Int. J. Appl. Math. Comput. Sci., 2024, Vol. 34, No. 4, 579–595
DOI: 10.61822/amcs-2024-0039

AN EXPERT SYSTEM FOR LEUKOCYTE CLASSIFICATION USING PROBABILISTIC DEEP FEATURE OPTIMIZATION VIA DISTRIBUTION ESTIMATION

MUHAMMAD AWAIS [a], TALLHA AKRAM [b,*], AREEJ ALASIRY [c], MEHREZ MARZOUGUI [c], JONGWOON PARK [d], BYOUNGCHOL CHANG [d]

[a]Department of Electrical and Computer Engineering
COMSATS University Islamabad
Wah Campus, Wah Cantt, 47040, Pakistan

[b]Department of Information Systems
Prince Sattam bin Abdulaziz University
Al-Kharj, 11942, Saudi Arabia
e-mail: t.akram@psau.edu.sa

[c]College of Computer Science
King Khalid University
King Abdullah Road, 61421, Abha, Saudi Arabia

[d]Department of Computer Science
Hanyang University
222, Wangsimni-ro, Seongdong-gu, Seoul, 04763, Republic of Korea

White blood cells (WBCs) are essential for immune and inflammatory responses, and their precise classification is crucial for diagnosing and managing diseases. Although convolutional neural networks (CNNs) are effective for image classification, their high computational demands necessitate feature selection to enhance efficiency and interpretability. This study utilizes transfer learning with EfficientNet-B0 and DenseNet201 to extract features, along with a Bayesian-based feature selection method with a novel optimization mechanism to improve convergence. The reduced feature set is classified using soft voting across multiple classifiers. Tests on benchmark datasets achieved over 99% accuracy with fewer features, surpassing or matching existing methods.

Keywords: convolutional neural networks, optimization, evolutionary algorithms, classification, machine learning.

1. Introduction

White blood cells, also called leukocytes, play a crucial role in maintaining the human immune system. They serve as the first line of defense against foreign agents, including bacteria, viruses, and parasites (Farag and Alagawany, 2018). Their versatile functions include detecting pathogens, orchestrating immune responses, aiding in tissue regeneration, and monitoring for diseases. These functions collectively improve the body's ability to defend itself and recover from injuries and ailments.

Leukocytes are primarily classified into two types of cells: granulocytes and agranulocytes. This classification depends on whether they possess minute membrane-bound vesicles, known as granules, within their cytoplasm. These granules contain several enzymes that play an important role in immune responses. Granulocytes are further divided into three types, namely neutrophils, basophils and eosinophils. Neutrophils are the most prevalent type of white blood cells (WBCs) that respond first to infections. They show strong phagocytic capabilities, enabling them to engulf and digest bacteria and other foreign substances effectively. Eosinophils

*Corresponding author

serve to regulate the immune response and combat specific allergens as well as parasites. Basophils release histamine and other compounds that participate in allergic reactions and the inflammatory process. Agranulocytes are the cells without the presence of granules inside their cytoplasm. They are further divided into lymphocytes and monocytes (Weatherspoon, 2024). Lymphocytes further include B cells and T cells which respectively produce antibodies and attack infected cells. Monocytes have the ability to migrate to tissue and turn into macrophages which are crucial for immunological monitoring and tissue healing (Mathur *et al*., 2013).

An essential step in the detection of various medical disorders is the analysis of WBCs. A traditional method for analyzing white blood cells, referred to as the WBC differential, involves the creation of blood smears. This procedure entails the application and coloring of a tiny layer of blood onto a glass slide, which is subsequently examined under a microscope. It involves an experienced haematologist or a laboratory expert who manually classifies and counts various types of WBCs in a given blood sample. This manual technique of WBC analysis is not only arduous and time-consuming, but it also has inaccuracies that are caused by a number of external factors. Flow cytometry, a sophisticated technique for WBC measurement, employs lasers to distinguish between various cell types. Image flow cytometry is a hybrid approach that combines high resolution image technology with flow cytometry. It can recognize and count individual cells based on their physical characteristics. Mass spectroscopy is another technique which used to identify specific proteins present on the surface of WBCs. More advanced biomedical analysis technologies such as lab-on-a-chip have been recently being investigated (Sakaguchi *et al*., 2022). While promising for accurate analysis, these methods have very high cost and still require specialized human intervention.

In recent years, image assisted technologies have brought a remarkable breakthrough in biomedical diagnostics (Zhang *et al*., 2021; He *et al*., 2020; Huang *et al*., 2022). Scientists have become more interested in using advanced computer vision and machine learning techniques for effective diagnosis of several medical conditions (Sun *et al*., 2023; Zhu, 2024; Lu *et al*., 2023). A breakthrough in this domain is made by convolutional neural networks (CNNs) that have emerged as a valuable tool in a wide variety of biomedical classification tasks. These networks have the capability to identify complex and subtle features in images. Several pre-trained deep CNN models have been introduced at the forefront of research, boasting high Top-1 accuracy across a spectrum of image object classes. GoogleNet (Szegedy *et al*., 2015), DenseNet (Howard *et al*., 2017), EfficientNet (Tan and Le, 2019), ResNet (He *et al*., 2016) and DarkNet (Redmon and Farhadi, 2018) are among the few names to mention. However, these networks require an extensive dataset for training from scratch to achieve an optimal accuracy for a particular task. In circumstances where such a dataset is not readily available, transfer learning emerges as a preferred technique, wherein a CNN model previously developed and pre-trained for one task is adapted for another task.

Transfer learning is typically executed in two manners: fine-tuning and feature extraction. Fine tuning involves adjustment of weights of the pre-trained model in accordance with the task-specific dataset. Conversely, feature extraction uses the pre-trained model to obtain a rich set of features from the input data, which are then employed to train outer classifiers for a particular task. An inherent limitation in employing deep CNNs as feature extractors is the high dimensionality of the resulting feature set, often marked by redundancy. Consequently, this imposes substantial storage and computational burdens on subsequent classifiers (Zhang *et al*., 2024). Furthermore, a classification model may lose accuracy due to overfitting caused by excessive redundant features. Feature selection is an important step in machine learning workflows. It improves the performance, efficiency and interpretability of machine learning models by selecting a manageable number of most discriminating features. The aim of the present research activities is to maximize the classification performance while reducing the computational and memory requirements with the help of smaller and more streamlined feature sets.

2. Literature review

In the published literature, machine learning methods for WBC classification consist of two categories of works. The first category includes the methods which involve extraction of handcrafted features from blood smear images and their classification using traditional classifiers. Some considerable contributions in this categories are the following. Tavakoli *et al*. (2021) proposed a method to segment the cytoplasm and nucleus of blood smear images. Segmentation of the nucleus is performed using Otsu's thresholding applied to the softmap of the image represented in CMYK and HLS color space. The cytoplasm is segmented by obtaining the convex hull of the nucleus. From the segmented nucleus and cytoplasm, multiple color and shape features are obtained and classified with the help of an SVM classifier. A 94.65% accuracy is achieved with the Raabin-WBC dataset. A clustering method is suggested by Sarrafzadeh *et al*. (2014) for separation of nucleus from the cytoplasm of WBCs. The work extracts color, shape, geometric, statistical and moment invariant features from nucleus and cytoplasm of leukocytes and classifies them with an SVM classifier with a one-to-one

architecture. The methodology achieves an accuracy of 93% on a dataset of 10 patients including 149 WBC blood smear images. Ko *et al*. (2011) suggested an image segmentation technique based on mean-shift clustering. This technique extracts a collection of geometrical, textural and shape features and their classification using a random forest classifier. Abdullah and Turan (2019) examined the effectiveness of six different machine learning methods on various statistical and geometric features. The multinomial logistic regression technique showcases improved performance.

In the work by Alruwaili (2021), a sequential linear discriminant analysis method designed to pinpoint distinct traits within blood smear images is used. These traits are then classified utilizing regression values, including partial F-values. Similarly, Su *et al*. (2014) proposed a method to locate a specific region of interest in HSI domain images of WBCs. Distinct colored pixels representing white blood cell nuclei and cytoplasmic granules were identified within the elliptical discriminating zone. Geometrical features, color attributes, and data based on local difference patterns (LDP) are additionally extracted and fed into multiple neural networks for WBC subtype classification. In the work of Sajjad *et al*. (2017), a framework is presented that relies on color K-means clustering in order to segment WBC's nucliei from microscopic blood smear images. Next, the segmented image is transformed to frequency domain to extract several statistical and texture features. Later a multi-class ensemble classifier is used for leukocyte classification at a mobile cloud.

