



# ConcatNeXt: An automated blood cell classification with a new deep convolutional neural network

Mehmet Erten<sup>1</sup> · Prabal Datta Barua<sup>2</sup> · Sengul Dogan<sup>3</sup>  · Turker Tuncer<sup>3</sup> · Ru-San Tan<sup>4,5</sup> · U. R. Acharya<sup>6</sup>

Received: 16 October 2023 / Revised: 25 February 2024 / Accepted: 17 July 2024 /  
Published online: 2 August 2024  
© The Author(s) 2024

## Abstract

Examining peripheral blood smears is valuable in clinical settings, yet manual identification of blood cells proves time-consuming. To address this, an automated blood cell image classification system is crucial. Our objective is to develop a precise automated model for detecting various blood cell types, leveraging a novel deep learning architecture.

We harnessed a publicly available dataset of 17,092 blood cell images categorized into eight classes. Our innovation lies in ConcatNeXt, a new convolutional neural network. In the spirit of Geoffrey Hinton's approach, we adapted ConvNeXt by substituting the Gaussian error linear unit with a rectified linear unit and layer normalization with batch normalization. We introduced depth concatenation blocks to fuse information effectively and incorporated a patchify layer.

Integrating ConcatNeXt with nested patch-based deep feature engineering, featuring downstream iterative neighborhood component analysis and support vector machine-based functions, establishes a comprehensive approach. ConcatNeXt achieved notable validation and test accuracies of 97.43% and 97.77%, respectively. The ConcatNeXt-based feature engineering model further elevated accuracy to 98.73%. Gradient-weighted class activation maps were employed to provide interpretability, offering valuable insights into model decision-making.

Our proposed ConcatNeXt and nested patch-based deep feature engineering models excel in blood cell image classification, showcasing remarkable classification performances. These innovations mark significant strides in computer vision-based blood cell analysis.

**Keywords** ConcatNeXt · Deep feature engineering · Nested patch division · Blood cell image classification · Computer vision

## 1 Introduction

Despite the widespread use of automated hematology analyzers, manual preparation of peripheral blood smears on glass slides for expert cytological interpretation remains relevant and useful for morphological confirmation of blood cell types [1] or identifying

specimens for further detailed characterization using advanced techniques [2]. A significant proportion of hematological diseases can be preliminarily diagnosed through peripheral blood smear examination of cell morphology [3]. For the accurate diagnosis of hematological malignancies, analysis of atypical cell morphology, in conjunction with molecular assessment, is obligatory. Peripheral blood smear examination has found applications in hematological diseases and beyond [4], often serving as a robust initial examination that may lead to further investigations [5].

The interpretation of peripheral blood smears is traditionally performed manually by trained laboratory professionals [6], which is labor-intensive and time-consuming. Digital microscopy and computer image analysis offer promising solutions for automated and objective morphological evaluation of blood cell images that can transform qualitative cytological assessments into reproducible quantitative readouts [7].

## 1.1 Literature review

We performed a nonsystematic review of the literature on works related to automated peripheral blood smear blood cell image classification. Patil et al. [8] employed canonical correlation analysis, recurrent neural networks, and convolutional neural networks (CNNs) to classify white blood cell images. On a balanced 4-class dataset comprising 623, 623, 620, and 624 images of eosinophils, lymphocytes, monocytes, and neutrophils, respectively, their model attained overall accuracy of 95.89%. Parab and Mehendale [9] used a CNN to classify red blood cell types using a 5000-image dataset divided into ten classes: normal, elliptocytes, spherocytes, microcytes, macrocytes, stomatocytes, tear-drop cells, sickle cells, Howell-Jolly cells, and codocytes. Their model attained excellent 98.50% accuracy. Liang et al. [10] classified blood cell images using a CNN-recursive network. Training their model on an augmented balanced dataset of 2483, 2497, 2499, and 2478 lymphocyte, eosinophil, neutrophil and monocyte images, respectively, they achieved 4-class classification accuracy of 90.79%. Nilufar et al. [11] deployed a joint histogram and Bhattacharya kernel in their blood cell classification model, which attained 92.04% overall accuracy. Su et al. [12] employed a hyperrectangular composite neural networks model for white blood cell classification. On a 450-image dataset, they attained an impressive 99.11% accuracy. Habibzadeh et al. [13] introduced a transfer learning approach for classifying white blood cells. On an imbalanced dataset of 88, 33, 21, and 207 eosinophil, lymphocyte, monocyte, and neutrophil images, respectively, they reported perfect 100.0% 4-class classification accuracy with ResNetV150. Almezghwi et al. [14] developed a model for classifying five white blood cell types that used image transformation and generative adversarial networks for data augmentation, and various deep networks for classification. Their best DenseNet-169-based model attained 98.8% validation accuracy. Gu and Sun [15] proposed an advanced deep learning model incorporating an attention mechanism into the YOLOv5 architecture, designated as AYOLOv5, aimed at improving the detection of blood cells. Their quantitative results from their study revealed that the AYOLOv5 model achieved a mean Average Precision (mAP) of 93.30%, marking a notable enhancement in performance when compared to the standard YOLOv5 model. Their study's reliance on the BCCD database for training and evaluation further validates the effectiveness of the AYOLOv5 model, offering a promising tool for the microscopic examination of blood samples with enhanced accuracy and reliability. Firat [16] presented a novel multibranch lightweight CNN-based method for the classification of microscopic peripheral blood cell images. Their approach incorporated the Inception module, Depthwise Squeeze-and-Excitation Block (DSEB), and

