Interdisciplinary Research in Medical Sciences Specialty
ISSN: 3062-4401
2023, Volume 3, Issue 2, Page No: 157-171
Copyright CC BY-NC-SA 4.0
Available online at: www.galaxypub.co/page/journals

Galaxy Publication

Region-Based Convolutional Neural Network with Radial Gradient Index Filtering for Automated Red Blood Cell Detection and Multi-Class Classification in Microscopic Images

Ivan Petrov[1], Alexey Smirnov[1], Dmitry Volkov[1]*

[1]Department of Translational Medicine, Faculty of Medicine, Sechenov University, Moscow, Russia.

***E-mail** ✉ d.volkov.research@yandex.ru

Received: 07 August 2023; **Revised:** 27 November 2023; **Accepted:** 28 November 2023

ABSTRACT

Biomedical image interpretation is fundamental to achieving robust imaging performance and supporting numerous clinical workflows. For accurate diagnosis of blood disorders involving red blood cells, these cells must be reliably detected and classified. Manual evaluation is labor-intensive and may introduce inaccuracies. Multi-label samples containing clusters of cells remain difficult to analyze because separating individual elements—especially when cells overlap or touch—is challenging. Modern biosensing technologies now enable highly capable biomedical imaging and support multiple medical applications. In this work, we design an intelligent neural architecture able to autonomously recognize and categorize red blood cells from microscopic imagery by employing region-based convolutional neural networks (RCNN) combined with advanced biosensors. The system effectively handles issues such as cell overlap or contact and can correctly identify diverse blood cell morphologies. Data augmentation was applied as a preliminary processing step to expand the dataset and boost computational performance. To further optimize image quality and suppress noise, we applied the Radial Gradient Index filtering technique for intensity normalization. When evaluated on medical imaging datasets, our approach demonstrated superior detection accuracy and lower loss compared with established models such as ResNet and GoogleNet. The model achieved 99% accuracy during training and 91.21% accuracy on test images. Overall, our method exceeded the performance of ResNet-50 and GoogleNet by roughly 10–15%. These findings indicate that AI-driven automated RBC detection can streamline hematological analysis, decrease human error, and promote timely disease identification.

Keywords: RBC detection, Object detection, Image processing, Deep learning

How to Cite This Article: Petrov I, Smirnov A, Volkov D. Region-Based Convolutional Neural Network with Radial Gradient Index Filtering for Automated Red Blood Cell Detection and Multi-Class Classification in Microscopic Images. Interdiscip Res Med Sci Spec. 2023;3(2):157-71. https://doi.org/10.51847/NVJ3VbiT3e

Introduction

The interpretation of microscopic medical images has become increasingly important for diagnosing and treating a wide range of clinical disorders. Among these, identifying and classifying blood cells is essential, especially for hematological diseases. Yet manual inspection of such small-scale images is labor-intensive and occasionally unreliable. To address these limitations, researchers have focused on developing advanced neural network systems capable of automatically recognizing red blood cells based on their morphological patterns while taking advantage of emerging biosensor technology. Using deep learning methodologies, our models consistently detect and categorize RBCs according to their structural attributes. By integrating state-of-the-art biosensors with deep learning frameworks, we aim to enhance both the precision and speed of blood cell assessment in microscopic images. This work may meaningfully impact diagnostic procedures and contribute to personalized healthcare advancements. Automated image-based blood cell analysis can reduce processing time, lower error rates, and ultimately lead to more reliable medical evaluations. For example, complete blood count analysis and microscopic review of blood preparations are central in diagnosing anemia, malaria, and leukemia, underscoring the clinical value of accurate RBC detection and classification. The Complete Blood Count (CBC) remains a standard method for quantifying the components of blood—platelets, Red Blood Cells (RBC), White Blood Cells (WBC), hemoglobin, and mean red cell indices—within routine physical assessments [1].

In efforts directed at automated RBC identification, blood smear evaluation remains an essential tool for diagnosis. Microscopic imaging enables early detection, analysis, and quantification of various blood-related conditions, particularly when parasites such as Babesia or Malaria infect RBCs directly [2]. Still, manually examining WBCs in leukemia or estimating parasitemia in thin smears is exhausting and vulnerable to counting errors [3]. Implementing algorithms that can accurately isolate and count red blood cells in microscopy images—and describe the distribution of small particles—would improve diagnostic reliability. Prior work, such as Khalid [4] and Rusul [5], introduced transfer learning strategies for automated detection of skin cancer and stroke due to the difficulty of manually distinguishing visually similar images. Likewise, deep learning systems are vital for identifying irregularities in hematological cells. When RBC abnormalities such as anemia are suspected, additional indices—including Mean Cell Volume (MCV), Mean Cell Hemoglobin (MCH), Red Blood Cell Distribution Width (RDW), Red Blood Cell (RBC), and Mean Corpuscular Hemoglobin Concentration (MCHC)—are evaluated to determine potential causes. Typically, healthy blood contains 4.2 million to 5.9 million RBCs per square centimeter, while normal White Blood Cell (WBC) counts range from 4,500 to 10,000 per microliter [6]. Elevated WBC levels beyond 30,000 cells per microliter may signal systemic infection, inflammation, allergy, leukemia, or tissue injury from burns [7]. Platelet measurements also hold diagnostic value; for example, platelet levels are closely monitored in dengue cases, and transfusion may be required if counts drop below critical thresholds [7].

The primary goal of this study is to introduce an integrated system capable of reliably detecting red blood cells within imaging datasets. By applying CBC-derived analysis to blood smear images, this framework can substantially enhance the assessment, diagnosis, and ongoing monitoring of numerous medical conditions, including anemia, leukemia, infections, and allergic responses. When abnormalities such as anemia are observed, additional RBC-related indices are evaluated to determine contributing factors. This work proposes a multi-model deep learning–driven approach for red blood cell identification. The deep learning section includes preprocessing, feature extraction, and image classification procedures [8]. A Region-based Convolutional Neural Network (RCNN) was implemented to accurately localize regions of interest and effectively describe RBCs in microscopic images. The main contributions of this study are summarized below.

1. We developed a framework integrating advanced biosensing technologies with deep learning techniques to improve the precision and overall performance of blood cell examination in microscopic medical imagery.
2. The Radial Gradient Index filtering procedure was adapted to enhance data normalization and suppress noise in microscopic inputs.
3. Data augmentation was applied to increase image resolution and address class imbalance within the dataset.
4. An RCNN-based model was employed for reliable RBC representation and detection.

Section 2 reviews existing work on RBC counting and segmentation. Section 3 outlines the methodology and neural network–based design. Section 4 evaluates accuracy, processing time, and RBC identification outcomes, along with experimental observations and results. Section 5 highlights limitations of the study and suggests future research directions. Section 6 provides the conclusion.

Related work

Recent advances in artificial neural networks—especially with the emergence of sophisticated machine learning and deep learning methods—have accelerated progress in this domain [9]. Deep learning, derived from ANN principles, mimics cognitive learning mechanisms of the human brain and is designed to approximate unknown functions through layered nonlinear structures. It learns directly from the properties of dataset samples and has been applied across diverse tasks such as image interpretation, handwritten character analysis, and speech recognition.