The second category consists of feature extraction and classification techniques that make use of deep neural networks. Here, a few noteworthy contributions are mentioned. Sharma *et al*. (2022) utilized DenseNet121 CNN along with image normalization and augmentation to classify several types of WBCs. The methodology is validated with different batch sizes and Adam optimizer. A maximum accuracy of 98.84% is reported for a Kaggle dataset. Yao *et al*. (2021) proposed a method which increases the model robustness using two-module transfer learning and deformable convolution. The proposed method is then compared with several classical CNN models on the proprietary dataset of the authors as well as BCCD dataset. In the work of Almezhghwi and Serte (2020), the process involves dataset augmentation through generative adversarial networks, followed by the utilization of transfer learning with the DenseNet169 network for WBC classification.

An important aspect of supervised learning based classification methods is the dimensionality reduction of the feature space (Yu *et al*., 2024). This is done by selecting the most dominant set of features. Most existing works on feature selection employ a filter-based approach, which overlooks the relationship between the classification algorithm and the feature subset (Kandukuri *et al*., 2023). In contrast, the wrapper approach takes into account the close relationship between the selected feature subset and accuracy. Population based and probabilistic methods have been extensively applied to solve a wide variety of combinatorial optimization problems (Iqbal *et al*., 2018; Malik *et al*., 2023).

In recent years, there has been lively interest in using iterative and evolutionary methods for feature selection. Several nature inspired evolutionary meta-heuristics have been applied to solve the feature optimization problem (Gupta *et al.*, 2020; Liu *et al.*, 2018; Ahmad *et al.*, 2023a; 2023b; Shahzad *et al.*, 2022). Improved (search space exploration and exploitation as well as algorithm design to avoid local optima are major challenges that must be addressed in order to attain improved accuracy utilizing these approaches.

2.1. Contributions. This work presents a refined methodology for WBC sub-type classification using deep transfer learning. In this work, the following contributions are made;

1. Transfer learning is employed for deep feature extraction using two prominent deep CNN models, i.e., DenseNet201 and EfficientNet-B0. An ensemble of feature vectors extracted from both the networks is then created.

2. A novel feature selection method is proposed that models the process as a binary combinatorial optimization problem with an objective to obtain the best feature set that can minimize the classification error rate. This difficulty is tackled by employing a probabilistic Bayesian approach that utilizes the estimation of distribution algorithm (EDA). In order to reduce the likelihood of reaching a sub-optimal solution, a custom EDA incorporates an efficient thresholding of probabilities of each variable of candidate solutions. The method effectively removes redundant and less powerful features, keeping only the most relevant set of deep features.

3. The set of selected features is subsequently utilized to train multiple outer classifiers with different kernel configurations.

4. The proposed method is validated on multiple standard datasets containing blood smear images of WBCs. Our method showcases superior or comparable performance, accompanied by a substantial reduction in the feature vector size, when compared with several existing approaches.

In the rest of the paper, Section 3 discusses all steps of proposed methodology for WBC sub-type classification. The experimental setup and results are

presented in Section 4 along with quantitative and statistical analysis. In Section 5, the main conclusions drawn from this study are discussed.

3. Materials and methods

3.1. Datasets. This study is performed on three publicly available datasets of WBC images, namely LISC, RABIN-WBC, and CUK.

LISC dataset. The LISC dataset (Rezatofighi and Soltanian-Zadeh, 2011) is composed of blood smear images belonging to 8 healthy subjects and sampled from 100 microscopic slides. The blood smear images were subjected to Gismo-Right staining prior to their analysis using an Axioskope 40 light microscope with a 100× magnification lens. Images were captured with a Sony SSCDC50AP digital camera and saved as BMP files with dimensions 720×560. The dataset consists of 180 training images and 170 testing images.

Raabin-WBC dataset. The Raabin-WBC dataset (Kouzehkanan *et al*., 2022) is a publicly available large dataset of WBCs images with cropped region of interest. This dataset comprises two labeled image sets, Train and Test-A, acquired from 56 normal peripheral blood smears. These sets capture four distinct white blood cell (WBC) types: lymphocytes, monocytes, neutrophils, and eosinophils. To enrich WBC diversity, an additional case of chronic myeloid leukemia (CML) was included for representation of basophils. The total image count is 14,514 across Train and Test-A. All smears were Giemsa-stained. Standard peripheral blood smears were captured using a Samsung Galaxy S5 camera and an Olympus CX18 microscope. Additionally, the CML slide was imaged using an LG G3 camera phone in conjunction with a Zeiss microscope with 100× magnification lens.

CUK dataset. This synthetic dataset was created by Jung *et al*. (2022) from the original dataset collected by the Catholic University of Korea. The dataset consists of 5000 images with equal distribution across all five classes of WBCs.

The distribution of cell classes within each dataset is comprehensively summarized in Table 1, whereas selected samples of images belonging to all WBC classes of these datasets is shown in Fig. 1. Table 2 summarizes key properties of WBCs in blood smear images, including nuclear and cytoplasmic morphology, along with nucleus-to-cytoplasm ratio (NCR).

3.2. Deep learning pipeline for WBC classification. The intricate process of classifying WBCs from images unfolds in a series of computational steps, visualized in Fig. 2. Each step plays a crucial role, as detailed in the following sections.

Table 1. Image distribution of the datasets used in this study.

Dataset		Number of WBC images					Dimensions
		Basophil	Neutrophil	Lymphocyte	Eosinophil	Monocyte	
LISC (Rezatofighi and Soltanian-Zadeh, 2011)	Training set	39	39	41	28	33	$720 \times 560 \times 3$
	Test set	16	17	18	11	15	
RAABIN-WBC (Kouzehkanan *et al*., 2022)	Training set	212	6231	2427	744	561	$575 \times 575 \times 3$
	Augmented training set	3180	6231	7304	6680	6083	
	Test set	89	2660	1034	322	234	
CUK (Jung *et al*., 2022)	Training set	800	800	800	800	800	$128 \times 128 \times 3$
	Test set	200	200	200	200	200	

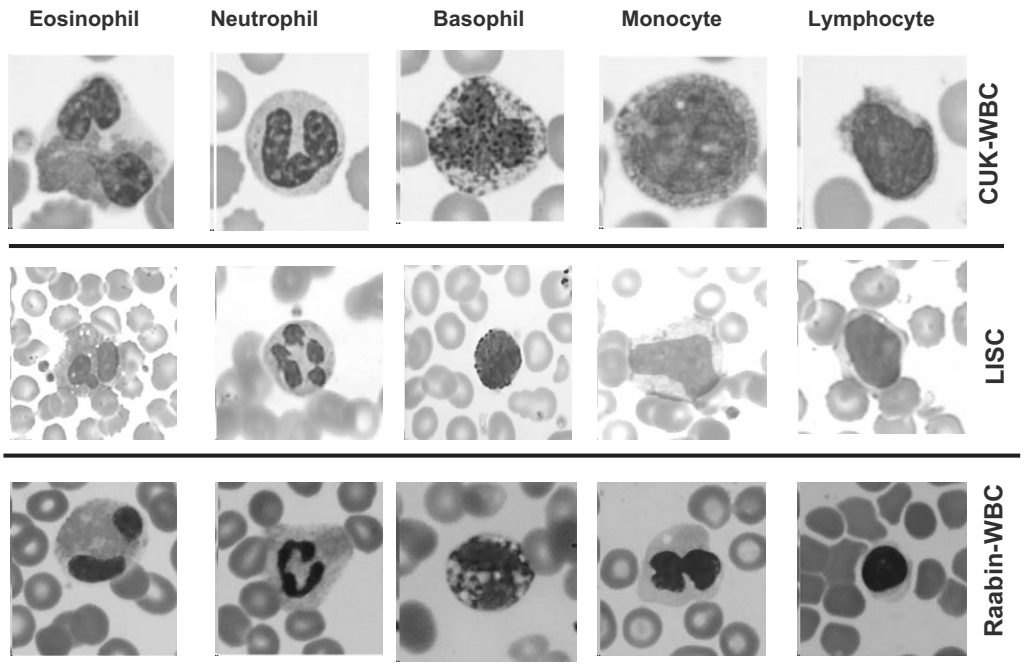

Fig. 1. Several samples of the Raabin-WBC, LISC and CUK datasets used in this study.

3.2.1. Preprocessing. The initial phase performs the contrast enhancement of both training and testing datasets. In most of the classical methods, contrast enhancement is accomplished by applying traditional histogram equalization of gray-scale images. However, when working with microscopic images, solely applying histogram equalization on RGB components may increase the intensity of regions with low intensity values, leading to challenges in discerning pertinent features from noise artifacts. With the goal of addressing this challenge, this work introduces contrast enhancement to the HSI image, taking into account the fact that in HSI, the intensity information is separated from hue and saturation. The partitioning is used to reduce the sensitivity of the RGB channels to changes caused by unfavorable lighting circumstances. This segmentation allows for the separate examination or combination of each channel, thereby permitting the extraction of subtle information relevant to the task of recognition.

