

# Automatic classification of 10 blood cell subtypes using transfer learning via pre-trained convolutional neural networks



Rabia Asghar<sup>a</sup>, Sanjay Kumar<sup>b</sup>, Paul Hynds<sup>a,c,\*</sup>

<sup>a</sup> Environmental Sustainability and Health Institute, Technological University (TU), Dublin, Ireland

<sup>b</sup> Department of Electronic and Computer Engineering, University of Limerick, Ireland

<sup>c</sup> Spatiotemporal Environmental Epidemiology Research (STEER) Group, Technological University (TU), Dublin, Ireland

## ARTICLE INFO

### Keywords:

Blood cell subtypes  
Machine learning  
Classification  
Feature extraction  
Pre-trained models  
Deep learning  
Image analyses  
Autoimmune diseases

## ABSTRACT

Human blood is primarily composed of plasma, red blood cells, white blood cells, and platelets. It plays a vital role in transporting oxygen and nutrients to all organs, and stores essential health-related data about the human body. Blood cells are utilized to defend the body against infections and disease. Hence, analysis of blood permits physicians to assess an individual's physiological condition. Blood cells are sub-classified into eight groups: Neutrophils, eosinophils, basophils, lymphocytes, monocytes, immature granulocytes (promyelocytes, myelocytes, and metamyelocytes), erythroblasts, and platelets (thrombocytes) on the basis of their nucleus, shape and cytoplasm. Traditionally, pathologists and hematologists have identified and examined these via microscopy prior to manual classification, with this manual approach being slow and prone to human error. Therefore, it is essential to automate this process. In the current study, transfer learning with a series of pre-trained Convolutional Neural Network (CNN) models—VGG16, VGG19, ResNet-50, ResNet-101, ResNet-152, InceptionV3, MobileNetV2 and DenseNet-201 was applied to the normal peripheral blood cells dataset (PBC). The overall accuracy achieved with individual CNNs ranged from 91.4 % to 94.7 %. Based on these pre-trained architectures, a CNN-based architecture has been developed to automatically classify all ten blood cell types. The proposed transfer learning CNN model was tested on blood cell images from the PBC, Kaggle and LISC datasets. Achieved accuracy was 99.91 %, 99.68 % and 98.79 %, respectively, across these three datasets. The presented CNN architecture outperforms all previous results reported in the scientific/medical literature with a high capacity for framework generalization in future applications of blood cell classification.

## 1. Introduction

Blood acquires oxygen from the lungs and transports it to all human organs and cells for metabolic operations. Blood cells also transport hormones and eliminate unnecessary materials for eventual elimination via the liver, kidneys, and intestine. Blood is comprised of plasma which forms the liquid portion, in addition to cell fragments. These cell fragments are composed of approximately 1 % white blood cells (WBCs), which are responsible for immunity, 40–50 % red blood cells (RBCs), which transport oxygen and carbon dioxide, while platelets have the crucial role of ensuring blood clotting [1,2]. WBCs can be divided into two general groups based on the presence of granules: granulocytes and agranulocytes (non-granulocytes). Lymphocytes and monocytes are classified as agranulocytes, while neutrophils, eosinophils, and basophils are considered granulocytes. Undeveloped WBCs, also known as

immature granulocytes (IGs), are expelled from bone marrow into the blood. The presence of IGs (promyelocytes, myelocytes, and metamyelocytes) in the blood signifies an early reaction to infection, swelling, or another issue related to the bone marrow such as leukemia (this is not the case in newborn infants or pregnant women). WBCs defend against infection, breaking down the foreign proteins present in bacteria, viruses, and fungi. WBCs fight infections and diseases by recognizing, identifying, and attaching themselves to foreign antigens [3]. Red blood cells (RBCs), also known as erythroblasts, help tissues produce energy by delivering appropriate oxygen. When energy is produced, waste in the form of carbon dioxide is formed. RBCs are responsible for transport of this waste stream to the lungs for exhalation. Erythroblasts are immature RBCs typically present in the blood of newborn infants (0–4 months), with their presence in human blood after the neonatal period (>4 months) typically indicating severe problems

\* Corresponding author. Environmental Sustainability and Health Institute, Technological University (TU), Dublin, Ireland.

E-mail address: [paul.hynds@tudublin.ie](mailto:paul.hynds@tudublin.ie) (P. Hynds).

such as damaged bone marrow, stress, or malignant tumors potentially leading to cancers. Platelets, also known as thrombocytes, are essential for immune system operation; their primary responsibility is to stop bleeding. If bleeding starts via an injury or blood vessel damage, the brain sends alert signals to the platelets - platelets subsequently inundate the wounded area, cluster, and form a clot, thus sealing the blood vessel. They also play an important role in tissue repair and remodeling to prevent tumor progression and leakage of vesicant fluids. The typical proportions of neutrophils in the blood are 0–6%. Eosinophils comprise 1–3%, basophils 0–1%, lymphocytes 25–33%, and monocytes 3–10% of the leukocytes suspended in the blood [4].

Accurate classification of blood cells represents a crucial research area for scientists in diagnosing diseases affecting blood cells. Blood cell classification using microscopic images can only be manually undertaken by medical professionals with the necessary experience and training. Blood is analyzed in two different ways, the first being a complete blood count (CBC) test that calculates the total percentage of RBCs, WBCs, and platelets. The second method is the peripheral blood smear(s) (PBS) test. These results represent the patient's overall health, whereby microscopic blood imagery are employed to identify the type (s) of RBCs, WBCs, and platelets, thus enabling early disease diagnosis. As presented above, each blood cell type serves a distinct purpose, with a change in the number (and subsequent proportion) of blood cell types potentially signalling an infection or disease. For example, a low WBC count may be reflective of several illnesses, including blood cancer, while a low RBC count typically results in onset of anemia [5]. Conventional blood cell type detection methods are slow and characterized by relatively low accuracy, highlighting the significance of accurate systems for rapid and precise blood cell analyses [6]. Microscopic blood smear images have been categorized using conventional machine learning (ML) techniques including support vector machines, decision trees, k-nearest neighbor, naive Bayes, and artificial neural networks [6]. The general process for traditional ML approaches includes pre-processing of blood smear images, segmentation for cell division, feature extraction, feature selection to remove undesired data, and subsequent classification [7]. Despite many promising results, feature extraction and selection typically affect the classification performance of classical ML algorithms. Selecting the optimal features and finding the appropriate feature extraction algorithm has become complex and time-consuming [7]. Several deep learning (DL) techniques including convolutional neural networks (CNNs) have recently been proposed as an appropriate solution [8] (See Section II). In contrast to conventional ML approaches, DL-based approaches comprise autonomous feature extraction and selection, with recent research revealing that CNN outperformed traditional ML techniques for classifying blood cells [8,9]. Currently, there is a significant demand for the utilization of convolutional neural networks to improve the performance of classification systems for various blood cells. Convolutional neural networks, unlike traditional image processing methods, work directly with raw imagery and automatically learn features.

Accordingly, the present study sought to:

- Develop a convolution neural network (CNN) based architecture with high generalization ability for classification of multiple blood cell types,
- To do this, we applied transfer learning to evaluate the individual performance of eight pre-trained CNN models: VGG16, VGG19, ResNet-50, ResNet-101, ResNet-152, InceptionV3, MobileNetV2, and DenseNet-201 on a microscopic image dataset [9].
- The datasets used for this study include: PBC, containing 17,092 images of size 360 x 363; Kaggle, with 12,500 images size 320 x 240; and LISC, comprising 10,000 images of size 720 x 576
- Subsequently, a CNN architecture for the classification of ten blood cells subtypes is presented and diagnosed.

- The proposed CNN architecture consists of eight convolutional layers, eight pooling layers, five fully connected hidden layers, and an output layer.

## 2. Related work

While a large volume of research has been undertaken in medical image classification and segmentation, few researchers have chosen to use manually crafted features for classification purposes.

Elhassan et al. proposed a two-step deep learning model for categorizing atypical lymphocytes and immature WBCs [10]. The problem of a significantly unbalanced distribution of WBCs in blood samples was addressed using the 'GT-DCAE WBC hybrid augmentation model' based on geometric transformation (GT) and a deep convolutional autoencoder (DCAE). A hybrid multi-classification model known as the "two-stage DCAE-CNN atypical WBC classification model" was developed to divide atypical WBCs into eight categories. The final model's average accuracy, sensitivity, and precision were 97 %, 97 %, and 98 %, respectively. Ahmad et al. presented an improved hybrid method for optimum deep feature extraction using DenseNet201 and Darknet53 [11]. Dominant characteristics were subsequently selected using an entropy-controlled marine predator algorithm (ECMPA) on a public dataset of 5000 images of five distinct WBC subtypes. The system obtained an overall mean accuracy of 99.9 % in concurrence with reducing the size of the feature vector by more than 95 %. Singh et al. [12] presented a white blood cell classifier using CNN and multiple optimizers including SGD, Adadelta, and Adam, with a batch size of 32 and 10 epochs. Best outcomes were obtained by using the Adam optimizer, yielding performance parameter values of 97 % accuracy, 99 % recall, and 98 % F1 score. Darrin et al. presented a method to automatically classify imbalanced red blood cells from videos (6–100 frames) to monitor the status of sickle cell anemia patients [5]. The convolutional neural network (CNN) model and a recurrent CNN were combined to achieve an accuracy of 97 % and an F1-score of 0.94 (94 %). Rabul and Salam suggested the use of Otsu's thresholding with Gray Level Co-occurrence Matrix (GLCM) features [13]. The R, G, and B channels from original RGB imagery were separated from the Kaggle BCCD public data set to conduct image subtraction between the blue-red and blue-green channels. Subsequently, WBCs were extracted from B-G channels using Otsu's thresholding and morphological filtering. The features were subsequently decorrelated using ANOVA and a zero-phase component analysis (ZCA) whitening procedure. KNN classification produced an accuracy of 94.25 %. Milkisa et al. proposed a training strategy for neural networks to highlight malaria-infected red blood cell pixels using the NH malaria dataset [14]. Masked images were used to highlight diseased regions by dividing images into R, G, and B channels, with the intensity of the red channel subsequently increased. The proposed approach achieved 97.2 % accuracy.