A number of contemporary studies have addressed the classification and counting of blood cells. Gangadhar *et al.* [10] reported the creation of an automated system for identifying and quantifying blood cells using holographic imaging coupled with deep learning. Their system achieved a notable overall accuracy of 97.4% when distinguishing RBCs, WBCs, and platelets. For red blood cell counting specifically, the model reached 98.2% accuracy, 98.5% precision, and 97.8% recall. In white blood cell enumeration, the system produced 96.1% accuracy, 96.7% precision, and 95.2% recall. Platelet analysis showed 98.9% accuracy, 99.1% precision, and 98.6% recall. Expert comparisons revealed strong concordance, with R2 values of 0.98 for RBCs, 0.96 for WBCs, and 0.99 for platelets. Performance remained stable across multiple samples, maintaining accuracy variability under 5.

A 2022 study by Sharma *et al*. [11] employed deep learning for blood cell classification and quantification. The authors showed that deep-learning models can reliably differentiate several blood cell types. DenseNet121 was applied for identifying white blood cell categories in microscopic images, achieving 98.84% accuracy, 99.33% precision, 98.85% sensitivity, and 99.61% specificity.

Raillon Camille *et al*. [12] focused on identifying and classifying Circulating Tumor Cells (CTCs). Their platform successfully distinguished CTCs from other blood components through microfluidic single-cell evaluation. Gaps observed in some RBC identifications indicated that additional preprocessing would enhance accuracy. The impedance chip demonstrated precise counting, showing mean errors of 1.0% for beads and 3.5% for cancer cells. Moreover, the technique effectively differentiated bead sizes, demonstrating the chip's capability to distinguish cancer cells from smaller blood cells using size-dependent information.

Jung *et al*. [13] designed a system for RBC counting and automated detection of malaria parasites in thin smear samples. Their method applied Otsu thresholding following green-channel processing to isolate infected RBCs. Notably, the referenced studies did not incorporate holographic imaging, which may leave datasets susceptible to artifacts or conditions that hinder correct cell detection. Several methods also struggled to recognize cells that were clustered or overlapping. The system's detection threshold showed exceptional sensitivity, capable of identifying parasites with a probability as low as 0.00066112%. It also reached 100% sensitivity and specificity, accurately identifying all P. vivax and P. falciparum samples. These findings highlight significant advantages for rapid malaria diagnosis and quantifying parasite levels compared to conventional microscopy.

Cruz Dela *et al*. [14] introduced a technique for counting RBCs, WBCs, and platelets. Their workflow included several preprocessing operations before converting the input image to a binary format, followed by segmentation and counting based on an optimal threshold derived from the histogram. Although they reported an accuracy of 95%, the approach could not successfully detect cells that overlapped. The reliance on iterative thresholding also risked discarding relevant information, which ultimately reduced segmentation performance.

To address the challenge of overlapping cells, Nenden *et al.* [15] utilized a distance transform–based strategy aimed specifically at clustered groups of cells. They began by computing central points using distance transformation, selecting the best center candidates according to boundary coverage, and estimating average cell size through single-cell extraction. An algorithm employing a single-cell mask was then developed to separate individual cells. Their dataset included expert-annotated ground truth, and the proposed method reached accuracies of 96% and 70% when using the Distance Transform technique. A notable limitation, however, was that the method assumed regular cell shapes and high-magnification clarity, conditions not always met—particularly in diseased samples. As a result, detecting irregular cell shapes remains essential. Furthermore, noise had a negative impact on their model's performance, leading to weaker outcomes.

Chen and Chung [16] introduced a circular detection approach called Randomized Circle Detection (RCD), which they argued performs better than traditional Hough Transform variants. Unlike accumulator-based methods, RCD randomly selects four edge pixels and verifies that they are not collinear to form a candidate circle. Once a circle is hypothesized, the algorithm evaluates its validity using a distance criterion. After determining radius and center, the method counts pixels lying near the circle boundary by measuring the distance between each pixel and the circle's edge; pixels falling below a fixed threshold are treated as boundary points. However, RCD shows reduced efficiency in large images with many edge pixels, and because it samples only four pixels, the likelihood of forming a true circle decreases. Its dependence on numerous preset parameters and threshold values—along with a fixed iteration count linked to pixel intensity—also makes RCD less suitable for detecting irregular circular shapes.

Given the strong performance of the Hough Transform, numerous researchers have applied it to circle detection in microscopic images for RBC and WBC quantification [17, 18]. Mahmood *et al*. [18] employed the technique for counting cells in microscopy images. They first converted the input into a color-space representation and segmented red and white cells based on luminance ranges of 130–150 and 80–100, respectively. The next phase used segmentation algorithms, followed by the Canny edge detector. Ultimately, the Hough Transform was used to detect and enumerate cells. Depending on the cell type and preprocessing methods used, accuracy ranged from 64% to 87%. Nevertheless, the approach required substantial memory and computational power to evaluate wide radius ranges, and it struggled with irregular or heavily overlapping cells, reducing its accuracy.

Mahmood and Mansor [19] studied ten images containing normal cells by first transforming them into HSV space and focusing on the Saturation ("S") channel. Wavelet-based segmentation combined with adaptive thresholding was then applied. Their method yielded approximately 96% accuracy compared with manual counts. Despite this, the approach worked best only when cells were normally shaped, had limited overlap, and were fairly uniform. Under such conditions, the Hough Transform contributed positively to performance.

Venkatalakshmi and Thilagavathi [20] applied a similar methodology to Mahmood and Mansor [19] for RBC counting using the Circular Hough Transform. Still, their model also demonstrated reduced flexibility when handling overlapping or irregularly formed cells. In [21], a filter was proposed to correct segmentation errors and refine image boundaries. Shan *et al*. [22] presented a lesion detection approach that incorporates speckle suppression and includes spatial and textural information, using anisotropic diffusion to reduce noise. Diaz-Pernil *et al*. [23] highlighted bio-inspired parallel methods for analyzing homology-based geometric elements in 2D images. However, many of these methods face limitations, particularly in extracting robust features or interpreting key patterns effectively.

Unlike the related studies summarized above—which often rely on metrics such as accuracy, precision, and recall—this work takes a different direction. It employs a region-focused deep learning architecture (RCNN) rather than conventional CNNs or autoencoder models used in prior research. A unique aspect of this study is the integration of biosensors into the RCNN design, distinguishing it from investigations that primarily depend on established architectures like ResNet or GoogLeNet. In preprocessing, this work adopts a specialized filtering method rather than conventional procedures such as thresholding or morphological processing employed in earlier papers. It additionally incorporates data augmentation to expand the training set, a feature frequently missing from most related research. Furthermore, RCNN offers superior handling of overlapping and irregularly shaped cells, an area where Hough Transform–based techniques commonly fall short. With its tailored RCNN model, biosensor integration, and advanced preprocessing, this methodology provides a more sophisticated and distinct solution compared with earlier approaches. Its emphasis on resolving overlapping cell issues further reinforces this distinction.

Proposed model

This section presents the RBC methodology composed of two major components: i) Data Processing and ii) RBC identification via an RCNN framework. Initially, the suggested preprocessing stage incorporates a filtering strategy that sharpens image clarity while reducing unwanted artifacts, along with augmentation operations that expand sample quantity and strengthen computational performance. Subsequently, an RCNN is trained to localize and classify red blood cells. The overall system keeps key visual information intact and improves detection effectiveness. **Figure 1** illustrates the layer-wise schematic of the proposed architecture.

