the data [23]. A lot of things, like rocks, sea turtles, corals, and ships, have barnacles permanently attached to them. One distinctive trait of barnacles, which are hermaphroditic creatures with both male and female reproductive systems, is the enormous size of their penis, which can extend up to 7 or 8 times the length of the rest of their body. Barnacles reproduce by sperm-cast and conventional copulation. Isolated barnacles engage in sperm-casting during mating.

The process of releasing the fertilised eggs into water accomplishes this. How barnacles release new generations of their own kind sheds light on how to apply BMO to optimise optimisation problems. When developing the selection method for parents to breed to produce new offspring, BMO follows the same approach as other evolutionary methods, such as GA. Problematically, unlike the GA, the solution process doesn't use any of the usual selection methods, such as tournaments or

roulette wheels. The following guidelines can be followed to select mating barnacles: Even though barnacles are hermaphroditic, one or more male barnacles can fertilise a female barnacle:

- To avoid problems, it is recommended that all barnacles mate with each other. The user can set the initial value of pl and perform selective breeding at will. Along with the maximum iterations and sum of barnacles, the charge of pl is a control parameter in these procedures that can be tuned to achieve better optimisation outcomes.
- When the range of pl is used for parents, the Hardy-Weinberg code is applied. Next, the sperm-cast is used to produce unique progeny.

The Hardy-Weinberg principle dictates the following for the production of unique progeny:

$$x_i^{N\_new} = px_{barnacle\_m}^{N} + qx_{barnacle\_d}^{N} \text{ for } k \leq pl \quad (3)$$

$$x_i^{N\_new} = \text{rand}\,() \times x_{barnacle\_m}^{N} \text{ for } k > pl \quad (4)$$

Where $k = |\,\text{barnacle}\_m - b\,\text{arncle}\,d\,|, p$ designates the random sum consistently distributed, $q = (1 - p)$, and $x_{barnacle\_m\,m}^{N}$ and $x_{barnacle\_d}^{N}$ show parents correspondingly. In addition, the random integer ranges from 0 to 1, as indicated by rand (), which is equivalent to 0~1. The percentages passed down from one set of barnacles to another are represented by p and q, respectively. For example, p is set to 0.80. As a result, the novel's progeny take on 80% of their mother's traits and 20% of their father's. Equation is utilised for the exploration of the proposed BMO, while Equation (4) is utilised for optimisation. Another interesting point is that the mother's barnacles are the only ones that can be involved in the exploration (sperm-cast) since they receive sperm from another barnacle somewhere else. The population of barnacles will be double its initial size the moment it begins to reproduce. Something has to be done to rein in these expansions. In BMO, as in GA, a sorting method is necessary to place the top half of the population with the best results from a given iteration. The modelled BMO method uses the fitness function (FF) to strike a balance between the classifier's accuracy (maximum) and the number of solutions (minimal) that use the chosen feature. The FF for the computational solution is represented by Equation (5):

$$\text{Fitness} = \alpha\gamma_R(D) + \beta\frac{|R|}{|C|} \quad (5)$$

At this point, $\gamma_R(D)$ signifies the classifier error rate of a obtainable $|R|$ characterises the features in the data. α, besides β display the two variable stars for the implication length, correspondingly. $\in [1,0]$ and $\beta = 1 - \alpha$.

3.4. Classification Using AOA-PWGAN-GP

3.4.1. GAN

By learning the distribution of data from the training set, GANs can be trained to produce high-quality images. There are two main components of a GAN: the generator (G) and the discriminator (D). After receiving the noise vector, the generator begins to produce samples. After that, it merges the created samples with the actual samples to feed into the discriminator, which must correctly differentiate between the two. As the two models compete, more accurate samples will be produced. Concurrently, the discriminator's capacity to discriminate will be amplified. When both the generator and the discriminator have played their cards well, they will achieve a Nash equilibrium [24]. You may utilise the GAN-generated samples to expand your image dataset, as they are in the same labelled class as the original samples. The equation shows the GAN's goal function (6):

$$\min_G \max_D V(D,G) = E_{x\sim p_{data\,(x)}}[\log(D(x))] + E_{z\sim p_\theta(z)}[\log(1 - D(G(z)))] \quad (6)$$

where $p_{data(x)}$ is the input noise equal to the probability distribution of the actual picture. G and D are in constant competition; G is working to improve its sample distribution capture and picture generation capabilities, while D is working to improve its image discrimination capabilities. Among the tested image generation algorithms, the original GAN produced the most lifelike results. But the original GAN had three major flaws: one, the generated images aren't stable because the values of the discriminative and generative models aren't stable during training. Two, the original GAN architecture isn't stable due to pattern collapse, which occurs when the data and the discriminator can't be effectively trained. Also, the generator doesn't produce diverse pictures, and (3) it isn't easy to get the typical GAN model to converge when you tweak its hyperparameters.

3.4.2. WGAN

One alternative to Jensen-Shannon that has been proposed to enhance GAN image quality is the Wasserstein distance metric, which compares the generated data to the true data distribution. Equation (7) shows the distance:

$$W(P_r, P_g) = \inf_{\gamma \in \prod (P_r, P_g)} E_{(x,y)\sim\gamma}[\| x - y \|] \quad (7)$$

where P_r is the data, P_g is the delivery of the produced data, and $\gamma \in \Pi(P_r, P_g)$ is the joint delivery of P_r and P_g. The loss meaning of exposed in Equation (8):

$$L(D) = E_{z\sim P_z}[f_w(G(z))] - E_{x\sim P_x}[f_w(x)] \quad (8)$$

Where z is the input noise and $E_{z\sim P_z}$ designates the likelihood distribution of the noise, $E_{x\sim P_r}$ denotes the probability image. f_w means that the WGAN discriminator neural network includes the w parameter. By using gradient clipping, also known as weight clipping, the discriminator can limit the neural network's parameter updates w and satisfy the Lipschitz condition. f_w to be in a certain range $[-c, c]$. Instead of directly comparing the two samples, the WGAN discriminator computes the Wasserstein distance to measure their difference. As a result, the generated sample becomes closer to the true distribution as the loss function value decreases, and the Wasserstein distance between the two samples approaches zero. Nevertheless, when using gradient clipping in WGAN, there is a possibility that the range. This might result in a variety of side effects, such as gradient explosion or disappearance, as well as an inappropriate generation process that includes the samples.