A method for classifying Multiple Myeloma (MM) and Acute Lymphoblastic Leukemia (ALL) using the SN-AM dataset was developed by Deepika et al. [15]. The model was trained using an optimized Dense Convolution Neural Network framework with a minimal number of parameters and computation time, and was capable of identifying the type of cancer present in cells with 97.2 % precision. Likewise, Ansari et al. attempted to create a deep learning model with a customized architecture for identifying acute leukemia using lymphocyte and monocyte imagery. A Generative Adversarial Network (GAN) was employed to increase the size of this dataset [16]. The proposed CNN model based on the Tversky loss function for categorizing acute leukemia images comprised six convolution layers, four dense layers, and a SoftMax activation function, achieving 99.5 % accuracy.

To identify various blood cells in microscopic blood images, Alkafrawi and Ismail developed an AlexNet-based classification model [17]. Using CNNs, tests were undertaken on a dataset of 17,000 blood smear samples obtained from the Hospital Clinic of Barcelona. Five convolutional layers, three maximum pooling layers and three fully connected

layers comprised the AlexNet model, which was characterized by 95.1 % accuracy.

Kareem et al. classified blood cells using two distinct scenarios, the first using CNN directly and the second using SVM [18]. A dataset containing 10,295 cell images was employed, with CNN obtaining an accuracy of 98.4 %, compared to 90.6 % via SVM.

Meanwhile, Miserlis et al. proposed an AI-based diagnosis for Peripheral Arterial Disease (PAD), developing 11 distinct ANN models [19], namely DenseNet201, ResNet50v2, EfficientNetB0, and EfficientNetB7, achieving 97.22 % precision. Arif et al. proposed a framework for automatic leukemia detection based on deep learning [20]. The framework comprises several layers, including convolutional layers, batch normalization, leaky ReLU, and max pooling layers, with AlexNet used for leukemia identification. The proposed framework accurately classified images as normal or leukemia-affected and achieved 98.05 % accuracy, 97.59 % specificity, 100 % recall, and a 99.06 % F1 score.

Devi & Ambary [21] tested several CNN architectures to detect and classify WBCs. AlexNet, VGG16, GoogleNet, and ResNet50 were tested and analyzed in the convolution layer. The VGG16 CNN architecture was trained with transfer learning and outperformed all others with an accuracy of 97.59 %. Alnawayseh et al. proposed differential WBC counting to assess the immune system status of patients [22]. A CNN model with three layers: a convolutional layer, a downsized pooling layer, and fully connected hidden layers was trained. Weights were automatically attributed to datasets based on 96.5 % accuracy.

You Only Look Once version 5 (YOLOv5) was used by Luong et al. as a WBC classification and counting method for diagnosis of blood related diseases [23]. Blood cells were labeled with 619 leukemia cells, 115 neutrophils, 80 lymphocytes, 23 eosinophils, and 73 monocytes, with a classification accuracy of 93 %.

To automatically extract RBC features, Liao and Zhang suggested an ultrasonic RF signal convolutional neural network [24]. The RBCA-VGG10 model, a modified VGG16 variant with deleting layers, achieved 92.40 % accuracy, outperforming other models including LeNet (82.25 %), AlexNet (83.10 %), GoogleNet (87.85 %), and ResNet (84.06 %). Likewise, Guan and Wang described a blood cell image classification algorithm based on Efficient Net that used EfficientNet-B7 as the classification model and Contrast Limited Adaptive Histogram Equalization (CLAHE) to enhance image quality during data pre-processing [25]. The algorithm achieved 99.6 % accuracy. LeukoX is a technique developed to identify and categorize WBCs based on physical characteristics [26]. The results of the Least Entropy Combiner (LEC) network were directly fed to the proposed LEC network for learning. With an accuracy of 96.67 %, the model outperformed individual networks, with kappa and Matthew's correlation coefficient (MCC) values of 0.9334 and 0.9550, respectively.

### 3. Methods

#### 3.1. Blood cell data

The datasets used for the current research study were extracted from Normal Peripheral Blood Cells, Kaggle [27] and LISC [28].

##### A. Normal Peripheral Blood Cells Dataset

The open-access PBC dataset [9,17] comprises 17,092 images of normal blood cells (Table 1), collected using the CellVision DM96 analyzer in the Core Laboratory of the Hospital Clinic of Barcelona. The dataset is delineated into the following eight categories: Neutrophils, eosinophils, basophils, lymphocytes, monocytes, immature granulocytes (promyelocytes, myelocytes, and metamyelocytes), erythroblasts, and platelets or thrombocytes (Fig. 1). Experienced clinical pathologists labeled the images, all of which are characterized by 360 x 363 pixels in JPEG format. The patients whose images were employed during blood collection were free of infections, hematologic, or oncologic diseases

**Table 1**

Blood cell types and associated image number comprising the PBC dataset.  
**B. Kaggle Dataset.** The dataset is comprised of 12,500 JPEG images sized 320 x 240 [27], with each classification represented by 3000 images. Four distinct categories of white blood cells are included: Eosinophils, Lymphocytes, Monocytes, and Neutrophils (Fig. 2).  
**C. LISC Dataset.** This dataset includes images collated from the peripheral blood of 8 normal subjects. A total of 400 samples were extracted from 100 microscope slides, and stored in BMP format [28]. Rotation augmentation is employed on these images, with rotation angles set at 90, 180, and 270°. After augmentation, the LISC dataset comprises 10,000 720 x 576 images. The dataset includes four distinct white blood cell types (Fig. 3).

Cell Type	Image #
Neutrophils	3329
Eosinophils	3117
Basophils	1218
Lymphocytes	1214
Monocytes	1420
Immature granulocytes (Metamyelocytes, Myelocytes and Promyelocytes)	2895
Erythroblasts	1551
Platelets (Thrombocytes)	2348
Total	17,092

and were not in receipt of any pharmaceutical treatments. This high-quality labeled dataset has been employed in several studies to train and evaluate deep learning and machine learning models.

For training our model, all images within three datasets were resized to 360\*360 and delineated into three partitions for training (60 %), testing (20 %) and validation (20 %).

#### 3.2. Transfer learning

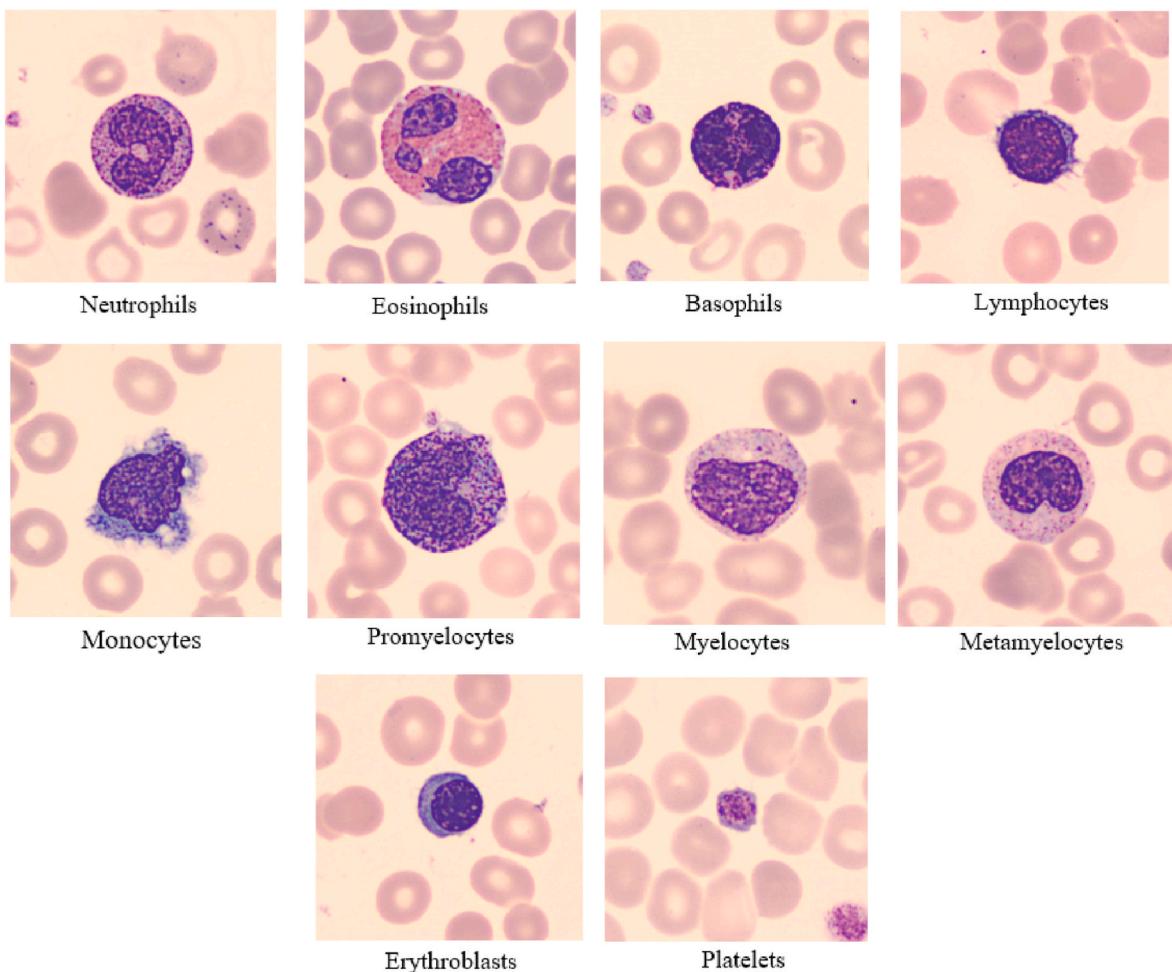
Transfer learning is a machine learning technique that enables utilization of an existing model (or several models), initially created for a particular task, to solve different tasks. This method brings several advantages by saving time and resources that would otherwise be required to construct "ground up" neural network models. It is particularly useful in computer vision and natural language processing fields, whereby it enhances the accuracy and performance of the resulting model(s).

