Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 5 (2023)

An Efficient Deep Learning Hybrid Model for Improving the Classification Accuracy of Blood Cell Images

1st **Rajalakshmi T**
Department of Computer and Information Science
Annamalai University
Chidambaram, India
rajilect@gmail.com

2nd **Senthil kumar C**
Department of Computer Science
Annamalai University PG Extension Center
Villupuram, India
csenthilkumar77@gmail.com

Abstract

Blood is the lifeline of human body and is also a key indicator to many underlying conditions or may predict the onset of serious conditions or disorders. Careful analysis with data inferred from the blood cell analysis is crucial for early detection and treatment of a number of life-threatening disorders. Amongst the various types of blood cells present in a blood cell image, WBCs or white blood cells give precise information about a number of such medical disorders. Analysis of WBC has been taken as the primary objective of this research work by invoking state-of-the-art image processing methods. A deep learning-based hybrid model has been proposed and implemented in this research work effective classification of WBCs from blood cell images obtained from benchmark dataset namely BCCD. The classification thus obtained has been compared with existing benchmark methods and in each case, the computed performance metrics indicate that proposed hybrid model outperforms existing methods.

Keywords: White Blood Cells, Classification, BCCD Dataset, Deep Learning, Accuracy.

1. INTRODUCTION

The Medical Health Care Sector has experienced a great deal of revolution in terms of innovative care provided to patients through rapid advancements in science and technology. This has given birth to a number state-of-the-art and cutting-edge tools for early detection and treatment of a number of medical conditions thereby contributing to significant reduction in global mortality rates. High resolution imaging modalities and their analysis through sophisticated and highly precise image processing methodologies and techniques have been predominant contributing factors for mortality reduction rate. Apart from conventional imaging modalities like MRI, CT, PET, Thermographs, Optical Imaging, etc. which are used to determine early onset of complex disorders and diseases, the simplest and most common test that people first resort is the blood test. Blood is the lifeline of the human body and is the first and foremost key indicator of any underlying conditions and abnormalities. Most people above a certain age resort to blood tests on a routing basis. A simple blood test may give a detailed analysis of various data of the body which may range from the hemogram down to allergens in the blood, functionalities of Liver Enzymes, Kidney Filtration processes, and a host of other data related to the human body functioning. One such report pertaining to the hemogram analysis gives a detailed account of the levels of hemoglobin in the blood, the volume of various types of blood cells which include red blood cells (RBC), White blood cells (WBC), Platelets, Monocytes, Lymphocytes etc. Their composition in the blood is a key indicator of normality or abnormal conditions. Analysis through microscope to analyze the blood cell images is an effective and most common methodology adopted. A sample image of blood cells depicting various types is illustrated in figure 1 shown below.

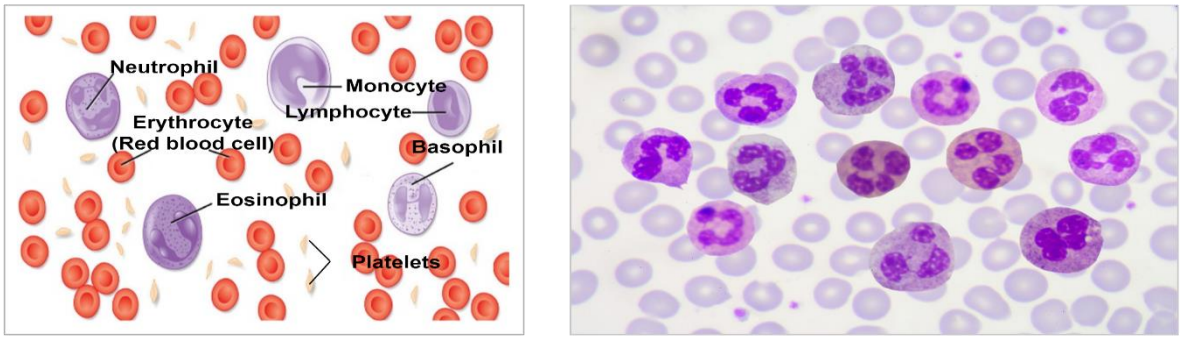

Figure 1 Illustration of Anatomy of blood cells and the typical blood smear (BCCD Dataset)

Each of the cell is characterized or classified into RBC, WBC and platelets with each of them being classified into many sub-classes of blood cell types. Study of each of these blood cell is very crucial to detect many underlying conditions in the body. The shape and concentration are of high essence. Unlike above image, where the concentarion of the cell is very less, in typical blood cell images, the concetration is extermely dense and hence identifying and classifiying them into their sub-clasess is quite a challenging issue. This is to a great extent eased by advent of state-of-the-art image processing techniques which are able to provide an automated process of segmenting and classifying them into their respective sub-classes. The concetration details of each of these cells could be done at the quickest time posible with high precision accuracy. A typical image processing methodology invovled in such a process is depicted in figure 2 shown below.

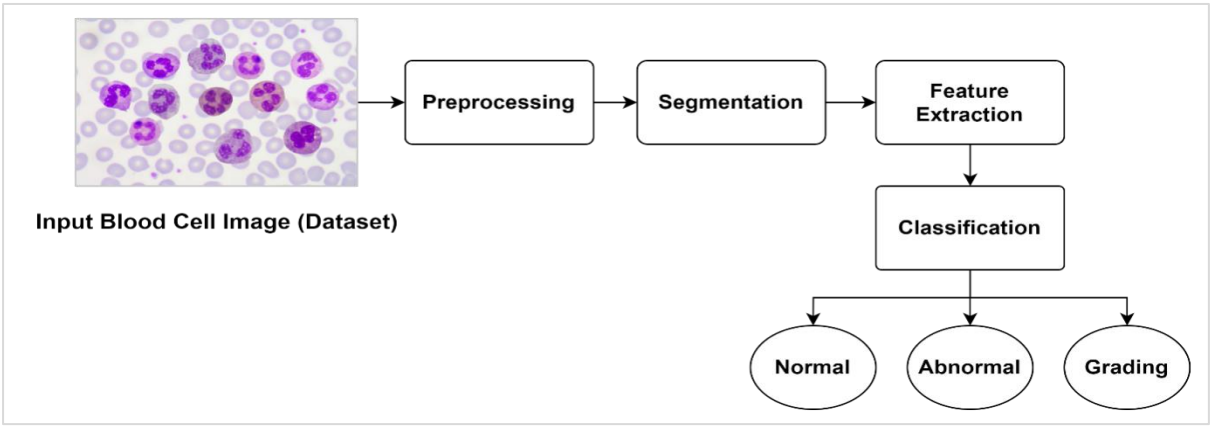

Figure 2 Illustration of typical Image Processing methodology for blood cell Classification