Pyramid Pooling Module (PPM) to efficiently and accurately classify blood cells. Their model was evaluated using three different datasets: BCCD, Raabin WBC, and PBC, achieving classification accuracies of 99.96% for BCCD, 99.22% for Raabin WBC, and 99.72% for PBC. Park et al. [17] developed a CNN-support vector machine algorithm for classifying circulating tumor cell (CTC) clusters based on morphological characteristics, without relying on immunofluorescence staining. The dataset was prepared using Wright–Giemsa staining, highlighting the morphological features of cells. This method demonstrated high sensitivity and specificity, achieving over 90% in classifying various configurations of CTC clusters. Murmu and Kumar [18] introduced a hybrid Deep CNN with Random Forest model for accurate malaria parasite detection in thin blood cell smear images. It utilized datasets from the National Library of Medicine (NLM), Kaggle, and the National Institutes of Health (NIH) comprising 27,558 cell images. Basophil, Eosinophil, and Erythroblast also exhibited high metrics, with Basophil reaching 99.94% accuracy, Eosinophil and Erythroblast both above 99.88% in accuracy, and IG with 99.3% accuracy. Chen et al. [19] developed Morphogo, a deep learning-based system for detecting circulating plasma cells (CPCs) in peripheral blood, aimed at improving diagnosis and monitoring of multiple myeloma (MM). Utilizing a comprehensive dataset comprising bone marrow and peripheral blood smears from patients, the system demonstrated high accuracy, sensitivity, and specificity in CPC detection. Morphogo achieved a 99.64% accuracy, 89.03% sensitivity, and 99.68% specificity, significantly outperforming manual microscopy. Dwivedi and Dutta [20] presented a novel CNN-based architecture named Microcell-Net, designed to enhance the classification of microscopic blood cell images across eight distinct classes. Their model aimed to address challenges such as inter-class and intra-class diversity, varying magnification levels, and image noise, which complicate the classification task. Utilizing a dataset from a public repository provided through Mendeley, consisting of 17,092 images, the Microcell-Net demonstrated exceptional performance, achieving a validation accuracy of 98.76% and a test accuracy of 97.65%. Bhuiyan and Islam [21] developed an ensemble learning-based deep neural network for the accurate classification of malaria parasites in red blood cell images. Utilizing a dataset from the National Institutes of Health, which comprised 27,558 cell images equally divided between parasitized and uninfected cells, their study demonstrated the effectiveness of combining adaptive weighted average and max voting ensemble techniques. Their study achieved an accuracy of 97.92%, outperforming other models and methods. Elhassan et al. [22] developed a two-stage hybrid model leveraging a deep convolutional autoencoder and deep CNN for classifying atypical white blood cells in Acute Myeloid Leukemia (AML). Utilizing the AML Cytomorphology LMU dataset, which includes 18,365 single-cell images from AML patients and controls, the model demonstrated impressive performance, achieving an average accuracy of 97%, sensitivity of 97%, and precision of 98%. Leng et al. [23] presented an enhanced Detection Transformer (DETR) model for leukocyte detection in microscopic blood images, named Improved-DETR. The study focused on leveraging the Pyramid Vision Transformer (PVT) and a Deformable Attention Module (DAM) to enhance detection accuracy and speed. Utilizing the Raabin dataset, which contains high-resolution images of real medical scenarios, the model achieved a mean average precision (mAP) of 96.10, outperforming traditional DETR and CNN. Barrera et al. [24] developed a method for creating synthetic images of white blood cells. Using generative adversarial networks (GANs), it produced images that closely mimic real cell morphology. Their approach aimed to enhance automatic recognition model training by generating a diverse range of high-quality, artificial leukocyte images, including those characteristic of leukemic conditions. Their study achieved an accuracy of 100.0%.

## 1.2 Literature gaps

From the literature review, we observed that prior publications were based on blood cell image datasets with relatively small sample sizes, with limited number of classes. Notably, modern CNN architectures, specifically ConvNeXt, ConvNeXt V2, or their variants, have not been studied. Moreover, while the researchers have reported classification performance on the blood cell image datasets, there was little focus on explainability, which is an important consideration for clinical adoption by users.

## 1.3 Motivation and our model

We aimed to develop an accurate automated computer vision-based model for detecting a broad range of blood cell types using a novel deep learning architecture. Among CNNs, residual networks (ResNet) stand out for their efficient classification performance, which encompassed skipped connections that overcame the problem of vanishing gradients [25]. We were inspired by the novel ConvNeXt architecture [26], which has been built on the ResNet architecture, to develop a new model with reduced operators and parameters without compromising classification accuracy. Moreover, we aspired to surpass machine learning models that have been trained on datasets with limited number of classes [27–29] by taking on the challenge of studying a much larger blood cell image dataset with expanded number of classes [30].

In this work, we introduced a new ConvNeXt- and swin transformer-inspired deep CNN, ConcatNeXt. We replaced ConvNeXt's Gaussian error linear unit (GELU) and layer normalization functions with rectified linear unit (ReLU) and batch normalization, respectively; applied depth concatenation blocks to the outputs of the layers for effective information fusion; and incorporated a patchify layer, akin to swin transformer [31]. We combined the proposed ConcatNeXt with a nested patch-based deep feature engineering, which enabled more efficient patch-based extraction of local features with reduced complexity compared with fixed-size. We employed effective downstream iterative neighborhood component analysis (INCA) [32]-based and SVM [33]-based feature selection and classification functions, respectively.

## 1.4 Innovation and contributions

**Innovations** We have proposed a novel lightweight deep CNN, ConcatNeXt, which could perform classification tasks independently or in combination with downstream nested patch-based deep feature engineering. These innovations are:

- We introduce ConcatNeXt, a pioneering lightweight deep CNN meticulously crafted for versatile classification tasks. The model reveals the inherent capability to operate independently and synergistically with downstream nested patch-based deep feature engineering.
- We have presented a nested patch-based deep feature engineering model, complemented by a compelling demonstration highlighting the transfer learning proficiency inherently embedded within ConcatNeXt.

**Contributions** Main contributions of this work are developing two innovative models, ConcatNeXt and ConcatNeXt-based deep model, leveraging pretrained ConcatNeXt,

nested patch division, INCA, and SVM. These models undergo comprehensive assessment on an extensive and diverse blood cell image dataset, yielding noteworthy results:

- Our developed system involves the introduction of ConcatNeXt, characterized by a simple architecture with exceptional performance.
- We have showcased that ConcatNeXt and ConcatNeXt-based deep models exhibit exemplary test and validation accuracies, both surpassing the remarkable threshold of 97%. This underscores the robust performance and reliability of these innovative models.

## 2 Materials and methods

### 2.1 Study dataset

We downloaded an open-access image dataset comprising 17092 RGB images of blood cells divided into eight categories: basophils (1218), eosinophils (3117), erythroblasts (1551), immature granulocytes (2895), lymphocytes (1214), monocytes (1420), neutrophils (3329), and platelets (2348) [3, 30]. The images depicting each blood cell type on a background of normal erythrocytes were resized to  $360 \times 363$  and partitioned into training and testing sets (Table 1).

### 2.2 ConcatNeXt network

In this study, our primary goal is to introduce a novel deep learning algorithm, thereby making a valuable contribution to the field of CNN research methodology. Additionally, it is to put forth an innovative, lightweight CNN model. Our contributions extend beyond proposing a deep learning model; we have also introduced a deep feature engineering model leveraging the suggested CNN architecture. This section provides comprehensive details about the proposed ConcatNeXt.

We developed a novel CNN architecture called ConcatNeXt by replacing the layer normalization and GELU activation functions of the ConvNeXt block with batch normalization and ReLU activation functions, respectively. We incorporated a depth concatenation operation alongside maximum pooling to increase the number of channels during down-sampling. Before the flatten (global average pooling) layer, we applied a patchify operator,

**Table 1** Distribution of the dataset used

Class	Training	Testing	Total
Basophils	914	304	1218
Eosinophils	2340	777	3117
Erythroblasts	1164	387	1551
Immature granulocytes	2172	723	2895
Lymphocytes	912	302	1214
Monocytes	1068	352	1420
Neutrophils	2497	832	3329
Platelets	1764	584	2348
Total	12,831 (75.07%)	4261 (24.93%)	17,092

inspired by the convolution-based downsampling in the swin transformer, and a patchify stem, utilizing a non-overlapping convolution with a  $4 \times 4$  sized filter. The configuration of our proposed ConcatNeXt is as follows:  $C = (96, 192, 384, 768)$ ,  $B = (2, 2, 2, 2)$ . Here,  $C$ : is the number of filters, and cap  $B$  is the mber of repetitions. The model schema and specifications are depicted in Fig. 1 and Table 2, respectively. The global average pooling layer (GAP) extracted 2112 features ( $768 + 768 + 384 + 192$ ) from each input. The model comprises 6.4 million trainable parameters and 74 operations, also called blocks in the MATLAB deep learning designer.