In this study, transfer learning was employed by utilizing several existing CNN models, including VGG16, VGG19, ResNet50, ResNet101, ResNet-152, InceptionV3, MobileNetV2 and DenseNet201. This approach enabled leveraging the pre-trained weights and architectures of these models, leading to substantial reductions in training time while enhancing model accuracy.

VGG16 [29] is a deep convolutional neural network known for its significant contributions to image classification. It comprises 16 wt layers, including 13 convolutional layers, five max-pooling layers, and three fully connected layers. The network utilizes small receptive fields of 3x3 for uniform feature extraction. Rectified Linear Unit (ReLU) activation functions introduce non-linearity, while max pooling layers reduce spatial dimensions. The final layers include three fully connected layers, with SoftMax activation for class probabilities.

VGG19 [30] is a convolutional neural network with 19 layers. It features a simple, uniform design using 3x3 convolutional filters and 2x2 max pooling layers. VGG19's stacked convolutional layers learn complex patterns and hierarchical representations. The final fully connected layers generate class probabilities. Despite its simplicity, VGG19 performs well in image classification and is a popular baseline model in computer vision research.

ResNet50 [31] is a robust deep convolutional neural network architecture. It uses residual connections for image classification and feature extraction tasks. With 50 wt layers, including convolutional, bottleneck, and fully connected layers, ResNet50 captures image features. The architecture employs 3x3 filters, bottleneck layers, and



**Fig. 1.** The PBC dataset includes images of normal peripheral blood cells organized into eight groups commonly seen in infections and regenerative anemias.

residual connections to enable gradient flow, address the “vanishing gradient” problem, and train deeper models. The final fully connected layers generate class probabilities.

ResNet-101 [32] is a deep neural network with 101 layers, including convolutional, pooling, and fully connected layers. It addresses the challenges of training deep networks using residual blocks with skip connections. These blocks facilitate gradient flow, also mitigating the “vanishing gradient” problem. ResNet-101 employs bottleneck layers to reduce computational complexity while preserving representation capacity. It comprises multiple stages with varying numbers of residual blocks, enabling extraction of hierarchical features.

ResNet-152 [32] is a deep convolutional neural network with 152 layers. It employs skip connections in residual blocks to solve training challenges in deep networks. The architecture includes bottleneck layers for efficiency and multiple stages with varying numbers of residual blocks to extract hierarchical features. The final fully connected layers generate class probabilities for image recognition. With both depth and feature-capturing capabilities, ResNet-152 frequently achieves high performance, enabling accurate and robust deep neural networks.

The InceptionV3 [33] model is a deep convolutional neural network architecture developed by researchers at Google Research. Through its use of inception modules, InceptionV3 employs parallel convolutional layers with differing filter sizes ( $1 \times 1$ ,  $3 \times 3$ ,  $5 \times 5$ ) and pooling operations to capture information at multiple scales and resolutions. The model incorporates batch normalization layers for efficient training and gradient flow. With a composition of multiple inception modules, fully connected layers, and a SoftMax activation function, InceptionV3 leverages pre-training on datasets such as ImageNet to learn hierarchical

representations and exhibit strong generalization capabilities.

MobileNetV2 [34] is a convolutional neural network for mobile and embedded vision applications. It achieves a balance between model size and accuracy using depth wise separable convolutions and linear bottlenecks. By employing inverted residual blocks with residual connections, batch normalization layers, and ReLU6 activations, MobileNetV2 captures and propagates information effectively. Pre-trained on ImageNet, it learns expressive features and demonstrates high performance on mobile devices. Its lightweight architecture makes it suitable for real-time applications in resource-constrained environments.

DenseNet201 [35] is a deep convolutional neural network comprising 201 layers. It employs dense connectivity, wherein each layer is connected to every other layer, ensuring efficient information flow and feature reuse. This network includes convolutional, pooling, and fully connected layers, along with bottleneck layers to reduce computational complexity. Accordingly, DenseNet201 frequently achieves high accuracy and improved gradient flow in computer vision tasks, and is often employed in image recognition and feature extraction applications.

### 3.3. Proposed CNN architecture

The developed CNN architecture comprises 22 layers, delineated into eight convolution layers responsible for feature extraction, 8 pooling layers followed by five fully connected hidden (dense) layers and an output layer for classification purposes. This architecture was specifically designed to process RGB images, aiming to train input images with dimensions of  $360 \times 360$  pixels, enabling classification of ten distinct

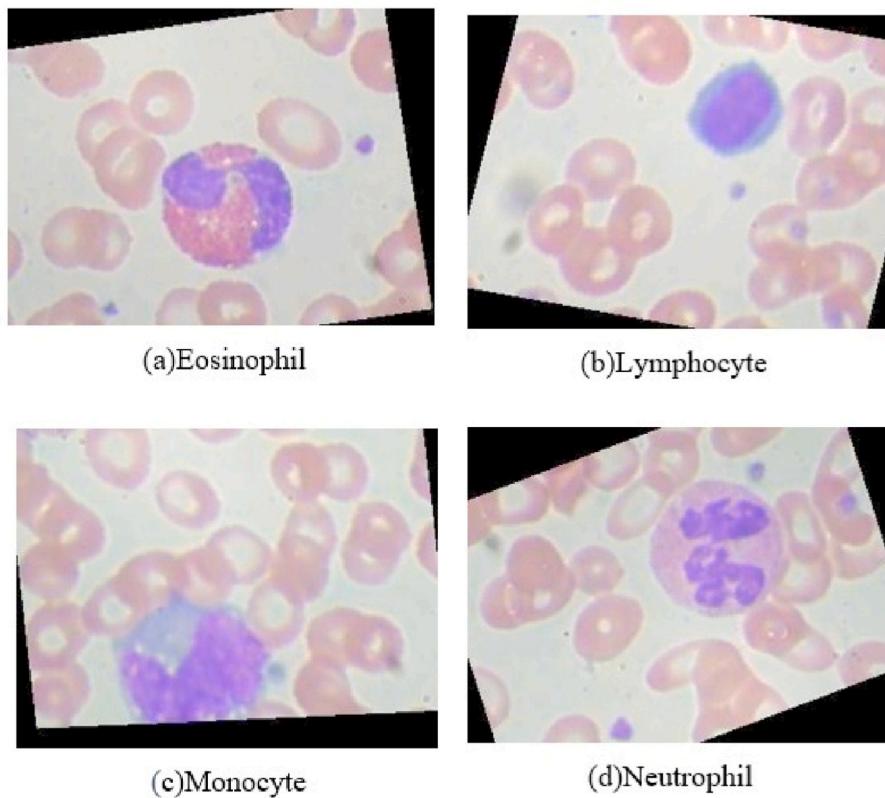


Fig. 2. White Blood cell images in the Kaggle dataset.