Input images obtained from benchmark datasets should be preprocessed prior to any further processing as they are prone to be corrupted with noise patterns especially speckle noise which is multiplicative in nature. A significant amount of impulsive noise may also be presented. In addition to preprocessing, the images may be contrast enhanced to improve the quality which may been degraded under poor illumination conditions etc. The preprocessed image is segmented to obtain the region of interest (ROI) alone rather than processing the whole image which may incur memory constraints and computational time overheads. Features which are key descriptors of the image are extracted and any redundant information is eliminated by using apt feature selection methods. The features are fed into the classifier where they are categorized into their target classes.

This research work primarily focuses on classifying WBC which are alternately termed as Leukocytes. The data related to WBC gives key indications on a number of disorders like Anemia, Immune Deficiency related disorders, Evan's syndrome etc. The sub-types of WBCs are five namely Neutrophils, Eosinophils, Basophils, Lymphocytes and Monocytes. With this objective, a deep learning model for efficient classification is proposed in this research work. The rest of the paper is organized into a systematic literature review in section 2 followed

by the formulation of the proposed work which is elaborately discussed in section 3. The experimentations are presented in section 4 followed by the concluding remarks in section 5 of the paper.

2. RELATED WORKS

Image processing technologies find widespread utility in a number of applications ranging from commercial utilities to healthcare sectors. A brief review of recent literature related to various segments of the workflow projected in figure 2 has been systematically presented in this section.

Manual classification of blood cells uses flow cytometry method and is usually very time consuming [5]. Sometimes it also leads to erroneous outputs due to inaccurate samples. Medical image classification is not just a simple process. Depending upon the application, it has to classify the type of cell, presence of a particular disease symptom, identify the stage and severity of malignancy. Hence computerized detection of blood cells was proposed initially. For automating this process in 1980, Computer Aided Detection (CAD) was introduced. But medical practitioners felt that it had higher false positive rates and researchers had to rely upon some other automation process [6]. Traditional learning and classification systems were then applied for this purpose of medical image classification. This process is shown in figure 3 below.

Figure 3. Traditional Classification process

The traditional process of blood cell image classification pre-processes the input using traditional methods like thresholding, resizing, cropping, denoising, filtering using filters like bilateral, global filters and non-local means filter. Etc. Features are then extracted using Principal Component Analysis, Partial Least Squares Analysis, Independent Component Analysis, Histogram of Oriented Gradients (HOG), Fishers Discriminant Ratio, Correlation Analysis, Local Binary Patterns (LBP), Scale Invariant Feature Transform (SIFT), Scale Up Robust Features (SURF), and Color Histograms [7]. Then classifiers like Linear Discriminant Analysis, Logistic Regression, K Nearest Neighbor, Naive Bayes, Decision Tree and Random Forest are used for the purpose of classification. Performance metrics like precision, Recall, specificity, F1 score and accuracy are calculated [8]. Now having reached the best of individual learning systems, we aim to create hybrid models by combining efficient classifiers that are compatible with each other in order to achieve better performance. Figure 4 shows the classification process using a hybrid classifier. It has some additional processes like data harmonization, data curation and data augmentation [9].

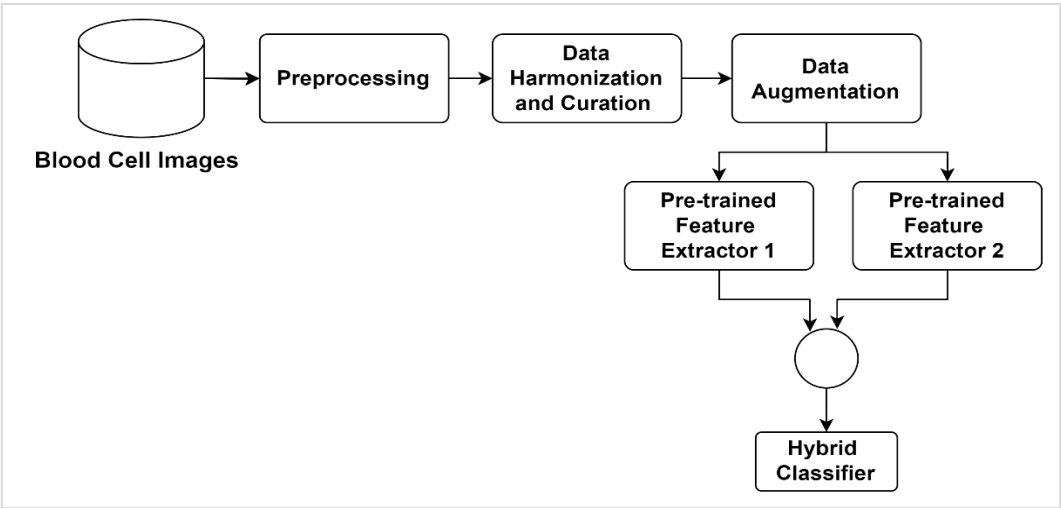

Figure 4. Hybrid Classification process

Traditional pre-processing process takes into account the quantitative characteristics of input images, whereas data harmonization is a process that looks out for the input image quality. There are various factors affect the input quality such as staining period of blood smear, illumination used while capturing the image, thickness of the blood film, any possible defects in the film etc [10]. These qualitative factors need to be addressed before features are extracted and training the system.

Data curation is similar to data harmonization. It is yet another process of checking the quality of input images before training the system. Using uncurated data leads to biased outputs, partially represented and inaccurate results. Data harmonization and curation is necessary because pathologists may use different optical microscopes and may follow different medical protocols. This leads to varied spatial and temporal resolution images. This needs to be compensated through normalization of images obtained from different systems while preserving the data. Data augmentation is the process of increasing the sample size of the inputs. It can be done in two ways either by over sampling them or using *data warping* operations like reflection, translation, rotation, scaling, mirroring, noise injection, cropping, averaging pixel values, filtering etc [11]. This is done to overcome the under fitting problem as medical data is not openly available much. Data augmentation is a proven solution to small data sets. It also solves the problem of '*Imbalance learning,*' which occurs when there are only a few samples in a particular class of inputs.