The following are the preprocessing steps performed in this work:

(i) transformation from RGB to HSI color space,

(ii) intensity matrix generation,

(iii) equalizing the computed intensity matrix using a histogram,

(iv) substitution of the current intensity values with the revised values,

(v) mapping back from HSI to RGB color space.

3.2.2. Feature extraction. An effective extraction of features is of paramount importance in classification tasks. This study utilizes feature extraction through transfer learning with two well-established deep convolutional neural networks: EfficientNetB0 and DenseNet201, as elaborated below.

EfficientNet. It was proposed by Google AI (Tan and Le, 2019). It distinguishes itself by achieving high accuracy in image recognition tasks while also being efficient in terms of memory and processing power. In contrast to traditional CNNs, which indiscriminately increase depth, width, or resolution in pursuit of higher accuracy, EfficientNet employs a unique compound scaling approach which scales all three dimensions of the network in a balanced manner, thus optimizing efficiency and performance. EfficientNet is family of models (B0-B7), where larger models (B1-B7) achieve higher accuracy at the cost of more parameters. This study adopts the EfficientNetB0 CNN model, characterized by three primary types of building blocks: the stem layer, mobile inverted bottleneck convolution (MBConv) blocks, and squeeze-and-excitation (SE) blocks. The stem layer is the initial stage, which processes the input image. Typically, it employs a combination of convolutional and pooling layers in order to compute basic features of the input image. Table 3 lists the details about the dimensions of the kernel used in convolution operations, as well as the resolution, channels, and number of layers in EfficientNetB0.

MBConv blocks are pivotal in EfficientNet's

Table 2. Some important visual properties of the microscopic images of WBCs.

Cell type	Visual properties	Nucleus	NCR
Neutrophils	Multi-lobed nucleus 3-5 lobes connected by thin threads	Pale pink with fine lilac granules	Low NCR, with the nucleus often appearing smaller than the cytoplasm
Lymphocytes	Round or slightly indented nucleus	Scant, clear to pale blue, with few or no visible granules	High NCR, with the nucleus occupying a large portion of the cell
Monocytes	Horseshoe-shaped or kidney--shaped nucleus	Abundant, pale grey to light blue, with fine azurophilic granules scattered throughout	Variable, but generally higher than neutrophils
Eosinophils	Bilobed nucleus: Connected by a thin thread, resembling spectacles or glasses	Abundant, light orange to red, with large, coarse, eosinophilic granules	Low to moderate, depending on the maturity of the cell
Basophils	Lobulated nucleus: Usually 2–4 lobes, often obscured by large basophilic granules	Scant, clear to pale blue, with large, dark blue to purple basophilic granules masking the nucleus	Very low, with the nucleus often barely visible

Table 3. Kernel size, resolution, channels, and number of layers of EfficientNetB0.

Stage	Operation	Input size	No. of channels	Layers
1	Conv 3×3	224×224	32	1
2	MBConv1, k3 $\times$ 3	112×112	16	1
3	MBConv5, k3 $\times$ 3	112×112	24	2
4	MBConv6, k5 $\times$ 5	56×56	40	2
5	MBConv6, k3 $\times$ 3	28×28	80	3
6	MBConv6, k5 $\times$ 5	14×14	112	3
7	MBConv6, k5 $\times$ 5	14×14	192	4
8	MBConv6, k3 $\times$ 3	7×7	320	4
9	Conv 1×1 & Pooling & FC	7×7	1280	1

architecture, facilitating efficient and potent feature extraction. They leverage an inverted residual connection, diverging from the standard residual connection, which typically involves adding input to output for residual mapping learning. In MBConv, this connection begins by expanding channel numbers via a lightweight layer, followed by depth-wise separable convolutions for feature extraction, and concludes with channel compression to reduce dimensionality. This approach significantly reduces computational complexity while preserving representational capacity.

At the heart of the MBConv block lies the depth-wise separable convolution, comprising two consecutive convolutions: depth-wise convolution, which applies distinct filters to each input channel independently, and point-wise convolution, merging output channels with a 1×1 convolution. Additionally, SE blocks are embedded within MBConv blocks. These blocks take feature maps from the MBConv block as input, where each channel denotes a specific feature. Global average pooling across width and height dimensions condenses spatial information into single values, effectively capturing overall channel activation. This yields a vector representing the importance of the features based on the activations on the map. SE blocks dynamically adjust the channel weights, helping the model distinguish between informative and less relevant features.

The methodology involved utilizing the pretrained EfficientNetB0 model to extract features, which had previously undergone training on the ImageNet dataset (ImageNet, 2024). In order to match the input layer requirements of EfficentNetB0, the input images are first resized to $(224 \times 224 \times 3)$ using the nearest neighbor interpolation method. Subsequently, augmentation of

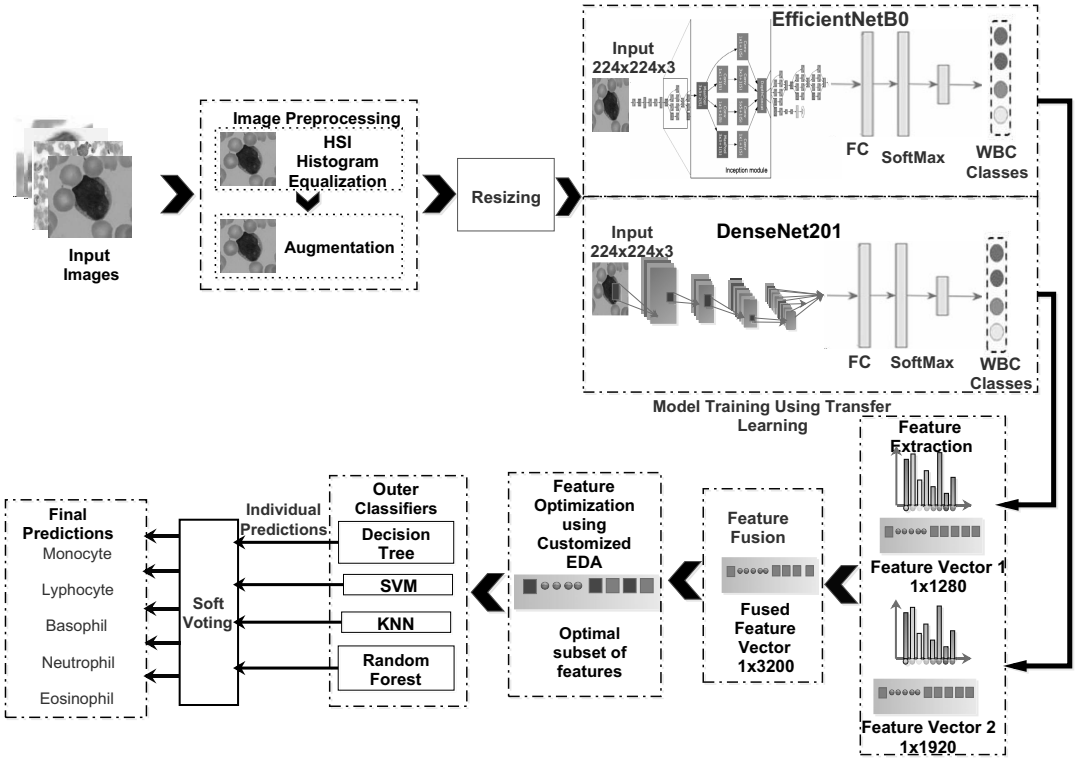

Fig. 2. Proposed system for WBC sub-type classification.

images is applied through random rotation of images in the range from 0 to 360 degrees and rescaling with a random factor in the range from 0.5 to 1. An alternative fully connected (FC) layer replaced the existing FC layer in the network to ensure that the output dimension matches the classes of WBC dataset under examination. New softmax and classification layers are added in the network replacing the existing corresponding layers. A total of 1280 deep features per training and testing image are obtained from the global-average pool layer of the network.

DenseNet201. The DenseNet architecture, comprising 201 layers, is trained using the ImageNet dataset (ImageNet, 2024). This architecture is carefully tailored to address the vanishing gradients problem. Layers are concatenated such that each layer leverages the collective knowledge gained from previous layers. As a result, the resulting network is small yet computationally efficient. The layerwise details of the DenseNet201 network are summarized in Table 4.

For the transfer learning of DenseNet201, a new fully connected layer (FC5) is embedded in the network to replace the "fc1000" layer, corresponding to five WBC subtypes. Additionally, a new softmax layer is incorporated into the classification layer, which does not contain any class identifiers. Before the initiation of training, the dimensions of input images are changed to 224×224 and enhanced using operations such as rotation and inverting. Through the utilization of the global average pooling layer within the trained network, features are extracted, resulting in the generation of a feature vector consisting of 1920 features per image.

