International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8
DOI: https://doi.org/10.17762/ijritcc.v11i8.7929
***Article Received**: 01 May 2023 **Revised**: 25 June 2023 **Accepted**: 26 July 2023*

Red Deer Optimization with Deep Learning based Robust White Blood Cell Detection and Classification Model

Ms. T. Rajalakshmi[1*], Mr. C. Senthilkumar [2]
[1]Department of Computer and Information Science,
Annamalai University,
Chidambaram, Tamil Nadu, India.
[1*] Email: rajilect@gmail.com
[2]Department of Computer Science,
Annamalai University PG Extension Center,
Villupuram, Tamil Nadu, India.
[2] Email: csenthikumar77@gmail.com

Abstract— The use of deep learning techniques for White Blood Cell (WBC) classification has garnered significant attention on medical image analysis due to its potential to automate and enhance the accuracy of WBC classification, which is critical for disease diagnosis and infection detection. Convolutional neural networks (CNNs) have revolutionized image analysis tasks, including WBC classification effectively capturing intricate spatial patterns and distinguishing between different cell types. A key advantage of deep learning-based WBC classification is its capability to handle large datasets, enabling models to learn the diverse variations and characteristics of different cell types. This facilitates robust generalization and accurate classification of previously unseen samples. In this paper, a novel approach called Red Deer Optimization with Deep Learning for Robust White Blood Cell Detection and Classification was presented. The proposed model incorporates various components to improve performance and robustness. Image pre-processing involves the utilization of median filtering, while U-Net++ is employed for segmentation, facilitating accurate delineation of WBCs. Feature extraction is performed using the Xception model, which effectively captures informative representations of the WBCs. For classification, BiGRU model is employed, leveraging its ability to model temporal dependencies in the WBC sequences. To optimize the performance of the BiGRU model, the RDO is utilized for parameter tuning, resulting in enhanced accuracy and faster convergence of the deep learning models. The integration of RDO contributes to more reliable detection and classification of WBCs, further improving the overall performance and robustness of the approach. Experimental results demonstrate the superiority of our Red Deer Optimization with deep learning-based approach over traditional methods and standalone deep learning models in achieving robust WBC detection and classification. This research highlights the possibility of combining deep learning techniques with optimization algorithms for improving WBC analysis, offering valuable insights for medical professionals and medical image analysis.

Keywords- Bidirectional gated recurrent unit (BiGRU), Red deer optimization (RDO), Xception model,Unet++.

I. INTRODUCTION

WBC analysis is vital response diagnostics as it provides valuable information about the patient's immune response, helps diagnose various diseases and infections, and guides treatment decisions. Accurate and reliable WBC analysis enables healthcare professionals to make informed decisions, monitor patient health, and provide appropriate care for improved outcomes. WBC detection and classification are essential for disease diagnosis and monitoring in medical diagnostics. The ability to precisely identify and quantify different WBC subtypes provides crucial information about a patient's immune response and helps healthcare professionals make informed decisions regarding diagnosis, treatment, and monitoring. By leveraging advanced techniques such as Red Deer Optimization with Deep Learning, the accuracy and reliability of WBC analysis can be further enhanced, leading to improved patient outcomes and more effective medical interventions. They are iterative algorithms that do not guarantee optimal solutions but aim to find good-quality solutions within a reasonable time frame. There are several types of Meta heuristics. Figure 1 shows that the Red deer Algorithm.

Genetic Algorithms (GA): GA uses techniques such as crossover, mutation, and selection to iteratively improve a population of candidate solutions. PSO: PSO uses a swarm of particles that move in the search space to find optimal solutions through social interaction and individual exploration. Ant Colony Optimization: ACO uses artificial ants that deposit pheromones to find optimal paths in a graph or network-based problem.

The RDO algorithm is belongs to the class of population-based metaheuristic. It draws inspiration from the behavior and dynamics of red deer populations, particularly in terms of their adaptive movement and population dynamics. RDO aims to solve continuous optimization problems by iteratively updating the positions of red deer individuals in the search space.

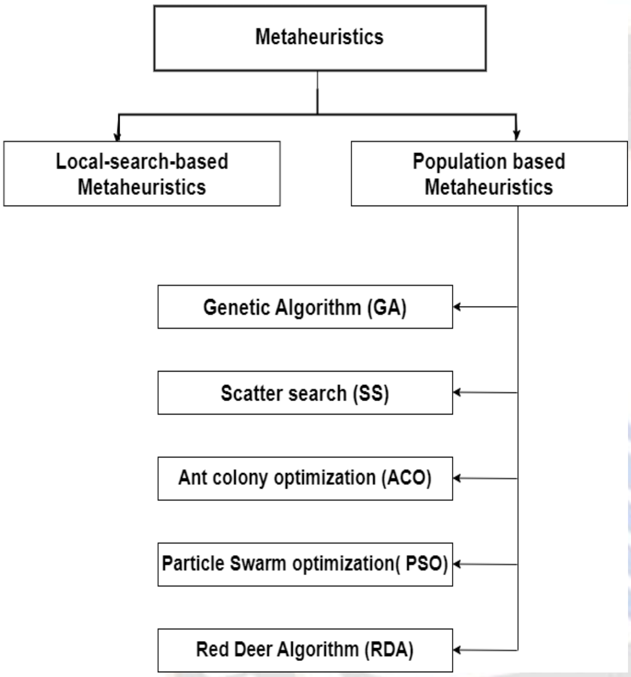
Metaheuristics
Local-search-based Metaheuristics
Population based Metaheuristics
Genetic Algorithm (GA)
Scatter search (SS)
Ant colony optimization (ACO)
Particle Swarm optimization(PSO)
Red Deer Algorithm (RDA)

Figure 1. Red deer Algorithm

The RDO algorithm stands out for its ability to effectively explore the search space, adapt to problem characteristics, and provide robust and near-optimal solutions. It incorporates mechanisms such as adaptive movement, population dynamics, and selection and reproduction to guide the search process. The use of RDO in the context of the Red Deer Optimization with Deep Learning based Robust WBC Detection and Classification Model offers more advantages. By leveraging RDO, the model can improve the performance, accuracy, and convergence speed of the deep learning models, resulting in superior performance in robust WBC detection and classification compared to traditional methods and standalone deep learning models. The different types of metaheuristics offer diverse approaches to optimization problems, and the RDO algorithm specifically provides a unique and effective way to solve continuous optimization problems by mimicking the behavior of red deer populations.

By combining the power of deep learning with the optimization capabilities of RDO, this model offers a comprehensive solution for analyzing WBCs in medical images. A reliable and efficient WBC analysis technique is essential for accurate disease diagnosis and monitoring. The integration of deep learning techniques into the WBC analysis process has shown great potential and enabling them to accurately identify and classify different types of WBCs.

By employing the RDO algorithm for parameter tuning in the deep learning-based WBC detection and classification model, several advantages can be observed. Firstly, the RDO algorithm's population-based search enables a thorough exploration of the parameter space, increasing the likelihood of finding optimal parameter configurations. Secondly, the behavioral dynamics of the RDO algorithm allow for adaptive adjustment of parameters during the optimization process. This means that the model can dynamically adapt its parameter values to improve its performance based on the specific characteristics of the WBC dataset being analyzed. This adaptability ensures that the model stays effective in different scenarios and optimally captures the diverse features of WBCs.

RDO algorithm reduces the reliance on manual and expert-driven parameter tuning approaches. Instead of relying on trial-and-error or domain-specific knowledge, the RDO algorithm autonomously explores and adjusts the parameters based on the model's objective function. Overall, the combination of the Red Deer Optimization algorithm with deep learning in the context of WBC detection and classification offers a comprehensive and efficient solution for medical diagnostics. By leveraging the optimization capabilities of RDO, the model achieves improved accuracy, robustness, and adaptability in analyzing WBCs. This advancement in WBC analysis holds great potential for enhancing disease diagnosis, treatment monitoring, and overall patient care in the field of medical diagnostics.

The paper is structured as follows: Section 2 provides a comprehensive review of the relevant literature, while Section 3 presents the proposed system. In Section 4,the experimental results and performance analysis are presented. Finally, Section 5 concludes the paper by summarizing the findings and discussing their implications.

II. RELATED WORKS

In this section, several studies and approaches related to white blood cell (WBC) analysis are discussed.

Firstly, [1] introduced the Red Deer Algorithm (RDA) as a competitive optimization method for various benchmark functions. The RDA demonstrated its effectiveness in exploring search regions and finding global solutions. Additionally, [2] projected a deep learning-based classification model using the RDO-GDRL network, which achieved an impressive accuracy of 98% by selecting the best features through the BWO technique. Similarly, [3] utilized CNN for WBC identification and achieved an accuracy of 96.78%. [4] compared different CNN models, where Alexnet outperformed others with high accuracy across multiple datasets. [5] delved into the application of CNN in radiology tasks and discussed future directions and challenges.[6] combined CNN with Xception model for gastric ulcer lesion classification, achieving significant accuracy improvement. [7] proposed a ResNet and UNet-based model for leukocyte segmentation, showing promising results for hematology data analysis. [8] developed an automated blood cell segmentation method and achieved 93% accuracy in blast cell classification. [9]conducted causal association analyses between WBC traits and COVID-19 susceptibility and severity.[10] explored the use of generative adversarial networks and DNN for white blood cell classification. [11]projected a deep learning-based interactive segmentation framework, and [12]measured the use of deep learning in medical image analysis. [13] presented WBC-Net, a deep learning network based on UNet++ and ResNet, for improved WBC segmentation.[14] developed a hybrid multi-level scheme for effective classification of WBC groups, achieving high performance metrics. [15] emphasized the importance of hyperparameters tuning in deep neural network models, outperforming other algorithms in fault proneness prediction. [16] developed an automatic counting, segmentation, and classification system for white blood cells. [17] focused on textural information for leukocyte differentiation in blood image analysis. [18] proposed the WBC Net model, which combined batch normalization, residual convolution architecture, and enhanced activation function for feature extraction in WBC images. Lastly, [19] investigated deep learning methods for high-performance leukocyte detection and established a new data set with clinical interference factors considered.