3.4.3. WGAN-GP

To address this issue, the WGAN-GP model was suggested; it uses a gradient penalty to teach the model to learn more refined decision boundaries, and its gradient penalty satisfies the Lipschitz requirement. In Equation (9), researchers can see the WGAN-GP loss function:

$$L(D) = \mathop{\mathbb{E}}_{\tilde{x}\sim P_g}[D(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim P_r}[D(x)] + \lambda \mathop{\mathbb{E}}_{\hat{x}\sim P_{\hat{x}}}[(\| \nabla \hat{x} D(\hat{x}) \|_2 - 1)^2] \quad (9)$$

where $\mathop{\mathbb{E}}_{\tilde{x}\sim P_g}[D(\tilde{x})] - \mathop{\mathbb{E}}_{x\sim P_r}[D(x)]$ is the purpose of WGAN, $\tilde{x} \sim P_g$ is the specimen of the generated data, besides $x \sim P_r$ is the sample data. $\lambda \mathop{\mathbb{E}}_{\hat{x}\sim P_{\hat{x}}}[(\| \nabla \hat{x} D(\hat{x}) \|_2 - 1)^2]$ is the incline penalty term, $\hat{x}$ is therein, $\hat{x} \leftarrow \varepsilon x + (1 - \varepsilon)\tilde{x}$ with accidental statistics $\varepsilon \sim U[0,1]$.

3.4.4. Progressive Training

With conventional GAN training complete, the discriminative model's structure remains unchanged, and the model's output images have a fixed resolution. Training the model becomes much more challenging as the resolution of the generated images increases, thanks to the 'zero-sum game' property of GANs. An approach to training that builds on itself was suggested: start with low resolution and gradually increase the model assembly. Once layers are added to the models, they will form an image. The training process is accelerated and stabilised, therefore, leading to the production of pictures of excellent clarity and quality.

3.4.5. Residual Block

The model's presentation is overly pretentious, given the depth of the neural network; typically, deeper networks perform better. The accuracy may reach a peak and then decline as the network gets deeper; this is known as gradient degradation. The ResNet model's essential component is the residual block, which was suggested. The residual block employs a mapping that sends inputs directly to outputs, producing features that pass through, allowing subsequent layers to influence them and ensuring steady network performance.

3.4.6. Progressive WGAN-GP

The generator and discriminator are the two primary components of the Progressive WGAN-GP (PWGAN-GP) concept. An upsampling layer, a layer, and a residual block make up the generator. Picture feature creation occurs in the generation model's residual block, whereas picture size scaling occurs in the upsampling layer. A downsampling module plus a residual module

make up the discriminant model. Picture feature extraction happens in the discriminator's residual module, while the downsampling layer shrinks the picture to a more manageable size. This uses the WGAN-GP loss function. The model's outstanding block layer in the discriminator and generator grows in size throughout training, as do the samples it generates. The training process begins by creating a low-resolution 4×4 version of the target picture. When the loss function's value reaches a stable state, the training is finished. The next step in continuing training is to add one layer to both the production and discriminant models' structures. The goal picture resolution of 256×256 is achieved by repeating this process. Since each training phase adds a new layer, it is not possible to immediately include the newly trained limits in the training process while the layer is still in its initialisation stage. This work describes a fusion method that incorporates the parameters of the previous layer into those of the new layer. Equation (10) displays the formula:

$$\text{Output } = \alpha \times L_{new} + (1 - \alpha) \times L_{old} \quad (10)$$

Output layer, α is the factor, L_{new} is the limit of the new layer, and L_{old} is a layer. The model factors in both the previous layer's parameters, multiplied by (α-1), and the new layer's parameters, multiplied by α. With each additional training, the value increases from 0 to 1. α can take values between 0 and 1.

3.4.6.1. Residual Block

A convolutional layer is necessary for the residual block to distinguish between the original and produced images by extracting input information. In determining the feature map, the layer integrates the activation function with the convolutional kernel. Algorithms (11) and (12) display the statistical description:

$$y_j^l = f(z_j^l) \quad (11)$$

$$z_j^l = \sum_{i \in M_j} x_i^{l-1} \times k_{ij}^l + b_j^l \quad (12)$$

where z_j^i is the production of the chin in the l-th layer, $f(\cdot)$ is the Relu purpose, z_j^l is the heaviness value of the j-th station in the l-th layer, x_i^{l-1} is the chin map of the $(l-1)$-th layer, M_j is the subset of map, and k_{ij}^l is layer l, * income the Difficulty action, b_j^i means the bias term. A two-layer residual block of the same design is utilised in this study. Batch normalisation and the LeakyReLU activation layer are part of it.

3.4.6.2. Upsampling

Here, the Upsampling layer employs the deep learning technique known as Transposed Convolution, which is essentially the inverse of convolution. Instead of getting the original values, this method may get the picture size and then move the space. The output and convolution kernel sizes determine the accuracy of Transposed Convolution. The output tensor size can be determined using the method described in Equation (13):

$$o' = i' + (k - 1) - 2p \quad (13)$$

Where o^' stands for the Transposed Convolution's output size, i^' for the input Transposed Convolution's size, k for the kernel's size, and p for the tensor's padding size.

3.4.6.3. Batch Normalisation Layer

In deep learning, batch normalisation stabilises neural network training and improves efficiency. When the input distribution to a layer changes during training, a problem known as internal covariate shift develops. Batch normalisation aims to alleviate this issue. A sluggish convergence or complete inability to converge may result from this. Batch Normalisation speeds up training by standardising inputs to each layer, thereby reducing internal covariate shift. The Batch Normalisation computation formula is given in Equations (14)– (17):

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m} \sum_{i=1}^{m} x_i \quad (14)$$

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m} (\sum_{i=1}^{m} x_i - \mu_{\mathcal{B}})^2 \quad (15)$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}} \quad (16)$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \quad (17)$$

Where m is the set batch size, x_i attitudes for the data of each batch, and $\mu_{\mathcal{B}}$ characterises the mini-batch mean. $\sigma_{\mathcal{B}}^2$ means $\hat{x}_i$ designates regularised. y_i reflects the Normalising coefficient, besides β is the bias term.