3.2.3. Feature fusion. In the next step, a serial concatenation is performed of the extracted feature vectors from both networks. Let A and B represent the feature vectors generated by the two deep CNNs, respectively, with sizes of $1 \times a$ and $1 \times b$. The fused feature vector Z has dimensions of $1 \times (a+b)$ and can be mathematically represented as

$$Z = [A, B]. \tag{1}$$

The above feature fusion step obtains a joint feature vector of size 1×3200 per training/testing image.

3.2.4. Feature selection using the estimation of the distribution algorithm. As a core contribution, this study endeavours to model the feature selection as a binary combinatorial optimization problem and solve it using a meta-heuristic technique. The optimization is aimed at reducing the number of features while maintaining a high level of classification accuracy. We propose a probabilistic Bayesian method for feature

Table 4. Layer-wise details of the DenseNet201 network.

Type of layer	Composition	Repeat factor	Output size
Input	–	–	224×224
Convolution	Conv(7×7), stride 2		112×112
MaxPool	(3×3), stride 2		56×56
Dense Block 1	Conv(1×1) Conv(3×3)	6	56×56
Transition Layer 1	Conv(1×1) Avg Pool(2×2), Stride 2	1	56×56 28×28
Dense Block 2	Conv(1×1) Conv(3×3)	12	28×28
Transition Layer 2	Conv(1×1) Avg Pool(2×2), Stride 2	1	28×28 14×14
Dense Block 3	Conv(1×1) Conv(3×3)	48	14×14
Transition Layer 3	Conv(1×1) Avg Pool(2×2), Stride 2	1	14×14 7×7
Dense Block 4	Conv(1×1) Conv(3×3)	32	7×7
Classification Layer	7×7 Global Avg. Pool 1000D fully Connected, softmax		1×1

selection that employs a customized algorithm for estimating distribution.

The estimation of distribution algorithm (EDAs) belongs to a category of stochastic optimization methods. These methods are based on probabilistic modeling of candidate solutions to a problem and progressively improving them. In contrast to standard evolutionary algorithms like genetic algorithms, which utilize mutation or crossover procedures to generate new populations, EDAs adopt a distinct method. Instead, new individuals in the population are formed by sampling from the probability distribution of the most highly picked candidates from the preceding population.

Furthermore, while other evolutionary algorithms rely on implicit distributions defined by variation variables to generate new candidates, EDAs express the relationships between candidates through an explicit probability distribution. This distribution is typically encoded by a model class such as a multivariate normal distribution, Bayesian network, or another suitable model.

To solve the feature selection combinatorial optimization problem, a binary EDA is applied in this work. Formally, the binary EDA starts with an initial population matrix represented as

$$\mathbb{P} = \begin{pmatrix} \vec{X}^1 \\ \vec{X}^2 \\ \vdots \\ \vec{X}^N \end{pmatrix} = \begin{pmatrix} x_1^1 & x_2^1 & \cdots & x_M^1 \\ x_1^2 & x_2^2 & \cdots & x_M^2 \\ \vdots & \vdots & \vdots & \vdots \\ x_1^N & x_2^N & \cdots & x_M^N \end{pmatrix}, \quad (2)$$

where N and M respectively denote the number of individual solutions of population and the number of variables in each individual. A candidate individual $\vec{X}^i$ is a binary row vector represented as $\vec{X}^i = (x_1, x_2, \ldots, x_M)$. The initial population is randomly generated from a uniform probability distribution.

Subsequently, the best individuals from the population are selected. The number of selected solutions is determined as $N_b = p_b \times N$, where p_b represents the best selection probability, an input parameter to the algorithm. Following selection, the probability distribution $P(\alpha_1, \alpha_2, \ldots, \alpha_M)$ is estimated for each best individual. This probability signifies the likelihood of '1' in each variable of the solution and represented as

$$P_1 = P(\alpha_1, \alpha_2, \ldots, \alpha_M | N_b) \quad (3)$$

Utilizing the probability distribution P_1, the remaining $N - N_b$ individuals are generated and merged with the previously selected best individuals to form a new population for the next iteration.

This work proposes a customized binary EDA for feature selection whose computational steps are outlined in Algorithm 1. In this algorithm, matrices are represented by boldface letters (e.g., $\mathbb{F}$), vectors are represented with an arrow above (e.g., $\vec{X}$) and scalar quantities are represented by normal or special characters (e.g., Γ). A list of symbols used in Algorithm 1 is described in Table 5.

The approach requires multiple key parameters as inputs which include the matrix $\mathbb{F}$ containing fused features, the vector $\vec{L}$ containing the labels of images, the

Table 5. List of symbols and functions of Algorithm 1.

Variable	Details	Variable	Details
$\mathbb{F}$	Matrix containing the fused features	$\vec{L}$	Vector of training labels
G	No. of EDA iterations	N	Size of population
M	No. of variables per individual	P_b	Probability of best selected individuals
P_{ut}, P_{lt}	Upper and lower threshold limits of probability	P_u, P_l	Upper and lower saturated values of probability
N_{th}	No. of iterations after which thresholding is applied	$\mathbb{P}$	Population matrix of size $N \times M$
$\vec{X}^*$	Global best individual vector of size $1 \times M$	Γ^*	Global best fitness
$\vec{X}^p$	Population best individual vector of size $1 \times M$	Γ^p	Population best fitness
N_b	No. of best selected features	$random(1:N, 1:M)$	Returns a matrix of size $N \times M$, of uniformly generated random numbers in [0,1]
$sort(\Gamma)$	Sorts the vector Γ in ascending order and returns the sorted values and their indexes	$\vec{I}$	Vector of sorted indexes
$\vec{\Gamma_s}$	Vector of sorted values	$Rep(N, \vec{P})$	Repeats N copies of $\vec{P}$ along the rows

total count of algorithm iterations (generations) denoted as G, the population size represented by N, and the size of each candidate solution vector denoted as M. The population size, N, is determined by the number of training or testing images utilized in the feature extraction step. Additionally, M represents the number of fused features, which in this case is equal to 3200. The initialization of main parameters is performed in Phase 1 of the algorithm including the probability of best selection P_b, upper and lower probability thresholds P_{ut}, P_{lt}, upper and lower probability values P_u, P_l, the number of iterations after which thresholding is applied N_th and several intermediate variables.

In Steps 4–10, an initial binary population is randomly generated in $\mathbb{P}$ by sampling a uniform distribution. Phase 3 includes the main execution of the algorithm that repeats for G iterations. For every iteration, the fitness value of each member of a population is calculated using the $CostFunction$ routine, Steps 15–18. In Step 20, the function routine $sort$ is called, which returns the fitness values sorted in ascending order and their corresponding indexes. These sorted values and their indexes are stored in column vectors $\vec{\Gamma_s}$ and $\vec{I}$ respectively. In Steps 21 and 22, the best population fitness Γ_p and individual $\vec{X_p}$ are obtained. In Steps 24–27, the best global fitness Γ^* is compared with the best population fitness Γ_p; if Γ^* is greater than Γ_p, then the best global individual Γ^* is replaced by the best population individual Γ_p. In Steps 29–32, the first N_b best individuals of population are retained in $\mathbb{P}$, whereas the remaining individuals are discarded. In Steps 33–35, the joint probability distribution of the best individuals is obtained. Each point $l \in \{1, \ldots, M\}$ of the distribution is computed by adding the values of the l-th column of $\mathbb{P}$ and dividing the sum by N_b. The probability distribution vector is stored in $\vec{P_1}$.

EDA with thresholding. In this work, we proposed an efficient thresholding of probabilities to avoid the local-optima convergence of the algorithm. The $Threshold$ routine of Algorithm 1 receives as inputs the joint probability distribution vector $\vec{P_1}$, the global best fitness, the number of iterations after which thresholding is applied, i.e., N_{th}, the upper and lower threshold limits of probability, i.e., P_{ut} and P_{lt} and maximum and minimum probability values P_u and P_l. In Steps 73–88, the best global fitness Γ^* is monitored in each iteration; if it remains unchanged for a predefined number of iterations N_{th}, then all values of probability vector $\vec{P_1}$ are clipped to P_u if they are greater than P_{ut}. In a similar manner, all values of $\vec{P_1}$, which are less than P_{lt} are made equal to P_l. The threshold values as well as the maximum and minimum probability values are experimentally obtained through several runs of the feature extraction pipeline. After updating the probabilities, in Steps 40–49, the new binary population matrix for next iteration is generated. The function Rep in Step 40 generates N copies of the updated probability vector $\vec{P_1}$ cascaded vertically. Finally, in Steps 51 and 52, the nonzero indexes of the best global individuals are obtained, which correspond to the features selected from $\mathbb{F}$ and returned by the algorithm.

Algorithm 1. EDA for deep feature selection.