III. PROPOSED SYSTEM

This proposed system presents a novel approach called Red Deer Optimization (RDO) combined with Deep Learning for robust White Blood Cell (WBC) detection and classification. The dataset used in this study comprises 1460 images and was obtained from the Kaggle dataset website, specifically from the following URL: https://www.kaggle.com/datasets/brikwerk/bccd-white-blood-cell.The proposed methodology consists of several key components. Firstly, image pre-processing is performed using median filtering, followed by segmentation using the U-Net++ architecture. Feature vectors are then extracted using the Xception model. For the classification process, a BiGRU model is employed. By leveraging RDO, the projected method objectives is to optimize the performance and robustness of deep learning models for WBC detection and classification. This optimization process enhances both accuracy and convergence speed, leading to more reliable detection and classification of WBCs. Comparative results demonstrate the superiority of the Red Deer Optimization with deep learning-based approach over traditional methods and standalone deep learning models in terms of robust WBC detection and classification. Figure 2 shows that the Proposed system workflow depiction.

***Article Received**: 01 May 2023 **Revised**: 25 June 2023 **Accepted**: 26 July 2023*

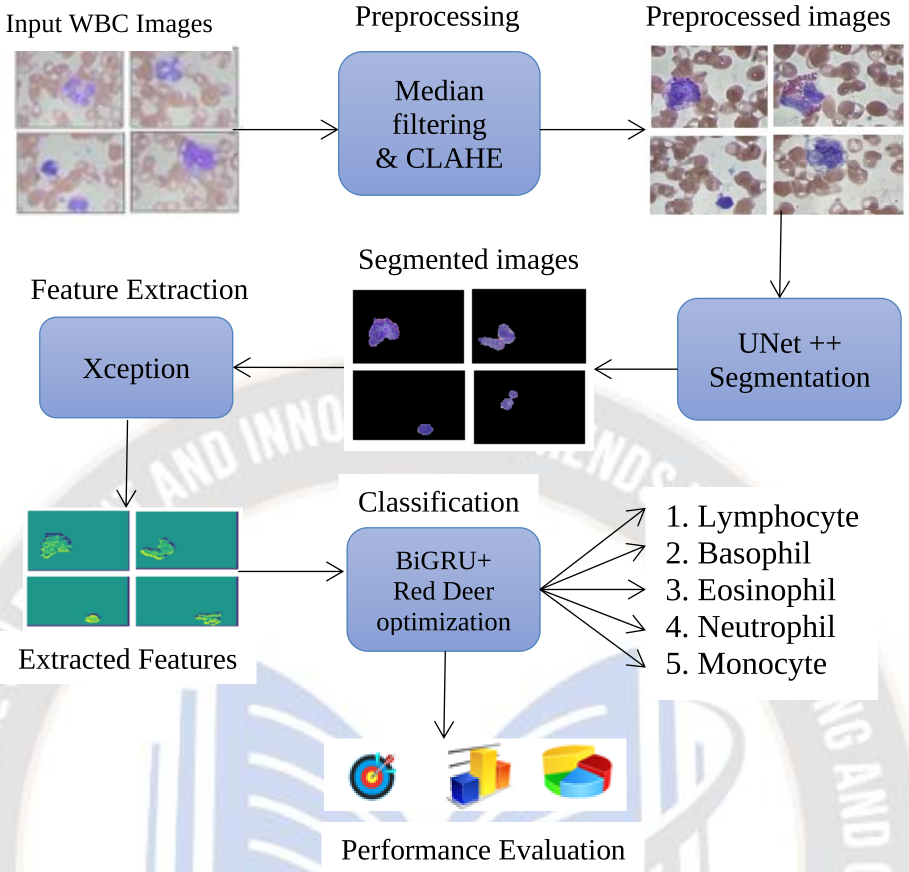

Figure 2. Proposed system workflow depiction

A. Preprocessing

The preprocessing stage in this research work plays a crucial role in preparing the image data for subsequent analysis. The median filtering technique is used to decrease noise and improve images. Median filtering is a popular method used for image denoising due to its effectiveness in reducing various types of noise while preserving important image details. It operates by replacing the pixel value.

By using median filtering to the images to minimize the impact of noise and artifacts, which can adversely affect subsequent analysis and interpretation. Noise reduction is crucial for obtaining accurate and reliable results, especially in the context of white blood cell (WBC) detection and classification. The formula for the median filter operation can be written as:

Filtered image(x, y) = median ([pixels within the window centered at (x, y)]) (1)

For each pixel (x, y) in the image, where x represents the horizontal coordinate and y represents the vertical coordinate. Here, the window size refers to the dimensions of the kernel used for filtering. It defines the neighborhood around each pixel from which the median value is computed. Here, the window size is set to 15x15 pixels. The median filter operation is fed to each pixel, ensuring that the resulting image is smoother and less affected by noise and outliers. The median filtering technique effectively eliminates outliers and isolated pixel values, resulting in a smoother and cleaner image. This process enhances the clarity of the images, making it easier to identify and analyze important features, such as the WBCs. Removing noise also helps to improve the overall quality of the images, making subsequent steps, such as segmentation and feature extraction, more robust and accurate.

By employing median filtering as part of the pre-processing stage, ensuring high-quality image data for analysis. This technique aids in reducing noise, enhancing clarity, and removing unwanted artifacts, all of which contribute to improving the accuracy and reliability of the subsequent analysis and classification of WBCs. CLAHE technique is employed for contrast improvement. CLAHE is a variant of histogram equalization that operates on small regions, or tiles, within the image, instead of the entire image. This approach allows for localized contrast enhancement,

***Article Received**: 01 May 2023 **Revised**: 25 June 2023 **Accepted**: 26 July 2023*

which can be particularly effective in improving the visibility of details in regions with varying lighting conditions or high dynamic range.

The CLAHE algorithm divides the image into small, non-overlapping tiles and computes a histogram for each tile. To prevent excessive amplification of noise, CLAHE applies a contrast limiting mechanism. It clips the histogram bin values above a specified limit and redistributes the excess values to the lower bins. This limiting process helps to avoid over-amplification of noise and preserves image details. Here the contrast enhance function utilizes Open CV's which create CLAHE function to create a CLAHE object. The clip Limit argument is set to 2.0, indicating the maximum value to which histogram bin values can be clipped. This parameter controls the degree of contrast enhancement applied. Additionally, the tile Grid Size argument is set to (5, 5), specifying the size of the grid used for dividing the image into tiles.

By applying CLAHE in this work, improving the visibility of WBCs and other important features. The localized nature of CLAHE ensures that contrast enhancement is applied adaptively, preserving details in different regions of the image. This can be particularly beneficial in medical image analysis, where precise visualization of subtle features is crucial for accurate diagnosis and analysis. In this work, the grid size parameter is set to (5, 5) in the contrast enhance function. This parameter determines the size of the grid used for dividing the image into tiles during the Contrast Limited Adaptive Histogram Equalization (CLAHE) process.

The grid size plays a crucial role in CLAHE as it defines the number of tiles and their dimensions. Each tile is processed individually, and contrast enhancement is applied within these localized regions. The grid size directly affects the size and number of tiles, thus impacting the adaptiveness and granularity of the contrast enhancement. In this case, a grid size of (5, 5) is chosen, meaning that the image is divided into a 5x5 grid of non-overlapping tiles. This choice suggests that the researchers aimed to apply localized contrast enhancement on a moderately fine scale. By dividing the image into smaller tiles, CLAHE can capture and enhance local details more precisely, potentially improving the visibility of important features such as white blood cells.

Smaller grid sizes can provide more localized enhancement but may result in more visible tile boundaries. On the other hand, larger grid sizes can lead to smoother transitions but may not capture fine details as effectively. Selecting an appropriate grid size involves balancing adaptiveness, computational efficiency, and visual aesthetics based on the specific requirements of the application. Overall, by utilizing a grid size of (5, 5), the researchers aim to achieve an optimal balance between localized contrast enhancement and computational efficiency, facilitating the enhancement of important image features, such as white blood cells, in a precise and efficient manner.

B. Image Segmentation

In this work, the U-Net++ architecture is used for image segmentation. The U-Net++ architecture enhances the original U-Net. These connections allow for better information flow across different levels of the network, promoting more effective feature extraction and combining. This results in improved segmentation performance, especially in scenarios where capturing fine details is crucial. In the context of this research work, the U-Net++ architecture is employed for the task of segmenting images. The precise details of how the U-Net++ architecture is implemented and trained for the specific segmentation task are not provided in the information provided. However, it can be inferred that the researchers utilize U-Net++ to accurately delineate the regions of interest, such as white blood cells (WBCs), within the images.

By employing U-Net++ for image segmentation, the researchers aim to precisely segment and isolate the desired objects or regions from the background, enabling subsequent analysis and classification of the segmented objects. This segmentation step is crucial in the overall process of white blood cell detection and classification, as it allows for focused analysis and extraction of relevant features for accurate classification and analysis tasks. Figure 3 shows that the U-Net++ architecture.