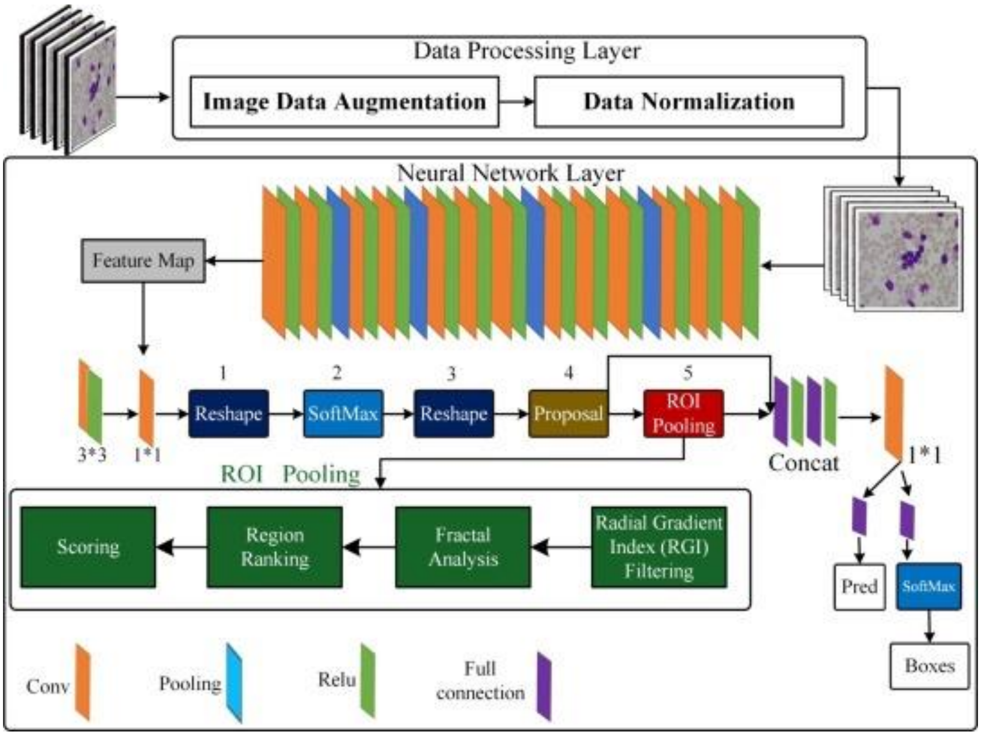


Figure 1. Visual outline of the designed red blood cell detection framework.

Data processing layer

(a) Image Data Augmentation: Within the Data Processing Layer, augmentation plays a central function. Standard deep-learning transformations—rotation, scaling, mirroring, and others—are applied to generate modified copies of the original images. This expansion of training variability supports improved generalization.

(b) Data Normalization: Normalization is also necessary, where pixel values are adjusted to a uniform scale, commonly between 0 and 1, to accelerate model optimization.

Neural network layer

(a) The Neural Network Layer is composed of multiple operations such as Convolution, Pooling, Relu activations, and related components. These progressively refine the representation of the input, extracting multilevel attributes needed for high-quality detection. Differently colored blocks symbolize distinct functional stages.

(b) Additional processes—including Reshape, SoftMax, Proposal, ROI Pooling, and Concat—mark structured stages tailored specifically for the object-detection workflow.

(c) ROI Pooling (Region of Interest Pooling) plays a crucial role by standardizing variable-dimension regions into a fixed-size feature map, enabling stable downstream classification and bounding-box refinement.

Reshape, SoftMax, Proposal, ROI (Region of Interest) pooling

(a) In this architecture, reshaping adjusts tensor dimensions while conserving numeric content. This realignment ensures compatibility with subsequent model layers.

(b) SoftMax functions as the probability-producing activation, transforming the model’s numeric outputs into interpretable class likelihoods.

(c) The Proposal stage suggests plausible object locations by producing candidate bounding regions that undergo further RCNN analysis for categorization and coordinate estimation.

(d) ROIs correspond to the proposed bounding regions that may contain target cells. ROI Pooling converts every ROI into a uniform representation, a critical preparatory step before classification and regression.

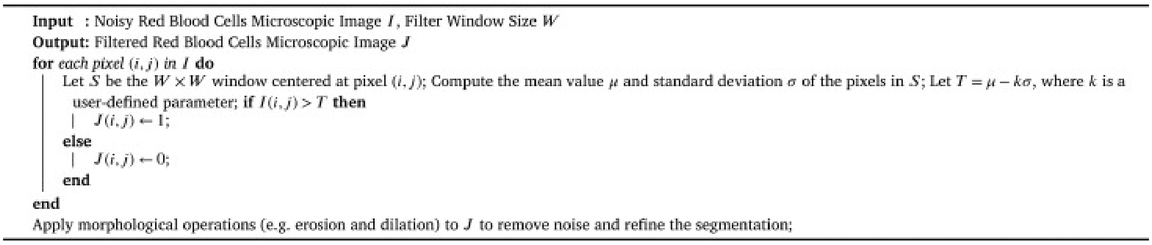
Post neural network operations

(a) These operations refine the model’s detections. Components such as Scoring, Region Ranking, Fractal Analysis, and Radial Gradient Index (RGI) Filtering enhance the reliability and precision of the object-recognition output.

(b) After executing these tasks, the system generates the final predictions (Pred) and bounding-box outputs (Boxes) that capture object identity and confidence values.

Pre-processing the data and detecting RBCs in clinical images

Even advanced image-processing approaches have constraints, particularly when they depend heavily on predefined rules. Deep-learning-based detection has shown significantly improved performance. Algorithm 1 summarizes the workflow used to denoise microscopic clinical images. Here, the segmentation assumes a binary scheme in which pixel values are categorized as 0 (background) or 1 (cell region). Final morphological cleanup removes residual noise from the segmented output.



```
Input  : Noisy Red Blood Cells Microscopic Image $I$, Filter Window Size $W$
Output: Filtered Red Blood Cells Microscopic Image $J$
for each pixel $(i, j)$ in $I$ do
    Let $S$ be the $W \times W$ window centered at pixel $(i, j)$; Compute the mean value $\mu$ and standard deviation $\sigma$ of the pixels in $S$; Let $T = \mu - k\sigma$, where $k$ is a
     user-defined parameter; if $I(i, j) > T$ then
        $J(i, j) \leftarrow 1$;
    else
        $J(i, j) \leftarrow 0$;
    end
end
Apply morphological operations (e.g. erosion and dilation) to $J$ to remove noise and refine the segmentation;
```

Algorithm 1. Improving Image Quality; Noise Suppression in Microscopy via Radial Gradient Index Filtering.

Let I denote a grayscale image with dimensions $m \times n$. A pixel with coordinates (i, j) has intensity (i, j) for $i=1,2,\ldots,m$ and $j=1,2,\ldots,n$.

Bilateral filtering smooths an image while retaining edges through weighted averaging that depends on spatial distance and similarity in brightness. Gaussian weighting functions determine these contributions.

Let $(i,,,q)$ represent the weight used when pixel (p,q) influences the filtering of pixel (i, j). The computed weights follow a Gaussian-based formulation:

$$w(i,j,p,q) = w_d(i,j,p,q) \cdot w_r(i,j,p,q) \tag{1}$$

where wd represents the spatial (distance) component of the weighting function, and w_r corresponds to the intensity-based (range) component.

The spatial weight wd quantifies how far pixel $(i,)$ is from pixel (p, q), and it is typically derived from a Gaussian formulation as follows:

$$w_d(i,j,p,q) = e^{\frac{-(i-p)^2+(j-q)^2}{2\sigma_d^2}} \tag{2}$$

where σd is the parameter that determines how broadly the filter responds across spatial distances.