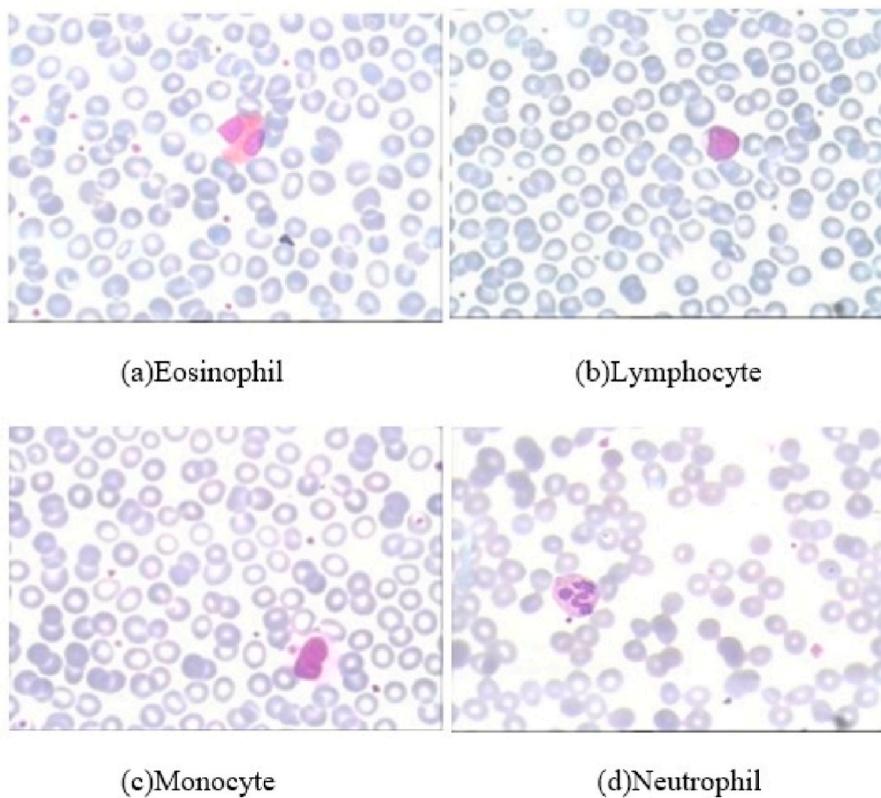


Fig. 3. White Blood cell Images in the LISC dataset.

classes. Each convolution block consists of a convolution layer, a max pooling layer, and a dropout layer. In a deep CNN, the convolutional layers apply filters to the original image or other feature maps.

When an input undergoes the same filter, it generates a feature map by performing a convolution operation on the input, passing the next output to the subsequent layer. Consequently, a convolution layer performs pixel-wise multiplication between a two-dimensional input array (image) and a two-dimensional weight array (kernel), yielding a single value for each multiplication. However, due to the repetitive application of the filter to the input array, the resulting output is represented by a two-dimensional array (feature map). Filters are employed to identify specific features within the input and systematically scan the entire input image using these filters. This enables filters to detect those particular features throughout the image.

The technique of pooling layers reduces the sampling of feature maps by summarizing the presence of features in specific map regions. By incorporating pooling layers, the size of feature maps is reduced. More specifically, after applying a nonlinear function (e.g., ReLU) to the feature maps generated by a convolutional layer, the pooling layer operates independently on each feature map, leading to creation of a new set of pooled feature maps.

The dropout technique is utilized to prevent overfitting. In a neural network, excessive weights indicate a more complex network that may overfit the training data. Dropout is a simple yet effective regularization approach by randomly removing nodes from the network during training. When applying dropout, employing a larger network with abundant training data and consideration of incorporating weight constraints is recommended.

After evaluating the performance of eight pre-trained models on the PBC dataset normal DIB, the authors found that these models did not achieve satisfactory accuracy for all blood cell classification. To address this, we propose the convolutional neural network (CNN) architecture presented in Fig. 4. This architecture comprises eight convolutional layers, eight pooling layers, five fully connected hidden layers, and an output layer. Parameters characterising the proposed CNN architecture are presented in Table 2.

The eight convolutional layers apply 3x3 convolutions with a stride of 1. These are followed by the Rectified Linear Unit (ReLU) activation function, which introduces non-linearity. Subsequently, MAXPOOL layers are used for 2x2 max pooling with a stride of 1. Dropout layers with a dropout rate of 0.25 were incorporated to improve the architecture's generalization ability.

**A. Convolutional Layers.** As stated, the presented architecture consists of eight convolutional layers, as follows:

- 1) First Layer: Kernel size: 3 x 3, number of filters: 32, Activation function: ReLU, Stride: 1, and input size: 360 x 360 (3 channels).
- 2) Second Layer: Kernel size: 3 x 3, number of filters: 64, Activation function: ReLU, Stride: 1
- 3) Third Layer: Kernel size: 3 x 3, number of filters: 64, Activation function: ReLU, Stride: 1.
- 4) Fourth Layer: Kernel size: 3 x 3, number of filters: 128, Activation function: ReLU, Stride: 1
- 5) Fifth Layer: Kernel size: 3 x 3, number of filters: 256, Activation function: ReLU, Stride: 1
- 6) Sixth Layer: Kernel size: 3 x 3, number of filters: 256, Activation function: ReLU, Stride: 1
- 7) Seventh Layer: Kernel size: 3 x 3, number of filters: 256, Activation function: ReLU, Stride: 1
- 8) Eighth Layer: Kernel size: 3 x 3, number of filters: 512, Activation function: ReLU, Stride: 1

#### B. Pooling Layers.

The developed architecture utilized max pooling with the same parameters for all eight pooling layers. The pooling layers were configured as follows: Pooling type: Maximum, Pooling Size: 2 x 2, Stride: 1, Dropout: 0.25.

#### C. Fully Connected Layers.

In our CNN architecture, the final layers consist of fully connected layers. Our developed methodology includes five fully connected hidden

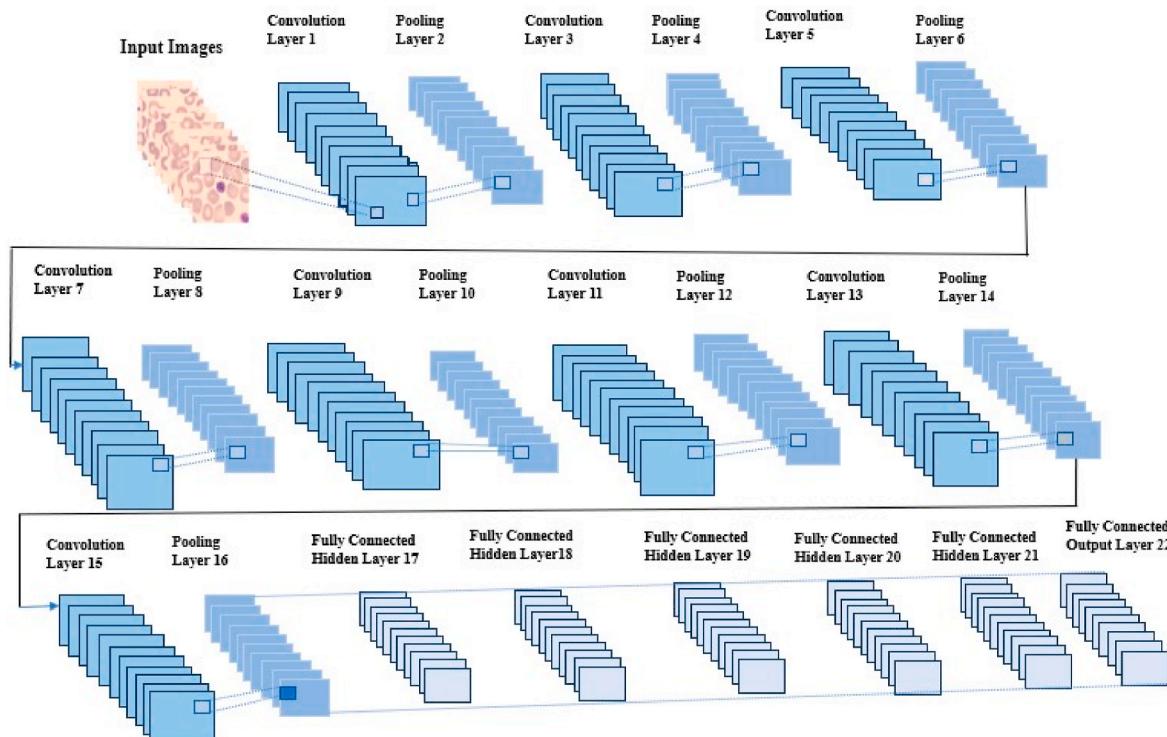


Fig. 4. Proposed CNN architecture.

**Table 2**

Parameters of the proposed CNN architecture.

Layer	Kernel Size	Filters	Stride	Activation	Layer Type	Pooling Size	Stride (Pooling)	Dropout	Nodes
Conv1	3x3	32	1	ReLU	Conv				
Conv2	3x3	64	1	ReLU	Conv				
Conv3	3x3	64	1	ReLU	Conv				
Conv4	3x3	128	1	ReLU	Conv				
Conv5	3x3	256	1	ReLU	Conv				
Conv6	3x3	256	1	ReLU	Conv				
Conv7	3x3	256	1	ReLU	Conv				
Conv8	3x3	512	1	ReLU	Conv				
Pool1					Pool	2x2	1	0.25	
Pool2					Pool	2x2	1	0.25	
Pool3					Pool	2x2	1	0.25	
Pool4					Pool	2x2	1	0.25	
Pool5					Pool	2x2	1	0.25	
Pool6					Pool	2x2	1	0.25	
Pool7					Pool	2x2	1	0.25	
Pool8					Pool	2x2	1	0.25	
Dense1				ReLU	Dense				512
Output				SoftMax	Dense				10

layers and one fully connected output layer, as follows.

- 1) Fully Connected Hidden Layer: Total nodes: 512, Activation: ReLU.
- 2) Fully Connected Output Layer: Total nodes: number of classes, Activation: SoftMax.