Hybrid systems include also additional measures like Receiver Operating Characteristics (ROC) which plots true positive rate against false positive rate and Area Under the Curve (AUC) which measures the area that falls under ROC. Blood cells usually have a variety of features with them that could be used for classification. Among them are color, intensity, shape, texture, geometrical, statistical, and morphological features [12]. Geometrical features include area of the blood cell, perimeter, orientation, and compactness, number of nuclei present in them. Morphological features include roundness, elongation, length to diameter ratio whereas statistical features include mean, variance, standard deviation, and skewness of the cell. It is usually advised to use many layers in feature extraction model instead of a single layer as lower layers learn less and simple features like edges and higher layers learn complex features. In order to improve the accuracy of the classifier, which is the aim of this paper, it is suggested to use pre-trained models as feature extractors instead of traditional feature extraction techniques available as this is the heart of the architecture that needs a lot of expertise and needs to be improved. These pre-trained models need not start from scratch as they are already trained and can perform faster too [13]. It is better suggested to use more than one feature extractor and take the best features among them to improve the rate of accuracy.

Artificial intelligence is a branch of science that deals with imparting knowledge to computers, and it has got numerous applications. Machine learning is a technology that helps us achieve artificial intelligence. Now comes deep learning, which is a subset of machine learning but varies from machine learning in many significant ways. Machine learning needs less training data and is not scalable in nature. It takes lesser time but gives lesser accuracy. It can be implemented on CPU and is for simple applications only. Deep Learning on the other hand needs large data for training and is easily scalable. It gives higher accuracy due to higher number of layers but consumes more time. GPU is needed for implementation and deep learning can be applied for complex applications also.

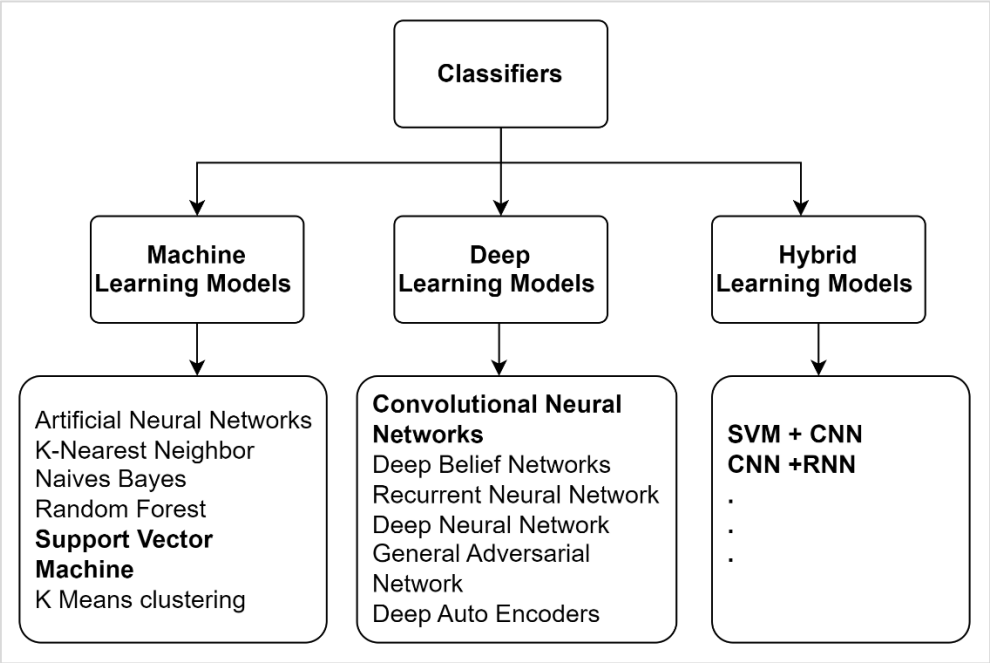

Figure 5. List of classifiers available

Figure 5 above shows the whole list of classifiers available among which Support Vector Machine (SVM) is proven to be the best among machine learning models and Convolutional Neural Networks (CNN) is the best under the category of deep learning and the combination of both CNN and SVM forms the best hybrid model. Convolutional neural networks have many pre-trained learning models such as Vgg16, Vgg19, DenseNet 121, DenseNet 169, ResNet 50, ResNet 121, Inception v2, Inception v3, MobileNet 224, MobileNet NasNet, Xception, etc. Among these DenseNet 169 is said to have the highest performance among all. Studies say that deep learning models have improved the diagnosis of tuberculosis diabetic retinopathy and skin cancer. It can be concluded that better performances are obtained by deep learning models compared to machine learning models. Whereas hybrid models perform better than deep learning models. Table 1 below shows the summary of works related to blood cell classification.

Table 1. Summary of Literature Survey

S.No.	Dataset Used	Hardware/ Software Employed	Preprocessing Technique	Feature Extraction Method	Classifier Used	Performance
[5]	MISP database - 100 images accuracy	Light microscope Axioscope 40 acromatic lens	Localizing WBC and cropping them	CNN	RF,SVM and KNN	Accuracy - 98.7%
[4]	Kaggle, 450 * 450	Tesla K80 GPU, Pytorch	Resizing to 299 * 299	Inception v3	XG boost	F1 score - 0.986
[13]	12,306 patients chest X-Ray	-	Self-supervised pre training	-	Multi instance contrastive learning	6.7% increase in accuracy 1.1% increase in mean AUC