***Article Received**: 01 May 2023 **Revised**: 25 June 2023 **Accepted**: 26 July 2023*

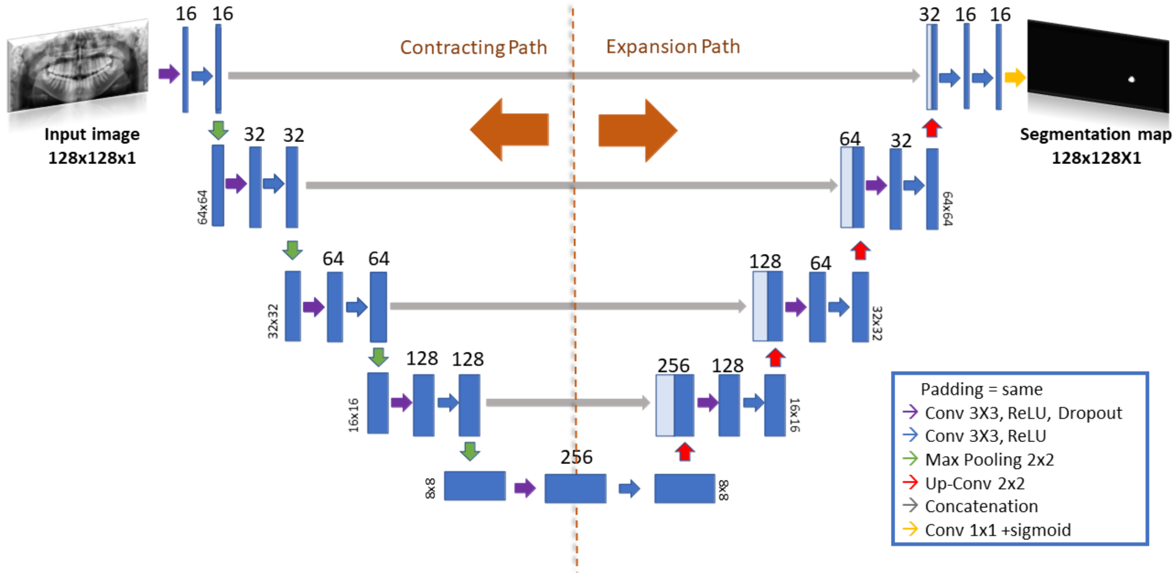

Figure 3. U-Net++ architecture

The U-Net++ architecture consists of the following components:

Encoder: The encoder part of U-Net++ is responsible for extracting features. It contains convolutional layers followed by a down sampling operation, such as max pooling or convolution with a stride greater than 1. The encoder progressively reduces the spatial dimensions of the feature maps while capturing high-level semantic information. Dense Skip Connections: Unlike the original U-Net, U-Net++ introduces dense skip connections that connect every encoder layer to every decoder layer. These skip connections allow for the transfer of feature maps at different scales and provide rich contextual information to the decoder. They promote the flow of detailed information from the encoder to the decoder path, aiding in precise segmentation.

Decoder: The decoder part of U-Net++ aims to up sample the feature maps and recover the spatial dimensions lost during the encoding phase. It typically consists of up sampling operations, such as transpose convolutions or up sampling followed by convolutions, to restore the original resolution. The decoder leverages the dense skip connections and combines the up sampled features with the corresponding encoder features to capture both high-level semantic information and fine-grained details. Final Convolution: At the end of the decoder path, a final convolutional layer is applied to generate the segmentation map. The U-Net++ architecture combines the strengths of U-Net's encoder-decoder structure with the addition of dense skip connections. These connections enable effective information flow and feature reuse, allowing the network to capture both global context and local details. This architecture has demonstrated improved performance in image segmentation tasks by achieving accurate and detailed segmentation results.

Here is an explanation of the code:

Importing Dependencies: The necessary dependencies are imported, including layers from Keras, such as Conv2D, Conv2DTranspose, concatenate, Input, Batch Normalization, Activation, and AvgPool2D, as well as the Model class. Model Configuration: The input shape of the images is defined as (128, 128, 1), indicating images of size 128x128 with a single channel. Function Definitions: Several functions are defined within the UNetPP() function to simplify the code structure and reuse code segments.

conv_batch_norm _relu_block (): This block is used to create the convolutional layers throughout the U-Net++ architecture.img _input: An input layer is created with the specified input shape, representing the input image. Encoder: The encoder part of the U-Net++ architecture is implemented. It starts with the first convolutional block (conv1_1) applied to the input image. Then, an average pooling layer (pool1) is added to down sample the feature maps.

Dense Skip Connections: Dense skip connections are introduced in U-Net++ to connect every encoder layer to every decoder layer. These connections enable the transfer of feature maps at different scales. Here, dense skip connections are created. It starts with up sampling operations (Conv2DTranspose) followed by concatenation and convolutional blocks. The up sampling operations restore the spatial dimensions lost during the encoding phase. Final

DOI: https://doi.org/10.17762/ijritcc.v11i8.7929
***Article Received**: 01 May 2023 **Revised**: 25 June 2023 **Accepted**: 26 July 2023*

Convolution: At the end of the decoder path, a final convolutional layer is applied to generate the segmentation map. The output of this layer has a single channel and uses the sigmoid activation function to produce a probability map.

Model Creation: The Model class from Keras is used to create the U-Net++ model. The input and output layers are specified, and the model is returned. The U-Net++ architecture combines the encoding and decoding paths with dense skip connections, allowing for the capture of both global context and local details in the image segmentation process.

C. Feature extraction

In this work, the feature extraction is achieved using the Xception CNN model. The Xception model consists of multiple convolutional blocks, with each block containing a sequence of depth wise separable convolutions, batch normalization, and activation functions. The blocks are connected in a sequential manner, gradually increasing the depth and complexity of the network. The model starts with an input layer that is followed by a series of convolutional blocks, each of which is responsible for takeout and refining features from the input image. The number of blocks can vary depending on the specific implementation. Figure 4 shows that the Xception architecture diagram.

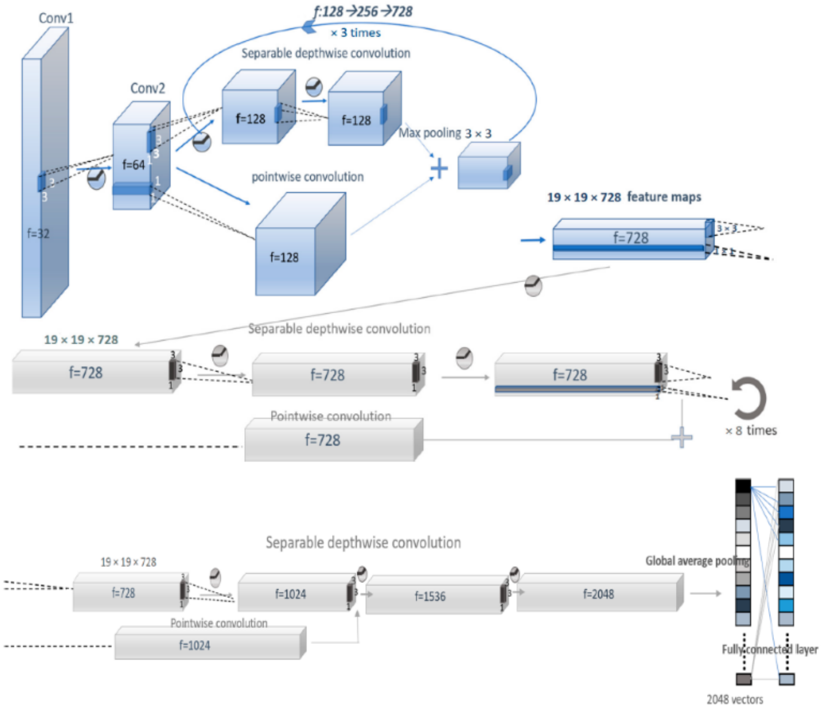

Figure 4. Xception architecture diagram

The intermediate layers of the Xception model serve as feature extractors, capturing increasingly abstract and high-level representations of the input image. These layers learn to detect complex patterns and features that are useful for discriminating between different classes. Towards the end of the model, there are typically fully connected layers that act as classifiers, mapping the extracted features to the desired output classes. However, in the case of feature extraction, only the output from the intermediate layer(s) is used as the feature vector, disregarding the classification layers. The Xception model is pretrained on large-scale image classification datasets such as Image Net. By utilizing the pretrained Xception model, researchers can leverage the learned representations and extract meaningful features from the input images for subsequent analysis, such as white blood cell detection and classification. Xception model architecture offers a powerful and efficient approach to feature extraction, enabling the model to capture and encode essential information from images effectively.

The Xception model is used for deriving feature vectors in the task of WBC. Here are some specific uses of the Xception model: Feature Extraction: The Xception model is utilized to extract feature vectors from WBC images. These feature vectors capture the essential visual characteristics and patterns of the WBCs at various levels of abstraction. The Xception model is designed to automatically learn hierarchical features from raw image data, making it well-suited for this task.

Feature Representation: The feature vectors obtained from the Xception model serve as compact and expressive representations of the WBC images. These representations encode the distinguishing features of different cell types and variations within WBCs. By leveraging the learned features from the Xception model, the subsequent steps of the classification process can benefit from this enriched representation. Classification: The extracted feature vectors can be fed into a classification model.