3.4.6.4. LeakyReLU

To train a neural network to fit different curves, one must first introduce nonlinear elements into the network using the activation function. Without the function, the output of every layer will be directly proportional to the input of the layer before it. The output can approximate any function by using a nonlinear activation function. One activation function that addresses the Dead ReLU issue is LeakyReLU. Equation (18) shows the mathematical explanation:

$$\text{LeakyReLU} = \begin{cases} x, x > 0 \\ \alpha x, x \leq 0 \end{cases} \quad (18)$$

For negative values, the LeakyReLU multiplies by 0.01. Typically, the value is 0.01. It can do operations between negative and positive infinity.

3.4.7. Aquila Optimiser

This research presents a PWGAN-GP model that uses AOA for hyperparameter tuning. The Aquila hunting behaviour inspired the Aquila Optimiser (AO), a metaheuristic optimisation procedure [25]. Aquila birds model their hunting style after the four main methods used by aquilas: they use their height to their advantage by swooping down vertically to catch prey floating in the air. When pursuing seabirds, aquila birds fly low to the ground in a contour-like pattern and attack with speed and glide-like manoeuvres. This is how aquila birds catch their prey, which includes foxes and other slow-moving animals. The avian hawk uses this strategy to pounce on its prey as it flies overhead.

3.4.7.1. Expanded Exploration

The Aquila Optimiser algorithm mimics the hunting technique of aquila birds by reaching enormous heights and then descending quickly, as seen in its expanded exploration u_1. The bird swoops down from high heights, surveying its hunting territory for potential prey and deciding where to set up shop. Equation (19) provides a mathematical depiction of this strategy:

$$u_1^{(h+1)} = u_{best}^{(h)} \times \left(1 - \frac{h}{H}\right) + \left(u_M^{(h)} - \text{ rand } \times u_{best}^{(h)}\right) \quad (19)$$

H is the maximum number of iterations in the algorithm, and h is the current iteration. Starting with a pool of possible solutions (u_1) yields the answer for the subsequent iteration, denoted as $\left(u_1^{(h+1)}\right)$. The appearance $\left(u_{best}^{(h)}\right)$ represents the best consequence up to the h^{th} equation $\left(1 - \frac{h}{H}\right)$an iteration count is used to fine-tune the depth of the search space. Additionally, the average value of keys at the h^"th" repetition is determined using Equation (20), where N is the population size, and D is the dimension size. $u_M^{(h)}$:

$$u_M^{(h)} = \frac{1}{N}\sum_{i=1}^{N} u_i(h), \text{ for all } j = 1,2, \ldots, D \quad (20)$$

3.4.7.2. Narrowed Exploration

Similar to how eagles hunt, the Aquila Optimiser procedure's limited exploration approach (u_2) involves fast, gliding assaults within a compact search region to seek prey. Finding a solution is the primary goal of this strategy, as shown mathematically in Equation (21). $\left(u_2^{(h+1)}\right)$ for the subsequent repetitions, signified as h:

$$u_2^{(h+1)} = u_{best}^{(h)} \times \text{Lev}(D) + u_R^{(h)} + (v - u) \times \text{ rand} \quad (21)$$

The Levy flying delivery for space, D, is the Aquila technique. In the range $[1, N]$, where N represents the population size, the accidental solution $\left(u_R^{(h)}\right)$ is found during the h^"th" repetition. The Levy flight distribution is determined by a fixed constant s, usually set to 0.01 [26]. The parameters u and v are randomly generated and can take values between 0 and 1. This computation can be expressed mathematically in Equation (22):

$$\text{Levy}(D) = s \times \frac{u \times \sigma}{|v|^{\frac{1}{\alpha}}} \tag{22}$$

The value of s is obtained from Equation (23) by setting a constant parameter a equal to 1.5:

$$\sigma = \left(\frac{\gamma(1+a) \times \sin\left(\frac{\pi a}{2}\right)}{\gamma\left(\frac{(1+a)}{2}\right) \times a \times 2^{\left(\frac{a-1}{2}\right)}} \right) \tag{23}$$

Equations (24) and (25), correspondingly, depict the spiral forms that fall inside the search range, denoted as v and u, respectively. You may find the details of these spiral forms in Equation (21):

$$v = r_1 + UD_1 \cos\left(-\omega D_1 + \left(\frac{3\pi}{2}\right)\right) \tag{24}$$

$$u = r_1 + UD_1 \sin\left(-\omega D_1 + \left(\frac{3\pi}{2}\right)\right) \tag{25}$$

The range of possible values for the variable r_1 over a set number of search repetitions is (1,20). There is no change to the values of W (0.005) and U (0.00565). $D_1 \in \mathbb{Z}$ has a range of D.

3.4.7.3. Expanded Exploitation

As part of its research phase, an avian hawk flies low and slowly descends on its prey after carefully surveying its whereabouts. An equation depicting this strategy, often called enlarged exploitation, is (26) in mathematics. u_3:

$$u_3^{(h+1)} = \left(u_{best}^h - u_M^{(h)}\right) \times \theta - \text{ rand } + \left((ub - lb) \times \text{ rand } + lb\right) \times \rho \tag{26}$$

Equation (8) has consequences in $\left(u_3^{(h+1)}\right)$, which signifies the repetition. In the h^{th} iteration, $\left(u_M^{(h)}\right)$ stands for the average value of Equation (20), and u_{best} (h) portrays the optimal solution discovered thus far. Typically, θ and ρ, the tuning parameters, are set to 0.1 each, whereas the random integer is drawn from 0 to 1. Both bounds are represented by the symbols ub and lb, respectively.