[3]	76 subjects, 47 cancer patients, 29 normal patients, 450 * 450	Tensorflow Nvidia Geforce GTX 1080 GPU	Normalization with ImageNet mean, Resizing to 380 * 380		VGG 16, Mobilenet, VGG + Mobilenet	VGG16 Accuracy - 80.77, Mobilenet accuracy – 88, VGG + Mobilenet - 96.7 %
[7]	BCCD data set, 12,500 images 320 * 240	-	Resizing 200 to 200, Global thresholding	DenseNet	LR + Mobilenet 224	Accuracy - 97.03%, Outperforms fully connected network by 25.78 %
[2]	15,920 images	Intel core i7 processor, Matlab R 2019a, DM 96 analyzer	-	Disruption Based Salp Swarm and Cat Swarm Optimization Algorithm	VGD1600	% WBC classification overall classification 99 %
[14]	LISC, WBC datasets, 260 images, 720 * 576	-	Gaussian filter	SIFT	deep CNN	Accuracy - 95.84%
[15]	ISBIC NMC 2019 data, 60 cancer patients	-	-	UNet architecture	RCNN + Restnet 152 v2	Accuracy - 96%
[11]	C-NMC 2019 data, 15140 images 450 * 450 size	-	Resize to 300 * 300, tensor flow crop function	-	EfficientNet B3	Accuracy - 99.31%
[16]	CIFAR 100 data set	Intel icore 7, Nvidia GPU	-	Generative Adversarial Network (GAN)	DenseNet169, VGG16 and Resnet	Accuracy - 98.8%
[17]	BCCD database, 12,447 JPEG images	-	Matrix Transformation	Deep convolutional GAN	Resnet	Accuracy - 91.7%
[25]	2500 blood smears	-	Resizing 200 * 200	Batch size tuning	Incremental training	Accuracy - 61.5% to 95%, Precision - 76.6% to 94.27%

[18]	-	-	Resizing	PCA	DCNN, SVM modified loss function	Accuracy - 96.1 to 98.42%
[19]	ASH image bank	Internet of Medical Things (IoMT) platform	-	Local and global features	Hybrid CNN	Accuracy – 99%
[20]	1.2 million images	Canon power shot G5 Digicam	-	CNN	Statistically Enhanced Salt Swarm Algorithm	Accuracy – 87.9%
[21]	Kaggle, LISC database	Intel core i7 processor, Matlab R 2019a	-	Alexnet, Googlenet	SVM	Accuracy – 99.73%
[22]	HPBC dataset	-	Bi-cubic interpolation	RMSP Optimizer	BCNet	Accuracy – 98.51%

The major drawback with medical images is that input could be of different sizes and differ from patient to patient depending upon the severity of the underlying disease. Hence obtaining quality data set is highly difficult [23]. Low contrast, overlapping components in blood cells are also mentionable challenges faced [24]. Over fitting becomes a problem because of small sized data set. One more challenge is the legal issues that could arise if these results are used on a clinical basis. Explaining the basis of decision making in case of misclassification becomes very difficult. *Confusable classes* are yet another problem that arises because of visual resemblance of certain classes of blood cells [25].

3. PROPOSED WORK

The proposed blood cell image classification model comprises a hybrid model which incorporates deep learning with machine learning algorithms. The proposed model has multiple steps in which pre-processing is performed in the first step. Wiener filter and CLAHE (Contrast Limited Adaptive Histogram Equalization) algorithms for noise removal and contrast enhancement respectively. The pre-processed image is segmented in the second step using k-means clustering which segments the interest area from the background. In the third step, the segmented image is fed into SqueezeNet model for optimal feature extraction. Finally, classification performed in the last step includes support vector machine. The final classification results provide the type of blood cell. Figure 6 presents the proposed model complete overview which covers all the above-mentioned steps.

3.1 Dataset description

The dataset used in the proposed work is benchmark BCCD White Blood Cell (WBC) dataset which is a publicly available dataset in Kaggle repository. The dataset includes 7 classes of blood cell images and the total number of images is 3500. For each class, the dataset has 700 images. The size of the image in the dataset is 84×84 pixels. For our convenience, 366 images are selected from the entire dataset under five classes like basophil, eosinophil, lymphocyte, monocyte, and neutrophil. 12 images for basophil, 364 image for eosinophil, 136 images for lymphocyte, 84 images for monocyte, 864 images for neutrophil.

3.2 Pre-processing of input image

In the pre-processing step, Wiener filter and CLAHE (Contrast Limited Adaptive Histogram Equalization) algorithms are used. Wiener filter is a mean square error (MSE)-optimal stationary linear filter which is suitable for images which are degraded due to additive noise and blurring. Wiener filter performs additive noise removal and blurring effects simultaneously. Using inverse filtering principle, Weiner filter effectively minimizes the mean square error. The blood cell image used in the proposed work has noise artifacts and it is

removed using Weiner filter. To remove the noise from the input image, Wiener filter evaluates the mean and variance around each pixel. Mathematically the calculation of mean and variance is formulated as

$$\text{Mean } \mu(A) = \frac{1}{m.n}\sum_{i=1}^{m}\sum_{j=1}^{n} a_{i,j} \tag{1}$$

$$\text{Variance } \sigma^2(A) = \frac{1}{m.n}\sum_{i=1}^{m}\sum_{j=1}^{n} a_{i,j}^2 - \mu^2(A) \tag{2}$$

where the input image is represented as A. The Wiener filter combines the mean and variance term to perform filtering which is mathematically formulated as

$$w_{i,j} = \mu(A) + \frac{\sigma^2(A)-v^2}{\sigma^2(A)}\left(a_{i,j} - \mu(A)\right) \tag{3}$$

where, $w_{i,j}$ represents the Wiener, μ represents the mean, σ represents the variance, and noise variance is indicated as v^2. The noise removed image is further processed through CLAHE algorithm for contrast enhancement.