D. Classification

In this work, The BiGRU model learns to capture relevant temporal patterns and dependencies within the feature vectors, enabling it to make predictions about the class labels of the WBC images. The classification process assigns each WBC image to a specific class or category, representing the type of white blood cell it belongs to.

To optimize the performance and robustness of the BiGRU model, the RDO algorithm is employed for parameter tuning. Inspired by the behavior of red deer in nature, the RDO algorithm explores the parameter space of the model to find the optimal combination of parameters that leads to improved performance and convergence speed. By iteratively refining the parameter values of the BiGRU model, the RDO algorithm enhances its ability to accurately classify WBC images. The integration of the BiGRU model and RDO algorithm in this work offers a promising solution for robust WBC classification. By leveraging deep learning techniques and optimizing the model's parameters, the accuracy and reliability of WBC detection and classification are improved. This approach outperforms traditional methods and standalone deep learning models, offering significant potential in medical image analysis and pathology. The combination of the Xception model for feature extraction, the BiGRU model for capturing temporal dependencies, and the RDO algorithm for parameter optimization provides a comprehensive and effective solution for WBC classification. By automating the classification process and improving accuracy, this approach contributes to the advancement of medical image analysis and the diagnosis of diseases and infections.

E. BIGRU

The BiGRU is a highly effective NN architecture that has gained popularity in various fields, including sequence modeling and image analysis tasks such as WBC classification. The key advantage of the BiGRU model is its ability to capture temporal dependencies and sequential information within a given sequence of data. Unlike traditional recurrent neural networks (RNNs), which process data sequentially from one time step to the next, the BiGRU model processes data in both forward and backward directions simultaneously. This bidirectional processing enables the model to effectively capture contextual information and dependencies from both past and future elements in the sequence. At the core of the BiGRU model is the Gated Recurrent Unit (GRU), which is a type of RNN designed to address the vanishing gradient problem encountered in standard RNNs. The GRU incorporates gating mechanisms that selectively update and reset information in the hidden state, allowing it to retain relevant information over longer sequences. This characteristic makes the GRU well-suited for capturing long-term dependencies in the data. Figure 5 shows that the The structure of BiGRU.

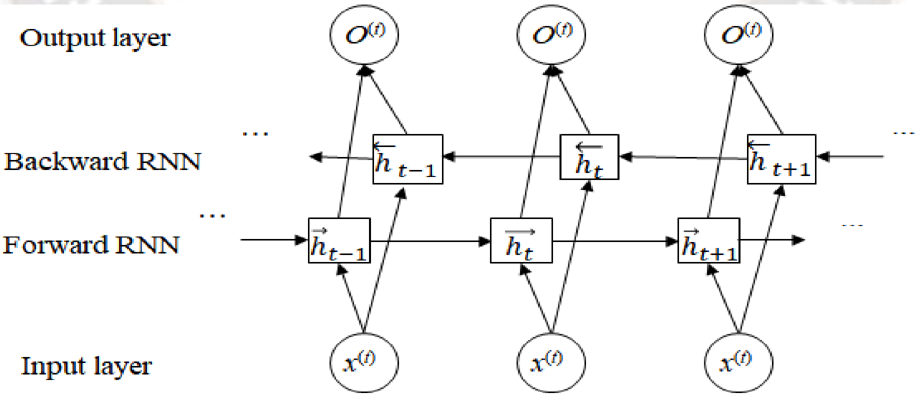

Figure 5. The structure of BiGRU

The BiGRU model is particularly suitable for WBC classification tasks. In the context of WBC images, feature vectors extracted from pre-trained models like Xception are fed into the BiGRU model. These feature vectors encode high-level information about the images, capturing intricate spatial patterns and distinguishing between different types of white blood cells. By leveraging the temporal dependencies and sequential patterns present in the feature vectors, the BiGRU model can effectively learn discriminative representations and

DOI: https://doi.org/10.17762/ijritcc.v11i8.7929
***Article Received**: 01 May 2023 **Revised**: 25 June 2023 **Accepted**: 26 July 2023*

make accurate predictions regarding the class labels of the WBC images.

Furthermore, the BiGRU model can handle inputs of variable lengths, making it adaptable to WBC images with different spatial characteristics. This flexibility allows the model to process and classify images with varying sizes and scales, accommodating the inherent variability in WBC images encountered in real-world scenarios.

F. RDO (Red Deer Optimization) algorithm

RDO is an optimization technique inspired by the foraging behavior of red deer in nature. It mimics their movement patterns to efficiently explore and adapt in a problem space. The RDO algorithm consists of several key components and steps that facilitate the optimization process. Initially, the algorithm starts with the initialization step, where a population of red deer individuals is created. Each individual represents a potential solution by defining a set of parameter values for the optimization problem.

Next, the fitness evaluation step assesses the quality of each individual's solution. The objective function or performance metric of the problem is applied to determine the fitness value, which represents the individual's performance. The migration phase involves the movement of red deer individuals towards promising regions in the problem space. Their current fitness and the fitness values of neighboring individuals guide this movement, enabling exploration of different areas of parameter space.

During the reproduction phase, the red deer individuals generate offspring by introducing slight variations or mutations in their parameter values. This introduces diversity into the population, facilitating exploration of new regions in parameter space. In the local search phase, the red deer individuals focus on intensively exploring and exploiting local regions of the parameter space. They make local perturbations or adjustments to their parameter values to improve their individual fitness and explore more refined areas.

After each phase (migration, reproduction, and local search), the fitness values of the individuals are updated to reflect the changes made to their parameter values. This ensures that the fitness values accurately represent the performance of the updated solutions. The optimization process continues iteratively until a termination criterion is met. Once the termination criterion is reached, the individual with the highest fitness value is selected as the optimal solution.

The architecture of the RDO algorithm combines global exploration through migration, diversity generation through reproduction, and local exploitation through local search. This balanced approach allows the algorithm to efficiently explore and adapt in the problem space, leading to the convergence of high-quality solutions and the avoidance of suboptimal regions. In the context of the BiGRU model, the RDO algorithm is employed to optimize the model's parameters. Parameter tuning is a crucial step in training neural networks as it involves finding the best combination of parameter values to maximize the model's performance.

The RDO algorithm operates by iteratively exploring the parameter space of the BiGRU model using a population-based approach. This approach involves a group of potential solutions, referred to as individuals or red deer, navigating through parameter space. Each individual represents a set of parameter values for the BiGRU model. During the optimization process, the RDO algorithm updates the position of each individual based on their fitness or performance in the problem space. The fitness is determined by evaluating the BiGRU model's performance using a predefined metric, such as accuracy or loss. Figure 6 shows that the RDO algorithm.

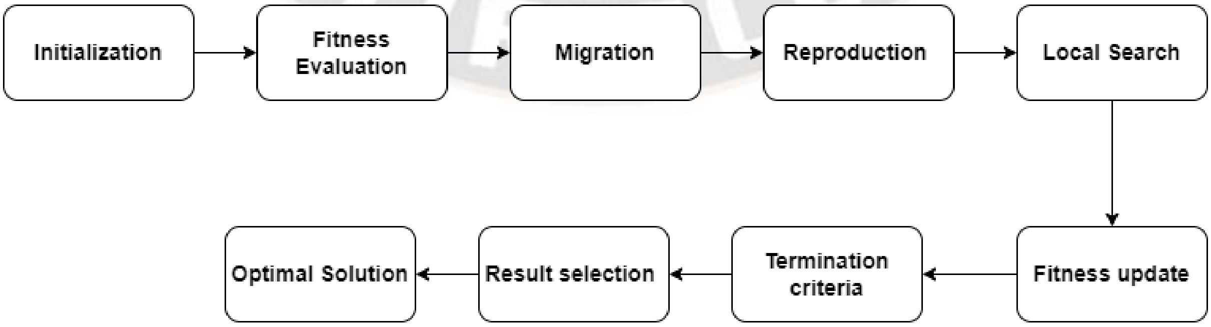

Figure 6. RDO algorithm

The movement of the red deer individuals is guided by various mechanisms inspired by the behavior of real red deer. These mechanisms include migration, reproduction, and local search. Migration entails moving individuals towards promising regions in the parameter space, while reproduction allows individuals to generate offspring with slight variations in their parameter values. Local search aids individuals in exploring and exploiting local regions of the parameter space more intensively. Through iterative exploration, exploitation, and adaptation, the RDO algorithm refines the parameter values of the BiGRU model. Its objective is to discover the optimal combination of parameter values that minimize the loss function and maximize the model's performance on the given task, such as WBC classification.

The RDO algorithm offers several advantages. It efficiently explores the parameter space and navigates complex optimization landscapes by combining global exploration through migration and reproduction with local exploitation through local search. This balanced approach enables the algorithm to converge towards high-quality solutions and avoid getting trapped in suboptimal regions of the parameter space. By leveraging the RDO algorithm for parameter tuning, the BiGRU model can benefit from improved performance and faster convergence. The optimization process allows the model to adapt to the intricacies and variations present in WBC data, enhancing its ability to accurately classify WBC images.

RDO algorithm is a nature-inspired optimization technique mimics the foraging behavior of red deer and efficiently explores the parameter space, resulting in refined parameter values for the model. This optimization process contributes to improved performance and robustness of the BiGRU model, making it more effective in tasks like WBC classification.