3.4.7.4. Narrowed Exploitation

As a hunting tactic, aquila birds take advantage of their prey's unpredictable flight patterns when they're on the ground to bring them down to size. Formula (27), which produces the h^{th} iteration indicated as $\left(u_4^{(h+1)}\right)$, uses this technique $\left(u_4^{(h)}\right)$ design. An excellent function specified in Equation (28) was a projected approach:

$$u_4^{(h+1)} = J \times u_{best}^{(h)} - \left(P_1 \times \text{ rand } \times u_1^{(h)}\right) - P_2 \times \text{Levy}(D) + \text{ rand } \times P_1 \tag{27}$$

Using equations (11) and (12), researchers can determine the path of an attack throughout a flight from its starting point to its destination (P2), as well as the pattern of motion for the Aquila bird's prey tracking (P1). The iteration counter (h) and the maximum iteration count (H) are used to execute the computations:

$$J(h) = h^{\frac{2 \times \text{ rand }(\ ) - 1}{(1-H)^2}} \tag{28}$$

$$P_1 = 2 \times \text{ rand } - 1 \tag{29}$$

$$P_2 = 2 \times \left(1 - \frac{h}{H}\right) \tag{30}$$

4. Results and Discussion

The efficacy of the DL models was evaluated using WEKA 3.8.6. The WEKA data mining programme is under the GNU General Public Licence. Data preparation, visualisation, and more are just some of the many features it offers alongside its extensive model library. The subsequent parameter of an individual system was used to perform the assessment:

- **Processor:** Intel(R) Core (TM) i7-97250H CPU @ 2.60 GHz
- **Memory:** 16 GB
- CPU based on the x64 instruction set.

4.1. Performance Metrics

The success of the proposed work is measured by the level of acceptance it receives. The proportion of WBC in the data set that was correctly identified is indicative of the precision. It reads as follows:

$$\text{Accuracy (ACC)} = \frac{\text{No.of correctly expressions}}{\text{Total no.of images}} \times 100 \quad (31)$$

$$\text{precision (PR)} = \frac{\text{TP}}{\text{TP+FP}} \times 100 \quad (32)$$

$$\text{F1} - \text{score (F1)} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision+Recall}} \times 100 \quad (33)$$

$$\text{Recall (RC)} = \frac{\text{TP}}{\text{TP+FN}} \times 100 \quad (34)$$

$$\text{Specificity (SP)} = \frac{\text{TN}}{\text{TN+FP}} \times 100 \quad (35)$$

4.2. Classification Validation

Table 1-5 presents the classification results for the five datasets using the proposed PWGAN-GP model. Table 1 and Figure 2 present the classification investigation results of a Blood Cell Detection (BCD) dataset using various models. The performance metrics, including accuracy (ACC), precision, and specificity (SP), are reported for each model.

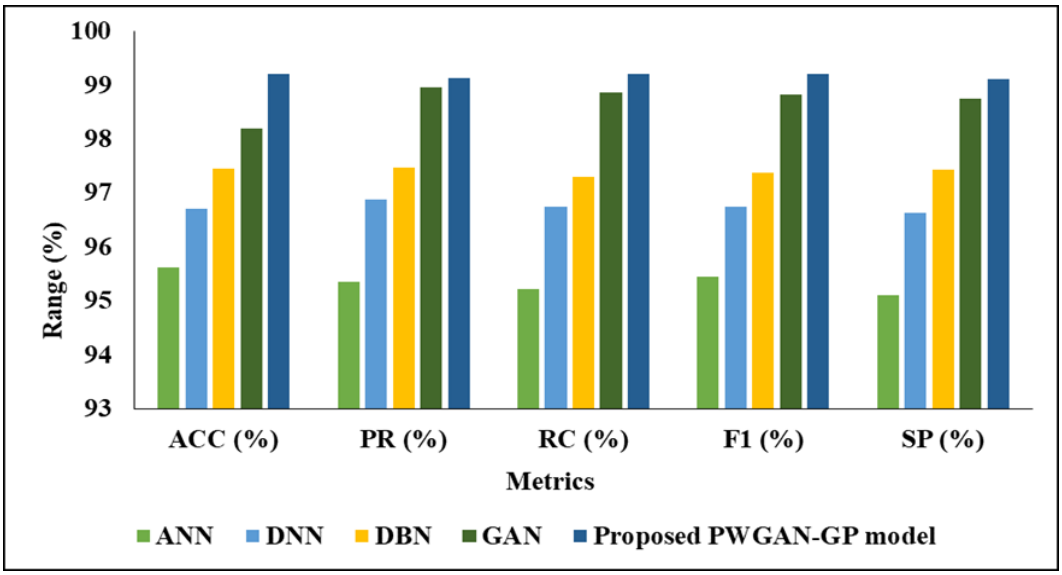

Figure 2: BCD dataset validation

The Artificial Neural Network (ANN) accomplished an accuracy of 95.62%, with specificity values of 95.35%, 95.23%, 95.46%, and 95.12% respectively. The Deep Neural Network (DNN) exhibited improved performance, achieving an accuracy of 96.72% and precision, recall, F1-score, and specificity of 96.88%, 96.75%, 96.76%, and 96.64%, respectively.

Table 1: Classification investigation of a blood cell detection (BCD) dataset

Models	ACC (%)	PR (%)	RC (%)	F1 (%)	SP (%)
ANN	95.62	95.35	95.23	95.46	95.12
DNN	96.72	96.88	96.75	96.76	96.64
DBN	97.45	97.47	97.31	97.39	97.44
GAN	98.21	98.97	98.87	98.83	98.76
Proposed PWGAN-GP model	99.22	99.14	99.21	99.21	99.11

The Deep Belief Network (DBN) demonstrated further improvement, achieving an accuracy of 97.45% and specificity values of 97.47%, 97.31%, 97.39%, and 97.44%, respectively. Notably, the Generative Adversarial Network (GAN) achieved impressive results with 98.21% accuracy, 98.97% precision, 98.87% recall, 98.83% F1-score, and 98.76% specificity. However, the proposed PWGAN-GP model outperformed all others, attaining the highest accuracy of 99.22%, precision of 99.14%, recall of 99.21%, F1-score of 99.21%, and specificity of 99.11%. These consequences underscore the efficacy of the proposed PWGAN-GP model in blood cell detection, surpassing existing methods in accuracy and performance across all evaluated metrics.