According to Table 2, the algorithm has been explained below.

*Stem:* Commencing the network is the Stem layer, which handles an input size of  $224 \times 224$ . Employing a  $4 \times 4$  operation featuring 96 channels and a stride of 4, this layer produces an output size of  $56 \times 56$ . In this layer, we implemented a patchify operator reminiscent of the approaches seen in swin transformer and ConvNeXt.

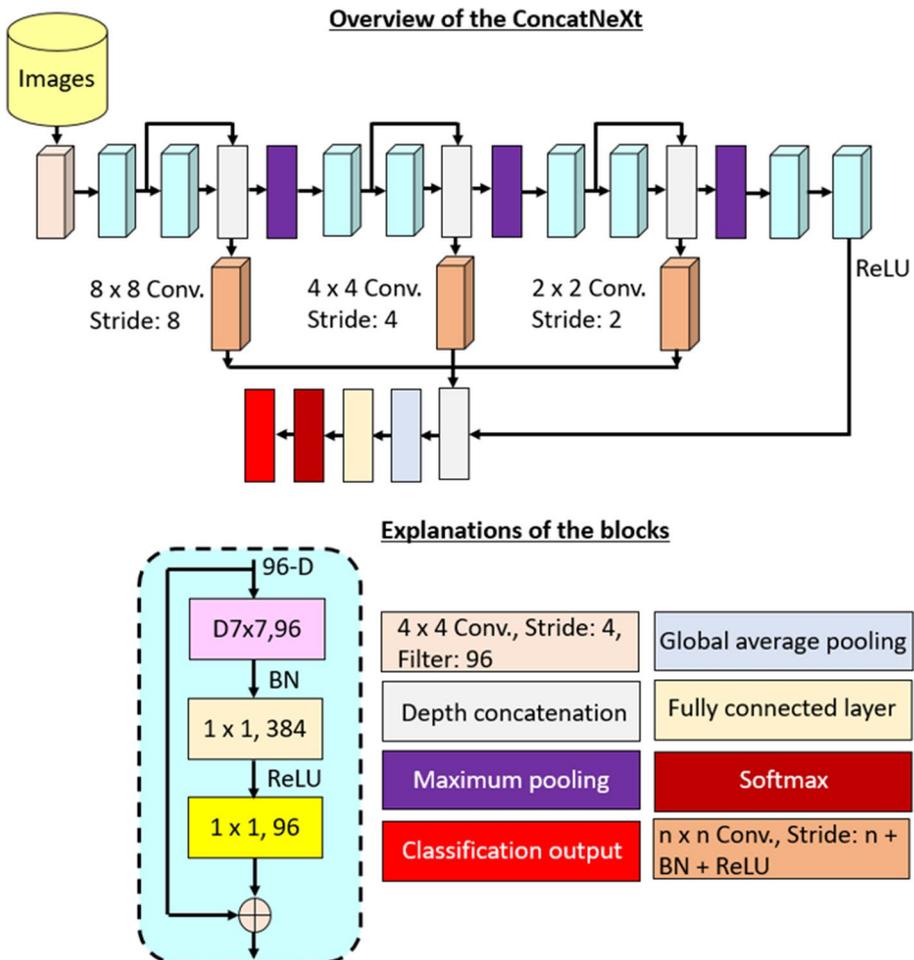


Fig. 1 Schema of the proposed ConcatNeXt. \*\*BN: batch normalization; D: Depthwise convolution

**Table 2** ConcatNeXt architecture specification

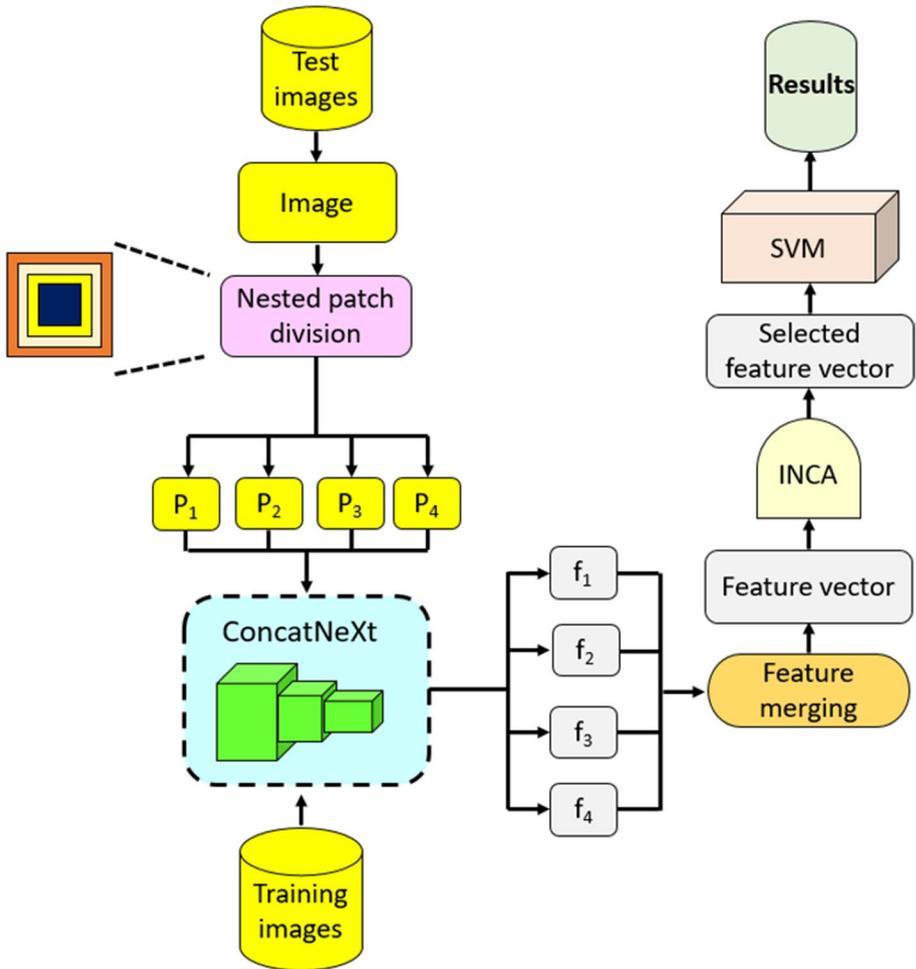
Layer	Input size	Operation	Output size
Stem	$224 \times 224$	$4 \times 4, 96$ , stride: 4	$56 \times 56$
Layer 1	$56 \times 56$	$\begin{bmatrix} D7 \times 7, 96 \\ 1 \times 1, 384 \\ 1 \times 1, 96 \end{bmatrix} \times 2$ $3 \times 3$ maximum pooling, stride: 2 Depth concatenation	$28 \times 28$
Layer 2	$28 \times 28$	$\begin{bmatrix} D7 \times 7, 192 \\ 1 \times 1, 768 \\ 1 \times 1, 192 \end{bmatrix} \times 2$ $3 \times 3$ maximum pooling, stride: 2 Depth concatenation	$14 \times 14$
Layer 3	$14 \times 14$	$\begin{bmatrix} D7 \times 7, 384 \\ 1 \times 1, 1536 \\ 1 \times 1, 384 \end{bmatrix} \times 2$ $3 \times 3$ maximum pooling, stride: 2 Depth concatenation	$7 \times 7$
Layer 4	$7 \times 7$	$\begin{bmatrix} D7 \times 7, 768 \\ 1 \times 1, 3072 \\ 1 \times 1, 768 \end{bmatrix} \times 2$	$7 \times 7$
Layer 5	$7 \times 7$	Layer 1 + $8 \times 8, 192$ , stride: 8 Layer 2 + $4 \times 4, 384$ , stride: 4 Layer 3 + $2 \times 2, 768$ , stride: 2 Layer 4 + ReLU, 768	$7 \times 7$
Output size	$7 \times 7$	Global average pooling, fully connected layer, softmax	Number of classes