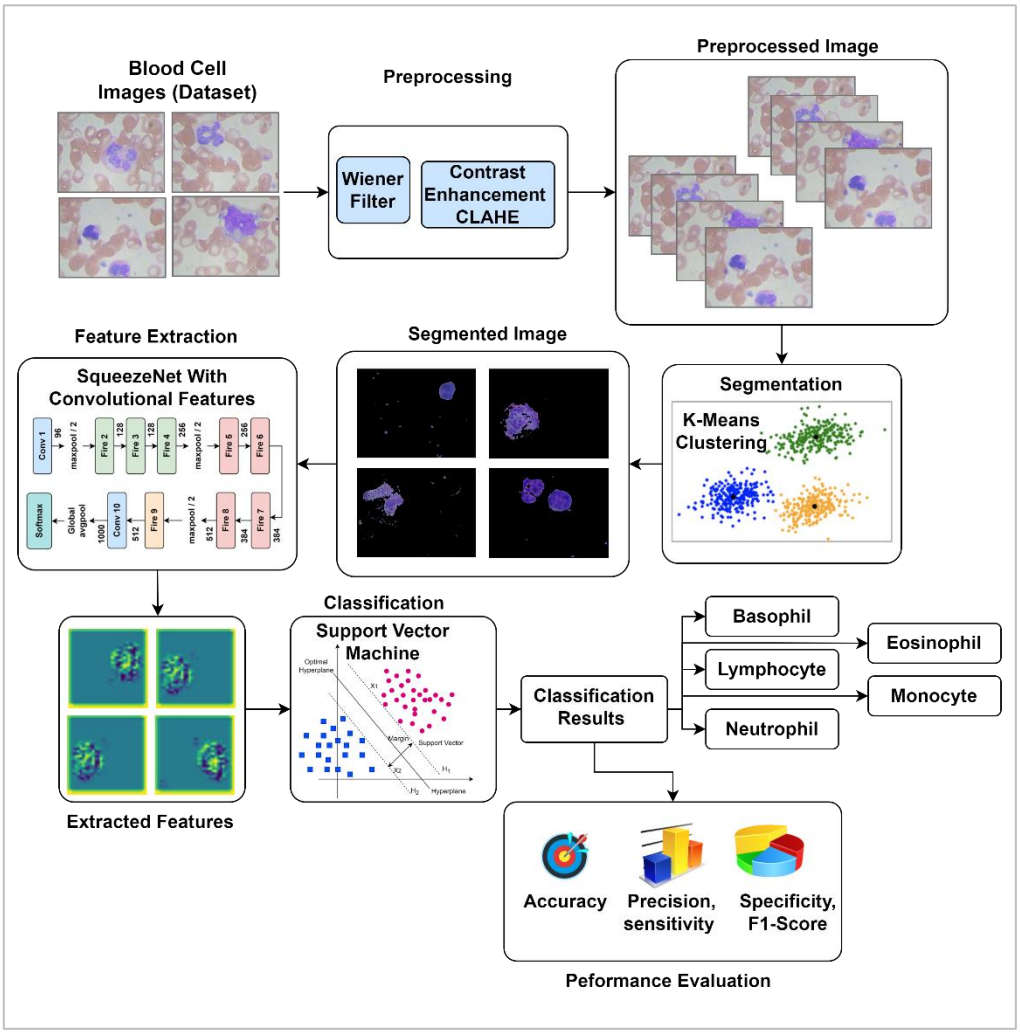

Figure 6. Proposed Hybrid Model for Blood Cell Image classification

CLAHE is a modified version of adaptive histogram equalization. An enhancement function is applied over neighborhood pixels to derive the transformation function in CLAHE. Traditional adaptive histogram equalization has contrast limitations and it is overcome by the CLAHE. CLAHE can be applied over gray scale and colored images. In the contrast enhancement process, clip limit, number of regions in row and column

direction, dynamic range and distribution parameter types are obtained initially. Then the image is divided into rectangular non-overlapping windows. Then a histogram is computed for each block and then excess portions are redistributed. Further cumulative distributive function is computed and it is used as a mapping function. Finally new image is obtained by interpolating the pixel values. The process of calculating histogram is mathematically formulated as

$$h(n) = \sum_{i=0}^{xx-1} \sum_{j=0}^{yy-1} g(n,i,j) \text{ for } n = 0,1,\dots,N-1 \quad (4)$$

$$g(n,i,j) = \begin{cases} 1 & if\ I(i,j) = n \\ 0 & otherwise \end{cases} \quad (5)$$

where n indicates the grey level, histogram bin, $h(n)$ indicates the nth bin histogram value, N indicates the number of histogram bins, dimension of image block is indicated as xx and yy. The pixel co-ordinates are indicated as i, j. The function which verifies whether the pixel value coordinates are equal to n is indicated as $g(n,i,j)$. The pixel value with coordinates (i,j) is indicated as $I(i,j)$.

3.3 Segmentation using K-Means clustering

The pre-processed image is further segmented using k-means clustering. Image segmentation in medical image processing is an essential process which extracts the necessary portion or region of interest from the background. The objects like organs or abnormal regions can be segmented effectively through clustering algorithms. One of the familiar clustering algorithms, k-means clustering algorithm is used in the proposed work for segmenting the white blood cell images. Compared to other clustering algorithms, k-means is computationally fast and simple to use. Also, k-means can able to work with diverse variables thus, k-means is used in the proposed model for segmentation.

Generally clustering divides, the dataset into specific groups. k-means clustering partitions the data into k-number of groups or disjoint cluster. The two phases in the k-means clustering calculates the k-centroid in the first phase. In the second phase, the clusters which has nearest centroid are selected. Euclidean distance is used to measure the distance of the nearest centroid. Further, grouping is performed and new centroids of each cluster is calculated between data point and center. Then new Euclidean distance is calculated for data point and center. The cluster which has minimum Euclidean distance is selected for further process. Using centroid and member objects, each cluster in the partition is defined and for each cluster centroid, the sum of distance from all the objects are minimized. A simple illustration of segmentation using k-means clustering is depicted in figure 7.

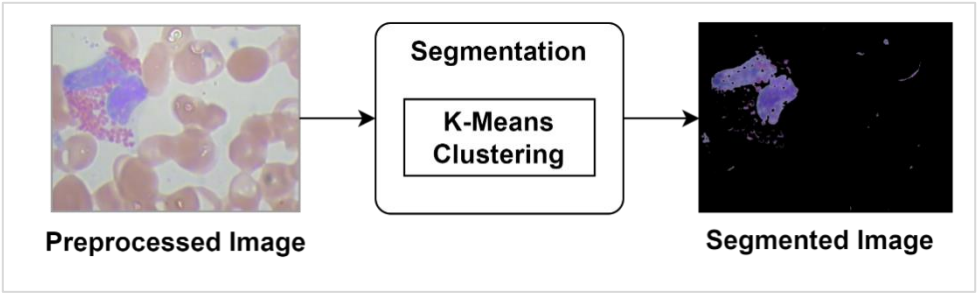

Figure 7. Segmentation using K-Means clustering

The steps followed in the segmentation process is given as follows. Consider a pre-processed image with $x \times y$ resolution. The image is then clustered into k-number of clusters. For each pixel of image, the Euclidean distance between pixel and center is calculated as follows.

$$d = \|p(x,y) - c_k\| \quad (6)$$

where d indicates the Euclidean distance, $p(x,y)$ indicates the input pixels, cluster centers are indicated as c_k. Next to distance calculation, all the pixels are assigned with nearest centers considering Euclidean distance. Once the pixels are assigned the new position of center is calculated as follows.

$$c_k = \frac{1}{k} \sum_{y \in c_k} \sum_{x \in c_k} p(x,y) \quad (7)$$

This process is repeated until the necessary condition is met. Then the cluster pixels are reshaped into an image to form the segmented image.