Table 2: Classification investigation of a complete blood count (CBC) dataset

Models	ACC (%)	PR (%)	RC (%)	F1 (%)	SP (%)
ANN	95.34	95.23	95.62	95.46	95.14
DNN	96.89	96.75	96.73	96.76	96.66
DBN	97.46	97.33	97.45	97.39	97.42
GAN	98.98	98.87	98.21	98.82	98.75
Proposed PWGAN-GP model	99.18	99.21	99.12	99.21	99.19

Table 2 and Figure 3 showcase the results of a classification investigation conducted on a Complete Blood Count (CBC) dataset using various models. Each model's presentation is evaluated. The ANN achieved accuracies of 95.34% and 95.23% for F1-score and specificity, respectively.

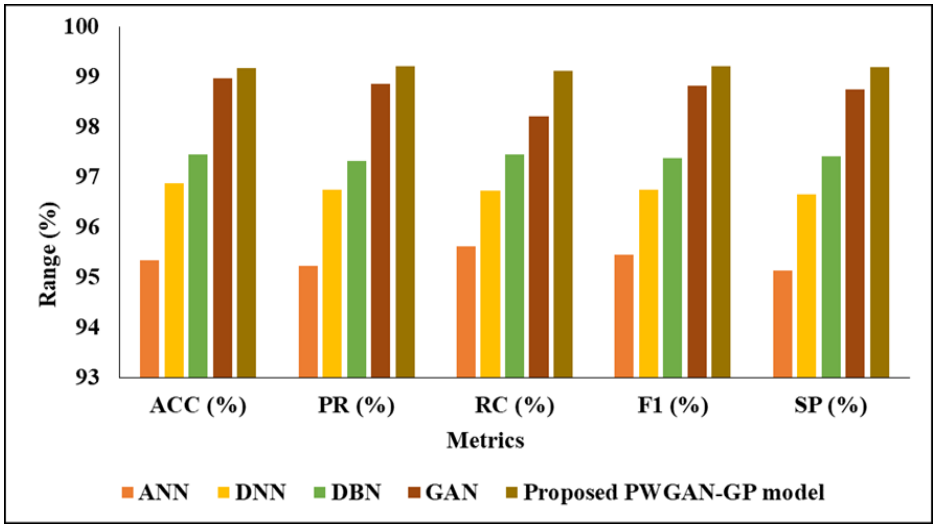

Figure 3: CBC dataset validation

The Deep Neural Network (DNN) demonstrated improved performance, achieving an accuracy of 96.89% and F1-scores of 96.75%, 96.73%, 96.76%, and 96.66%, respectively. The Deep Belief Network (DBN) showed further improvement, achieving an accuracy of 97.46%, an F1-score of 97.45%, and specificity values of 97.33%, 97.45%, 97.39%, and 97.42%, respectively. The Generative Adversarial Network (GAN) yielded impressive results, achieving 98.98% accuracy, 98.87% precision, 98.21% recall, 98.82% F1-score, and 98.75% specificity. Notably, the proposed PWGAN-GP model outperformed all others, achieving the highest accuracy of 99.18%, precision of 99.21%, recall of 99.12%, F1-score of 99.21%, and specificity of 99.19%. These findings underscore the effectiveness of the PWGAN-GP model for classification tasks, particularly on CBC datasets, where it achieves superior performance across all evaluated metrics.

Table 3: Classification of white blood cells (WBC) dataset

Models	ACC (%)	PR (%)	RC (%)	F1 (%)	SP (%)
ANN	93.8	86.2	86.7	86.3	86.4
DNN	94.4	87.5	87.3	87.4	87.7
DBN	95.9	88.6	88.9	88.2	88.7
GAN	96.9	90.4	91.4	89.8	89.6
Proposed PWGAN-GP model	98.7	98.6	98.1	98.5	98.7

Table 3, along with Figure 4, presents the consequences of classifying a WBC dataset using various models. Each model's presentation is evaluated based on accuracy. They achieved accuracy of 93.8%, F1-scores of 86.2%, 86.7%, 86.3%, and 86.4%, and specificities of 86.2%, 86.7%, 86.3%, and 86.4%, respectively. The Deep Neural Network (DNN) exhibited improved performance, achieving an accuracy of 94.4% and F1-scores of 87.5%, 87.3%, 87.4%, and 87.7%, respectively. The Deep Belief Network (DBN) demonstrated further improvement, achieving 95.9% accuracy and 88.7% accuracy, respectively.

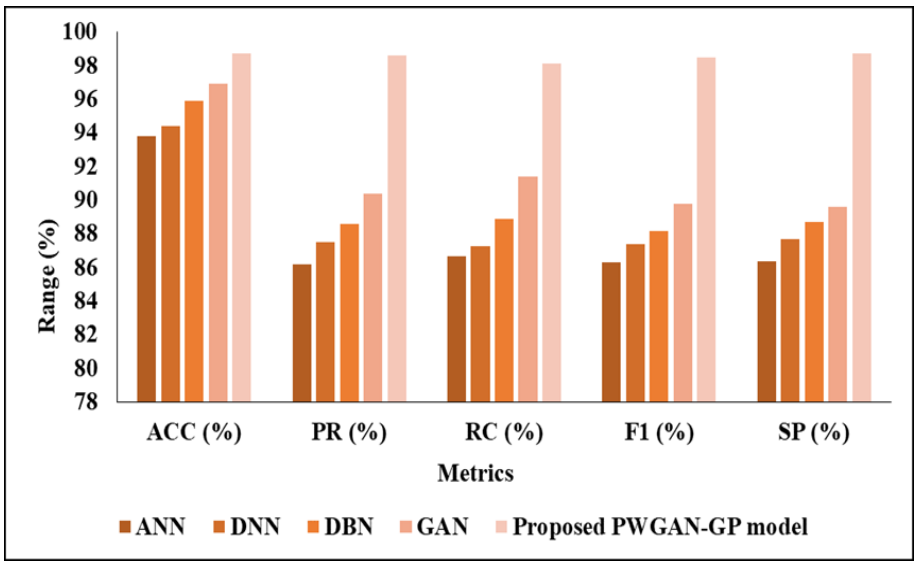

Figure 4: WBC dataset validation

The Generative Adversarial Network (GAN) yielded impressive results, achieving 96.9% accuracy, 90.4% precision, 91.4% recall, 89.8% F1-score, and 89.6% specificity. Notably, the proposed PWGAN-GP model outperformed all others, achieving the highest accuracy of 98.7%, exactness of 98.6%, recall of 98.1%, F1-score of 98.5%, and specificity of 98.7%. These findings highlight the efficiency of the PWGAN-GP model in precisely classifying white blood cells, showcasing superior performance across all evaluated metrics.