The range component wr evaluates how close the intensity values of pixels $(i,)$ and $(p,)$ are, and it is obtained through an additional Gaussian expression given by:

$$w_r(i,j,p,q) = e \;\; \frac{(I\,(i,j) - I\,(p,q))^2}{2\sigma_r^2} - 22\text{☒}\sigma 2r \tag{3}$$

where σr specifies the extent to which intensity differences influence the filter response.

The output image I' is then computed by taking a weighted average of the intensities of surrounding pixels, using the previously defined weights w:

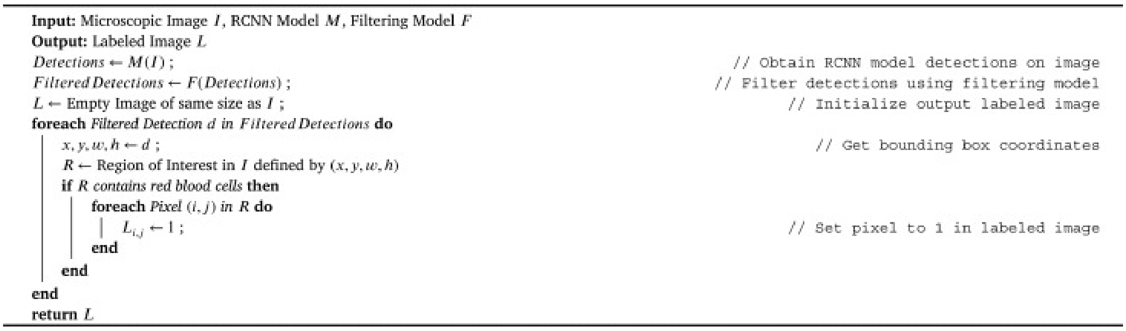
$$I'\,(i,j) = \frac{1}{W\,(i,j)} \sum_{p=1}^{m} \sum_{q=1}^{n} w\,(i,j,p,q) I\,(p,q) \tag{4}$$

while $W(i,j)$ denotes the total accumulated weight from all surrounding pixels.

In Equation (1), the term $w(i,j,p,q)$ is obtained by multiplying the spatial component $wd(i,j,p,q)$ with the intensity-based component $wr(i,j,p,q)$. Equation (2) specifies how wd is derived, Equation (3) provides the formulation for wr, and Equation (4) outlines how the filtered image I' is produced.

The Radial Gradient Index filtering method described in Algorithm 1 takes advantage of separable kernels and recursive operations for efficient computation. The filter is repeatedly applied until the desired smoothing is reached. By tuning σd and σr, one can regulate both the degree of noise reduction and how well edges are preserved. The Radial Gradient Index procedure serves as a computational strategy to enhance or extract particular structures within digital images. It works by determining the gradient at each pixel—representing the magnitude of local intensity variation—and arranging this gradient information radially around a focal region. Through additional actions such as thresholding or convolution on this indexed gradient map, the method can emphasize certain features, including edges or surface patterns. This makes it a valuable tool for preliminary image enhancement and targeted feature extraction, with applications across image processing, computer vision, and pattern recognition.

Algorithm 2 provides the procedure for assigning labels to red blood cells. The RCNN identifies RBCs in the microscopic input and returns bounding boxes for each detection. The filtering module then removes incorrect detections, and the remaining valid bounding boxes are used to mark RBC locations in the output image. Labeling is completed by setting every pixel inside each retained bounding box to 1 within the generated labeled image.



```
Input: Microscopic Image $I$, RCNN Model $M$, Filtering Model $F$
Output: Labeled Image $L$
$Detections \leftarrow M(I)$ ;                                    // Obtain RCNN model detections on image
$FilteredDetections \leftarrow F(Detections)$ ;                  // Filter detections using filtering model
$L \leftarrow$ Empty Image of same size as $I$ ;                  // Initialize output labeled image
foreach Filtered Detection $d$ in $FilteredDetections$ do
    $x, y, w, h \leftarrow d$ ;                                   // Get bounding box coordinates
    $R \leftarrow$ Region of Interest in $I$ defined by $(x, y, w, h)$
    if $R$ contains red blood cells then
        foreach Pixel $(i, j)$ in $R$ do
            $L_{i,j} \leftarrow 1$ ;                              // Set pixel to 1 in labeled image
        end
    end
end
return $L$
```

Algorithm 2. Labeling red blood cells using rcnn model with filtering.

Let I denote a grayscale microscopic image of size $m \times n$. Each pixel is given an intensity $I(i,j)$ for i=1,2,…,m and j=1,2,…,n.

Image segmentation procedures are then applied to identify RBCs. Segmentation refers to partitioning the image into meaningful parts, such as separating objects from the background. For RBC detection, a commonly used approach is threshold-based segmentation. With this method, a threshold T is first selected to differentiate between foreground (RBCs) and background regions. Each pixel is then categorized accordingly, as shown in Equation (5):

$$\mathrm{B}\,(\mathrm{i},\mathrm{j}) = \begin{cases} 1, \text{if } \mathrm{I}(\mathrm{i},\mathrm{j}) > \mathrm{T} \\ 0, \text{otherwise} \end{cases} \{0, \text{otherwise}$$

$$\mathrm{F}\,(\mathrm{i},\mathrm{j}) = \begin{cases} 1, \text{if I } (\mathrm{i},\mathrm{j}) \leq \mathrm{T} \\ 0, \text{otherwise} \end{cases} \qquad (5)$$

where $B(i,j)$ and $F(i,j)$ represent the binary background and foreground images, respectively.

Once thresholding is performed, the segmentation can be improved by incorporating morphological procedures such as erosion and dilation. These steps assist in eliminating residual noise or tiny artifacts in the foreground and in merging adjacent components that should form a single object.

Let S denote the structuring element applied during the morphological processes. The erosion and dilation operations are expressed in Equation (6) as follows:

$$\mathrm{B}_{\text{eroded}}(\mathrm{i},\mathrm{j}) = \min_{(\mathrm{p},\mathrm{q}) \in \mathrm{S}} \mathrm{B}(\mathrm{i}+\mathrm{p},\mathrm{j}+\mathrm{q}) \qquad (6)$$

$$\text{Fdilated}(\text{i}, \text{j}) = \underset{(\text{p}, \text{q}) \in \text{S}}{\text{mix}} \text{F}(\text{i} + \text{p}, \text{j} + \text{q})$$

where $B_{\text{ero}}\text{d}_{\text{e}}$d and $F\text{d}_{\text{ilate}}$d denote the background and foreground binary outputs after the erosion and dilation steps have been applied.