*Layers 1–5:* We introduce the primary building block and the downsampling process within these layers. The main block encompasses three convolutions with filter sizes of  $7 \times 7$ ,  $1 \times 1$ , and  $1 \times 1$ , akin to ConvNeXt. For downsampling, a  $3 \times 3$  maximum pooling operation is employed. These layers undergo repetition twice, constituting the fundamental feature map generation layers.

*Output:* The ultimate layer concludes with the production of classification outcomes. Various patchify layers ( $8 \times 8$ ,  $4 \times 4$ , and  $2 \times 2$  sized patchify convolutions were employed for downsampling) were used to concatenate all layers before the GAP layer, and the primary innovation in our proposal lies in this operator. The operations involved encompass GAP, a fully connected layer, and Softmax operators.

### 2.3 ConcatNeXt combined with nested patch-based deep feature engineering

We combined the proposed ConcatNeXt with nested patch-based deep feature engineering; the latter encompassed iterative neighborhood component analysis (INCA)-based feature selection and support vector machine (SVM)-based classification (Fig. 2). Comprehensive explanation of the model is provided in the following subsections.



**Fig. 2** Block diagram of the presented deep feature engineering model. From the input image, four patches (P) were created, each of which was input to the pretrained ConcatNeXt for deep feature extraction. The GAP of ConcatNeXt generated four feature vectors (f), each of length 2112, which were concatenated to form the final feature vector of length 8448 ( $=2112 \times 4$ ). INCA function selected the optimal length of the most informative features, which were in turn fed to SVM function for classification using robust tenfold cross-validation

### 2.3.1 Feature extraction

The steps of nested patch-based deep feature extraction are:

- Step 1:* Read each test image from the test dataset and resize each image to  $224 \times 224$ .
- Step 2:* Apply nested patch division to each image. In Eq. (1), we have defined the used nested patch division.

$$P_k = I(c - counter \times k + 1 : c + counter \times k),$$

$$k \in \{1, 2, 3, 4\}, counter = \frac{224}{8}, c = \frac{224}{2} \quad (1)$$

where  $I$  represents blood cell image; and  $P$ , patch. The sizes of the generated patches were  $56 \times 56$ ,  $112 \times 112$ ,  $168 \times 168$ , and  $224 \times 224$ . In this aspect, the fourth patch is the original image.

*Step 3:* Extract deep features from each patch using the GAP layer of the pretrained ConcatNeXt. In Eq. (2), we have depicted the feature extraction process. where  $f$  represent the feature vector.

$$f_k = \text{ConcatNeXt}(P_k, \text{GAP}) \quad (2)$$

*Step 4:* Create the final feature vector by merging the generated feature vectors. The mathematical formulation for the feature merging process is given in Eq. (3). where  $F$  represents the final feature vector of length of 8448.

$$F(g + 2112 \times (k - 1)) = f_k(g), g \in \{1, 2, \dots, 2112\} \quad (3)$$

### 2.3.2 Feature selection

For this step, we deployed INCA [32], an enhancement of neighborhood component analysis, that uses an iterative classification accuracy calculator to select the optimal number of the most discriminative features in the final feature vector. The steps are:

*Step 5:* Compute the qualified/sorted indexes of the generated features using neighborhood component analysis. In Eqs. (3) and (4), we have defined min–max normalization and the qualified index creation.

$$F = \frac{F - \min(F)}{\max(F) - \min(F)} \quad (4)$$

$$index = \text{fscnca}(F, y) \quad (5)$$

where  $index$  represents qualified/sorted indexes of the features;  $y$ , actual/real outputs; and  $\text{fscnca}(\cdot, \cdot)$ , neighborhood component analysis function. Min–max normalization was applied prior.

*Step 6:* Choose features iteratively based on the generated indexes. The iterative selection process is defined in Eq. (6). where  $\text{fsel}$  represents selected feature vectors;  $NI$ , number of images;  $\text{stval}$ , implied start value of the loop; and  $\text{finval}$ , finite value of the loop. Here, the loop range was set from 100 to 1000,

$$\text{fsel}^{\text{stval}+1}(d, h) = F(d, \text{index}(h)),$$

$$d \in \{1, 2, \dots, NI\}, h \in \{1, 2, \dots, t\},$$

$$d \in \{1, 2, \dots, NI\}, h \in \{1, 2, \dots, t\}, \quad (6)$$

*Step 7:* Compute classification accuracies of the selected feature vectors. The mathematical expression of this step is given in Eq. (7). where  $ac$  represents classification

accuracy; and  $\varphi(., .)$ , classification accuracy calculation function. Here, the cubic SVM classifier was used as the classification accuracy calculation function.

$$ac(t - stval + 1) = \varphi(fsel^{t-stval+1}, y) \quad (7)$$

*Step 8:* Generate the selected feature vector with the optimal length of the most discriminative features according to classification accuracy. The selection of the best feature vector is defined in Eqs. (8) and (9). where  $idx$  represents the index of the selected feature vector with maximum accuracy; and  $feat$ , the selected feature vector.

$$idx = \max(ac) \quad (8)$$

$$feat = fsel^{idx+stval-1} \quad (9)$$

### 2.3.3 Classification

The selected feature vector was input into the SVM classifier with the following configuration: one-vs-all coding, third-degree polynomial kernel function, and a box constraint level of 1. A tenfold cross-validation (CV) strategy was adopted. The final step of this deep feature engineering model is the classification, as outlined below.

*Step 9:* Classify the selected feature vector using the SVM classifier with a tenfold CV. In Eq. (10), we have defined the classification process.