Table 4: Classification of the Kaggle blood cell images (KBC) dataset

Models	ACC (%)	PR (%)	RC (%)	F1 (%)	SP (%)
ANN	96.82	96.91	96.71	96.72	96.73
DNN	96.93	97.22	97.52	97.43	96.32
DBN	97.41	97.92	97.71	97.45	97.61
GAN	97.62	98.13	98.03	97.63	97.94
Proposed PWGAN-GP model	98.82	98.44	98.32	98.41	98.52

Table 4 and Figure 5 outline the outcomes of the classification task performed on the Kaggle Blood Cell Images (KBC) dataset employing various models. Each model's efficacy is evaluated across key metrics (F1) and specificity (SP). The Artificial Neural Network (ANN) achieved accuracies of 96.82%, 96.72%, and 96.73% for precision, recall, and F1, respectively. The Deep Neural Network (DNN) exhibited a similar performance, with an accuracy of 96.93% and precision, recall, F1-score, and specificity of 97.22%, 97.52%, 97.43%, and 96.32%, respectively. The Deep Belief Network (DBN) demonstrated further improvement, achieving an accuracy of 97.41% along with precision, recall, F1-score, and specificity values of 97.92%, 97.71%, 97.45%, and 97.61% respectively.

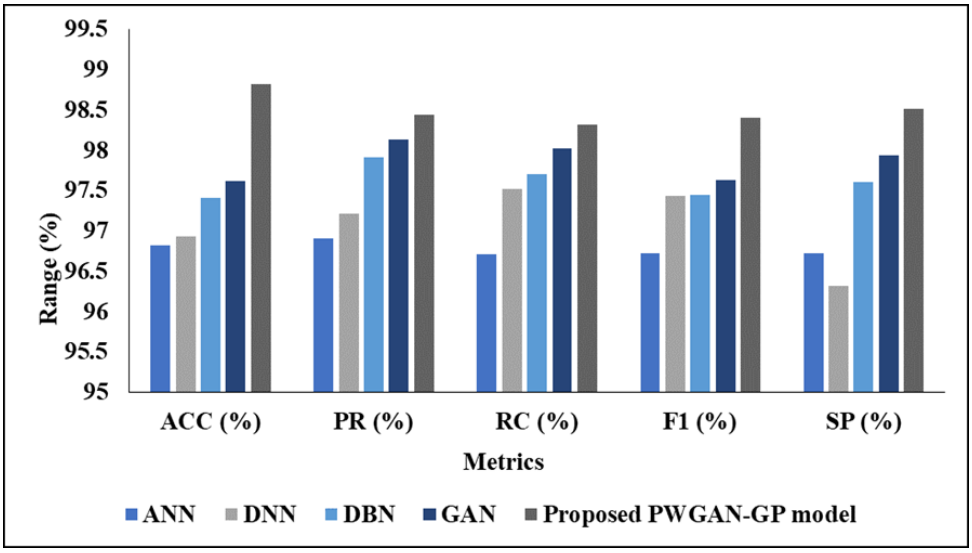

Figure 5: KBC validation

The Generative Adversarial Network (GAN) yielded commendable results, attaining an accuracy of 97.62% and specificity values of 98.13%, 98.03%, 97.63%, and 97.94%. Notably, the proposed PWGAN-GP model outperformed all others, achieving the highest accuracy of 98.82% and specificity values of 98.44%, 98.32%, 98.41%, and 98.52% respectively. These results underscore the efficiency of the PWGAN-GP, which precisely orders blood cell images and demonstrates superior performance across all evaluated metrics.

Table 5: Classification analysis of leukocyte images for segmentation, besides the classification (LISC) dataset

Models	ACC (%)	PR (%)	RC (%)	F1 (%)	SP (%)
ANN	87.30	90.3	90.56	88.50	91.50
DNN	90.67	93.45	91.67	91.39	92.67
DBN	92.34	95.78	94.78	92.39	96.27
GAN	95.43	95.12	95.09	95.23	95.22
Proposed PWGAN-GP model	98.91	97.02	98.05	97.20	98.73

Table 5 and Figure 6 illustrate the consequences of the classification analysis conducted on the Leukocyte Images for Segmentation dataset utilising various models.