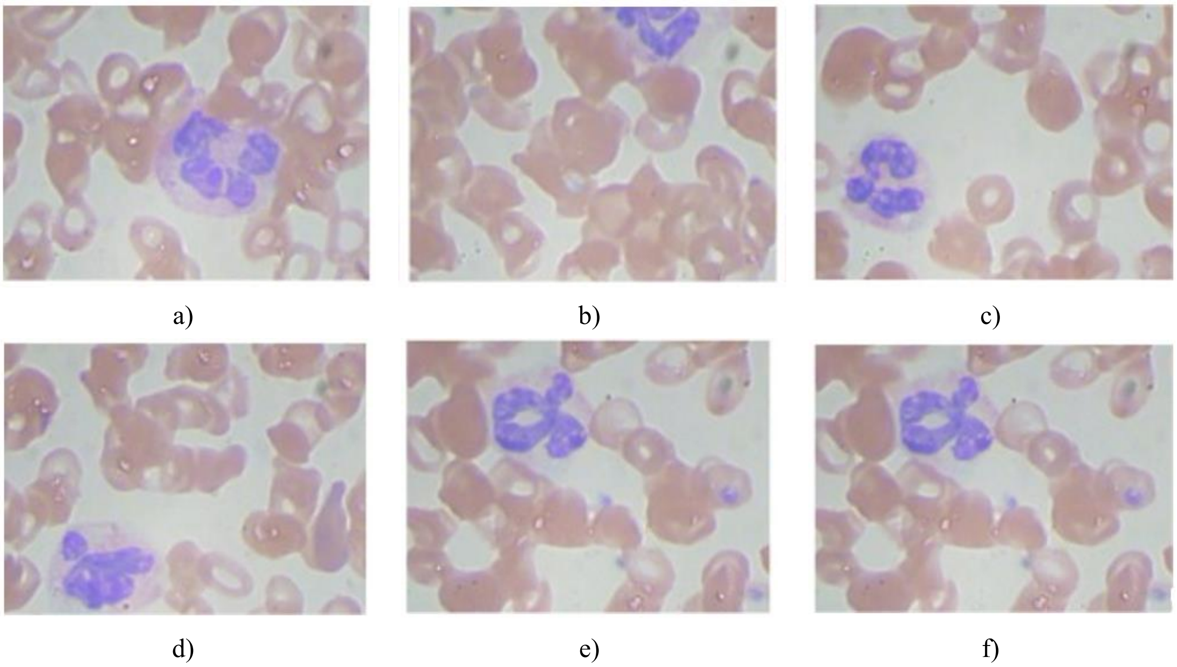
To produce the final segmented output, the processed foreground and background binary images are merged as follows:

$$\text{R}(\text{i}, \text{j}) = \begin{cases} 1, \text{if } \text{F}_{\text{dilated}}(\text{i}, \text{j}) = 1 \text{ and } \text{B}_{\text{eroded}}(\text{i}, \text{j}) = 0 \\ 0, \text{otherwise} \end{cases} \quad (7)$$

where R denotes the binary mask corresponding to the identified red blood cells.

In Equation (5), the background and foreground binary images are specified after applying the threshold. Equation (6) provides the definitions of erosion and dilation, and Equation (7) yields the final segmentation by merging the foreground and background binary outputs.

In practical settings, both the threshold T and the structuring element S are selected according to image characteristics and the precision level needed for segmentation. RCNN has emerged as a significant approach in image analysis and speech-related tasks. By sharing weights, the model decreases the overall parameter count and computational burden. RCNN is particularly effective when handling multidimensional inputs, with images serving as the most common data type. Unlike classical feature-engineering methods, RCNNs bypass the complexity of manual feature extraction and reconstruction. A multi-layer perceptron is structured to recognize patterns within 2D image data and can adapt to transformations including translation, rotation, and scaling. To enhance recognition accuracy, we utilize an RCNN-driven strategy. Typically, RCNN architectures include several feature-extraction layers, as illustrated in **Figure 1**. The first layer ingests the image and generates an input representation that corresponds to the training image dimensions. Subsequent convolution and subsampling layers extract informative patterns while minimizing redundant processing. A fully connected layer then selects relevant features and converts the 2D feature representation into 1D, enabling evaluation through various classifier metrics. Finally, the classifier differentiates the cells based on extracted characteristics and assigns categories accordingly. Representative blood sample images are presented in **Figure 2**.



a) b) c)

d) e) f)

Figure 2. Samples of clinical microscopic images.

Training process of RCNN

Object detection in a standard RCNN framework is typically carried out in four stages. First, candidate regions are generated within the image through a region-proposal strategy. Next, RCNN feature extraction is applied to those regions, producing fixed-length feature vectors. Then, each region is classified using its extracted attributes. The RCNN ultimately refines and merges bounding boxes using regression to produce accurate object locations. In our proposed configuration for precise RBC quantification and categorization, we concentrate on the Region of Interest (ROI) and combine multiple object cues to ensure reliable detection. RCNN (Region-based Convolutional Neural Network) serves as a deep learning system specialized for object identification in images. Its training procedure is divided into the following steps:

- **Dataset Preparation:** The first stage involves assembling the dataset. This includes gathering numerous images containing the target objects or regions, marking those images with bounding boxes, and splitting the dataset into training and testing subsets. All input object anchors are mapped to the conv5-3 feature map, and the RPN network performs a forward pass to determine anchor scores and bounding box regression outputs.
- **Feature Extraction:** RCNN employs a pre-trained Convolutional Neural Network for feature extraction. This CNN is typically trained on a large database, and its parameters remain unchanged during RCNN training. Proposal box locations are computed using anchor coordinates along with the bounding box regression offsets.
- **Region Proposal:** The next step is to generate possible object regions. A selective search algorithm is used to produce a set of candidate locations likely to contain targets. These are known as region proposals. Coordinates that exceed image boundaries are clipped by setting their minimum and maximum values to 0 and the image's width or height, respectively.
- **Region of Interest (RoI) Pooling:** The proposed regions are fed into the CNN for feature extraction. Because proposal sizes differ, they must first be normalized to a fixed size through RoI pooling. Regions failing to meet minimum size constraints (based on height or width) are removed.
- **Classification:** Features from each proposal are then passed through fully connected layers that assign each region to an object category or label it as background. After this filtering process, the remaining proposals are ranked by their object scores. A predefined number, pre_nms_topN (e.g., 6000), is then selected.
- **Localization:** In addition to classification, RCNN predicts bounding box coordinates to tightly enclose the object within each proposal. Following non-maximum suppression, the surviving proposals are pruned once again to obtain the final detections. The top post_nms_topN proposals (e.g., 300) based on foreground probability after NMS are retained.
- **Training:** RCNN training utilizes a multi-task loss function that integrates both classification loss and localization loss. Classification loss measures the deviation between predicted and true categories, while localization loss evaluates the difference between predicted and actual bounding box parameters.
- **Fine-tuning:** After initial training, the RCNN can be fine-tuned on a domain-specific dataset to enhance performance for that particular application.
- **Testing:** During evaluation, the RCNN receives new images and generates object proposals. The network assigns classes and predicts bounding box coordinates. The final result consists of bounding boxes around detected objects together with their category labels.

Experiments

Dataset

The dataset used in this study was obtained from the Kaggle repository and includes microscopic blood-cell imagery. It consists of 12,500 augmented JPEG images spanning four categories of blood cells: Eosinophil, Lymphocyte, Monocyte, and Neutrophil, with roughly 3000 samples for each class. The class annotations are supplied through CSV files. In addition to the augmented set, there are 410 pre-augmentation images, each labeled into two sub-groups (WBC and RBC) using XML annotations.1

Experimental setup

(i) To validate the contribution of the proposed technique, we conducted several trials using a desktop machine equipped with an Intel Pentium Core™ i7-6700 (3.7 GHz) CPU and 16 GB RAM. A k-fold validation procedure was employed to assess model robustness. The dataset was divided into 60/20/20 segments for training, validation, and testing. Ten equivalent sub-groups were randomly drawn from the full dataset. For each cycle, nine were used for training, and one for evaluation, and this rotation was repeated 10 times. Final metrics were derived by averaging all iterations.

(ii) The first test evaluates the proposed framework by validating it across the full image set.