#### 3.4. Evaluation parameters

We evaluated the performance of existing and developed deep learning models using four key metrics: accuracy, recall, precision, and F-measure. These metrics, essential for evaluation of model performance, are derived using a confusion matrix, with the associated equations used to calculate each of these metrics detailed in [Table 3](#) and below.

## 4. Results and discussion

### 4.1. Pre-trained model compilation and results

In the first phase of evaluation, the eight pre-trained architectures, including VGG16, VGG19, ResNet50, ResNet101, ResNet-152, Inception V3, MobileNetV2 and DenseNet201 were assessed on the normal DIB of the PBC dataset ( $N = 3418$ ). These architectures were trained using sparse categorical cross-entropy loss with Adam as the gradient-based optimizer. The pre-trained weights utilized were derived from the ImageNet dataset classification. The training focused on the last dense layers and spanned 150 epochs, with a learning rate of 0.001.

The performance parameters achieved with the pre-trained architectures are summarized in [Table 4](#). As shown, the VGG-16 architecture misclassified 100 of 3418 images. Specifically, misclassifications occurred in the following categories: 15 eosinophil images, 11 lymphocyte images, 13 monocyte images, 13 neutrophil images, 9 basophil images, 15 immature granulocyte (metamyelocytes, myelocytes, and promyelocytes) images, 11 erythroblast images, and 13 platelet (thrombocytes) images. The overall accuracy achieved with

VGG-16 was 93.75 %. Similarly, the VGG-19 architecture misclassified a total of 102 images, including 14 eosinophils, 17 lymphocytes, 16 monocytes, 18 neutrophils, 8 basophils, 10 immature granulocytes, 8 erythroblasts, and 11 platelets. The overall accuracy attained with VGG-19 was 93.12 %.

The ResNet-50 architecture misclassified 136 images. These included 30 eosinophils, 6 lymphocytes, 13 monocytes, 14 neutrophils, 28 basophils, 15 immature granulocytes, 15 erythroblasts, and 15 platelets. The overall accuracy achieved using ResNet-50 was 91.8 %. The ResNet-101 architecture misclassified a total of 133 images, including 39 eosinophils, 7 lymphocytes, 11 monocytes, 13 neutrophils, 30 basophils, 9 immature granulocytes, 13 erythroblasts, and 11 platelets. The overall classification accuracy obtained using ResNet-101 was 92.01 %. ResNet-152 architecture misclassified 89 images, distributed across blood type categories as follows: 15 eosinophils, 4 lymphocytes, 7 monocytes, 18 neutrophils, 20 basophils, 6 immature granulocytes, 9 erythroblasts, and 10 platelets. The overall accuracy achieved with ResNet-152 was 94.26 %. In the case of the InceptionV3 architecture, 118 images were misclassified, involving 17 eosinophils, 7 lymphocytes, 20 monocytes, 14 neutrophils, 17 basophils, 16 immature granulocytes, 11 erythroblasts, and 16 platelets. The overall accuracy achieved with InceptionV3 was 92.8 %. The MobileNetV2 architecture misclassified a total of 142 images, including 15 eosinophils, 9 lymphocytes, 19 monocytes, 20 neutrophils, 18 basophils, 30 immature granulocytes, 16 erythroblasts, and 15 platelets. The overall accuracy achieved with MobileNetV2 was 91.37 %. Lastly, utilizing the DenseNet201 architecture, a total of 88 images were misclassified. These included 15 eosinophils, 5 lymphocytes, 18 monocytes, 7 neutrophils, 9 basophils, 11 immature granulocytes, 7 erythroblasts, and 16 platelets. The overall accuracy achieved with DenseNet201 was 94.72 %.

### 4.2. Developed CNN architecture

To improve architecture performance, the authors extracted and employed advantages from existing architectures in developing a new, bespoke and generalizable CNN architecture. During the presented study three key parameters were considered for training: the loss function, optimizer, and evaluation metrics. The sparse categorical cross-entropy loss function was utilized for our CNN architecture in concurrence with the Adam optimizer. The training process involved introducing the training dataset to the developed architecture and training for 150 epochs, with best weights saved based on the loss function. The training ratio was 60 %, with testing and validation ratios both 20 %. Subsequently, the proposed convolutional neural network architecture was evaluated using all the blood cell images from the PBC dataset's normal DIB category.

**Table 3**

Confusion matrix and formulae for employed evaluation parameters.

Positive		Negative	
Positive	True Positive (TP)	False Positive (FP)	
Negative	False Negative (FN)	True Negative (TN)	

Accuracy =  $TP + TN / (TP + TN + FP + FN)$ .

Precision =  $TP / (TP + FP)$ .

Recall =  $TP / (TP + FN)$ .

F Measure =  $2 * (precision * recall) / (precision + recall)$ .

**Table 4**

Test results of pre-trained models on PBC dataset normal DIB (N = 3418).

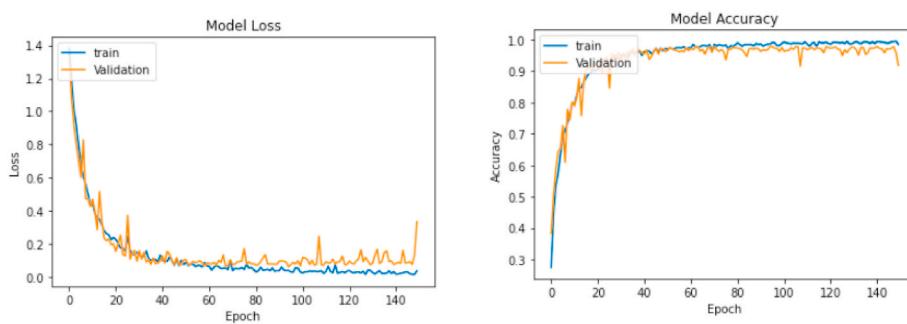
Model	Type	Truth	Classified	Accuracy	Precision	Recall	F-measure
VGG16	Eosinophils	412	427	0.92	0.89	0.85	0.90
	Lymphocytes	416	427	0.971	0.97	0.97	0.97
	Monocytes	414	427	0.952	0.95	0.98	0.97
	Neutrophils	414	427	0.93	0.93	0.88	0.90
	Basophils	419	428	0.92	0.92	0.82	0.87
	IG (Metamyelocytes, Myelocytes, and Promyelocytes)	412	427	0.90	0.91	0.81	0.85
	Erythroblasts	417	428	0.94	0.95	0.97	0.96
	Platelets	414	427	0.961	0.96	0.97	0.96
VGG19	Eosinophils	413	427	0.90	0.90	0.84	0.85
	Lymphocytes	410	427	0.97	0.96	0.97	0.97
	Monocytes	411	427	0.90	0.92	0.82	0.84
	Neutrophils	409	427	0.94	0.95	0.98	0.96
	Basophils	420	428	0.91	0.90	0.83	0.87
	IG (Metamyelocytes, Myelocytes, and Promyelocytes)	417	427	0.97	0.93	0.95	0.95
	Erythroblasts	420	428	0.94	0.96	0.95	0.97
	Platelets	416	427	0.92	0.92	0.83	0.85
ResNet50	Eosinophils	397	427	0.942	0.94	0.92	0.93
	Lymphocytes	421	427	0.982	0.98	0.99	0.98
	Monocytes	414	427	0.972	0.97	0.99	0.98
	Neutrophils	413	427	0.932	0.93	0.93	0.93
	Basophils	400	428	0.95	0.94	0.81	0.85
	IG (Metamyelocytes, Myelocytes, and Promyelocytes)	412	427	0.930	0.93	0.82	0.87
	Erythroblasts	413	428	0.930	0.93	0.82	0.87
	Platelets	412	427	0.94	0.94	0.81	0.88
ResNet-101	Eosinophils	388	427	0.90	0.90	0.81	0.83
	Lymphocytes	420	427	0.95	0.94	0.94	0.96
	Monocytes	416	427	0.90	0.90	0.80	0.81
	Neutrophils	414	427	0.92	0.9	0.96	0.94
	Basophils	398	428	0.91	0.92	0.80	0.85
	IG (Metamyelocytes, Myelocytes, and Promyelocytes)	418	427	0.95	0.90	0.92	0.93
	Erythroblasts	415	428	0.93	0.94	0.92	0.94
	Platelets	416	427	0.90	0.90	0.81	0.83
ResNet-152	Eosinophils	412	427	0.930	0.93	0.82	0.87
	Lymphocytes	423	427	0.98	0.98	0.99	0.98
	Monocytes	420	427	0.93	0.93	0.80	0.86
	Neutrophils	409	427	0.96	0.96	0.99	0.98
	Basophils	408	428	0.90	0.91	0.81	0.85
	IG (Metamyelocytes, Myelocytes, and Promyelocytes)	421	427	0.961	0.96	0.97	0.96
	Erythroblasts	419	428	0.95	0.97	0.96	0.98
	Platelets	417	427	0.930	0.93	0.82	0.87
InceptionV3	Eosinophils	410	427	0.930	0.93	0.82	0.87
	Lymphocytes	420	427	0.961	0.96	0.97	0.96
	Monocytes	407	427	0.95	0.95	0.98	0.96
	Neutrophils	413	427	0.94	0.93	0.97	0.95
	Basophils	411	428	0.90	0.91	0.81	0.85
	IG (Metamyelocytes, Myelocytes, and Promyelocytes)	411	427	0.92	0.92	0.82	0.86
	Erythroblasts	417	428	0.91	0.92	0.81	0.85
	Platelets	411	427	0.92	0.931	0.825	0.87
MobileNetV2	Eosinophils	412	427	0.93	0.93	0.82	0.87
	Lymphocytes	418	427	0.94	0.95	0.95	0.95
	Monocytes	408	427	0.87	0.89	0.81	0.85
	Neutrophils	407	427	0.93	0.93	0.80	0.86
	Basophils	410	428	0.90	0.92	0.825	0.87
	IG (Metamyelocytes, Myelocytes, and Promyelocytes)	397	427	0.87	0.89	0.80	0.86
	Erythroblasts	412	428	0.97	0.97	0.99	0.98
	Platelets	412	427	0.90	0.91	0.80	0.83
DenseNet201	Eosinophils	412	427	0.942	0.94	0.92	0.93
	Lymphocytes	422	427	0.982	0.98	0.99	0.98
	Monocytes	409	427	0.972	0.97	0.99	0.98
	Neutrophils	420	427	0.932	0.93	0.93	0.93
	Basophils	419	428	0.95	0.94	0.81	0.85
	IG (Metamyelocytes, Myelocytes, and Promyelocytes)	416	427	0.930	0.93	0.82	0.87
	Erythroblasts	421	428	0.93	0.93	0.82	0.87
	Platelets	411	427	0.90	0.94	0.81	0.88