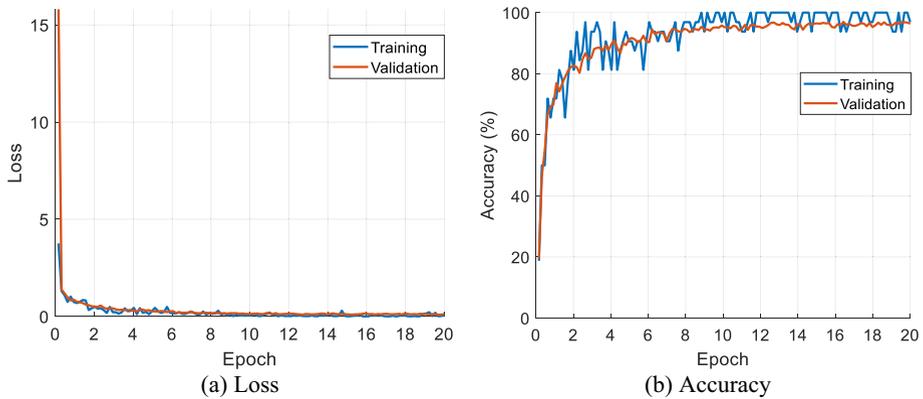
$$result = SVM(feat, y) \quad (10)$$

The above 10 steps define our proposed ConcatNeXt-based deep feature engineering model.

## 3 Performance analysis

### 3.1 Experimental setup

The model was designed in MATLAB (2023a) deep network designer using m files and implemented on a personal computer with the following specifications: Intel @i9 9th generation central processing unit; GeForce RTX 2070 graphics processing unit; 64 GB RAM; 512 GB HDD; and Windows 11 operating system. The pretrained ConcatNeXt model was saved as a mat file and then applied to the nested patches of images in the training set. Training parameters settings were initial learning rate, 0.01; minimum batch size, 32; maximum epoch, 20; and solver, stochastic gradient descent with momentum. We adopted a split ratio of 70:30 for the training and validation sets; accordingly, the distribution of our training, validation, and test sets was approximately 52.5%, 22.5%, and 25%, respectively. The ConcatNeXt extracted 2112 features from each of the four nested patches, concatenated into a final feature vector of length 8448. Setting a loop range of 100 to 1000, INCA generated a selected feature vector that contained 412 of the most informative among the 8448 extracted features, which was then input to downstream SVM for classification using a tenfold CV. The parameters of the used SVM are given as follows.



**Fig. 3** Loss and accuracy curves of the ConcatNeXt

**Table 3** Performance evaluation

Class	ConcatNeXt				ConcatNeXt plus nested patch-based deep feature engineering			
	Rec (%)	Pre (%)	F1 (%)	Acc (%)	Rec (%)	Pre (%)	F1 (%)	Acc (%)
Basophil	98.03	97.39	97.70	-	98.36	99.34	98.84	-
Eosinophil	99.74	100.00	99.87	-	100.00	99.87	99.94	-
Erythroblast	97.93	97.43	97.68	-	98.45	99.22	98.83	-
Immature granulocyte	96.82	93.21	94.98	-	97.23	96.70	96.97	-
Lymphocyte	98.68	98.35	98.51	-	100	99.67	99.83	-
Monocyte	96.02	97.13	96.57	-	98.58	98.02	98.30	-
Neutrophil	95.55	98.73	97.13	-	97.84	98.19	98.01	-
Platelet	99.83	99.83	99.83	-	100.00	99.83	99.91	-
Overall	97.83	97.76	97.79	97.77	98.81	98.85	98.83	98.73

\*\*Acc: accuracy; F1: F1 score; Pre: precision; Rec: recall

Kernel: Cubic polynomial order, box constraint level: 1, coding: one-vs-all, validation: ten-fold cross-validation.

### 3.2 Results

ConcatNeXt attained 100% and 97.43% accuracies during training and validation, respectively, with a validation loss of 0.0942 (Fig. 3). On the testing set, ConcatNeXt attained excellent 8-class classification performance using standard metrics: 97.83% unweighted average recall, 97.76% unweighted average precision, 97.79% overall F1 score, and 97.77% overall accuracy (Table 3). Moreover, the ConcatNeXt-based deep feature engineering model also attained excellent 98.81% unweighted average recall, 98.85% unweighted average precision, 98.83% overall F1-score, and 98.73% overall classification accuracy on the

testing set (Table 3). For both ConcatNeXt and ConcatNeXt-based deep feature engineering model, there were low rates of misclassification (Fig. 4).

In Fig. 4, 1, 2, 3, 4, 5, 6, 7, and 8 denote “Basophil”, “Eosinophil”, “Erythroblast”, “Immature granulocyte”, “Lymphocyte”, “Monocyte”, “Neutrophil”, and “Platelet” classes, respectively. For both experiments, the largest number of misclassifications were images in the “Neutrophil” class misclassified as the “Immature granulocyte” class, both of which are closely related (a neutrophil is a mature granulocyte).

### 4 Discussion

In our proposed model, ConcatNeXt is a novel, lightweight convolutional neural network (CNN) that excelled in the classification of blood cell images. The ConcatNeXt is simple and obtained exceptional performance, showcasing a strategic balance between the two. The model comprises two key components: the introduced ConcatNeXt and a deep feature engineering model based on ConcatNeXt.

Within ConcatNeXt, a new block has been introduced, creating a streamlined and straightforward CNN with a mere 74 operations. This renders ConcatNeXt a straightforward yet effective deep learning model, featuring approximately 6.4 million learnable parameters.

The proposed ConcatNeXt is the foundation for a nested patch deep feature engineering model, highlighting the model’s robust transfer learning capabilities. This research integrates deep learning and feature engineering seamlessly, strategically weaving connectivity between architecture and functionality.

ConcatNeXt achieved outstanding accuracies of 100% in training, 97.43% in validation, and 97.77% in testing on a substantial blood cell image dataset. Building upon this network’s GAP, additional components such as nested patch division, INCA, and SVM were introduced to create an advanced deep feature engineering model. This new model achieved an impressive overall 8-class classification accuracy of 98.73%. The final choice of SVM was based on a comprehensive comparison with seven other standard classifiers, including linear discriminant analysis, k-nearest neighbors, decision tree, artificial neural