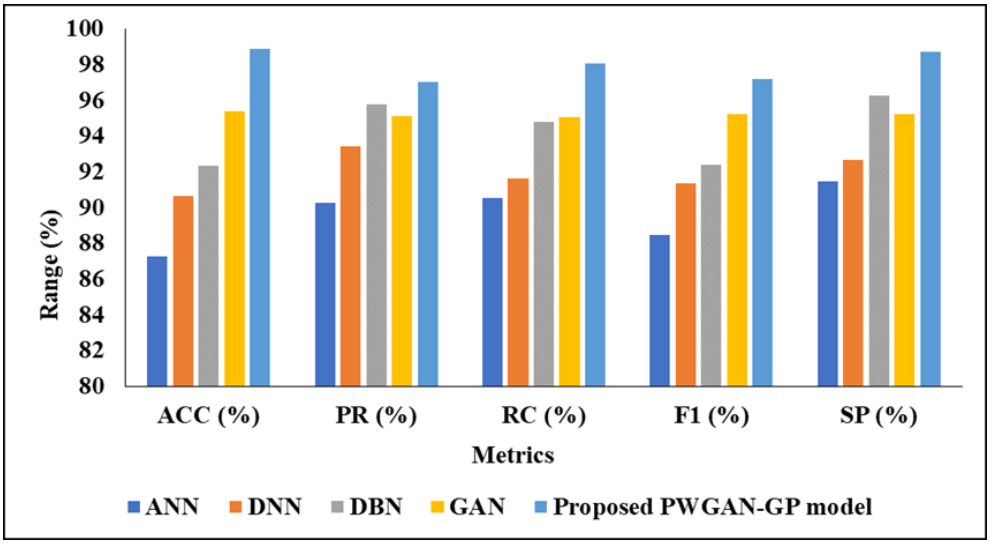

Figure 6: LISC validation

Each model's performance is assessed across multiple metrics. The ANN achieved an accuracy of 87.30%, with F1-scores of 90.3%, 90.56%, 88.50%, and 91.50%, respectively. The Deep Neural Network (DNN) demonstrated improved performance, with accuracies of 90.67%, 91.67%, and 91.39%, and F1-scores of 91.67% and 92.67%, respectively. The Deep Belief Network (DBN) demonstrated further improvement, achieving an accuracy of 92.34%, along with F1-Scores and specificities of 95.78%, 94.78%, 92.39%, and 96.27%, respectively. The Generative Adversarial Network (GAN) yielded notable results, attaining accuracies of 95.43%, 95.23%, and 95.22%, respectively. Remarkably, the proposed PWGAN-GP model outperformed all others, achieving the highest accuracy of 98.91% and specificity values of 97.02%, 98.05%, 97.20%, and 98.73%, respectively. These discoveries underscore the efficacy of the PWGAN-GP in accurately classifying leukocyte images, demonstrating superior performance across all evaluated metrics.

5. Conclusion

To sum up, this research introduces a revolutionary approach to determining the value of white blood cells (WBCs), which are an important part of evaluating the human immune system. This study uses advanced deep learning algorithms to greatly enhance classification accuracy and resilience, unlike typical diagnostic methods that rely heavily on expert pathologists. The study employs five extensive datasets encompassing lymphocytes, monocytes, segmented neutrophils, and eosinophils, ensuring a varied and representative assortment of WBC classifications. To improve data quality, new preprocessing methods were used, including Edge Image Enhancement to remove noise and the BMO methodology for comprehensive feature extraction. The suggested system's main part is an improvement generator model based on a liberal Wasserstein approach (PWGAN-GP). This model uses progressive training to improve image resolution gradually. Its well-designed loss function also makes the samples more realistic and stable during training. The PWGAN-GP architecture includes the Aquila Optimiser Algorithm (AOA) to strengthen the classification framework further. This makes optimisation more efficient, and the model works better. The new PWGAN-GP model consistently outperformed existing approaches across all five datasets, attaining outstanding accuracies of 99.22% for BCD, 99.18% for CBC, 98.7% for WBC, 98.82% for KBC Cell Images, and 98.91% for LISC. These results clearly show that the method is superior and can provide highly reliable WBC categorisation across a wide range of situations. In the future, research will focus on improving deep learning architectures, finding additional clinical applications for them, and making model outputs easier to interpret so they can support real-world medical decisions. This work not only improves diagnostic accuracy by advancing automated WBC analysis but also sets the stage for faster, easier immune system testing across all healthcare settings.

Acknowledgement: The authors sincerely acknowledge the support and contributions of New Horizon College of Engineering, Quest Technologies, Saranathan College of Engineering, and IPS Health in the successful completion of this research. Their guidance and resources were invaluable to this study.

Data Availability Statement: Data supporting this work may be obtained from the corresponding authors upon reasonable request and with adherence to relevant ethical and institutional regulations.

Funding Statement: The authors confirm that no external funding or financial assistance was received for conducting this research or preparing this manuscript.

Conflicts of Interest Statement: The authors declare that they have no known competing financial interests or personal relationships that could have influenced the work reported in this study.

Ethics and Consent Statement: This research was conducted in compliance with ethical standards and approved by the appropriate institutional ethics committee or review board. Informed consent was obtained from all participants prior to their involvement in the study.

References

1. A. M. Patil, M. D. Patil, and G. K. Birajdar, “White blood cells image classification using deep learning with canonical correlation analysis,” *IRBM*, vol. 42, no. 5, pp. 378–389, 2021.
2. H. Kutlu, E. Avci, and F. Özyurt, “White blood cells detection and classification based on regional convolutional neural networks,” *Medical Hypotheses*, vol. 135, no. 2, p. 109472, 2020.
3. Y. Y. Baydilli and Ü. Atila, “Classification of white blood cells using capsule networks,” *Computerized Medical Imaging and Graphics*, vol. 80, no. 3, p. 101699, 2020.
4. M. Toğaçar, B. Ergen, and Z. Cömert, “Classification of white blood cells using deep features obtained from Convolutional Neural Network models based on the combination of feature selection methods,” *Applied Soft Computing*, vol. 97, no. 12, p. 106810, 2020.