3.4 Feature extraction using SqueezeNet

The feature extraction in the proposed model is performed using SqueezeNet algorithm. The SqueezeNet is a pretrained Convolutional Neural Network (CNN) algorithm which works better than traditional CNN algorithms with minimum parameters. Figure 8 depicts the complete architecture of SqueezeNet model. The building block comprises a fire module which includes convolution layer with squeeze and expand layers.

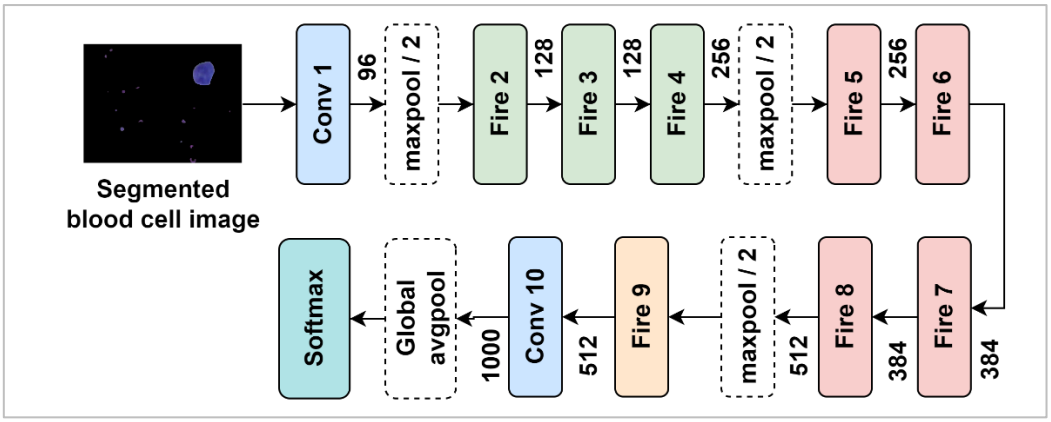

Figure 8. SqueezeNet Architecture

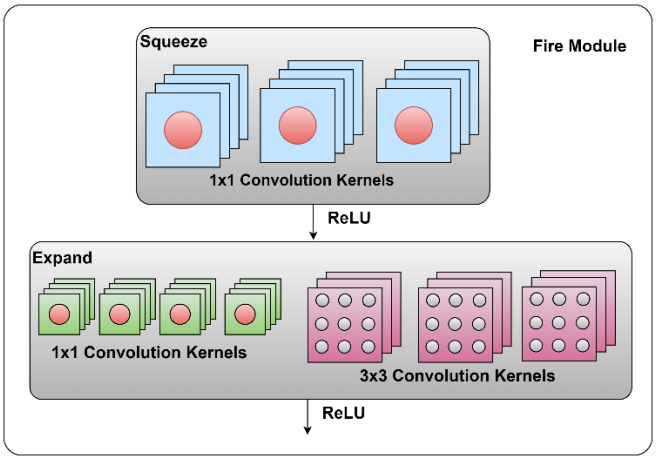

Figure 9. Fire block in SqueezeNet

The segmented input is fed into the convolution layer. The convolution layer has squeeze layer which has a single filter. Further the features are fed into expand layer which has 1×1 convolution kernels and 3×3 convolutional kernels. Using these layers, the spatial information is extracted at various scales. A detailed illustration of convolution layer or fire block in SqueezeNet architecture is depicted in figure 9. Followed by convolution layer, fire modules are included in the architecture which performs spatial information extraction. In between fire layers, max pooling layers are included to reduce the feature dimensions. All the squeeze and expand layers are connected using Rectified linear Unit (ReLU) activation function. To avoid data overfitting, a dropout layer is included next to fire 9 module.

Compared to traditional CNN architectures, improved performance can be attained through SqueezeNet. The major reason for this performance enhancement is its filter size. In the traditional CNN architectures, convolution layer will use 3×3 filters, but in the SqueezeNet 1×1 filters are used; thus, the network weights are greatly reduced. Next, by reducing the number of input channels into 3×3 filters, improved performance is attained in the SqueezeNet model. The last part of the network includes down sampling with large activation maps. Due

to this, the direct relationship between activation map size and classification accuracy is established. The fully connected dense layer in the conventional architecture is removed in the SqueezeNet model and replaced with convolution layer. Due to this, the number of output channels is made equal to the number of data classes. The complete layer details of SqueezeNet architecture are presented in table 2.

Table 2 SqueezeNet model

Layer	Filter/Input size	Stride	Padding
Input Image	224×224×1	-	-
Conv 1	1×1	2	-
Max Pooling	3×3	2	[0 0 0 0]
Fire 2	96 (1×1)	2	-
Fire 3	128 (1×1)	2	-
Fire 4	128 (3×3)	2	-
Max Pooling	3×3	2	[0 0 0 0]
Fire 5	256 (3×3)	2	-
Fire 6	256 (3×3)	2	-
Fire 7	384 (3×3)	2	-
Fire 8	384 (3x3)	2	-
Max Pooling	3×3	2	[0 0 0 0]
Fire 9	512 (3×3)	2	-
Dropout	50%	-	-
Conv 10	512 (3×3)	2	-
Max Pooling	3×3	2	[0 0 0 0]
FC1	1000	-	-

3.5 Classification using Support vector machine (SVM)

The final step in the proposed model includes classification operation where support vector machine is used as classifier to classify the extracted features using sqeezeNet model. SVM is an efficient machine learning algorithm which is widely used for classification and regression analysis. Data samples are mapped into high dimensional space. A set of hyperplanes in the classifier divides the data into different partitions. Figure 10 depicts a simple illustration for support vector machine classification.