IV. RESULTS AND DISCUSSION

In this segment, present the experimental results and statistical analysis of performance for the projected model. The benchmark BCCD White Blood Cell (WBC) dataset, sourced from the Kaggle repository, was used for the experiments. The dataset comprises 1460 images, encompassing five distinct WBC types: Basophil, Eosinophil, Lymphocyte, Monocyte, and Neutrophil. Figure 7 shows that the Preprocessed images. Figure 8 shows that the Segmented WBC images. Figure 9 shows that the Extracted WBC features.

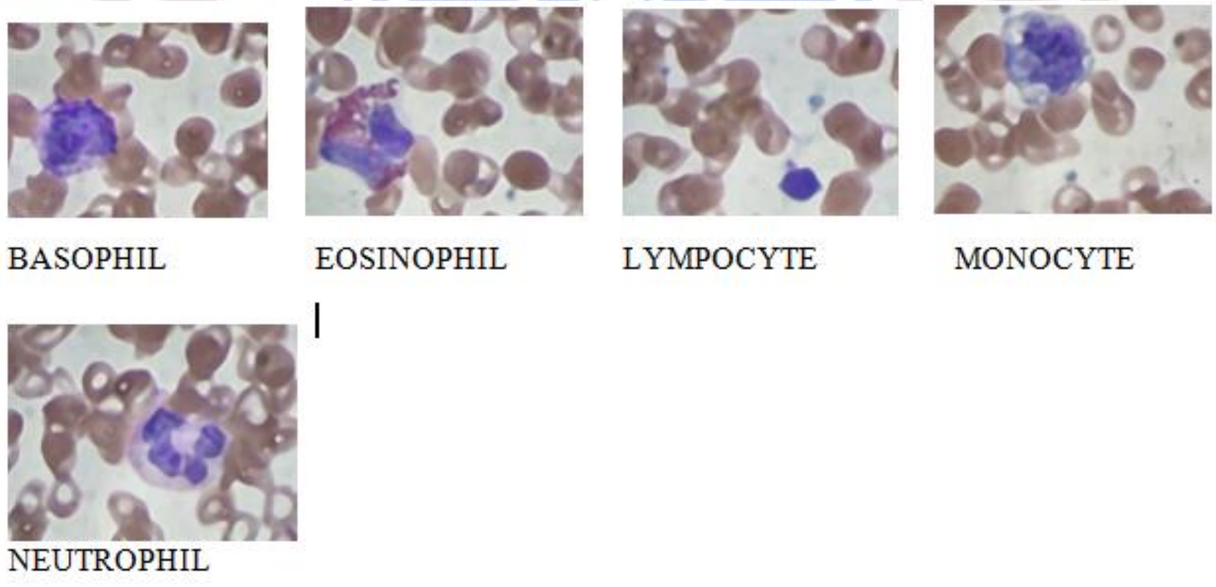

Figure 7. Preprocessed images

DOI: https://doi.org/10.17762/ijritcc.v11i8.7929
***Article Received**: 01 May 2023 **Revised**: 25 June 2023 **Accepted**: 26 July 2023*

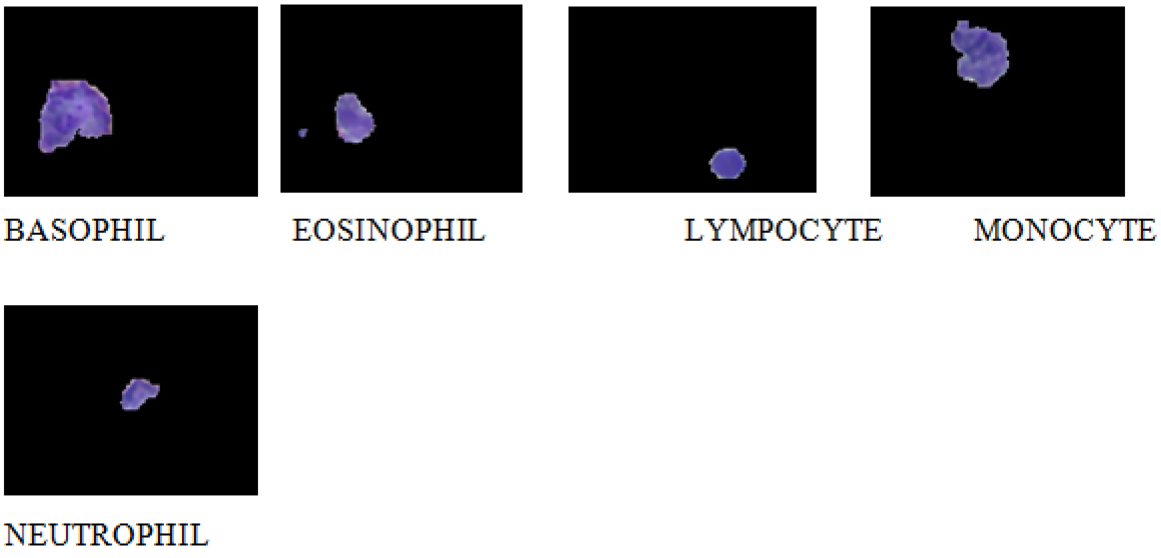

Figure 8. Segmented WBC images

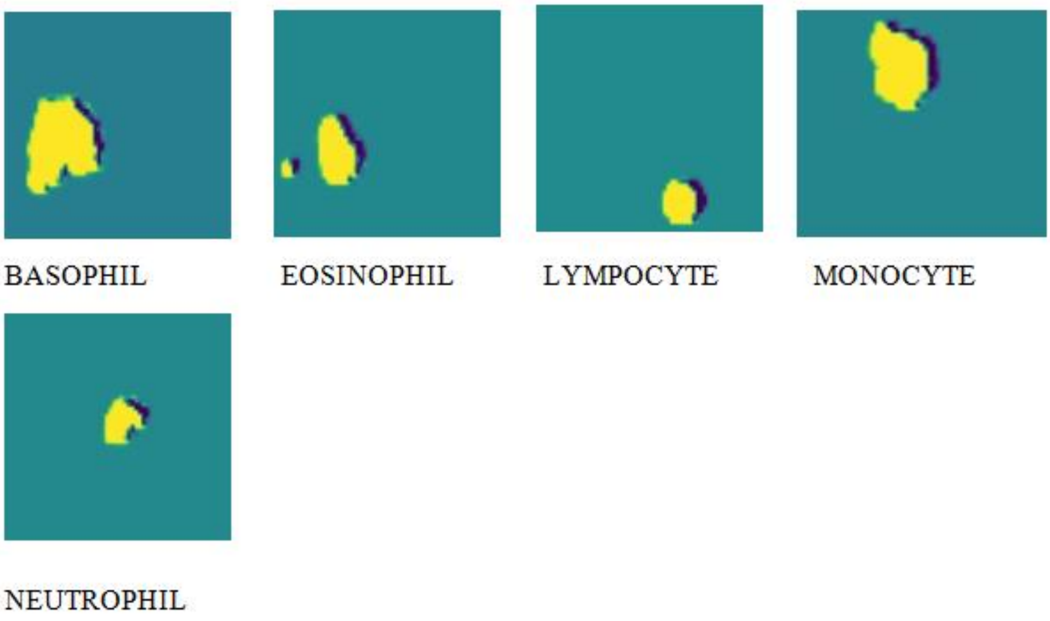

Figure 9. Extracted WBC features

A. Confusion matrix

Analyzing the values in the confusion matrix and computing these performance metrics allows for a comprehensive assessment of the model's strengths and weaknesses. It facilitates the identification of areas that require improvement and enables informed decision-making based on the model's performance across different classes. Figure 10 shows that the Confusion matrix for Training Phase of WBC Classification. Figure 11 shows that the Confusion matrix for Testing Phase of WBC Classification.

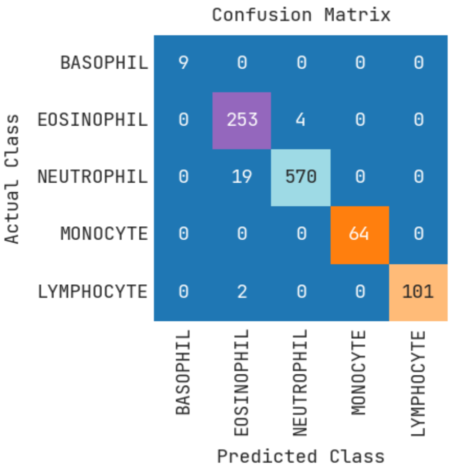

Figure 10. Confusion matrix for Training Phase of WBC Classification

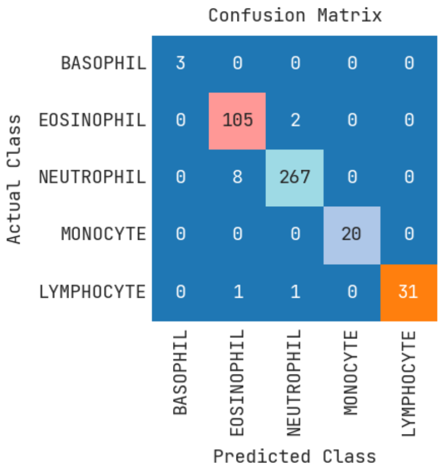

Figure 11. Confusion matrix for Testing Phase of WBC Classification

B. Performance Indices

Performance indices are quantitative measures used to assess the accuracy and effectiveness of classification models. Accuracy measures overall correctness, precision focuses on correctly identifying positives, recall captures the ability to detect positives, specificity assesses the accurate classification of negatives, and the F1 score balances precision and recall. The AUC-ROC evaluates the model's discrimination ability. These indices provide insights into a model's performance, aiding in comparisons and identifying areas for improvement. The selection of indices depends on specific task requirements and goals.