5. A. T. Sahlol, P. Kollmannsberger, and A. A. Ewees, "Efficient classification of white blood cell leukemia with improved swarm optimization of deep features," *Scientific Reports*, vol. 10, no. 1, p. 2536, 2020.
6. K. Almezhghwi and S. Serte, "Improved classification of white blood cells with the generative adversarial network and deep convolutional neural network," *Computational Intelligence and Neuroscience*, vol. 2020, no. 1, p. 6490479, 2020.
7. D. Ryu, J. Kim, D. Lim, H. S. Min, I. Y. Yoo, D. Cho, and Y. Park, "Label-free white blood cell classification using refractive index tomography and deep learning," *BME Frontiers*, vol. 2021, no. 7, p. 9893804, 2021.
8. S. Tavakoli, A. Ghaffari, Z. M. Kouzehkanan, and R. Hosseini, "New segmentation and feature extraction algorithm for classification of white blood cells in peripheral smear images," *Scientific Reports*, vol. 11, no. 1, p. 19428, 2021.
9. H. Chen, J. Liu, C. Hua, Z. Zuo, J. Feng, and B. Pang, "Transmixnet: an attention based double-branch model for white blood cell classification and its training with fuzzified training data," *in Proc. IEEE Int. Conf. Bioinformatics and Biomedicine (BIBM)*, Houston, Texas, United States of America, 2021.
10. C. K. Jinka, B. Naseeba, S. S. Abbaraju, T. P. B. Reddy, K. Lokesh, B. T. R. Tatigunta, and N. P. Challa, "White Blood Cells Classification using CNN," *EAI Endorsed Transactions on Pervasive Health and Technology*, vol. 9, no. 1, pp. 1-8, 2023.
11. R. Al-Qudah and C. Y. Suen, "Improving blood cells classification in peripheral blood smears using enhanced incremental training," *Computers in Biology and Medicine*, vol. 131, no. 4, p. 104265, 2021.
12. T. T. Teoh, "CNN for White Blood Cell Classification," in Convolutional Neural Networks for Medical Applications, *Springer*, Singapore, 2022.
13. S. P. M. Raj, V. H. Deep, D. Chaudhary, and V. P. Singh, "A Pioneering and Streamlined Pixel Based Approach of Hyperspectral Image Classification for White Blood Cell Using Convolution Neural Network," *Annals of the Romanian Society for Cell Biology*, vol. 25, no. 5, pp. 260–269, 2021.
14. R. Ahmad, M. Awais, N. Kausar, and T. Akram, "White Blood Cells Classification Using Entropy-Controlled Deep Features Optimization," *Diagnostics*, vol. 13, no. 3, p. 352, 2023.
15. T. S. Almurayziq, E. M. Senan, B. A. Mohammed, Z. G. Al-Mekhlafi, A. Alshammari, M. Alturki, and A. Albakar, "Deep and Hybrid Learning Techniques for Diagnosing Microscopic Blood Samples for Early Detection of White Blood Cell Diseases," *Electronics*, vol. 12, no. 8, p. 1853, 2023.
16. K. G. Kannan, T. R. G. Babu, R. Praveena, P. Sukumar, G. Sudha, and M. Birunda, "Classification of WBC cell classification using fully connected convolution neural network," in *Journal of Physics: Conf. Series*, vol. 2466, no. 1, p. 012033, 2023.
17. S. S. R. Bairaboina and S. R. Battula, "Ghost-ResNeXt: An Effective Deep Learning Based on Mature and Immature WBC Classification," *Applied Sciences*, vol. 13, no. 6, p. 4054, 2023.
18. N. Dong, Q. Feng, J. Chang, and X. Mai, "White blood cell classification based on a novel ensemble convolutional neural network framework," *The Journal of Supercomputing*, vol. 80, no. 6, pp. 249–270, 2023.
19. R. Saikia and S. S. Devi, "White Blood Cell Classification based on Gray Level Co-occurrence Matrix with Zero Phase Component Analysis Approach," *Procedia Computer Science*, vol. 218, no. 1, pp. 1977–1984, 2023.
20. C. Cheuque, M. Querales, R. León, R. Salas, and R. Torres, "An efficient multi-level convolutional neural network approach for white blood cells classification," *Diagnostics*, vol. 12, no. 2, p. 248, 2022.
21. S. Dorafshan, R. J. Thomas, and M. Maguire, "Benchmarking image processing algorithms for unmanned aerial system-assisted crack detection in concrete structures," *Infrastructures*, vol. 4, no. 2, p. 19, 2019.
22. U. Obeta, D. Deko, and E. Mantu, "Deep learning-based diagnostic techniques for cancer: Extensive testing and clinical application insights," *AVE Trends in Intelligent Health Letters*, vol. 1, no. 1, pp. 28–37, 2024.
23. F. S. Alrayes, M. Maray, A. Gaddah, A. Yafoz, R. Alsini, O. Alghushairy, H. Mohsen, and A. Motwakel, "Modeling of Botnet Detection Using Barnacles Mating Optimizer with Machine Learning Model for Internet of Things Environment," *Electronics*, vol. 11, no. 20, p. 3411, 2022.
24. B. Karthikeyan, R. A. Devi, and M. Munshi, "Utilizing deep learning for the classification of brain tumours using MRI," *AVE Trends in Intelligent Health Letters*, vol. 1, no. 1, pp. 1–9, 2024.
25. Y. Lu, X. Tao, N. Zeng, J. Du, and R. Shang, "Enhanced CNN Classification Capability for Small Rice Disease Datasets Using Progressive WGAN-GP: Algorithms and Applications," *Remote Sensing*, vol. 15, no. 7, p. 1789, 2023.
26. M. Varshney, P. Kumar, M. Ali, and Y. Gulzar, "Using the Grey Wolf Aquila Synergistic Algorithm for Design Problems in Structural Engineering," *Biomimetics*, vol. 9, no. 1, p. 54, 2024.
27. S. R. Bose, J. A. Jeba, V. K. Kishore, G. Gnanaguru, and T. Shynu, "Deep learning-driven acute lymphoblastic leukemia detection using CT scan imaging," *AVE Trends in Intelligent Health Letters*, vol. 1, no. 2, pp. 110–124, 2024.
28. R. M. Das, A. Thirumalraj, and T. Rajesh, "An Improved ARO Model for Task Offloading in Vehicular Cloud Computing in VANET," *Research Square*, 2023. Available: https://www.researchsquare.com/article/rs-3291507/v1 [Accessed by 31/11/2024].

29. S. Karthik, E. S. Soji, S. S. Priscila, L. S. Deve, P. Paramasivan, and A. S. Kumar, “Enhanced performance evaluation of vector images using SVA for tumor detection and sizing in brain MRI scans,” *AVE Trends in Intelligent Health Letters*, vol. 1, no. 2, pp. 51–68, 2024.
30. S. Baswaraju, V. U. Maheswari, K. K. Chennam, A. Thirumalraj, M. V. V. P. Kantipudi, and R. Aluvalu, “Future food production prediction using AROA based hybrid deep learning model in agri-sector,” *Human-Centric Intelligent Systems*, vol. 3, no. 4, pp. 521–536, 2023.
31. GitHub, https://github.com/draaslan/blood -dataset, 2020. [Accessed by 30/06/2020].
32. GitHub, https://githvub.com/MahvmudulAlamCocmpl ete-Blood-Cell-Count -Dataset, 2021. [Accessed by 20/06/2021].
33. X. Zheng, “Data for: Fast and robust segmentation of cell images by self-supervised learning.” Mendeley, 2018. Available Online: http://www.doi.org/10.17632/w7cvnmn4c5.1 [Accessed by 15/05/2019].
34. Kaggle, https://www.kaggle.com/paultiothymooney/blood-cells, 2019. [Accessed by 15/05/2019].
35. Leukocyte Images, https://users.cecs.anu.edu.au/hrezatofighi/Leukocytehtm, 2019. [Accessed by 15/05/2019].