(iii) The second test involves training and evaluating a classifier for blood-cell structure without applying the proposed method. Here, multiple established feature-extraction techniques were used, such as local directional pattern, local binary pattern, local directional pattern variance, wavelet-based methods, the Hough transform, SIFT, curvelet-based approaches, deformable models, template-matching strategies, and unitary image transforms.

(iv) The last phase compares the performance of the new model against earlier feature-extraction approaches.

Accuracy and computational cost of deep-learning-based augmentation

This section focuses on enhancing the dataset size through augmentation to reduce overfitting. **Table 1** outlines the augmentation configuration adopted for the training images. After augmentation, microscopic images were processed using an R-CNN, and the results were compared with ResNet and GoogleNet variants. R-CNN delivered better accuracy while also requiring less processing time.

Table 1. Parameter settings for feature extraction.

Parameter	Value	Description
fill_mode	nearest	Pixel-filling method for areas outside the transformed image boundaries (nearest-neighbor interpolation)
rescale	1/255	Pixel value normalization (scales intensity values from [0–255] to [0–1])
rotation_range	20°	Maximum rotation angle in degrees (random rotations uniformly sampled from ±20°)
width_shift_range	0.2	Fractional width shift for random horizontal translations (±20% of image width)
height_shift_range	0.1	Fractional height shift for random vertical translations (±10% of image height)
shear_range	0.1	Shear intensity (shear angle in radians ≈ 0.1 rad ≈ 5.7°)
zoom_range	0.5	Random zoom range [1–0.5, 1+0.5] → zoom between 50% and 150% of original size
horizontal_flip	True	Randomly applies horizontal flipping (mirroring) with 50% probability
vertical_flip	False	Vertical flipping disabled (not applied)

Results and discussion

The main outcomes show strong accuracy, short runtime, improved handling of overlapping cells, and higher robustness when combined with preprocessing. The comparative evaluation confirms that the proposed method provides advantages over existing techniques. Since RCNN requires uniform input dimensions, all microscopic images were resized to 24×24. The RCNN yielded 99.06% accuracy during training and 91.21% accuracy during testing. The loss rates were also low—2.017% for training and 3.312% for testing. Execution times were 2091 s for training and 1474 s for testing.

Figure 3 illustrates that most models achieve accuracy values close to or exceeding 90%, with RCNN showing the highest performance. The training–testing accuracy differences across models remain minimal, suggesting reliable generalization. However, ResNet-18 and ResNet-152 reveal slightly larger gaps, hinting at mild overfitting.

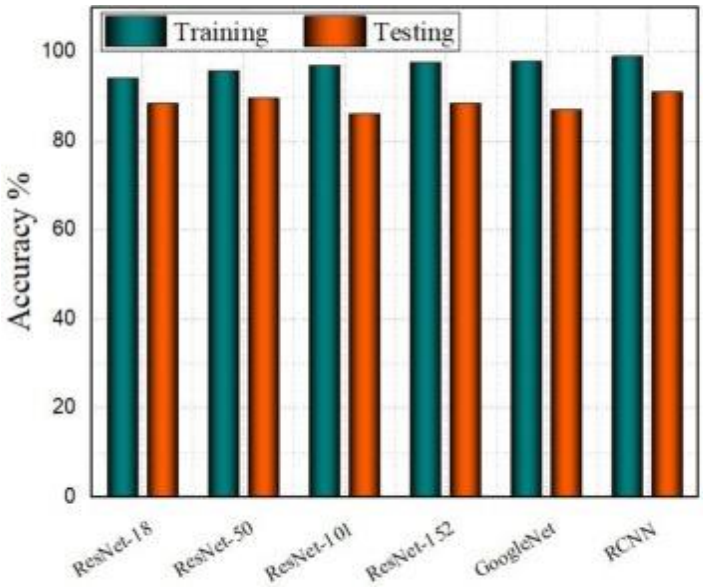


Figure 3. Training and testing accuracy for resnet-18, ResNet-50, ResNet-101, ResNet-152, GoogleNet, and RCNN.

Figure 4 presents model loss comparisons. RCNN displays the smallest losses in both training and testing. ResNet-50 and ResNet-101 follow with relatively low training losses. ResNet-152 shows moderate loss values, while ResNet-18 exhibits the highest loss among the ResNet series, though still low overall. The smaller differences between training and testing losses for ResNet-18, ResNet-50, ResNet-101, and RCNN indicate stronger generalization performance.

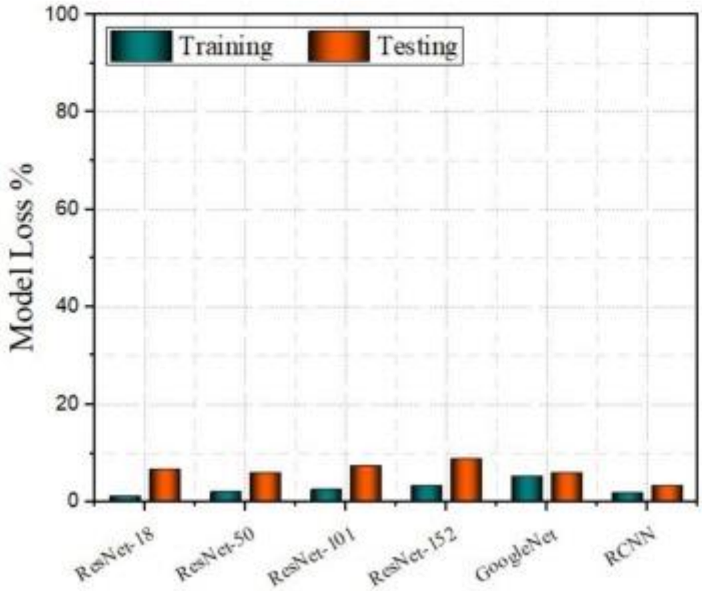


Figure 4. Loss Values for ResNet-18, ResNet-50, ResNet-101, ResNet-152, GoogleNet, and RCNN.

Figure 5 reports computational time. ResNet-152 requires the most time—11315 s for training and 9576 s for testing. GoogleNet also shows significant runtimes (5201 s and 2001 s). RCNN is the fastest among all compared methods, finishing in 2091 s for training and 1474 s for testing. A visual sample of red-blood-cell detection appears in **Figure 6.**