#### 4.3. PBC dataset normal DIB - results

The developed architecture's loss and accuracy achieved during each epoch for the PBC dataset's normal DIB are presented in Fig. 3. Network performance was evaluated using the cross-entropy loss function, commonly employed to assess the effectiveness of convolutional neural networks. The cross-entropy value increases when the predicted value differs from the actual value. Ideally, the cross-entropy value should be

zero. As shown in Fig. 5, the cross-entropy value reached its minimum of 0.026 after 140 epochs i.e., proximal to zero. The maximum error, which combines training and validation, was 0.057 (5.7 %). The highest training accuracy recorded was 0.993 after 142 epochs, while the maximum validation accuracy achieved was 0.985 (Fig. 4).

Using the proposed CNN model, the weights corresponding with minimum loss were saved and used to predict labels for the testing dataset. Results are presented in Table 5. In the PBC dataset's normal



**Fig. 5.** Receiver operating curves (ROC) representing loss and accuracy of proposed CNN architecture on PBC dataset.

**Table 5**

Confusion matrix for Peripheral Blood Cells dataset normal DIB.

Class	Eosinophil	Lymphocyte	Monocyte	Neutrophil	Basophils	IG	Erythroblast	Platelets
<b>Eosinophils</b>	426	0	0	0	0	0	1	0
<b>Lymphocytes</b>	0	427	0	0	0	0	0	0
<b>Monocytes</b>	0	0	427	0	0	0	0	0
<b>Neutrophils</b>	0	0	0	427	0	0	0	0
<b>Basophils</b>	1	0	0	0	427	0	0	0
<b>IG</b>	0	0	0	0	0	427	0	0
<b>Erythroblasts</b>	0	0	0	0	0	0	428	0
<b>Platelets</b>	0	0	0	0	0	0	0	427

DIB category, misclassifications were observed for the eosinophil ( $n = 1$ ) and basophil ( $n = 1$ ) classes, equating to a total of 2 images out of 3418 being misclassified. This is likely attributable to the similarity in shape and size between these two cell types. However, all images within the other six classes were correctly classified with 100 % accuracy.

Associated accuracy, precision, recall, and F-measure rates for each class are presented in [Table 6](#). Precision rates were 99 % for eosinophils, 100 % for lymphocytes, monocytes, neutrophils, immature granulocytes (metamyelocytes, myelocytes, and promyelocytes), erythroblasts, and platelets (thrombocytes), and 99.3 % for basophils. F-measure rates were 99.3 % for eosinophils, 100 % for neutrophils, lymphocytes, monocytes, immature granulocytes, erythroblasts, platelets, and 98 % for basophils. Recall rates achieved were 99.4 % for eosinophils, 100 % for lymphocytes, monocytes, neutrophils, immature granulocytes, erythroblasts, platelets, and 98.5 % for basophils. The mean (combined) accuracy across all classes was 99.91 %.

The impact of varying dropout rates, optimizers, and loss functions on model performance can be seen in [Table 7](#). Using a 0.25 dropout rate and the Adam optimizer, the model achieved a 0.09 % loss rate and 99.91 % accuracy after 150 epochs. Using the same optimizer with a Sparse Categorical Cross-entropy loss function and extending training to

**Table 7**

Effect of dropout rates, optimizers, loss function and epochs on model performance.

Dropout Rate	Optimizer	Loss Function	Epochs	Loss Rate (%)	Accuracy (%)
0.25	Adam	Sparse Categorical Crossentropy	150	0.09	99.91
0.25	Adam	Sparse Categorical Crossentropy	200	0.71	99.29
0.20	SGD	Categorical Crossentropy	150	1.32	98.68
0.15	SGD	Categorical Crossentropy	130	1.36	98.64
0.10	RMSprop	Categorical Crossentropy	100	1.42	98.58

200 epochs, the loss rate increased to 0.71 % and accuracy decreased very slightly to 99.29 %. A 0.20 dropout rate combined with a Stochastic Gradient Descent (SGD) optimizer resulted in a 1.32 % loss rate and 98.68 % accuracy after 150 epochs, and a marginally higher 1.36 % loss

**Table 6**

Test results of proposed CNN architecture.

	Type	Observed	Classified	Accuracy (%)	Precision (%)	Recall (%)	F measure (%)
<b>PBC Dataset</b>	Eosinophils	426	427	0.995	0.99	0.994	0.993
	Lymphocytes	427	427	100	100	100	100
	Monocytes	427	427	100	100	100	100
	Neutrophils	427	427	100	100	100	100
	Basophils	427	428	0.998	0.993	0.985	0.98
	IG	427	427	100	100	100	100
	Erythroblasts	428	428	100	100	100	100
	Platelets	427	427	100	100	100	100
<b>Kaggle Dataset</b>	Eosinophils	623	625	0.99	0.992	0.99	0.991
	Lymphocytes	624	625	0.996	0.99	0.99	0.993
	Monocytes	625	625	100	100	100	100
	Neutrophils	625	625	100	100	100	100
	Platelets	627	627	100	100	100	100
<b>LISC Dataset</b>	Eosinophils	498	500	0.984	0.982	0.98	0.985
	Lymphocytes	495	500	0.977	0.97	0.974	0.98
	Monocytes	500	500	100	100	100	100
	Neutrophils	499	500	0.987	0.985	0.998	0.98

rate with 98.64 % accuracy after 130 epochs. Lastly, a 0.10 dropout rate with the Root Mean Square Propagation (RMSprop) optimizer led to a 1.42 % loss rate and 98.58 % accuracy after 100 epochs.

#### 4.4. Kaggle Dataset - results

A graphical representation of the developed model's accuracy and loss for the Kaggle dataset is provided in Fig. 6. After 148 epochs, the cross-entropy loss function reached a minimum value of 0.0365. Notably, the model achieved a maximum validation accuracy of 98.3 % and a maximum training accuracy of 99.12 %. The associated confusion matrix for the Kaggle dataset is provided in Table 8. Of the 2500 images, 2 eoxinophile images, and 1 lymphocyte image were misclassified i.e., total of 3 misclassifications. The model achieved a mean combined accuracy of 99.68 %. The performance metrics for blood cell classification using the proposed CNN architecture across all three datasets are presented in Table 6.

#### 4.5. LISC dataset - results

Fig. 7 presents model loss and accuracy for the LISC dataset. The cross-entropy loss function reached its minimum value of 0.0376 after 146 epochs. Validation accuracy peaked at 98.5 %, while training accuracy achieved a maximum of 97.56 %. The confusion matrix for the LISC dataset is presented in Table 9. Of 2000 images, 8 were misclassified, including 2 eosinophils, 5 lymphocytes, and 1 neutrophil. A mean combined accuracy of 98.79 % was achieved.

A comparison of outcomes between the method developed in the current study and similar recent studies from the existing literature is presented in Table 10. As shown, Ahmad et al. [11] utilized a Deep CNN, achieving an accuracy of 99.6 % on a public dataset containing 5000 images, while Dhar et al. [36] obtained 97.81 % precision on the PBC DIB dataset, 96.2 % on ALL DIB, and 98.9 % on the Kaggle dataset. Atici and Kocer [37] applied a CNN to the PBC DIB dataset, achieving 99.31 % accuracy. Mehedi et al. [38] employed CNN on the ABIDE dataset, obtaining 90 % accuracy. Safua et al. [39] employed the WBC Kaggle and LISC datasets, which contain 12,500 and 260 images, respectively, achieving 98.08 % accuracy on Kaggle and 84.52 % on LISC. Tsutsui et al. [40] attained 98.67 % accuracy using the LISC dataset with ResNet50 and VGG16. Sangeetha et al. [41] achieved an accuracy of 98.7 % on the Kaggle dataset using CNN. Saidani et al. [42] used the IEEE Dataport dataset ( $N = 3539$ ), in concurrence with a CNN model = to achieve a classification accuracy of 99.8 %, precision of 99 %, recall of 99 %, and an F1-score of 99 %. Meanwhile, Anita et al. [43] utilized the Kaggle dataset ( $N = 12500$ ). Their final models, VGG16 and MobileNet, both achieved an accuracy of 99.36 %, precision of 99 %, recall of 99 %, and an F1-score of 99 %. Song et al. [44] worked with a private dataset

comprising 9069 samples. They used a CNN model that attained an accuracy of 97.9 %, precision of 97 %, recall of 97 %, and an F1-score of 98 %. T. Uyar et al. [45] analyzed data from the LISC and BCD datasets, with sample sizes of 400 and 2340 respectively. They applied a CNN model, achieving accuracies of 98.0 % and 98.3 %, precisions of 98.2 % and 96.6 %, recalls of 98 % and 96.9 %, and F1-scores of 98.1 % and 96.3 %.