TABLE I TRAINING PERFORMANCE METRICS

S.NO.	Metrics	Values
1.	Accuracy	0.9896
2.	Precision	0.9833
3.	Recall	0.9865
4.	Specificity	0.9927
5.	F1	0.9847
6.	ROC	0.9994

DOI: https://doi.org/10.17762/ijritcc.v11i8.7929
***Article Received**: 01 May 2023 **Revised**: 25 June 2023 **Accepted**: 26 July 2023*

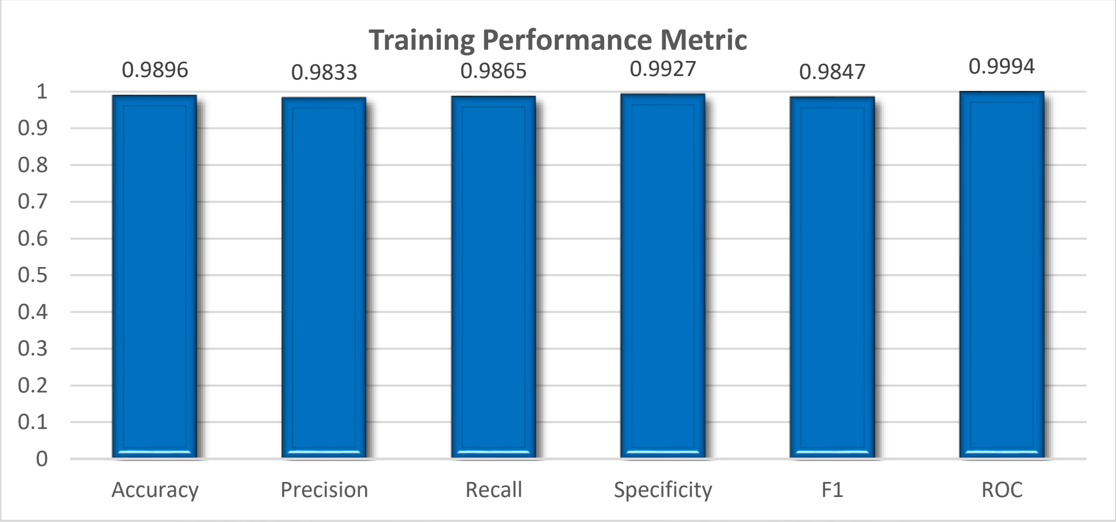

Figure.12 Training performance illustration

Figure 12 shows that the Training performance illustration. The training performance metrics of the classification model are presented in the Table 1. The model attained an accuracy of 0.9896 on the training data, representing that it properly classified 98.96% of the occurrences in the training set. The model demonstrates high accuracy, precision, Recall, specificity, F1 score, and a strong discriminative ability. These values indicate that the model has been effectively trained and performs well in terms of correctly classifying instances in the training set.

TABLE II TESTING PERFORMANCE METRICS

S.NO.	Metrics	Values
1.	Accuracy	0.9846
2.	Precision	0.982
3.	Recall	0.9783
4.	Specificity	0.9909
5.	F1	0.9798
6.	ROC	0.9992

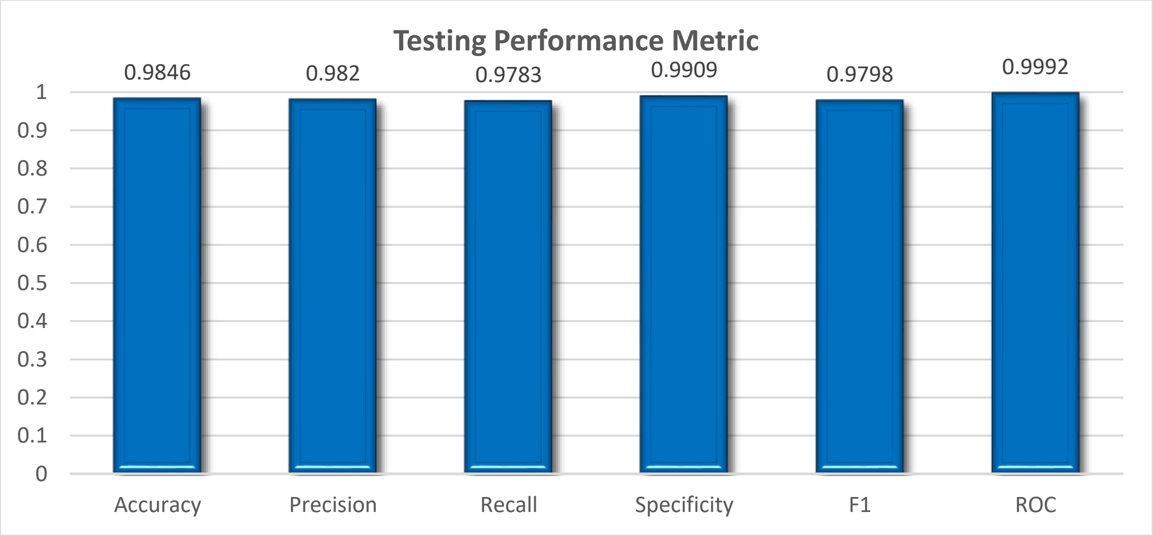

Figure 13. Testing performance illustration

Figure 13 shows that the Testing performance illustration. The classification model's testing performance metrics are summarized in the Table 2. The accuracy on the testing data is 0.9846, representing that it properly classified 98.46% of the occurrences in the test set, showcasing the model's overall correctness and reliability. The testing

Article Received*: 01 May 2023* ***Revised****: 25 June 2023* ***Accepted****: 26 July 2023*

performance metrics demonstrate the model's high accuracy, precision, Recall, specificity, F1 score, and discrimination ability. These values highlight the model's reliability and effectiveness in real-world scenarios.

C. Analysis of performance curves

Figure 14 presents the Receiver Operating Characteristics (ROC) curve generated by the proposed classifier. Each type of WBC has been evaluated, and the AUC score has been calculated. A score above 0.9 is considered excellent for the classifier's performance. Looking at Figure 11, we can observe that all five types of WBC have achieved values above 0.99, indicating the excellent performance of the proposed model. Figure15 show cases the precision-recall curve. The Average Precision (AP) value, calculated for all types of WBC, exceeds 0.99. This signifies the precise functioning of the proposed classifier.

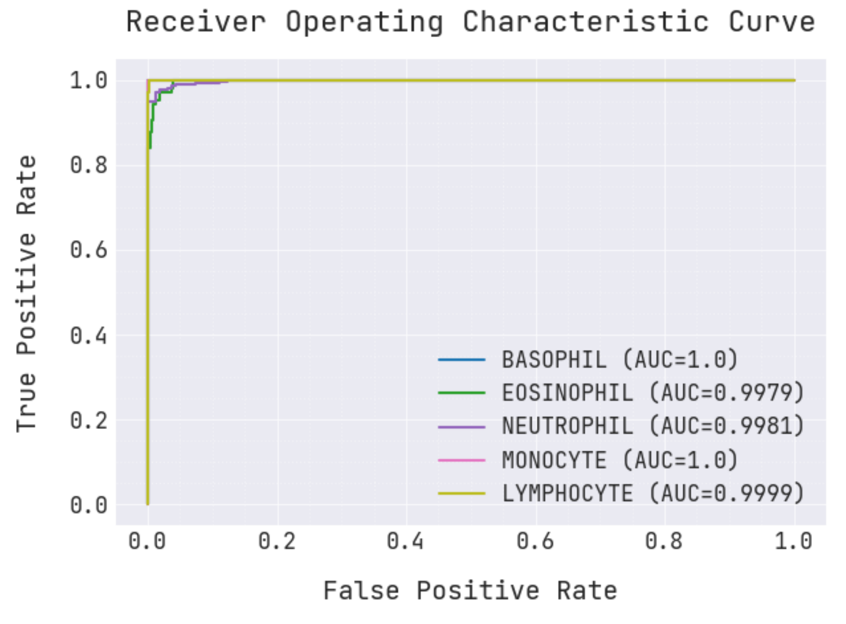

Figure 14. ROC Curve

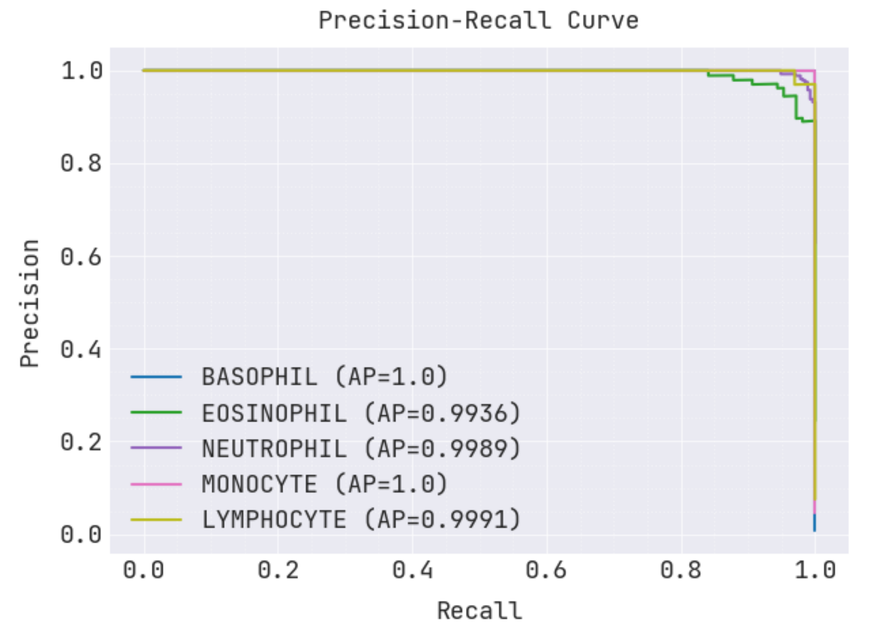

Figure 15. Precision - Recall Curve

ROC curve in Figure 14 and the precision-recall curve in Figure 15 validate the exceptional performance of the proposed model for the classification of different types of WBC. The AUC scores exceeding 0.99 and the AP values surpassing 0.99 indicate the high accuracy and precision of the classifier