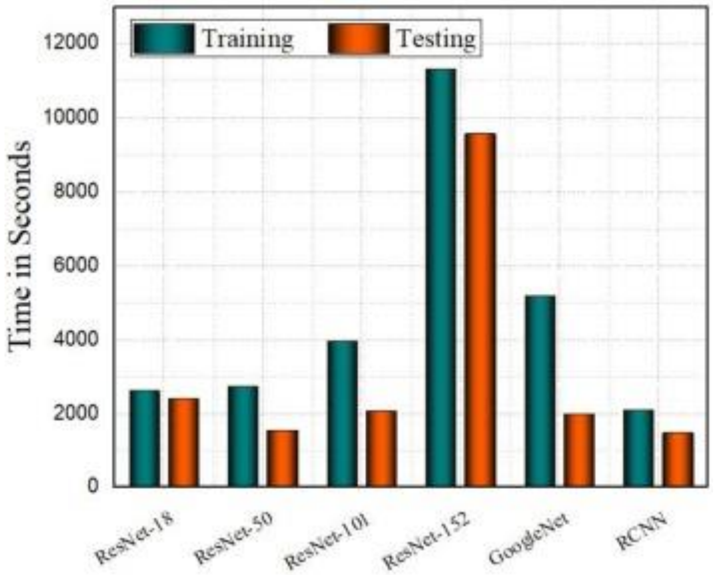
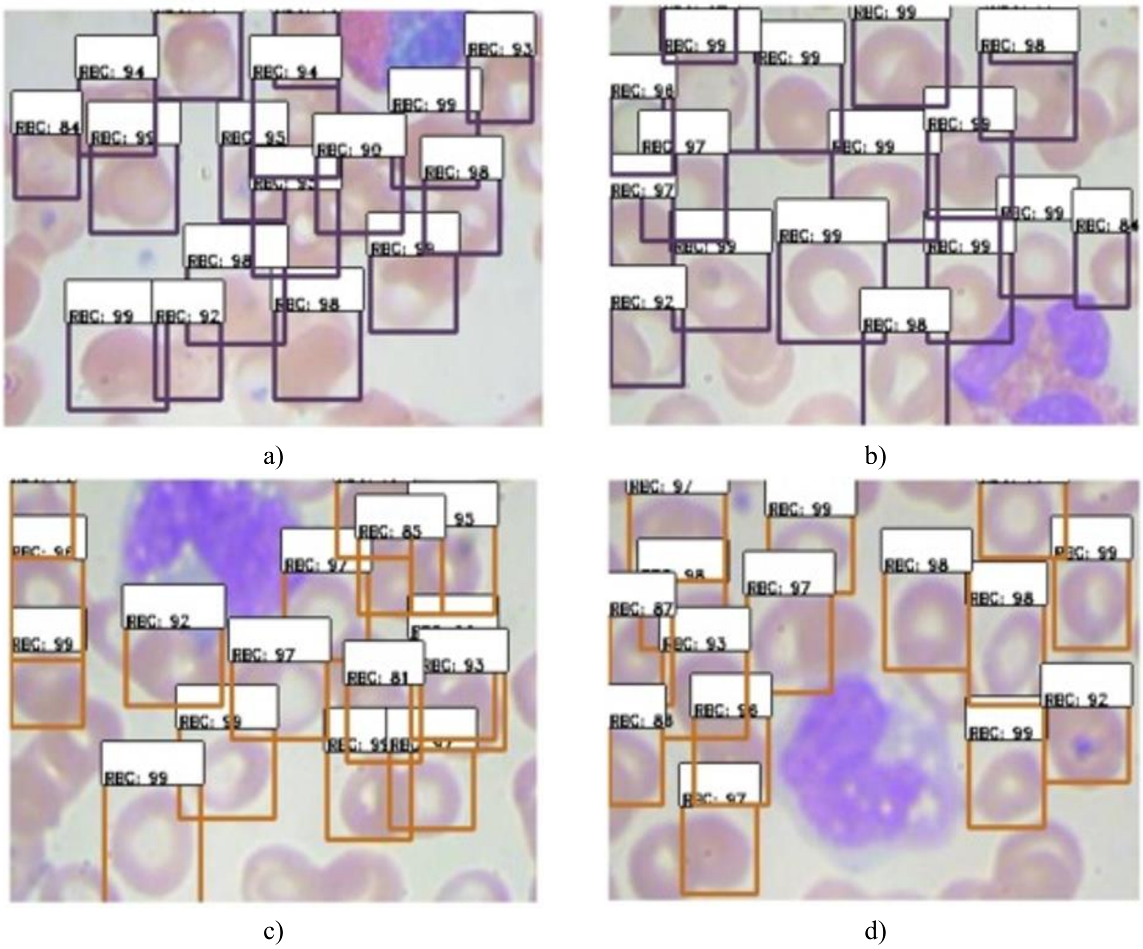


Figure 5. Illustrates the runtime for four ResNet architectures, GoogleNet, and RCNN. The Y-axis spans 0 to 12000 seconds, reflecting execution duration, while the X-axis lists each deep learning model. Teal bars correspond to training time, and orange bars indicate testing time.

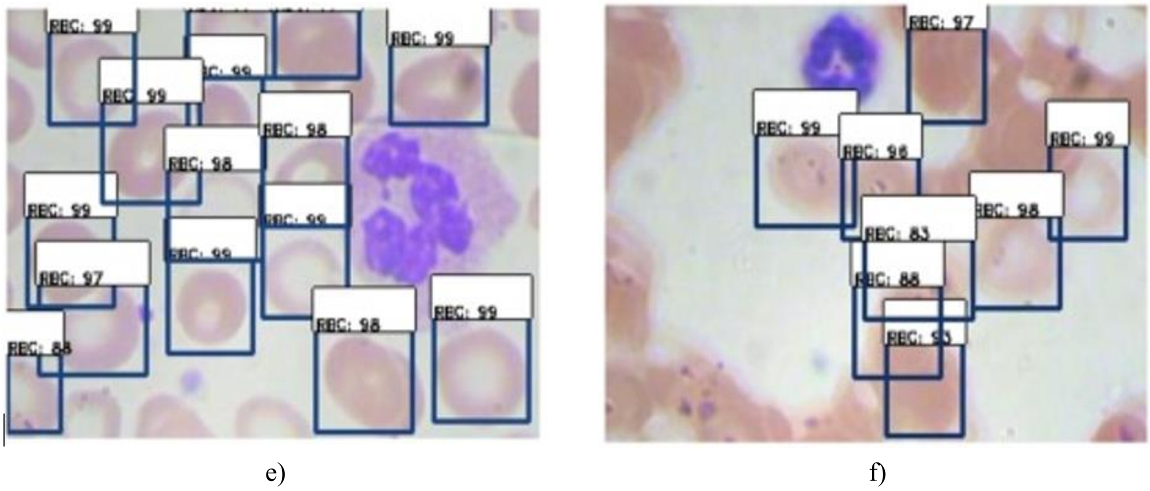


a)

b)

c)

d)



e) f)

Figure 6. Presents the detection of Red Blood Cells. Dark brown boxes (top two images) mark accurate detections, light brown boxes (middle two images) highlight missed cells or false negatives, and dark blue boxes (bottom two images) denote false positives occurring in background regions.

Observations

*ResNet Series***:** Increasing depth from ResNet-18 through ResNet-152 tends to reduce training loss and raise training accuracy. Still, ResNet-152 displays slightly higher test loss than ResNet-50 and ResNet-101, suggesting potential overfitting in the deepest variant.

*GoogleNet***:** Relative to the ResNet family, GoogleNet exhibits higher losses in both training and testing. While its accuracy is competitive, it does not surpass the top-performing models. Its training duration is longer than most ResNets but less than ResNet-152.

*RCNN***:** RCNN records the highest training accuracy among all models and achieves strong test accuracy. However, its losses exceed those of many ResNet versions, and its training time falls between GoogleNet and ResNet-152. The proposed framework exceeded typical models such as ResNet-50 and GoogLeNet by over 10–15% in accuracy, demonstrating its strong performance on the same dataset.

Limitations of current research and future directions

The following points outline the constraints of the current study and suggest directions for future work to build more dependable systems for blood cell assessment:

1. The training and testing samples were limited in number or diversity, which may restrict generalization. Broader datasets spanning additional cell or tissue types and varied imaging methods should be explored in subsequent studies.
2. The images analyzed were confined to particular cell categories, magnifications, or imaging styles. Including more varied image groups—such as additional blood cell sub-types, including platelets—would help uncover broader model limits.
3. The study focused solely on cell detection and classification, with no extended morphological evaluation. Future investigations could assess detailed structural features rather than simply identifying or counting cells.
4. Most evaluations were conducted under ideal imaging conditions, leaving performance under noise or artifacts uncertain. Upcoming work should assess resilience in suboptimal or degraded images.
5. Comparisons were restricted to deep learning models, without assessing traditional image processing approaches. Advancing techniques for separating or reconstructing overlapping cells may provide significant improvements.
6. Distinguishing overlapping cells remains a challenge, highlighting a current limitation. Subsequent research should move beyond patch-level inspections toward full blood smear analysis.