In comparison to the most recent studies in this area (Table 10), the current study presents an architecture that significantly enhances the accuracy of blood cell classification, achieving exceptional accuracy of 99.91 % on the PBC DIB dataset, 99.68 % on Kaggle, and 98.79 % on the LISC dataset, thus surpassing all previous work in this area [9].

## 5. Conclusion

In this study, we initially applied a series of eight pre-trained models: VGG16, VGG19, ResNet-50, ResNet-101, ResNet-152, InceptionV3, MobileNetV2, and DenseNet-201. Inspired by these architectures and aiming for enhanced performance, we developed and presented a CNN-based architecture designed to categorize blood cell subtypes, including red blood cells, white blood cells, and platelets. This proposed architecture employs transfer learning and comprises eight convolutional layers, eight pooling layers, five fully connected hidden layers, and an output layer. We performed classification training and testing using microscopic blood cell images obtained from three datasets: PBC dataset normal DIB, Kaggle, and LISC. During testing, our proposed algorithm demonstrated optimal classification performance, achieving 99.91 % accuracy for the PBC dataset, 99.68 % accuracy for the Kaggle dataset, and 98.79 % accuracy for the LISC dataset. The efficacy of the proposed architecture is underscored by its outcomes, which outperform the majority of, if not all, previous results reported in the literature for the same datasets. Looking ahead, our presented architecture has potential applications in classifying myriad human cells and tissues, assisting pathologists and clinicians in making rapid and accurate diagnoses.

## Ethical statement

No ethical clearance was deemed necessary in carrying out this work as all data were and are open access, with individual informed consent and patient details not employed.

## CRediT authorship contribution statement

**Rabia Asghar:** Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Sanjay Kumar:** Writing – review & editing, Supervision, Resources, Project administration, Conceptualization. **Paul Hynds:**

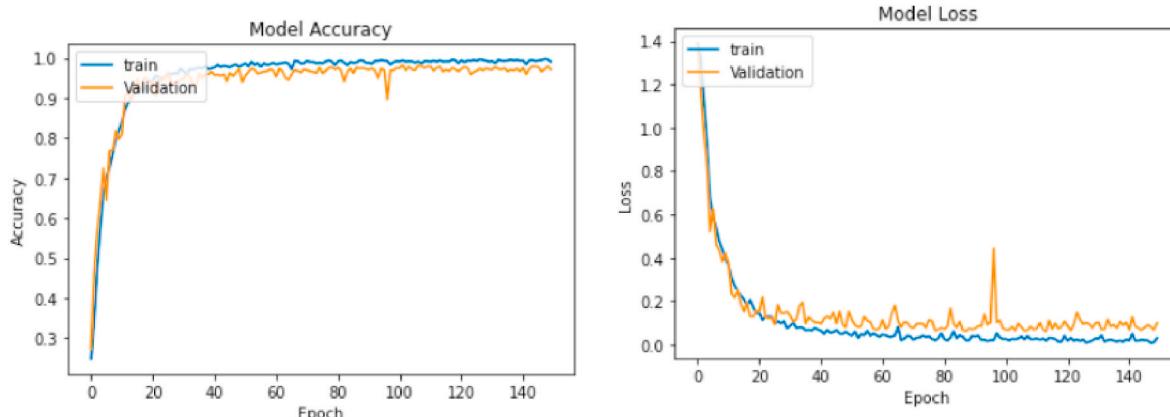
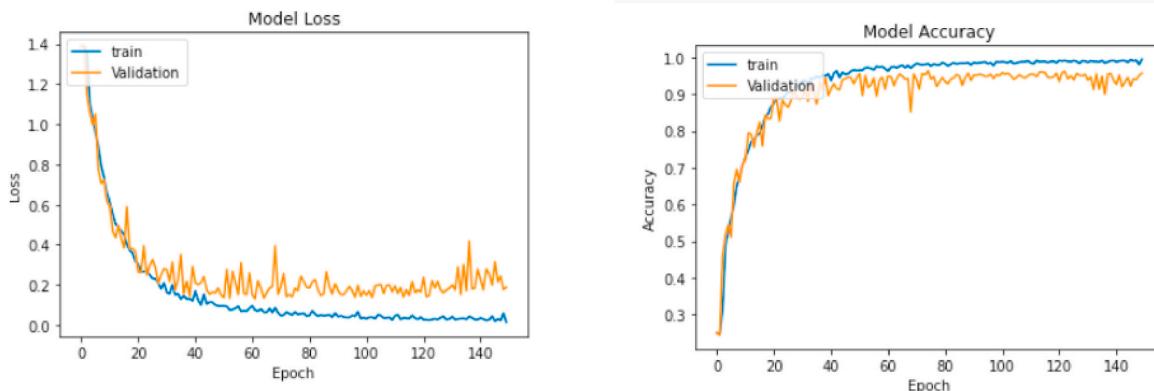


Fig. 6. Receiver operating curves (ROC) representing loss and accuracy of proposed CNN architecture on the Kaggle dataset.

**Table 8**

Confusion matrix for Kaggle dataset.

Class	EOSINOPHIL	LYMPHOCYTE	MONOCYTE	NEUTROPHIL
EOSINOPHIL	623	0	0	2
LYMPHOCYTE	1	624	0	0
MONOCYTE	0	0	625	0
NEUTROPHIL	0	0	0	625

**Fig. 7.** Receiver operating curves (ROC) representing loss and accuracy of proposed CNN architecture on the LISC dataset.**Table 9**

Confusion matrix for LISC dataset.

Class	EOSINOPHIL	LYMPHOCYTE	MONOCYTE	NEUTROPHIL
EOSINOPHIL	498	0	0	2
LYMPHOCYTE	3	495	0	2
MONOCYTE	0	0	500	0
NEUTROPHIL	1	0	0	499

**Table 10**

Comparison of proposed CNN architecture with similar recent (2023/24) studies.

Author	Dataset	Images	Architect.	Accuracy (%)	F1 score (%)	Recall(%)	Precision (%)
Ahmad et al. [11]	Public dataset	5000	Deep CNN	99.6	–	–	–
Prasenjit et al. [39]	PBC DIB, ALLIDB1, Kaggle	17092,108,100	Watershed algorithm	–	97.95,96.15,	98.1,96.1,98.6	97.81,96.2,98.8
Atici et al. [40]	PBC DIB	17,092	CNN	99.31	96.87	97.37	97.25
Rahman et al. [41]	ABIDE	2939	CNN	90	90	91	88
Safua et al. [42]	Kaggle, LISC	12500,260	CNN	98.08,84.52	–	–	–
Tsutsui et al. [43]	LISC	257	ResNet-50, VGG-16	98.67	99.12	92.31	94.44
Sangeetha et al. [44]	Kaggle	12,500	CNN	98.70	–	–	–
Saidani et al. [45]	IEEE Dataport	3539	CNN	99.8	99	99	99
Anita et al. [46]	Kaggle	12500	VGG16	99.36	99	99	99
Song et al. [47]	Private Dataset	9069	CNN	97.9	97	97	98
Uyar et al. [48]	LISC, BCD	400,2340	CNN	98.0, 98.3	98.2,96.6	98,96.9	98.1,96.3
Presented Methodology	PBC DIB,Kaggle, LISC	17,092,12500,10,000	CNN	99.91,99.68,98.79	99.6,99.0,98.50	99.6,99.4,99.8	99.77,99.3,98.0

Writing – review &amp; editing, Supervision.

**Declaration of competing interest**

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Acknowledgements**

No external funding was acquired for this study.