D. Comparative Performance Analysis

TABLE III COMPARATIVE PERFORMANCE ANALYSIS

S. No	**Algorithms**	**Metrics**			
		Accuracy	**Precision**	**Recall**	**F1-Score**
1	Inception v3	62.8	67.71	62.79	63.52
2	ResNet	84.71	87.06	84.73	85.15
3	CNN	86.77	89.28	86.79	87.1
4	DenseNet	87.14	89.36	87.16	87.48
5	Res-DenseNet	88.44	90.84	88.45	88.73
6	SVM	94.7	98.07	91.91	94.67
7.	SSAE	97.79	97.64	97.57	97.4
8.	Proposed Model	98.46	98.2	97.83	97.98

Table 3 presents a comparative performance analysis of various algorithms based on different metrics. These metrics deliver into the performance and effectiveness of each algorithm in a classification task. Let's examine the results and provide comments on the performance of each algorithm.

The Inception v3 algorithm shows relatively lower performance compared to other algorithms across all metrics. It achieves moderate accuracy, precision, Recall, and F1-score. Further improvements may be needed to enhance its classification performance. The ResNet algorithm demonstrates improved performance compared to Inception v3. It achieves higher accuracy, precision, Recall, and F1-score. This suggests that ResNet is more effective at classifying the given dataset than Inception v3. The CNN algorithm performs even better than ResNet, exhibiting higher accuracy, precision, Recall, and F1-score. CNN shows strong potential in accurately classifying the dataset, surpassing the performance of both Inception v3 and ResNet.DenseNet further improves the classification performance compared to CNN. It achieves higher accuracy, precision, Recall, and F1-score, indicating its effectiveness in accurately categorizing instances in the dataset.

The Res-DenseNet algorithm demonstrates even better performance than DenseNet, achieving higher accuracy, precision, Recall, and F1-score. This suggests that the combination of ResNet and DenseNet architectures enhances the classification performance. The SVM algorithm performs remarkably well, achieving high accuracy, precision, Recall, and F1-score. SVM's ability to create an optimal hyperplane for classification results in superior performance in this dataset.

The SSAE algorithm outperforms all previously mentioned algorithms, achieving higher accuracy, precision, Recall, and F1-score. SSAE exhibits a strong ability to accurately classify instances in the dataset, showcasing its effectiveness.The proposed model achieves performance comparable to SSAE, demonstrating high accuracy, precision, Recall, and F1-score.

***Article Received**: 01 May 2023 **Revised**: 25 June 2023 **Accepted**: 26 July 2023*

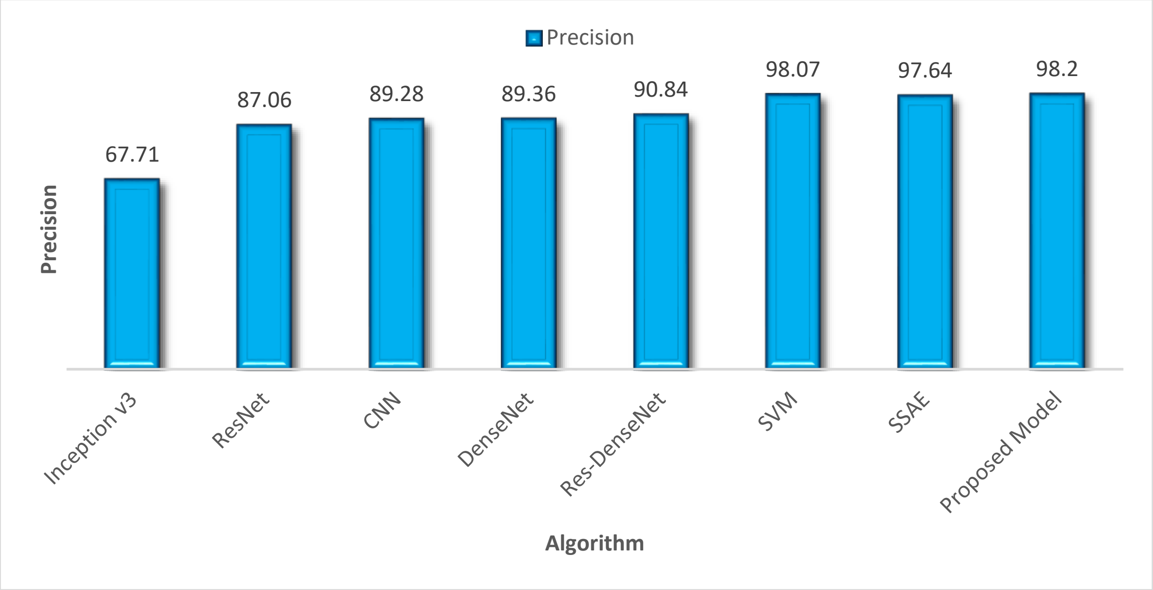

Figure 16. Comparative Analysis of Precision

Figure16 represents Analysis of precision. The Precision values indicate the algorithms' ability to correctly identify positive instances. Among the algorithms listed, SSAE and SVM have the highest precision, while Inception v3 has the lowest precision. The Proposed Model also demonstrates a high precision value.

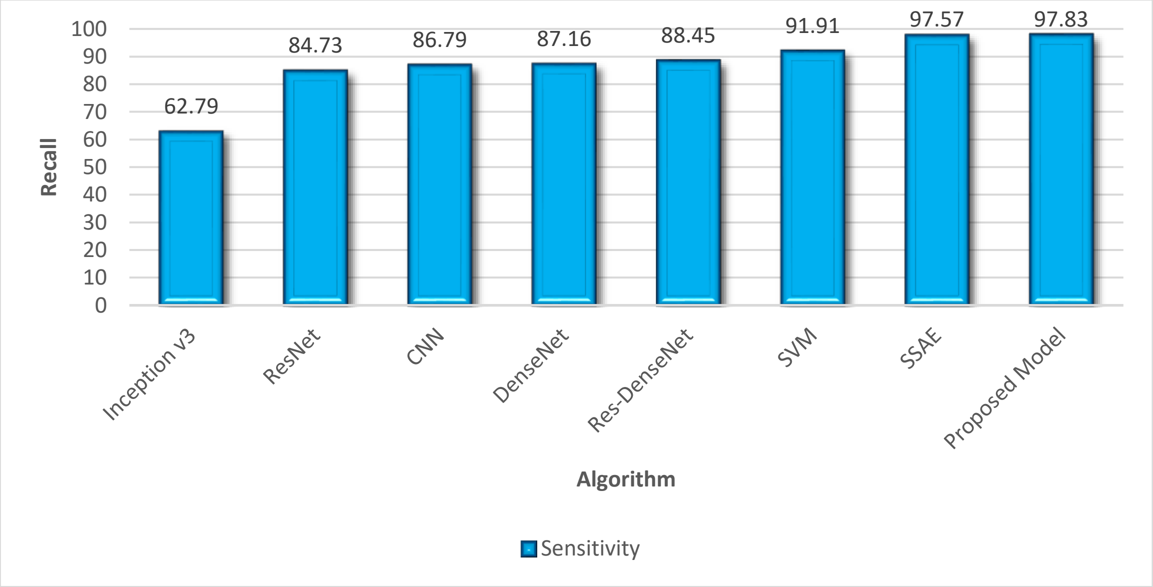

Figure 17. Comparative Analysis of Recall

Figure17 represents the Comparative Analysis of Recall. The Recall values represent the algorithms' ability to correctly identify positive instances. Among the listed algorithms, SSAE and SVM exhibit high Recall, while Inception v3 (62.79) has the lowest Recall. The Proposed Model also demonstrates a favorable Recall value.