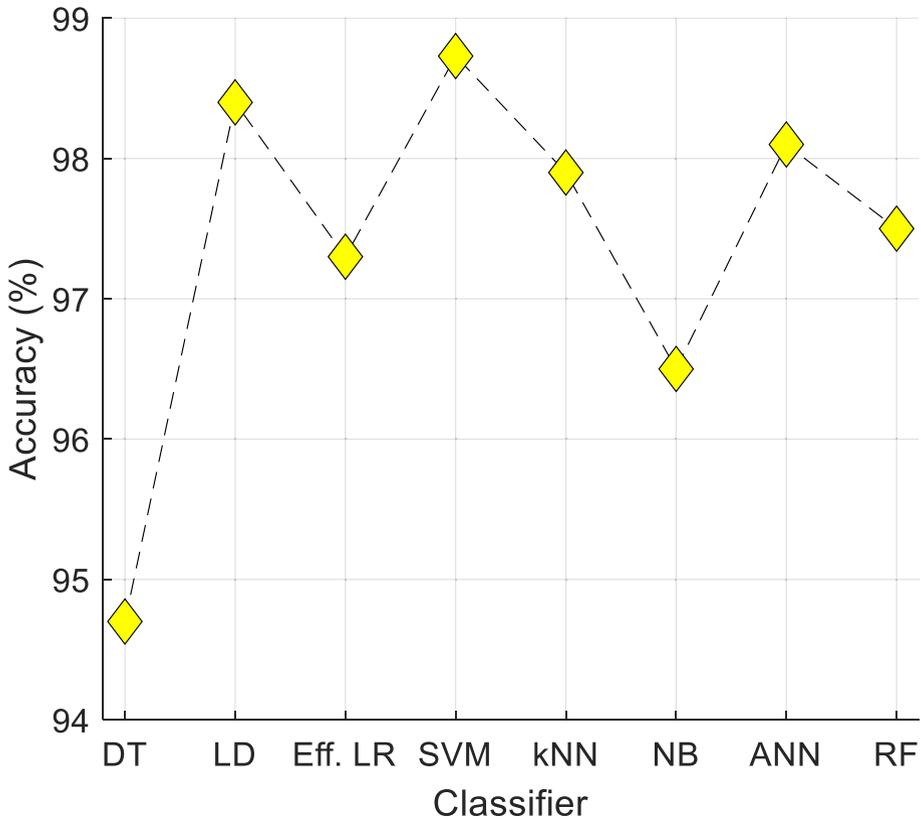
1	298			5	1			
2	1	775					1	
3			379	1	2	2	2	1
4	5		5	700	2	5	6	
5		2		298	2			
6		2	12		338			
7	2	1	33		1	795		
8							1	583
	1	2	3	4	5	6	7	8

(a) ConcatNeXt

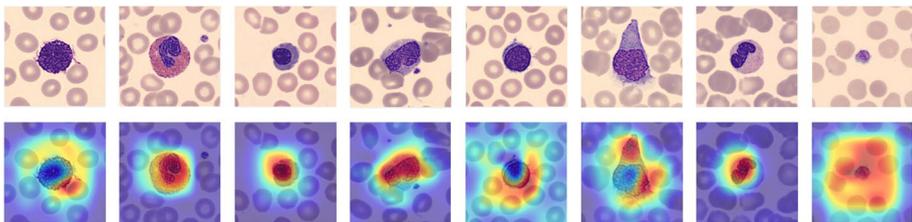
1	299		1	3			1	
2		777						
3			381	1	1	2	1	1
4	2		2	703		4	12	
5				302				
6		1		3	347	1		
7				17	1	814		
8								584
	1	2	3	4	5	6	7	8

(b) ConcatNeXt-based deep feature engineering model

Fig. 4 Confusion matrixes



**Fig. 5** Comparison of accuracies of various classifiers using the ConcatNeXt+INCA features. \*\* DT: Decision Tree, LD: Linear Discriminant, Eff. LR: Efficient Logistic Regression, SVM: Support Vector Machine, kNN: k Nearest Neighbor, NB: Naïve Bayes, ANN: Artificial Neural Network, RF: Random Forest



**Fig. 6** Original blood cell images (top row) and their corresponding heat maps generated by applying Grad-CAM to ConcatNeXt-based deep feature engineering model (bottom row). \*\*Columns 1 to 8 depict images corresponding to “Basophils”, “Eosinophils”, “Erythroblasts”, “Immature granulocytes”, “Lymphocytes”, “Monocytes”, “Neutrophils”, and “Platelets”, respectively

network, random forest, efficient logistic regression, and naïve Bayes, conducted using the MATLAB Classification Learner Toolbox. The SVM classifier outperformed these classifiers, achieved an overall accuracy of 98.73% (see Fig. 5).

## 4.1 Comparative results

In order to show superior classification performance of the proposed ConcatNeXt for the used blood image cell dataset, we performed another series of experiments to compare the classification accuracy of our proposed ConcatNeXt network with other benchmark CNNs: ResNet18 [34], ResNet50 [34], ResNet101 [34], MobileNetV2 [35], DarkNet53 [36], Xception [37], ShuffleNet [38], DenseNet201 [39], and InceptionV3 [40]. Applying these models to the same blood cell image dataset, we obtained results that were inferior to our proposed ConcatNeXt network and ConcatNeXt-based deep feature engineering model (Table 4).

Among the benchmark models, DenseNet201 achieved the highest classification accuracy, reaching 96.13%. However, it's crucial to highlight the substantial computational difference between DenseNet201 and our proposed ConcatNeXt. While DenseNet201 utilized 708 operations and featured a total of 20 million trainable parameters, ConcatNeXt employed only 74 operations and had 6.4 million trainable parameters. Despite this significant contrast in computational complexity, ConcatNeXt not only demonstrated efficiency but also achieved superior classification performance compared to the other benchmark models.

Among published literature, our work bears the most resemblance to Acevedo et al. [3], who developed a two-stage models that combined VGG16 or Inceptionv3 networks with SVM or Softmax classifiers. The model underperformed our proposed ConcatNeXt network and ConcatNeXt-based deep feature engineering (Table 5).

Table 5 demonstrates a detailed comparison between our proposed ConcatNeXt network and ConcatNeXt-based deep feature engineering model and the approach presented by [3, 20, 41]. The comparison includes various models implemented by Acevedo et al., each combining different CNN architectures with either SVM or Softmax classifiers using fivefold cross-validation. Our proposed methods, ConcatNeXt and ConcatNeXt combined with nested patch-based deep feature engineering with tenfold cross-validation, achieved superior classification accuracies of 97.77% and 98.73%, respectively. Notably, our models outperformed Acevedo et al.'s best-performing model, which employed VGG16 with

**Table 4** Comparison of image classification performance of established convolutional neural networks and our proposed ConcatNeXt network and ConcatNeXt-based model

Convolutional neural network	Overall accuracy (%)	Unweighted average recall (%)	Unweighted average precision (%)	Overall F1 score (%)
ResNet18	87.15	86.26	87.46	86.75
ResNet50	94.25	93.53	94.34	93.90
ResNet101	94.10	93.28	94.16	93.66
MobileNetv2	93.97	93.13	94.00	93.52
DarkNet53	95.01	94.30	94.89	94.56
Xception	92.89	91.90	92.81	92.33
ShuffleNet	90.06	89.12	89.68	89.31
DenseNet201	96.13	95.72	96.20	95.94
Inceptionv3	93.36	92.57	93.06	92.78
ConcatNeXt	97.77	97.83	97.76	97.79
ConcatNeXt, INCA, SVM	98.73	98.81	98.85	98.83

**Table 5** Comparison of our proposed ConcatNeXt network and ConcatNeXt-based deep feature engineering model with Acevedo et al. [3]

Method	Model	Accuracy (%)
Acevedo et al. [3]	VGG16+SVM+ fivefold CV	87.40
	Inceptionv3+SVM+ fivefold CV	90.50
	VGG16+ Softmax + fivefold CV	96.20
	Inceptionv3+ Softmax + fivefold CV	94.90
Dwivedi and Dutta [20]	Microcell-Net+ 88:12 split ratio	97.65
Tseng and Huang [41]	CNN	90.10
Our proposed method	ConcatNeXt+ Softmax + 70:30	97.77
	ConcatNeXt+ INCA + SVM + tenfold CV	98.73

Softmax and achieved an accuracy of 96.2%. This emphasizes the effectiveness of our proposed ConcatNeXt-based models in blood cell image classification.