**References**

- [1] Tamang T, Baral S, Paing MP. Classification of white blood cells: a Comprehensive study using transfer learning based on convolutional neural networks. *Diagnostics* 2022;22(12):2903. <https://doi.org/10.3390/diagnostics12122903>. 12.
- [2] Lu N, Tay HM, Petchakup C, He L, Gong L, Maw KK, Hou HW. Label-free microfluidic cell sorting and detection for rapid blood analysis. *Lab Chip* 2023;23(5):1226–57.
- [3] Rustam F, Aslam N, De La Torre Díez I, Khan YD, Mazón JLV, Rodríguez CL, Ashraf I. White blood cell classification using texture and RGB features of oversampled microscopic images. *Healthcare* 2022, November;10(11):2230. MDPI.
- [4] Gavas E, Olpadkar K. Deep CNNs for peripheral blood cell classification. arXiv preprint arXiv:2110.09508 2021. <https://doi.org/10.48550/arXiv.2110.09508>.
- [5] Darrin M, Samudre A, Sahun M, Atwell S, Badens C, Charrier A, Helfer E, Viallat A, Addad VC, Roisin SG. Classification of red cell dynamics with convolutional and

- recurrent neural networks: a sickle cell disease case study. *Sci Rep* 2023;13(1):745. 2023.
- [6] Çelebi S, Burkay Çöte M. Red and white blood cell classification using Artificial Neural Networks. Yıldırım Beyazıt University, AIMS Bioeng 2018;5(3):179–91. PhD Thesis.
- [7] Tseng TR, Huang HM. Classification of peripheral blood neutrophils using deep learning. *Cytometry* 2023;103(4):295–303.
- [8] Sahlot AT, Kollmannsberger P, Ewees AA. Efficient classification of white blood cell leukemia with improved swarm optimization of deep features. *Sci Rep* 2020;10(1):2536.
- [9] Asghar R, Kumar S, Shaukat A, Hynds P. Classification of white blood cells (leucocytes) from blood smear imagery using machine and deep learning models: a global scoping review. *PLoS One* 2024;19(6):e0292026. <https://doi.org/10.1371/journal.pone.0292026>.
- [10] Elhassan TA, Mohd Rahim MS, Siti Zaiton MH, Swee TT, Alhaj TA, Ali A, Aljurf M. Classification of atypical white blood cells in acute myeloid leukemia using a two-stage hybrid model based on deep convolutional autoencoder and deep convolutional neural network. *Diagnostics* 2023;13(2):196.
- [11] Ahmad R, Awais M, Kausar N, Akram T. White blood cells classification using entropy-controlled deep features optimization. *Diagnostics* 2023;13(3):352.
- [12] Singh R, Sharma A, Sharma N, Gupta R. Impact of Adam, adadelta, SGD on CNN for white blood cell classification. In: 2023 5th International conference on Smart systems and inventive technology (ICSSIT). IEEE; 2023. p. 1702–9.
- [13] Saikia R, Devi SS. White blood cell classification based on gray level co-occurrence matrix with zero phase component analysis approach. *Procedia Comput Sci* 2023; 218:1977–84.
- [14] Yebasse M, Cheoi KJ, Ko J. Malaria disease cell classification with highlighting small infected regions. *IEEE Access* 2023;11:15945–53.
- [15] Kumar D, Jain N, Khurana A, Mittal S, Satapathy SC, Senkerik R, Hemanth JD. Automatic detection of white blood cancer from bone marrow microscopic images using convolutional neural networks. *IEEE Access* 2020;8:142521–31. <https://doi.org/10.1109/ACCESS.2020.3012292>.
- [16] Ansari S, Navin AH, Sangar AB, Gharamaleki JV, Danishvar S. A customized efficient deep learning model for the diagnosis of acute leukemia cells based on lymphocyte and monocyte images. *Electronics* 2023;12(2):322.
- [17] Alkafrawi IM, Dakhell ZA. Blood cells classification using deep learning technique. In: 2022 international conference on engineering & MIS (ICEMIS); 2022, July. p. 1–6. IEEE.
- [18] Muhamad HA, Kareem SW, Mohammed AS. A comparative evaluation of deep learning methods in automated classification of white blood cell images. In: 2022 8th International engineering conference on sustainable technology and development (IEC). IEEE; 2022, February. p. 205–11.
- [19] Miserlis D, Munian Y, Bohannon WT, Wechsler M, Montero-Baker M, Ferrer-Cardona L, Alamaniotis M. Convolutional neural network analysis of tissue remodeling and myopathy in peripheral arterial disease. In: 2022 13th international conference on information, intelligence, systems & applications (IISA). IEEE; 2022, July. p. 1–8.
- [20] Arif R, Akbar S, Farooq AB, Hassan SA, Gull S. Automatic detection of leukemia through convolutional neural network. In: 2022 international conference on frontiers of information technology (FIT); 2022, December. p. 195–200. IEEE.
- [21] Devi GM, Neelambari V. Computer-aided diagnosis of white blood cell leukemia using VGG16 convolution neural network. In: 2022 4th international conference on inventive research in computing applications (ICIRCA). IEEE; 2022, September. p. 1064–8.
- [22] Alnawayseh SE, Al-Sit WT, Alrababah H, Yasin NS, Fatima M, Mahmood N. Classification of white blood cells empowered with auto encoder and CNN. In: 2022 international conference on cyber resilience (ICCR). IEEE; 2022, October. p. 1–7.
- [23] Luong DT, Anh DD, Thang TX, Huong HTL, Hanh TT, Khanh DM. Distinguish normal white blood cells from leukemia cells by detection, classification, and counting blood cells using YOLOv5. In: 2022 7th national scientific conference on applying new technology in green buildings (ATiGB). IEEE; 2022, November. p. 156–60.
- [24] Liao Z, Zhang Y. Red blood cell aggregation classification based on ultrasonic radiofrequency echo signals by an improved convolutional neural network. In: 2022 3rd international conference on computer vision, image and deep learning & international conference on computer engineering and applications (CVIDL & ICCEA); 2022, May. p. 195–8. IEEE.
- [25] Guan Y, Wang Z. Blood cell image recognition algorithm based on EfficientNet. In: 2022 IEEE international conference on mechatronics and automation (ICMA). IEEE; 2022, August. p. 1640–5.
- [26] Ghosh S, Majumder M, Kudeshia A. Leukox: leukocyte classification using least entropy combiner (lec) for ensemble learning. *IEEE Transactions on Circuits and Systems II: Express Briefs* 2021;68(8):2977–81. <https://doi.org/10.1109/TCSII.2021.3064389>. 2977–2981, Aug. 2021.
- [27] <https://www.kaggle.com/paultimothymooney/blood-cells>. [Accessed 13 December 2023].
- [28] Rezatofighi SH, Soltanian-Zadeh H. Automatic recognition of five types of white blood cells in peripheral blood. *Comput Med Imag Graph* 2011;35(4):333–43.
- [29] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* 2014. <https://doi.org/10.48550/arXiv.1409.1556>.
- [30] Murthy VN, Maji S, Manmatha R. Automatic image annotation using deep learning representations. In: Proceedings of the 5th ACM on international conference on multimedia retrieval; 2015, June. p. 603–6.
- [31] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770–8.
- [32] Li Y, Zeng J, Zhang J, Dai A, Kan M, Shan S, Chen X. Kinnet: fine-to-coarse deep metric learning for kinship verification. In: Proceedings of the 2017 workshop on recognizing families in the wild; 2017, October. p. 13–20.
- [33] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 2818–26.
- [34] Sandler M, Howard A, Zhu M, Zhmoginov A, Chen LC. Mobilenetv2: inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 4510–20.
- [35] Wang SH, Zhang YD. DenseNet-201-based deep neural network with composite learning factor and precomputation for multiple sclerosis classification. *ACM Trans Multimed Comput Commun Appl* 2020;16(2s):1–19.
- [36] Dhar P, Suganya Devi K, Satti SK, Srinivasan P. Efficient detection and partitioning of overlapped red blood cells using image processing approach. *Innovat Syst Software Eng* 2022;1–13.
- [37] Atıcı H, Kocer HE. Mask R-CNN based segmentation and classification of blood smear images. *Gazi Mühendislik Bilimleri Dergisi* 2023;9(1):128–43.
- [38] Mehedi ST, Anwar A, Rahman Z, Ahmed K, Islam R. Dependable intrusion detection system for IoT: a deep transfer learning based approach. *IEEE Trans Ind Inf* 2022;19(1):1006–17. <https://doi.org/10.1109/NTIC55069.2022.10100517>.
- [39] Safuan SNM, Tomari MRM, Zakaria WNW. Cross validation analysis of convolutional neural network variants with various white blood cells datasets for the classification task. *International Journal of Online & Biomedical Engineering* 2022;18(2).
- [40] Tsutsui S, Su Z, Wen B. Benchmarking white blood cell classification under domain shift. In: ICASSP 2023–2023 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE; 2023, June. p. 1–5.
- [41] Sangeetha M, Tamizarasu K, Devi RM, Arokiaraj RM, Sudha K, Kumar KK. An automated multi-level convolutional neural network approach for classification of white blood cells. In: 2023 14th international conference on computing communication and networking technologies (ICCCNT). IEEE; 2023, July. p. 1–7.
- [42] Saidani O, Umer M, Alturki N, Alshardan A, Kiran M, Alsabai S, Ashraf I. White blood cells classification using multi-fold pre-processing and optimized CNN model. *Sci Rep* 2024;14(1):3570.
- [43] Davamani KA, Jawahar M, Anbarasi LJ, Ravi V, Al Mazroa A, Robin CR. Deep transfer learning technique to detect white blood cell classification in regular clinical practice using histopathological images. *Multimed Tool Appl* 2024;1:25.
- [44] Song H, Wang Z. Automatic classification of white blood cells using a semi-supervised convolutional neural network. *IEEE Access* 2024;2024.
- [45] Özcan ŞN, Uyar T, Karayeşen G. Comprehensive data analysis of white blood cells with classification and segmentation by using deep learning approaches. *Cytometry* 2024. <https://doi.org/10.1002/cyto.a.24839>.