Conclusion

This work introduces a multi-modal methodology that fuses deep learning with technological tools to locate relevant regions within microscopic images and detect red blood cells based on tissue characteristics. The framework consists of three major components: data preprocessing, feature extraction, and image classification. Preparatory steps such as filtering and augmentation help refine the dataset, lowering noise and improving computational performance. Feature extraction, performed using RCNN combined with biosensor-derived information, enables the model to capture meaningful representations and compare its effectiveness against ResNet and GoogleNet. Classification then highlights regions of interest guided by tissue properties.

The resulting model achieves high accuracy and is suitable for supporting the diagnosis and management of various medical conditions. Overall, the proposed system delivers strong performance with reduced computational time. This integrated multi-modal pipeline offers a promising avenue for advancing microscopic image-based diagnostics.

Acknowledgments: This project was supported by the Deanship of Scientific Research at Najran University (Grant No: NU/RG/SERC/12/3) and research funding from the Yangtze Delta Region Institute (Huzhou), University of Electronic Science and Technology of China (Grant No. U03210068).

Conflict of Interest: The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Riaz Ullah Khan reports article publishing charges and statistical analysis were provided by Yangtze Delta Region Institute (Huzhou), University of Electronic Science and Technology of China. Sultan Almakdi reports article publishing charges was provided by Najran University. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Financial Support: None

Ethics Statement: None

References

1. K.T. Navya, K. Prasad, B.M.K. Singh, Analysis of red blood cells from peripheral blood smear images for anemia detection: a methodological review, Med. Biol. Eng. Comput. 60 (9) (2022) 2445–2462.
2. F.V. Nurçin, E. Imanov, Selective hole filling of red blood cells for improved marker-controlled watershed segmentation, Scanning 2021 (1) (2021) 1–9.
3. S. Savkare, S. Narote, Automatic system for classification of erythrocytes infected with malaria and identification of parasite's life stage, Proc. Technol. 6 (2012) 405–410.
4. D. Khalid, A. Al-Saedi, S. Savas¸, Classification of skin cancer with deep transfer learning method, J. Comput. Sci. IDAP-2022 (2022) 202–210.
5. R. Ali, J. Alhatemi, S. Savas¸, Transfer learning-based classification comparison of stroke, J. Comput. Sci. (2022) 192–201, IDAP-2022: International Artificial Intelligence and Data Processing Symposium.
6. G.C. Kabat, M.Y. Kim, J.E. Manson, L. Lessin, J. Lin, S. Wassertheil-Smoller, T.E. Rohan, White blood cell count and total and cause-specific mortality in the women's health initiative, Am. J. Epidemiol. 186 (1) (2017) 63–72.
7. O. Malik Mohamed, Normal reference value of blood cell count, red, white and..., Al Neelain Med. J. 3 (8) (2013) 100–109.
8. S. Savas¸, N. Topalog˘ lu, Ö. Kazcı, P.N. Kos¸ ar, Performance Comparison of Carotid Artery Intima Media Thickness Classification by Deep Learning Methods, HORA2019 - International Congress on Human-Computer Interaction, Optimization and Robotic Applications, vol. 4, SETSCI, Ürgüp, Turkey, 2019, pp. 125–131.
9. R.U. Khan, S. Almakdi, M. Alshehri, R. Kumar, I. Ali, S.M. Hussain, A.U. Haq, I. Khan, A. Ullah, M.I. Uddin, Probabilistic approach to COVID-19 data analysis and forecasting future outbreaks using a multi-layer perceptron neural network, Diagnostics 12 (12) (2022) 2539.

10. A. Gangadhar, H. Sari-Sarraf, S.A. Vanapalli, Deep learning assisted holography microscopy for in-flow enumeration of tumor cells in blood, RSC Adv. 13 (7) (2023) 4222–4235.
11. S. Sharma, S. Gupta, D. Gupta, S. Juneja, P. Gupta, G. Dhiman, S. Kautish, Deep learning model for the automatic classification of white blood cells, Comput. Intell. Neurosci. 2022 (1) (2022) 1–13.
12. C. Raillon, J. Che, S. Thill, M. Duchamp, B.X. Desbiolles, A. Millet, E. Sollier, P. Renaud, Toward microfluidic label-free isolation and enumeration of circulating tumor cells from blood samples, Cytometry, Part A 95 (10) (2019) 1085–1095.
13. J. Yoon, W.S. Jang, J. Nam, D.C. Mihn, C.S. Lim, An automated microscopic malaria parasite detection system using digital image analysis, Diagnostics 11 (3) (2021) 527.
14. D. Cruz, C. Jennifer Valiente, L.C. Castor, C.M.T. Mendoza, B.A. Jay, L.S.C. Jane, P.T.B. Brian, Determination of blood components (WBCs, RBCs, and Platelets) count in microscopic images using image processing and analysis, in: HNICEM 2017 - 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management, vol. 2018-Janua, Institute of Electrical and Electronics Engineers Inc., 2017, pp. 1–7.
15. N.S. Fatonah, H. Tjandrasa, C. Fatichah, Automatic leukemia cell counting using iterative distance transform for convex sets, Int. J. Electr. Comput. Eng. 8 (3) (2018) 1731–1740.
16. T.C. Chen, K.L. Chung, An efficient randomized algorithm for detecting circles, Comput. Vis. Image Underst. 83 (2) (2001) 172–191.
17. M. Maitra, R. Kumar Gupta, M. Mukherjee, M. Maitra, R. Kumar Gupta, M. Mukherjee, Detection and counting of red blood cells in blood cell images using Hough transform, Int. J. Comput. Appl. 53 (16) (2012) 13–17.
18. N.H. Mahmood, P. Lim, S. Mazalan, M. Abdul Razak, Blood cells extraction using color based segmentation technique, Int. J. Life Sci. Biotechnol. Pharma Res. 2 (2) (2013) 233–240.
19. N.H. Mahmood, M.A. Mansor, Red blood cells estimation using Hough transform technique, Signal Image Process. 3 (2) (2012) 53.
20. B. Venkatalakshmi, K. Thilagavathi, Automatic red blood cell counting using Hough transform, in: IEEE Conference on Information and Communication Tech- nologies, ICT 2013, 2013, pp. 267–271.
21. D.R. Chen, R.F. Chang, W.J. Wu, W.K. Moon, W.L. Wu, 3-D breast ultrasound segmentation using active contour model, Ultrasound Med. Biol. 29 (7) (2003) 1017–1026.
22. J. Shan, H.D. Cheng, Y. Wang, Completely automated segmentation approach for breast ultrasound images using multiple-domain features, Ultrasound Med. Biol. 38 (2) (2012) 262–275.
23. D. Díaz-Pernil, A. Berciano, F. Peña-Cantillana, M.A. Gutiérrez-Naranjo, Bio-inspired parallel computing of representative geometrical objects of holes of binary 2D-images, Int. J. Bio-Inspir. Comput. 9 (2) (2017) 77–92.