## 4.2 Explainable results

To enhance the explainability of our proposed ConcatNeXt-based deep feature engineering model, we applied gradient-weighted class activation map (Grad-CAM) [42–44] a commonly employed technique for visualizing interpretable results, to selected blood cell images from each class. On the generated heat maps (Fig. 6), red-colored areas depicting regions of the images that contributed the most to the classification results are seen to correspond visually to the locations of the distinct blood cells. These heat maps provide valuable insights into ConcatNeXt's decision-making process, and indirect support for its validity and efficacy in automatically identifying relevant features for accurate classification. Of note, the ROIs are centered, further affirming the effectiveness of the nested patch division-based deep feature engineering model in achieving enhanced classification performances.

Based on the Grad-CAM results depicted in Fig. 6, our proposed ConcatNeXt exhibits a notable focus on the region of interest (ROI) corresponding to each cell category. These results serve as validation for the high classification performance achieved by the presented ConcatNeXt. Notably, the ROIs are centered, further affirming the effectiveness of the nested patch division-based deep feature engineering model in achieving enhanced classification performances.

## 4.3 Advantages and limitations

The ConcatNeXt attained 97.77% test accuracy on the dataset; the ConcatNeXt-based deep feature engineering model, 98.73%. The latter attained perfect 100% recall for the Eosinophil, Lymphocyte, and Platelet classes.

**Advantages** Both models attained high classification performance (Table 3). Moreover, we have provided explainable results using Grad-CAM, which would provide valuable insights into model decision-making. The proposed ConcatNeXt is a computationally lightweight CNN, featuring relatively low numbers of 6.4 million trainable parameters and 74 operations.

**Limitation** The study dataset comprised mainly white blood cell classes. In the future, we plan to expand our dataset to include a more diverse range of abnormal red blood cells as well.

## 5 Conclusions

We have introduced a novel CNN architecture, ConcatNeXt, drawing inspiration from ConvNeXt and the swin transformer. Also, we proposed a nested patch-based deep feature engineering model, contributing significantly to enhanced classification performance. A key highlight of our work lies in the simplicity and efficiency of ConcatNeXt. Boasting only 74 operations and 6.4 million trainable parameters, our model as a lightweight and efficient CNN architecture delivering remarkable performance. Furthermore, the presented ConcatNeXt is a straightforward CNN.

Using a big 8-class blood cell image dataset, ConcatNeXt and the ConcatNeXt-based deep feature engineering model achieved outstanding multi-class classification accuracies of 97.77% and 98.73%, respectively, surpassing state-of-the-art models (Table 4). Additionally, by incorporating Grad-CAM explanation results, our models revealed valuable insights into the decision-making process, of further validating the model's capability to focus on critical regions of the images.

The proposed ConcatNeXt and nested patch-based deep feature engineering models have showcased high classification performances, making substantial contributions to the field of blood cell classification. We plan to use these models for the classification of various organs/cells using different imaging modalities.

**Author contributions** Conceptualization ME, PDB, SD, TT, RST, URA; formal analysis, ME, PDB, SD, TT, RST, URA; investigation, ME, PDB, SD, TT; methodology, ME, PDB, SD, TT; software, SD, TT; project administration, URA.; resources, ME; supervision, URA; validation, ME, PDB, SD, TT, RST, URA; visualization, ME, PDB, SD, TT; writing—original draft, ME, PDB, SD, TT, RST, URA; writing—review and editing, ME, PDB, SD, TT, RST, URA; All authors have read and agreed to the published version of the manuscript.

**Funding** Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK). The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

**Data availability** The data used in this study were downloaded from [3, 30].

## Declarations

**Ethical approved** Not applicable

**Competing interests** The authors have no relevant financial or non-financial interests to disclose.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Adewoyin A (2014) Peripheral blood film-a review. *Annals of Ibadan postgraduate medicine* 12(2):71–79
2. Craig F (2017) The utility of peripheral blood smear review for identifying specimens for flow cytometric immunophenotyping. *Int J Lab Hematol* 39:41–46
3. Acevedo A, Alf3rez S, Merino A, Puigv3 L, Rodellar J (2019) Recognition of peripheral blood cell images using convolutional neural networks. *Comput Methods Programs Biomed* 180:105020
4. Merino A, Vlagea A, Molina A, Egri N, Laguna J, Barrera K, Bold3 L, Acevedo A, D3az-Pav3n M, Sibina F (2020) Atypical lymphoid cells circulating in blood in COVID-19 infection: morphology, immunophenotype and prognosis value. *J Clin Pathol* 2:104
5. Alf3rez S, Merino A, Bigorra L, Mujica L, Ruiz M, Rodellar J (2015) Automatic recognition of atypical lymphoid cells from peripheral blood by digital image analysis. *Am J Clin Pathol* 143(2):168–176
6. Fischbach FT, Dunning MB (2009) *A manual of laboratory and diagnostic tests*. Lippincott Williams & Wilkins
7. Merino A, Puigv3 L, Bold3 L, Alf3rez S, Rodellar J (2018) Optimizing morphology through blood cell image analysis. *Int J Lab Hematol* 40:54–61
8. Patil A, Patil M, Birajdar G (2021) White blood cells image classification using deep learning with canonical correlation analysis. *IRBM* 42(5):378–389
9. Parab MA, Mehendale ND (2021) Red blood cell classification using image processing and CNN. *SN Computer Science* 2(2):1–10
10. Liang G, Hong H, Xie W, Zheng L (2018) Combining convolutional neural network with recursive neural network for blood cell image classification. *IEEE Access* 6:36188–36197
11. Nilufar S, Ray N, Zhang H (2008) Automatic blood cell classification based on joint histogram based feature and Bhattacharya Kernel. In: 2008 42nd Asilomar conference on signals, systems and computers. IEEE, pp 1915–1918. <https://doi.org/10.1109/ACSSC.2008.5074762>
12. Su M-C, Cheng C-Y, Wang P-C (2014) A neural-network-based approach to white blood cell classification. *Sci World J*. <https://doi.org/10.1155/2014/796371>
13. Habibzadeh M, Jannesari M, Rezaei Z, Baharvand H, Totonchi M (2018) Automatic white blood cell classification using pre-trained deep learning models: ResNet and Inception. In: Tenth international conference on machine vision (ICMV 2017). *Int Soc Optics Photonics* 1069612. <https://doi.org/10.1117/12.2311282>
14. Almezghwi K, Serte S (2020) Improved classification of white blood cells with the generative adversarial network and deep convolutional neural network. *Computational Intelligence and Neuroscience*. <https://doi.org/10.1155/2020/6490479>
15. Gu W, Sun K (2024) AYOLOv5: Improved YOLOv5 based on attention mechanism for blood cell detection. *Biomed Signal Process Control* 88:105034
16. F3rat H (2024) Classification of microscopic peripheral blood cell images using multibranch lightweight CNN-based model. *Neural Comput Appl* 36(4):1599–1620
17. Park J, Ha S, Kim J, Song J-W, Hyun K-A, Kamiya T, Jung H-I (2024) Classification of circulating tumor cell clusters by morphological characteristics using convolutional neural network-support vector machine. *Sens Actuators, B Chem* 401:134896
18. Murmu A, Kumar P (2024) DLRFNet: deep learning with random forest network for classification and detection of malaria parasite in blood smear. *Multimedia Tools Appl* 1–23. <https://doi.org/10.1007/s11042-023-17866-6>
19. Chen P, Zhang L, Cao X, Jin X, Chen N, Zhang L, Zhu J, Pan B, Wang B, Guo W (2024) Detection of circulating plasma cells in peripheral blood using deep learning-based morphological analysis. *Cancer*. <https://doi.org/10.1002/cncr.35202>
20. Dwivedi K, Dutta MK (2023) Microcell-Net: A deep neural network for multi-class classification of microscopic blood cell images. *Expert Systems*:e13295. <https://doi.org/10.1111/exsy.13295>
21. Bhuiyan M, Islam MS (2023) A new ensemble learning approach to detect malaria from microscopic red blood cell images. *Sensors Int* 4:100209
22. Elhassan TA, Mohd Rahim MS, Siti Zaiton MH, Swee TT, Alhaj TA, Ali A, Aljurf M (2023) 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* 13(2):196
23. Leng B, Wang C, Leng M, Ge M, Dong W (2023) Deep learning detection network for peripheral blood leukocytes based on improved detection transformer. *Biomed Signal Process Control* 82:104518