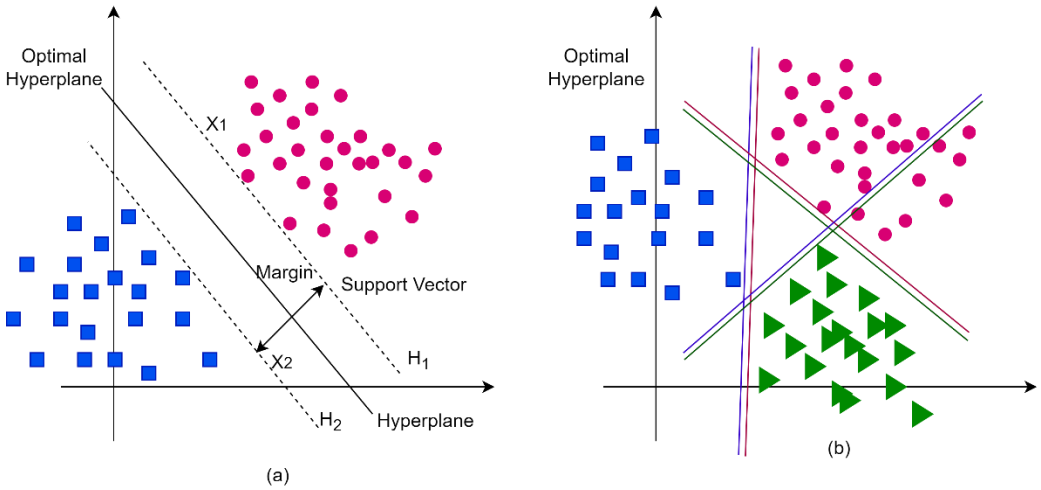

Figure 10. SVM classifier (a) Binary classification (b) Multi-class Classification

Generally, SVM is used to binary classification. However, in the proposed work, multiple class of data samples are used. Thus, a multi-class classification is performed in this proposed model. The hyperplane in SVM separate the data into two groups. Whereas in multi-class classification the data is divided into multiple groups. In the proposed model, the output is classified into five classes as basophil, eosinophil, lymphocyte, monocyte and neutrophil.

4. RESULTS AND DISCUSSION

The experimentation of proposed model utilizes benchmark BCCD White Blood Cell (WBC) dataset which is a publicly available dataset in Kaggle repository. The dataset includes 3500 images of 7 classes of blood cell images. For experimental convenience, five classes of data are considered. Basophil, Eosinophil, Lymphocyte, Monocyte and Neutrophil images are considered for experimentation. The experimentation is performed in Python tool and the necessary packages required to run the experimentation are tensorflow==2.10.0, keras, tqdm, opencv-python, scikit-image, numpy, pandas, scipy, matplotlib, seaborn, scikit-learn, pyqt5, prettytable, and cmapy. The hyperparameters for the feature selection and classification model used in the experimentation are listed in table 3.

Table 3. Hyperparameter details

S.No	Hyperparameter	Range/Type
Squeezenet	Learning rate	0.01
	Batch size	30
	Number of Epochs	30
SVM	Penalty factor (c)	1
	Gamma (γ)	0.01
	Kernel	RBF

Figure 11 depicts the samples used as input image from the dataset. Five classes of data are considered for analysis. The input image is pre-processed using Wiener filter and CLAHE algorithm. The additive noise in the input image is removed using Wiener filter and contrast is enhanced using CLAHE. Figure 12 depicts the pre-processed image output for all the classes. After pre-processing, the essential portions are segmented using k-means clustering. Figure 13 depicts the segmented output for the given pre-processed image.