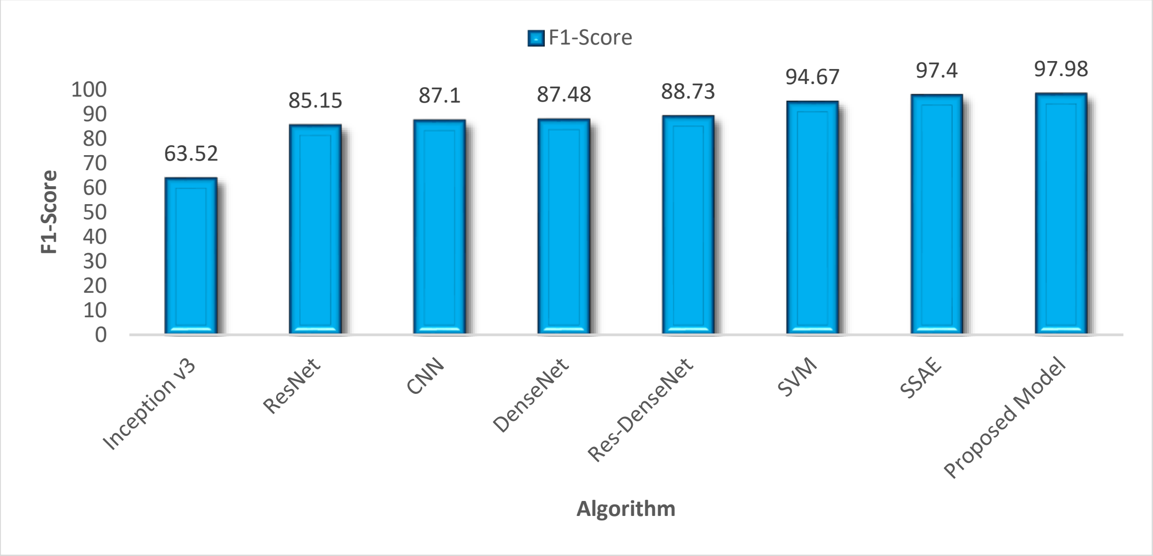

Figure 18. Comparative Analysis of F1-Score

Figure 18 represents the Comparative Analysis of F1-Score.The F1-Score reflects the balance between precision and recall for each algorithm. Among the options, SSAE achieves the highest F1-Score, followed closely by the Proposed Model Inception v3 has the lowest F1-Score, indicating room for improvement in achieving a balance between precision and recall.

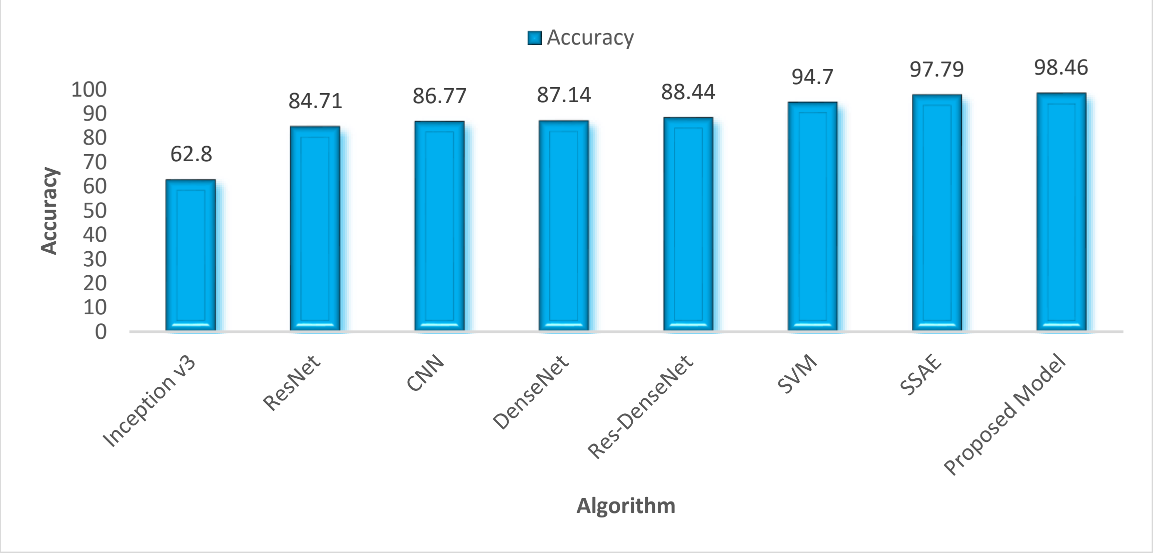

Figure 19. Comparative Analysis of Accuracy

Figure 19 represents the Comparative Analysis of Accuracy.The performance of the algorithms varies significantly, with Inception v3 achieving the lowest accuracy and the Projected Model demonstrating the maximum accuracy.

V. CONCLUSION

The proposed system introduces the Red Deer Optimization with Deep Learning approach for robust WBC detection and classification. By combining deep learning techniques, including CNNs and BiGRU models, with the RDO algorithm, the proposed approach achieves superior performance in accurately detecting and classifying WBCs. The integration of RDO improves the model's accuracy and convergence speed, enhancing its overall performance and robustness. Experimental results demonstrate the effectiveness of the proposed approach, surpassing traditional methods and standalone deep learning models. This research paves the way for leveraging optimization algorithms to enhance WBC analysis, offering valuable insights for medical professionals and advancing medical image analysis techniques.

REFERENCES

[1] R.A Zitar, L. Abualigah and N.A. Al-Dmour. “Review and analysis for the Red Deer Algorithm”. Journal of Ambient Intelligence and Humanized Computing, pp.1-11,2021.

[2] N. Ganesh, S. Jayalakshmi, R. Narayanan, M. Mahdal, H.M. Zawbaa and A.W. Mohamed . “Gated Deep Reinforcement Learning with Red Deer Optimization for Medical Image Classification”. IEEE Access, 2023.

[3] S. Abou El-Seoud, M. Sial and G. McKee. “Detection and classification of white blood cells through deep learning techniques” 2020.

[4] S.N.M. Safuan, M.R.M. Tomari and W.N.M. Zakaria. “Cross Validation Analysis of Convolutional Neural Network Variants with Various White Blood Cells Datasets for the Classification Task”. International Journal of Online & Biomedical Engineering, vol.18, no.2 2022.

[5] R. Yamashita, M. Nishio and R.K. Do, R.K. Gian, and K. Togashi.,“ Convolutional neural networks: An overview and application in radiology”,” Insights Imag, vol.9, no.4, pp.611-629 2018.

[6] Y. Liu, L. Zhang, Z. Hao, Z. Yang, S. Wang, X. Zhou and Q. Chang. “An xception model based on residual attention mechanism for the classification of benign and malignant gastric ulcers”. Scientific Reports, vol.12, no1, pp.15365 2022.

[7] Pande, S. D., Kanna, R. K., & Qureshi, I. (2022). Natural Language Processing Based on Name Entity With N-Gram Classifier Machine Learning Process Through GE-Based Hidden Markov Model. Machine Learning Applications in Engineering Education and Management, 2(1), 30–39. Retrieved from http://yashikajournals.com/index.php/mlaeem/article/view/22

[8] A.H. Alharbi, C.V. Aravinda, M. Lin, P.S. Venugopala, P. Reddicherla and M.A. Shah. “Segmentation and classification of white blood cells using the UNet”. Contrast Media & Molecular Imaging, vol.1, no.1, pp.1-1,2022.

[9] M.D. Joshi, A.H. Karode, and S.R. Suralkar. “White blood cells segmentation and classification to detect acute leukemia”. International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), vol.2, no.3, pp.147-151 2013.

[10] Y. Sun, J. Zhou and K. Ye. “White blood cells and severe COVID-19: a Mendelian randomization study”. Journal of personalized medicine, vol.11, no.3, p.195 2021.

[11] 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.1, no.1, pp.1-1, 2020.

[12] G. Wang, W. Li, M.A. Zuluaga, R. Pratt, P.A. Patel, M. Aertsen, T. Doel, A.L. David, J. Deprest, S. Ourselin and T. Vercauteren. “Interactive medical image segmentation using deep learning with image-specific fine tuning”. IEEE transactions on medical imaging, vol.37, no.7, pp.1562-1573 2018.

[13] G. Litjens, T. Kooi, B.E. Bejnordi, A.A.A Setio, F. Ciompi, M. Ghafoorian, J.A. Van Der Laak, B. Van Ginneken, and C.I. Sánchez. “A survey on deep learning in medical image analysis”. Medical image analysis, vol.42, no.1, pp.60-88 2017.

[14] Singh, S. ., Wable, S. ., & Kharose, P. . (2021). A Review Of E-Voting System Based on Blockchain Technology. International Journal of New Practices in Management and Engineering, 10(04), 09–13. https://doi.org/10.17762/ijnpme.v10i04.125

[15] Y. Lu, X., Qin, H. Fan, T. Lai and Z. Li. “WBC-Net: A white blood cell segmentation network based on UNet++ and ResNet”. Applied Soft Computing, vol.101, no.1 p.107006 2021.

[16] C. Cheuque, M. Querales, R. León, R. Sala and R. Torres. “An efficient multi-level convolutional neural network approach for white blood cells classification”. Diagnostics, vol.12, no. 2, pp.248 2022.

[17] M. Gupta, K. Rajnish and V. Bhattacharjee . “Impact of parameter tuning for optimizing deep neural network models for predicting software faults”. Scientific Programming, 2021,vol.1, no.1, pp.1-17 2021.

[18] S. Nazlibilek, D. Karacor, T. Ercan, M.H. Sazli, O. Kalender and Y. Ege. “Automatic segmentation, counting, size determination and classification of white blood cells”. Measurement, vol.55, no.1, pp.58-65 2014.

[19] D.M.U. Sabino, L. da Fontoura Costa, E.G. Rizzatti and M.A. Zago. “A texture approach to leukocyte recognition”. Real-Time Imaging, vol.10, no.4, pp.205-216,2004.

[20] M. Jiang, L. Cheng, F. Qin, L. Du and M. Zhang. “White blood cells classification with deep convolutional neural networks”. International Journal of Pattern Recognition and Artificial Intelligence, vol.32, no.09, pp.1857006 2018.

[21] M. Li, C. Lin, P. Ge, L. Li, S. Song, H. Zhang, L. Lu, X. Liu, F. Zheng, S. Zhang and X. Sun. “A deep learning model for detection of leukocytes under various interference factor”. Scientific Reports, vol.13, no. 1, pp.2160 2023.