1: **Inputs:** $\mathbb{F}, \vec{L}, G, N, M$
2: **Phase 1: Parameter Initialization**
$P_b \leftarrow 0.4,\ P_{ut} \leftarrow 0.85,\ P_{lt} \leftarrow 0.15\ P_u \leftarrow 0.75,\ P_l \leftarrow 0.25$
$N_{th} \leftarrow 5,\ i \leftarrow 1,\ \Gamma_{temp} \leftarrow 0,\ N_c \leftarrow 0$
$\mathbb{P}(1:N, 1:M) \leftarrow 1$
$\Gamma^* \leftarrow inf, \vec{X^*}(1:M) \leftarrow 0,$
$\Gamma_p \leftarrow inf, \vec{X_p}(1:M) \leftarrow 0.$
$N_b \leftarrow \lceil N \times P_b \rceil$
3: **Phase 2: Generate Initial Random Population**
4: $\mathbb{R}(1:N, 1:M) \leftarrow random(1:N, 1:M)$
5: **for** $x = 1:N$ **do**
6: **for** $y = 1:M$ **do**
7: **if** $\mathbb{R}(x,y) \leq 0.5$ **then**
8: $\mathbb{P}(x,y) \leftarrow 1$
9: **else**
10: $\mathbb{P}(x,y) \leftarrow 0$
11: **end if**
12: **end for**
13: **end for**
14: **Phase 3: Execution**
15: **while** $i < G$ **do**
16: **Calculate Fitness of all solutions**
17: **for** $j = 1:N$ **do**
18: $\vec{X} \leftarrow \mathbb{P}(j, 1:M)$
19: $\Gamma(j) \leftarrow CostFunction(\mathbb{F}, \vec{L}, \vec{X})$
20: **end for**
21: obtain the population best solution
22: $[\vec{I}, \vec{\Gamma_s}] \leftarrow sort(\Gamma)$
23: $\Gamma_p \leftarrow \vec{\Gamma_s}(1)$
24: $\vec{X_p} \leftarrow \mathbb{P}(\vec{I}(1), 1:M)$
25: **update global best solution**
26: **if** $\Gamma^* > \Gamma_p$ **then**
27: $\Gamma^* \leftarrow \Gamma_p$
28: $\vec{X^*} \leftarrow \vec{X_p}$
29: **end if**
30: **Compute the joint probability distribution of N_b best solutions**
31: $\vec{I_b} \leftarrow \vec{I}(1:N_b)$
32: $\mathbb{P}(1:N_b, 1:M) \leftarrow \mathbb{P}(\vec{I_b}, 1:M)$
33: $\mathbb{P}(N_b+1:N, 1:M) \leftarrow 0$
34: $\vec{P_1}(1, 1:M) \leftarrow 0$
35: **for** $l = 1:M$ **do**
36: $\vec{P_1}(1,l) \leftarrow \frac{\sum \mathbb{P}(1:N_b, l)}{N_b}$
37: **end for**
38: **apply the thresholding to update probabilities**
39: $\Gamma_{temp} \leftarrow \Gamma_p$
40: $\vec{P_1} \leftarrow Threshold(\vec{P_1}, \Gamma_{temp}, \Gamma^*, N_c, N_{th}, P_{ut}, P_{lt}, P_u, p_l)$
41: **generate new population for next iteration**
42: $\mathbb{P}_2 \leftarrow Rep(N, \vec{P_1})$
43: $\mathbb{R} \leftarrow random(1:N, 1:M)$
44: **for** $x = 1:N$ **do**
45: **for** $y = 1:M$ **do**
46: **if** $\mathbb{R}(x,y) < \mathbb{P}_2(x,y)$ **then**
47: $\mathbb{P}(x,y) \leftarrow 1$
48: **else**
49: $\mathbb{P}(x,y) \leftarrow 0$
50: **end if**
51: **end for**
52: **end for**
53: **end while**
54: **extract Index of Best Features**
55: $\vec{I} \leftarrow 1:M$
56: $\vec{S} \leftarrow \vec{I}(\vec{X^*} == 1))$
57: **OUTPUT:** $\vec{S}$

58: **Function:** $CostFunction$
59: **Inputs:** $\mathbb{F}, \vec{L}, \vec{X}$
60: **Parameters:** $\alpha_1 = 0.99, \alpha_2 = 0.01, k = 5, h_o = 0.2$
61: **if** (sum($\vec{X} == 1$) $== 0$) **then**
62: $cost = \inf$
63: **else**
64: $\mathbb{F}_2 \leftarrow \mathbb{F}(:, (\vec{F} == 1))$
65: $\mathbb{F}_{train}, \vec{L_{train}}, \mathbb{F}_{test}, \vec{L_{test}} \leftarrow partition(\mathbb{F}_2, \vec{L}, h_o)$
66: $Model \leftarrow trainKNN(\mathbb{F}_{train}, \vec{L_{train}}, k)$
67: $\vec{L_{pred}} \leftarrow predict(Model, \mathbb{F}_{test})$
68: $acc \leftarrow sum(\vec{L_{pred}} == \vec{L_{test}})/length(\vec{L_{test}})$
69: $err \leftarrow 1 - acc$
70: $f_s \leftarrow sum(\vec{X} == 1)$
71: $f_t \leftarrow length(\vec{X})$
72: $cost \leftarrow \alpha_1 \times err + \alpha_2 \times (\frac{f_s}{f_t})$
73: **end if**
74: **Return:** $cost$

75: **Function:** $Threshold$
76: **Inputs:** $\vec{P_1}, \Gamma_{temp}, \Gamma^*, N_c, N_{th}, P_{ut}, P_{lt}, P_u, p_l$
77: **if** $\vec{\Gamma_{temp}} \neq \vec{\Gamma^*}$ **then**
78: $\vec{\Gamma_{temp}} = \vec{\Gamma^*}$
79: $N_c \leftarrow 0$
80: **else**
81: $N_c \leftarrow N_c + 1$
82: **end if**
83: **if** $N_c > N_{th}$ **then**
84: $N_c \leftarrow 0$
85: **for** m=1:M **do**
86: **if** $\vec{P_1}(m) \geq P_{ut}$ **then**
87: $\vec{P_1}(m) \leftarrow P_u$
88: **else if** $\vec{P_1}(m) < P_{lt}$ **then**
89: $\vec{P_1}(m) \leftarrow P_l$
90: **end if**
91: **end for**
92: **end if**
93: **Return:** $\vec{P_1}$

The $CostFunction$ routine, for the whole population, calculates the fitness value. The individual binary solution vector $\vec{X}$ is passed as an input to this function along with matrix $\mathbb{F}$ and vector $\vec{L}$. In Step 60, the indexes of nonzero elements of $\vec{x}$ are used to extract the corresponding features from $\mathbb{F}$. The extracted features are kept in the field $\mathbb{F}_2$. In Step 61, matrix $\mathbb{F}_2$ along with the vector $\vec{L}$ are partitioned into two parts, i.e., training and testing with a ratio h_0. Next, the training features are applied to train a KNN classifier with $k = 5$ neighbors. After training, the testing feature set $\mathbb{F}_{train}$ is applied to KNN and predicted labels are obtained in $\vec{L}_{pred}$. The fitness is calculated as

$$a_c = \frac{n_{pred}}{n_{test}} \times 100(\%), \tag{4}$$

$$\Gamma = \alpha_1 \times (1 - a_c) + \alpha_2 \times \frac{q_s}{q_t}. \tag{5}$$

The variables n_{pred} and n_{test} represent the total number of correctly predicted and tested samples, respectively, by the KNN classifier. The weight coefficients α_1 and α_2 satisfy the equation $\alpha_1 + \alpha_2 = 1$; q_s represents the number of features selected out of q_t features of $\vec{\mathbb{X}}$.

3.2.5. Classification. The selected features of the fused feature matrix $\mathbb{F}$, identified by the proposed EDA in Algorithm 1, are subsequently employed for supervised learning with various outer classifiers. In this study, several classifiers are utilized, including random forest classifiers, K-nearest neighbor (K-NN), support vector machines (SVMs), and decision trees (DTs), each with various kernel settings. The models provided individual predictions, and their performance was evaluated using test data.

3.2.6. Soft voting. In this work, the individual class-wise predictions of all classifiers are combined using a soft voting technique to obtain the final predictions. Although each classifier might have different levels of confidence in their predictions, soft voting takes into account the collective knowledge of all classifiers to make a more informed decision. In the proposed soft-voting, a simple averaging of probabilities for each class across all classifiers is calculated. Finally, we choose the class with the highest average probability as the final prediction.

4. Performance results

4.1. Experimental setup. The WBC classification pipeline illustrated in Fig. 2 was developed using Matlab 2023, operating on a Core i7 CPU with 8 GB of RAM and a 64-bit operating system. The proposed pipelined was trained and validated separately for the three datasets mentioned in Section 3.1. To mitigate the issue of class imbalance, following the contrast enhancement step, the preprocessed images from selected datasets underwent an augmentation phase, involving random rotation within the range from 0 to 360 degrees and resizing within the interval $[0.5, 1]$. Table 6 shows the class distribution after performing the augmentation on the selected datasets.