24. Barrera K, Merino A, Molina A, Rodellar J (2023) Automatic generation of artificial images of leukocytes and leukemic cells using generative adversarial networks (syntheticcellgan). *Comput Methods Programs Biomed* 229:107314
25. Jain R, Nagrath P, Kataria G, Kaushik VS, Hemanth DJ (2020) Pneumonia detection in chest X-ray images using convolutional neural networks and transfer learning. *Measurement* 165:108046
26. Liu Z, Mao H, Wu C-Y, Feichtenhofer C, Darrell T, Xie S (2022) A convnet for the 2020s. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* 11976–11986. <https://doi.org/10.48550/arXiv.2201.03545>
27. Yao X, Sun K, Bu X, Zhao C, Jin Y (2021) Classification of white blood cells using weighted optimized deformable convolutional neural networks. *Artif Cells, Nanomed, Biotechnol* 49(1):147–155
28. Khan A, Eker A, Chefranov A, Demirel H (2021) White blood cell type identification using multi-layer convolutional features with an extreme-learning machine. *Biomed Signal Process Control* 69:102932
29. Baghel N, Verma U, Nagwanshi KK (2021) WBCs-Net: type identification of white blood cells using convolutional neural network. *Multimedia Tools Appl* 1–17. <https://doi.org/10.1007/s11042-021-11449-z>
30. Acevedo A, Merino A, Alférez S, Molina A, Boldú L, Rodellar J (2020) A dataset of microscopic peripheral blood cell images for development of automatic recognition systems. *Data in Brief*, ISSN: 23523409, 30. <https://doi.org/10.1016/j.dib.2020.105474>
31. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo (2021) Swin transformer: Hierarchical vision transformer using shifted windows. In: *Proceedings of the IEEE/CVF international conference on computer vision* 10012–10022. <https://doi.org/10.1109/ICCV48922.2021.00986>
32. Tuncer T, Dogan S, Özyurt F, Belhaouari SB, Bensmail H (2020) Novel multi center and threshold ternary pattern based method for disease detection method using voice. *IEEE Access* 8:84532–84540
33. Vapnik V (1998) The support vector method of function estimation. In: *Nonlinear Modeling*. Springer 55–85. [https://doi.org/10.1007/978-1-4615-5703-6\\_3](https://doi.org/10.1007/978-1-4615-5703-6_3)
34. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* 770–778. <https://doi.org/10.1109/CVPR.2016.90>
35. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C (2018) Mobilenetv2: Inverted residuals and linear bottlenecks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* 4510–4520. <https://doi.org/10.1109/CVPR.2018.00474>
36. Redmon J, Farhadi A (2017) YOLO9000: better, faster, stronger. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* 7263–7271. <https://doi.org/10.1109/CVPR.2017.690>
37. Chollet F (2017) Xception: Deep learning with depthwise separable convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition, (CVPR)*, Honolulu, HI, pp 1251–1258. <https://doi.org/10.1109/CVPR.2017.195>
38. Zhang X, Zhou X, Lin M, Sun J (2018) Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: *Proceedings of the IEEE conference on computer vision and pattern recognition, Salt Lake City, UT, USA*, pp 6848–6856. <https://doi.org/10.1109/CVPR.2018.00716>
39. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition* 4700–4708. <https://doi.org/10.1109/CVPR.2017.243>
40. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826. <https://doi.org/10.1109/CVPR.2016.308>
41. Tseng TR, Huang HM (2023) Classification of peripheral blood neutrophils using deep learning. *Cytometry A* 103(4):295–303
42. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-cam: Visual explanations from deep networks via gradient-based localization. In: *Proceedings of the IEEE international conference on computer vision*, 618–626. <https://doi.org/10.1109/ICCV.2017.74>
43. Jahmunah V, Ng EYK, Tan R-S, Oh SL, Acharya UR (2022) Explainable detection of myocardial infarction using deep learning models with Grad-CAM technique on ECG signals. *Comput Biol Med* 146:105550
44. Loh HW, Ooi CP, Seoni S, Barua PD, Molinari F, Acharya UR (2022) Application of explainable artificial intelligence for healthcare: a systematic review of the last decade (2011–2022). *Computer Methods and Programs in Biomedicine* 107161. <https://doi.org/10.1016/j.cmpb.2022.107161>

## Authors and Affiliations

Mehmet Erten<sup>1</sup> · Prabal Datta Barua<sup>2</sup> · Sengul Dogan<sup>3</sup>  · Turker Tuncer<sup>3</sup> · Ru-San Tan<sup>4,5</sup> · U. R. Acharya<sup>6</sup>

✉ Sengul Dogan  
sdogan@firat.edu.tr

Mehmet Erten  
mehmeter23@gmail.com

Prabal Datta Barua  
Prabal.Barua@usq.edu.au

Turker Tuncer  
turkertuncer@firat.edu.tr

Ru-San Tan  
tanrsnhc@gmail.com

U. R. Acharya  
Rajendra.Acharya@usq.edu.au

<sup>1</sup> Laboratory of Medical Biochemistry, Fethi Sekin City Hospital, Elazığ, Turkey

<sup>2</sup> School of Business (Information System), University of Southern Queensland, Brisbane, Australia

<sup>3</sup> Department of Digital Forensics Engineering, Technology Faculty, Firat University, Elazığ, Turkey

<sup>4</sup> Department of Cardiology, National Heart Centre Singapore, Singapore, Singapore

<sup>5</sup> Duke-NUS Medical School, Singapore, Singapore

<sup>6</sup> School of Mathematics, Physics and Computing, University of Southern Queensland, Springfield, Australia