Following augmentation, the datasets were randomly partitioned into training and validation sets with a ratio of 70:30. Subsequently, all images were resized to dimensions of $224 \times 224 \times 3$ to meet the input layer specifications of EfficientNetB0 and DenseNet201. The training part was then used for transfer learning of both deep CNNs. The parameters setting for training of networks is shown in Table 7. We investigated different parameter settings in numerous training iterations and selected the one that yielded the best training performance. The training segments of each dataset are individually utilized for transfer learning with deep CNNs.

After performing transfer learning of networks using training datasets, the fused feature vector of size 3200 is obtained and subjected to customized EDA for feature selection.

Figure 3 displays the error rate performance of three algorithms: the classical genetic algorithm (GA), the standard estimation of distribution algorithm (EDA), and our proposed EDA with thresholding. The plotted error rate for each algorithm is an average of multiple Monte Carlo iterations of the algorithm on the same fused feature vector. During each Monte Carlo iteration, the algorithms run for $G = 100$ iterations (generations). The plots demonstrate that our proposed EDA with thresholding not only attains a lower error rate, but also exhibits rapid convergence, reaching the minimum error rate with fewer iterations. In Fig. 4, the error rates achieved by the customized EDA and classical GA are plotted as a function of population size N with a fixed value of $G = 50$ iterations. The plots indicate that the error rates of both algorithms improve as the population size increases from 5 to 10. However, beyond this range, the error rates of both algorithms remain relatively stable. Figures 3 and 4 collectively reveal that for all values of N and G, the proposed EDA with efficient thresholding scheme shows better convergence and feature optimization in comparison with classical population based meta-heuristics.

4.2. Classification results. Table 8 displays the WBC classification results obtained by various individual classifiers employing different kernel settings on the LISC dataset. The table quantifies the classification performance using a number of key metrics which include accuracy, precision, recall, F1 score, and sensitivity. The reported classification results are achieved through the proposed EDA, which performs approximately 90% feature reduction, selecting a reduced set of 326 optimal features per image from an initial set of 3200 fused features. Although the original images of LISC are relatively blurred and exhibit a high similarity between the WBC regions of interest and other blood entities, as shown in Figure 1, all chosen classifiers demonstrate a notable classification performance. Among the classifiers evaluated, the random forest classifier demonstrates comparatively superior performance, attaining a peak accuracy of 98.8%, a precision of 0.982, a recall of 0.991, an F1 score of 0.941, and a sensitivity of 0.996.

Table 9 presents the classification results obtained by the proposed pipeline using the CUK-WBC dataset. The table illustrates that both the random forest and KNN coarse classifiers perform well, achieving accuracy rates exceeding 99%, along with high values for other performance metrics. Similarly, in Table 10, the classification performance results are showcased for the Raabin-WBC dataset. Among these, the KNN classifier with a coarse kernel stands out, achieving an accuracy of 99.3%, a precision of 0.99, a recall of 0.992, an F1 score of 0.996, and a sensitivity of 0.995.

In Table 11, the comprehensive outcomes of the proposed WBC-classification pipeline are showcased. These results are derived through soft voting, wherein

Table 6. Class-wise distribution of images in augmented LISC, RAABIN-WBC and CUK datasets used in this study.

Dataset		Number of WBC images				
		Basophil	Neutrophil	Lymphocyte	Eosinophil	Monocyte
LISC	Original training set	39	39	41	28	33
	Augmented training set	361	361	359	372	367
RAABIN-WBC	Original training set	212	6231	2427	744	561
	Augmented training set	3180	6231	7304	6680	6083
CUK	Original training set	800	800	800	800	800
	Augmented training set	800	800	800	800	800

Table 7. Parameter settings for training of EfficientNetB0 and DenseNet201.

Parameter	Value	Parameter	Value
Kernel	sdgm	Learning rate	1×10^{-4}
Environment	Auto	MiniBatch size	20
MaxEpochs	5	Validation frequency	30
Dropout rate	0.1	Stride size	1

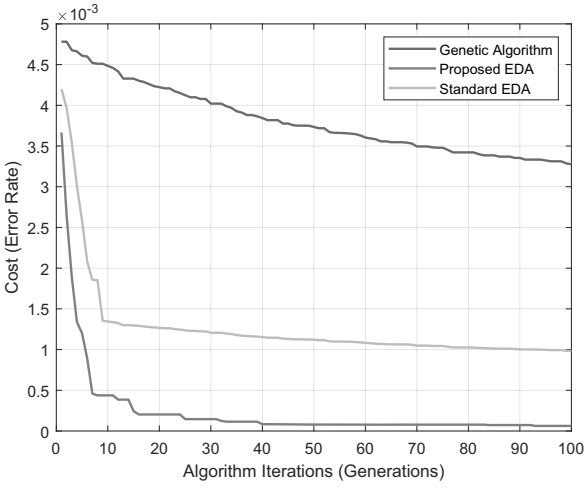

Fig. 3. Error rate comparison of feature selection using a classical genetic algorithm, standard EDA and proposed EDA with thresholding.

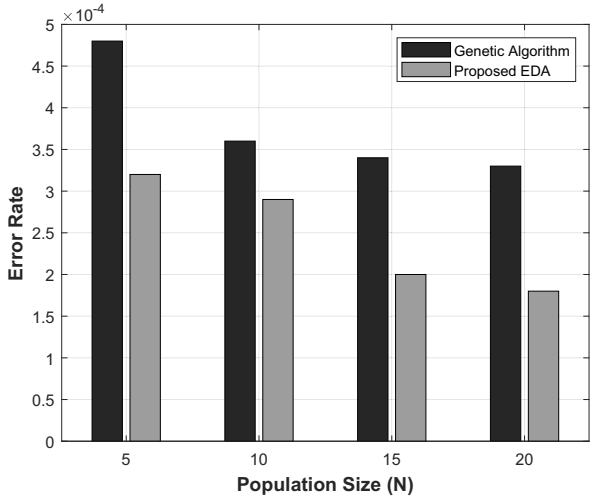

Fig. 4. Error rate of the genetic algorithm and proposed EDA with $G = 50$ iterations and different population sizes N.

the probability distributions of all classifiers are averaged. This approach allows for the incorporation of diverse confidence levels from individual classifiers. The results clearly highlight that soft voting leverages the collective knowledge of all classifiers, leading to more informed decisions. Notably, the proposed pipeline employing soft voting achieves outstanding classification performance with the Raabin-WBC dataset, attaining a maximum accuracy of 99.6%, a precision of 0.992, a recall of 0.995, an F1-score of 0.997, and a sensitivity of 0.994. Remarkable performance is also observed across other datasets.

Table 12 displays a comparison between the accuracy of our proposed method and that of previous studies on WBC classification employing deep learning networks and comparable datasets. Our method demonstrates comparable or superior accuracy performance compared with prior works, despite utilizing fewer features. This reinforces the viability and effectiveness of our approach.

4.3. Statistical analysis. The statistical analysis of performance results of our classification system is performed using the analysis of variance (ANOVA) (Turner and Thayer, 2001). ANOVA compares the means of several distributions to determine if there are any statistically significant differences between them. In

Table 8. Performance results of the proposed WBC classification system on the LISC dataset.

Classifier	No. of selected features	Accuracy %	Precision	Recall	F1 score	Sensitivity
Decision tree (medium)		78.6	0.83	0.826	0.82	0.84
KNN (coarse)		98.7	0.97	0.94	0.94	0.942
KNN (cosine)		98.6	0.99	0.995	0.996	0.994
Random forest	326	98.8	0.982	0.991	0.996	0.996
SVM (Gaussian)		96.2	0.972	0.94	0.941	0.98
SVM (regression)		86.2	0.869	0.876	0.872	0.867

Table 9. Performance results of the proposed WBC classification system on the CUK dataset.

Classifier	No. of selected features	Accuracy %	Precision	Recall	F1 score	Sensitivity
Decision tree (medium)		80.6	0.851	0.842	0.852	0.85
KNN (coarse)		99.1	0.987	0.98	0.986	0.992
KNN (cosine)		98.9	0.98	0.985	0.986	0.984
Random forest	326	99.2	0.992	0.981	0.996	0.986
SVM (Gaussian)		97.8	0.976	0.964	0.981	0.981
SVM (regression)		94.2	0.969	0.976	0.942	0.947

our framework, we performed statistical analysis on the classification accuracy of SVM (Gaussian), KNN (coarse) and random forest classifiers on all the selected datasets.

Before performing ANOVA, we conducted the Shapiro–Wilk test (Shapiro and Wilk, 1965) to ensure the assumption of normality, and Bartlett's test (Bartlett, 1937) to verify the homogeneity of variances. We used a significance level of 1% for these tests, which corresponds to $\alpha = 0.01$. The means of the accuracy values for the SVM (Gaussian), KNN (coarse), and random forest classifiers are denoted as μ_1, μ_2, and μ_3, respectively.

The null hypothesis assumes that the data are normally distributed if the Shapiro–Wilk p-values are greater than α; otherwise, the alternative hypothesis is accepted. The computed Shapiro–Wilk p-values for the SVM, KNN, and random forest classifiers are $p_1 = 0.684$, $p_2 = 0.723$, and $p_3 = 0.7123$, respectively. Additionally, Bartlett's test yielded a chi-squared probability $p_{ch} = 0.823$. These p-values indicate that the values of accuracy are normally distributed having homogeneous variances.

Table 13 presents the key metrics results of ANOVA which include degrees of freedom (df), sum of squared deviations (SS), mean squared error (MSE), F-statistic and the obtained p-value of 0.685, which is greater than α. Therefore, we conclude that there are no significant differences between the means of the three classifiers, suggesting that the classification accuracy values of these classifiers are statistically identical.

As an example, Fig. 5 shows the accuracy confidence interval plots for three selected classifiers for the LISC dataset. The lighter bars represent the average accuracy, while the black bars indicate the 99% confidence limits for each classifier. Additionally, the medium bars illustrate the lower and upper quantile points derived from the statistical tests mentioned earlier. Similar plots were obtained for other datasets but not shown here due to space reasons.

The data indicate that the SVM Gaussian and random forest classifiers provide superior average accuracy, along with broader confidence interval sizes, in comparison with the coarse KNN classifier. In addition, the quartile points for each classifier are inside their corresponding confidence ranges. The elevated p-values linked to these quantile points lead to the acceptance of the null hypothesis, suggesting that there are no substantial disparities in the accuracy distributions of the classifiers. This indicates that although there may be changes in the average accuracy, these variations do not have a substantial statistical impact, confirming the strength and dependability of the classifiers' performance under the conditions that were tested.

5. Conclusions

White blood cell (WBC) identification and classification are indispensable for the diagnosis of numerous blood disorders, particularly leukemia. In the pursuit of accuracy, modern methodologies for leukemia detection that utilize transfer learning on deep neural networks heavily depend on feature extraction to surmount the obstacle posed by high-dimensional data.

This study details the revised pipeline for WBC classification. Initially, deep transfer learning serves as the primary feature extractor. Following this, a highly effective probabilistic deep feature selection algorithm is applied. This algorithm incorporates an

Table 10. Performance results of the proposed WBC classification system on the Raabin-WBC dataset.

Classifier	No. of selected features	Accuracy %	Precision	Recall	F1 score	Sensitivity
Decision tree (medium)		90.3	0.901	0.922	0.914	0.905
KNN (coarse)		99.3	0.990	0.992	0.996	0.995
KNN (cosine)		99.1	0.99	0.99	0.992	0.989
Random forest	326	99.1	0.992	0.991	0.995	0.996
SVM (Gaussian)		98.8	0.986	0.984	0.989	0.991
SVM (regression)		97.2	0.989	0.979	0.982	0.987

Table 11. Overall results of the proposed WBC-classification pipeline after performing soft-voting of individual predictions of classifiers.

Data set	Accuracy	Precision	Recall	F1 score	Sensitivity
CUK	98.9	0.986	0.991	0.984	0.98
LISC	99.4	0.991	0.993	0.994	0.991
Raabin-WBC	99.6	0.992	0.995	0.997	0.994

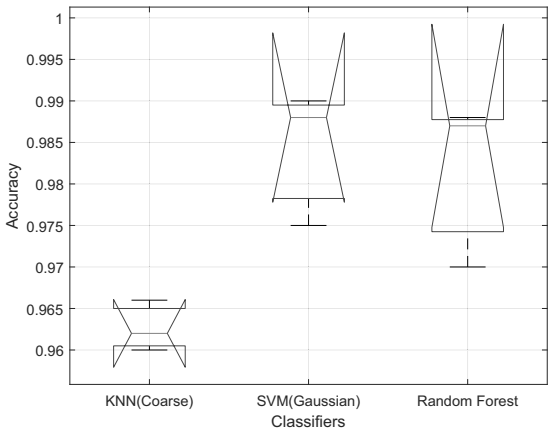

Fig. 5. Confidence interval for selected classifiers on the LISC dataset.

advanced thresholding mechanism within an estimation of distribution framework, successfully avoiding local optima and significantly reducing the size of the feature vector while maintaining classification accuracy.

Comparative analysis shows that the proposed algorithm performs better in terms of the error rate in feature selection compared with traditional population-based metaheuristics. Additionally, the proposed pipeline for WBC classification can be integrated into advanced clinical-grade systems, such as automated image flow cytometry. This integration provides a valuable tool for both scientific research and medical diagnostics.

Acknowledgment

The authors extend their appreciation to the Deanship of Research and Graduate Studies at King Khalid University for funding this work through the Large Research Project under the grant no. RGP2/283/45. This work was also supported by the Human Resources Program in Energy Technology of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and was granted financial resources from the Ministry of Trade, Industry, and Energy, Republic of Korea (no. 20204010600090).

References

Abdullah, E. and Turan, M.K. (2019). Classifying white blood cells using machine learning algorithms, *International Journal of Engineering Research and Development* **11**(1): 141–152.

Ahmad, R., Awais, M., Kausar, N. and Akram, T. (2023a). White blood cells classification using entropy-controlled deep features optimization, *Diagnostics* **13**(3): 352–369.

Ahmad, R., Awais, M., Kausar, N., Tariq, U., Cha, J.-H. and Balili, J. (2023b). Leukocytes classification for leukemia detection using quantum inspired deep feature selection, *Cancers* **15**(9): 2507–2524.

Almezhghwi, K. and Serte, S. (2020). Improved classification of white blood cells with the generative adversarial network and deep convolutional neural network, *Computational Intelligence and Neuroscience* **2020**(01): 6490479.

Alruwaili, M. (2021). An intelligent medical imaging approach for various blood structure classifications, *Complexity* **2021**(01): 5573300.

Bartlett, M.S. (1937). Properties of sufficiency and statistical tests, *Proceedings of the Royal Society of London A: Mathematical and Physical Sciences* **160**(901): 268–282.

Farag, M.R. and Alagawany, M. (2018). Erythrocytes as a biological model for screening of xenobiotics toxicity, *Chemico-Biological Interactions* **279**: 73–83.

Gupta, D., Agrawal, U., Arora, J. and Khanna, A. (2020). Bat-inspired algorithm for feature selection and white blood cell classification, *in* X. Yang (Ed), *Nature-Inspired*

Table 12. Comparative analysis of the performance between the proposed method and existing approaches (×: Not done, NA: Information not available.)

Deep framework	Feature size	Classifier	Selection algorithm	Accuracy %
GoogleNet, ResNet-50	755	Quadratic discriminant analysis	Maximal information coefficient, ridge regression model	97.95
PatternNet fused ensemble of CNNs	NA	CNN	×	99.90
ResNet and inception	NA	ResNet and inception	Hierarchical approach	99.84
AlexNet	1000	CNN	×	98.4
EfficientNetB0 and DenseNet201(proposed)	**326**	**Soft voting of DT, KNN, RF, SVM**	**Customized EDA**	**99.6**

Table 13. ANOVA statistical results of the overall accuracy of classifiers.

V-Source	SS	df	MSE	F-statistics	p-Value
Between	6.815×10^{-5}	2	2.6258×10^{-5}	0.34	0.685
Within	6.2136×10^{-4}	6	8.8614×10^{-5}	–	–
Total	6.1259×10^{-4}	8	–	–	–

Computation and Swarm Intelligence, Academic Press, Cambridge, pp. 179–197.

He, B., Lu, Q., Lang, J., Yu, H., Peng, C., Bing, P., Li, S., Zhou, Q., Liang, Y. and Tian, G. (2020). A new method for CTC images recognition based on machine learning, *Frontiers in Bioengineering and Biotechnology* **8**: 897–907.

He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep residual learning for image recognition, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVR), Las Vegas, USA*, pp. 770–778.

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. and Adam, H. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications, *arXiv:* 1704.04861.

Huang, H., Wu, N., Liang, Y., Peng, X. and Shu, J. (2022). Slnl: A novel method for gene selection and phenotype classification, *International Journal of Intelligent Systems* **37**(9): 6283–6304.

ImageNet (2024). ImageNet project image database, `http://www.image-net.org`.

Iqbal, M., Naeem, M., Ahmed, A., Awais, M., Anpalagan, A. and Ahmad, A. (2018). Swarm intelligence based resource management for cooperative cognitive radio network in smart hospitals, *Wireless Personal Communications* **98**: 571–592.

Jung, C., Abuhamad, M., Mohaisen, D., Han, K. and Nyang, D. (2022). WBC image classification and generative models based on convolutional neural network, *BMC Medical Imaging* **22**(1): 1–16.

Kandukuri, U.R., Prakash, A.J., Patro, K.K., Neelapu, B.C., Tadeusiewicz, R. and Pławiak, P. (2023). Constant Q-transform-based deep learning architecture for detection of obstructive sleep apnea, *International Journal of Applied Mathematics and Computer Science* **33**(3): 493–506, DOI: 10.34768/amcs-2023-0036.

Ko, B., Gim, J. and Nam, J. (2011). Cell image classification based on ensemble features and random forest, *Electronics Letters* **47**(11): 638–639.

Kouzehkanan, Z.M., Saghari, S., Tavakoli, S., Rostami, P., Abaszadeh, M., Mirzadeh, F., Satlsar, E.S., Gheidishahran, M., Gorgi, F., Mohammadi, S. and Hosseini, R. (2022). A large dataset of white blood cells containing cell locations and types, along with segmented nuclei and cytoplasm, *Scientific Reports* **12**(1): 1123–1137.

Liu, J., Lin, Y., Li, Y., Weng, W. and Wu, S. (2018). Online multi-label streaming feature selection based on neighborhood rough set, *Pattern Recognition* **84**: 273–287.

Lu, S., Liu, S., Hou, P., Yang, B., Liu, M., Yin, L. and Zheng, W. (2023). Soft tissue feature tracking based on deep matching network, *Computer Modeling in Engineering and Sciences* **136**(1): 363–379.

Malik, S., Akram, T., Awais, M., Khan, M.A., Hadjouni, M., Elmannai, H., Alasiry, A., Marzougui, M. and Tariq, U. (2023). An improved skin lesion boundary estimation for enhanced-intensity images using hybrid metaheuristics, *Diagnostics* **13**(7): 1285.

Mathur, A., Tripathi, A.S. and Kuse, M. (2013). Scalable system for classification of white blood cells from Leishman stained blood stain images, *Journal of Pathology Informatics* **4**(2): 15–20.

Redmon, J. and Farhadi, A. (2018). YOLOv3: An incremental improvement, *arXiv:* 1804.02767.

Rezatofighi, S.H. and Soltanian-Zadeh, H. (2011). Automatic recognition of five types of white blood cells in peripheral

blood, *Computerized Medical Imaging and Graphics* **35**(4): 333–343.

Sajjad, M., Khan, S., Jan, Z., Muhammad, K., Moon, H., Kwak, J.T., Rho, S., Baik, S.W. and Mehmood, I. (2017). Leukocytes classification and segmentation in microscopic blood smear: A resource-aware healthcare service in smart cities, *IEEE Access* **5**: 3475–3489.

Sakaguchi, K., Akimoto, K., Takaira, M., Tanaka, R.-i., Shimizu, T. and Umezu, S. (2022). Cell-based microfluidic device utilizing cell sheet technology, *Cyborg and Bionic Systems* **2022**: 9758187.

Sarrafzadeh, O., Rabbani, H., Talebi, A. and Banaem, H.U. (2014). Selection of the best features for leukocytes classification in blood smear microscopic images, *Medical Imaging 2014: Digital Pathology, San Diego, USA*, pp. 159–166.

Shahzad, A., Raza, M., Shah, J.H., Sharif, M. and Nayak, R.S. (2022). Categorizing white blood cells by utilizing deep features of proposed 4B-additionNet-based CNN network with ant colony optimization, *Complex & Intelligent Systems* **8**: 3143–3159.

Shapiro, S.S. and Wilk, M.B. (1965). An analysis of variance test for normality (complete samples), *Biometrika* **52**(3–4): 591–611.

Sharma, S., Gupta, S., Gupta, D., Juneja, S., Gupta, P., Dhiman, G. and Kautish, S. (2022). Deep learning model for the automatic classification of white blood cells, *Computational Intelligence and Neuroscience* **2022**: 7384131.

Su, M.-C., Cheng, C.-Y. and Wang, P.-C. (2014). A neural-network-based approach to white blood cell classification, *The Scientific World Journal* **2014**: 796371.

Sun, T., Lv, J., Zhao, X., Li, W., Zhang, Z. and Nie, L. (2023). In vivo liver function reserve assessments in alcoholic liver disease by scalable photoacoustic imaging, *Photoacoustics* **34**: 100569.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015). Going deeper with convolutions, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, USA*, pp. 1–9.

Tan, M. and Le, Q. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks, *Proceedings of the 36th International Conference on Machine Learning, Long Beach, USA*, pp. 6105–6114.

Tavakoli, S., Ghaffari, A., Kouzehkanan, Z.M. and Hosseini, R. (2021). New segmentation and feature extraction algorithm for classification of white blood cells in peripheral smear images, *Scientific Reports* **11**(1): 19428.

Turner, J.R. and Thayer, J. (2001). *Introduction to Analysis of Variance: Design, Analysis & Interpretation: Design, Analyis & Interpretation*, Sage, Thousand Oakes.

Weatherspoon, D. (2024). What to know about white blood cells, *Medical News Today*, `https://www.medicalnewstoday.com/articles/327446`.

Yao, X., Sun, K., Bu, X., Zhao, C. and Jin, Y. (2021). Classification of white blood cells using weighted optimized deformable convolutional neural networks, *Artificial Cells, Nanomedicine, and Biotechnology* **49**(1): 147–155.

Yu, Y., Wan, M., Qian, J., Miao, D., Zhang, Z. and Zhao, P. (2024). Feature selection for multi-label learning based on variable-degree multi-granulation decision-theoretic rough sets, *International Journal of Approximate Reasoning* **169**: 109181.

Zhang, C., Ge, H., Zhang, S., Liu, D., Jiang, Z., Lan, C., Li, L., Feng, H. and Hu, R. (2021). Hematoma evacuation via image-guided para-corticospinal tract approach in patients with spontaneous intracerebral hemorrhage, *Neurology and Therapy* **10**: 1001–1013.

Zhang, R., Tan, J., Cao, Z., Xu, L., Liu, Y., Si, L. and Sun, F. (2024). Part-aware correlation networks for few-shot learning, *IEEE Transactions on Multimedia* **26**: 9527–9538.

Zhu, C. (2024). Computational intelligence-based classification system for the diagnosis of memory impairment in psychoactive substance users, *Journal of Cloud Computing* **13**(1): 119.

Muhammad Awais received his PhD degree from VLSI Lab, Department of Electronics & Telecommunications, Politecnico di Torino, Italy, in 2014. He is currently an associate professor at the Department of Electrical & Computer Engineering, COMSATS University Islamabad, Wah Campus, Pakistan. His research interests include optimization and signal processing for deep learning, advanced VLSI and FPGA/SoC design of high performance digital architectures from various domains including wireless communication, machine learning and signal processing.

Tallha Akram is an associate professor in the Department of Information Systems, Prince Sattam bin AbdulAziz University, KSA. He is a co-founder of Convex Solutions Pvt (Ltd), where his methodological work focuses on statistical machine learning, artificial intelligence, and data democratization to improve analytics adaptation. He graduated in 2008 from the University of Leicester, UK, with distinction, and his thesis was on the stability analysis of human neurons. In addition, he received his doctorate from Chongqing University, China, in 2014, and his thesis was focused on human attention models. During his stay in China, he got an opportunity to work in Rockwell Automation Lab, the American provider of industrial automation and information technology, where he managed to develop state-of-the-art computer vision/ AI algorithms.

Areej Alasiry holds a BSc degree in information systems from King Khalid University, Abha, Saudi Arabia, an MSc degree (Hons) in advanced information systems (2010), and a PhD degree in computer science and information systems from Birkbeck College, University of London, UK (2015). She is currently an assistant professor at the College of Computer Science, King Khalid University. Her main research interests include machine learning and data science.

Mehrez Marzougui received his PhD degree in electronics from the University of Monastir, Monastir, Tunisia, in 2005. From 2001 to 2005, he was a research assistant with the Electronics and Micro-Electronics Laboratory. From 2006 to 2012, he was an assistant professor with the Electronics Department, University of Monastir. Since 2013, he has been an assistant professor with the Engineering Department, College of Computer Science, King Khalid University, Abha, Saudi Arabia. His research interests include hardware/software cosimulation, image processing, and multiprocessor systems on chips.

Jongwoon Park is currently pursuing his PhD degree at the Department of Computer Science, Hanyang University, Seoul, Korea. His scientific interests include pattern recognition, electronics devices, and machine learning.

Byoungchol Chang is currently affiliated with the Department of Electronic Computer Engineering, Hanyang University, South Korea. His research interest include computer vision pattern recognition, electronics devices, and machine learning.

Received: 7 May 2024
Revised: 5 July 2024
Re-revised: 26 August 2024
Accepted: 30 August 2024