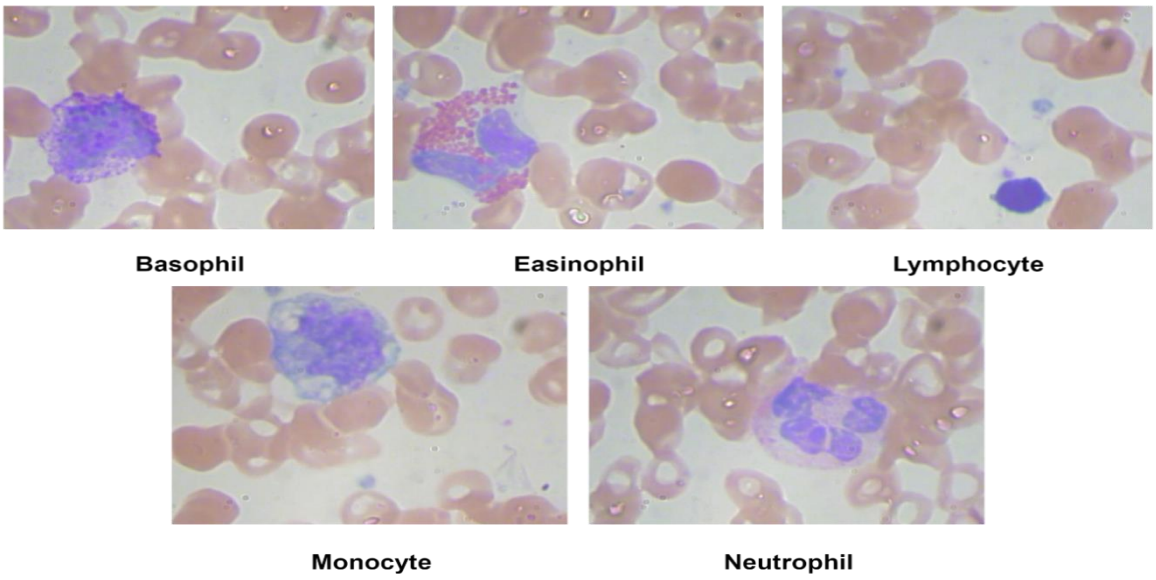

Figure 11. Input images from dataset

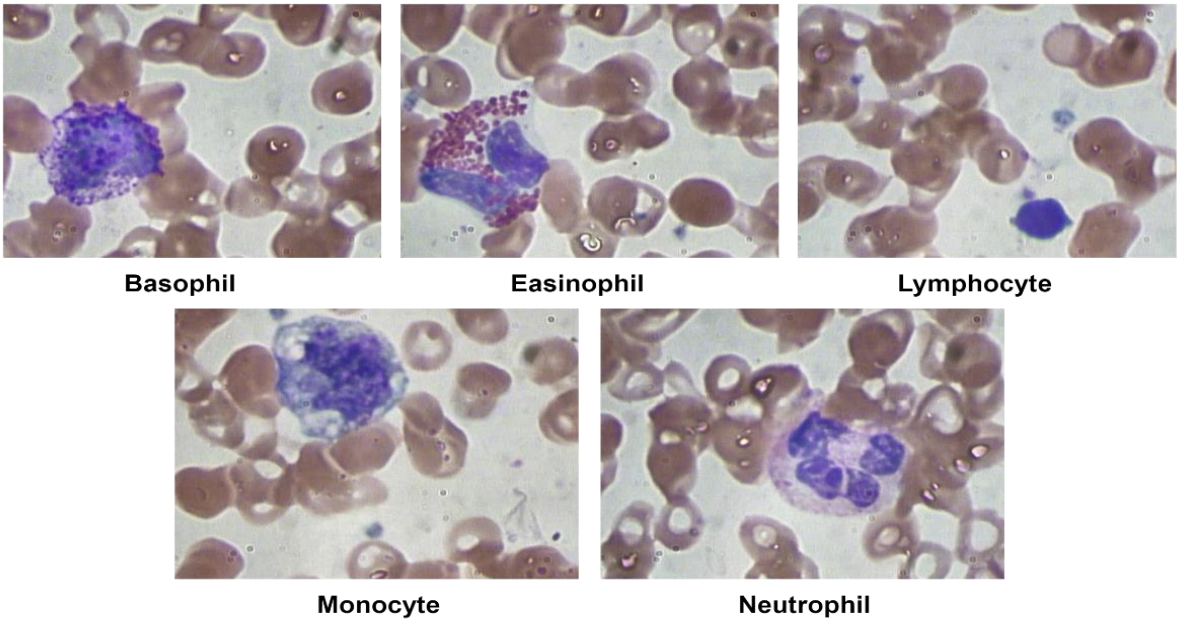

Figure 12. Pre-processed Input images

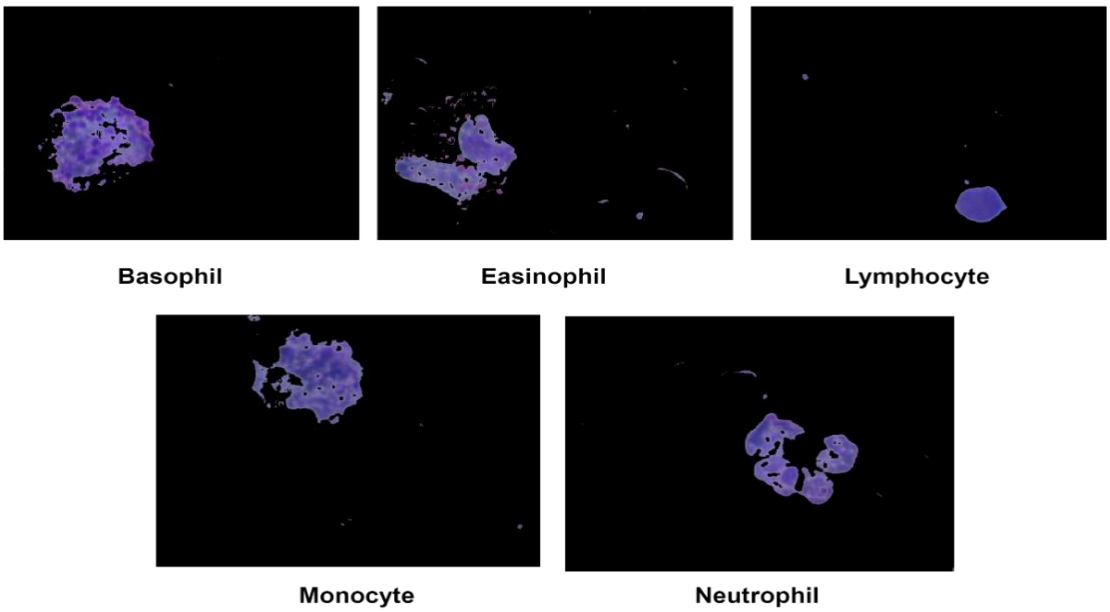

Figure 13. Segmented Input images

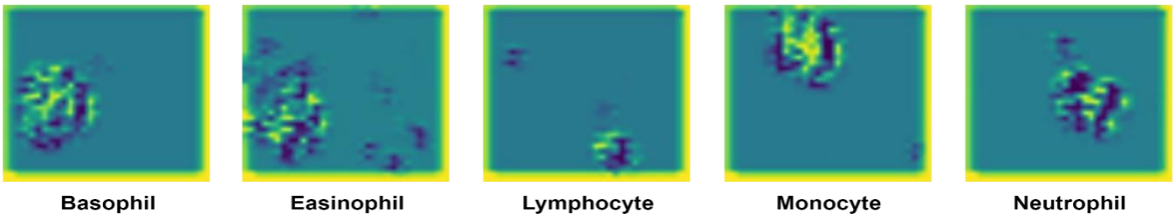

Figure 14. Extracted features

Figure 14 depicts the feature extracted output for five class of images. The feature extraction is performed through SqueezeNet algorithm. Finally classification is performed using support vector machine. The confusion matrix obtained by classification model is presented in figure 15. From the confusion matrix, based on the True Negative (TN), False Positive (FP), True Positive (TP), and False Negative (FN) values are obtained. From the obtained values, performance metrics like accuracy, Recall, specificity, precision, and f1-score are calculated. The essential mathematical formulations for the performance metrics are presented as follows.

$$Accuracy = (TP + TN)/(TP + TN + FP + FN) \quad (8)$$

$$Recall = TP/(TP + FN) \quad (9)$$

$$Specificity = TN/(TN + FP) \quad (10)$$

$$Precision = TP/(TP + FP) \quad (11)$$

$$F1 = 2 \times \frac{(Specificity \times Sensitivity)}{(Specificity + Sensitivity)} \quad (12)$$

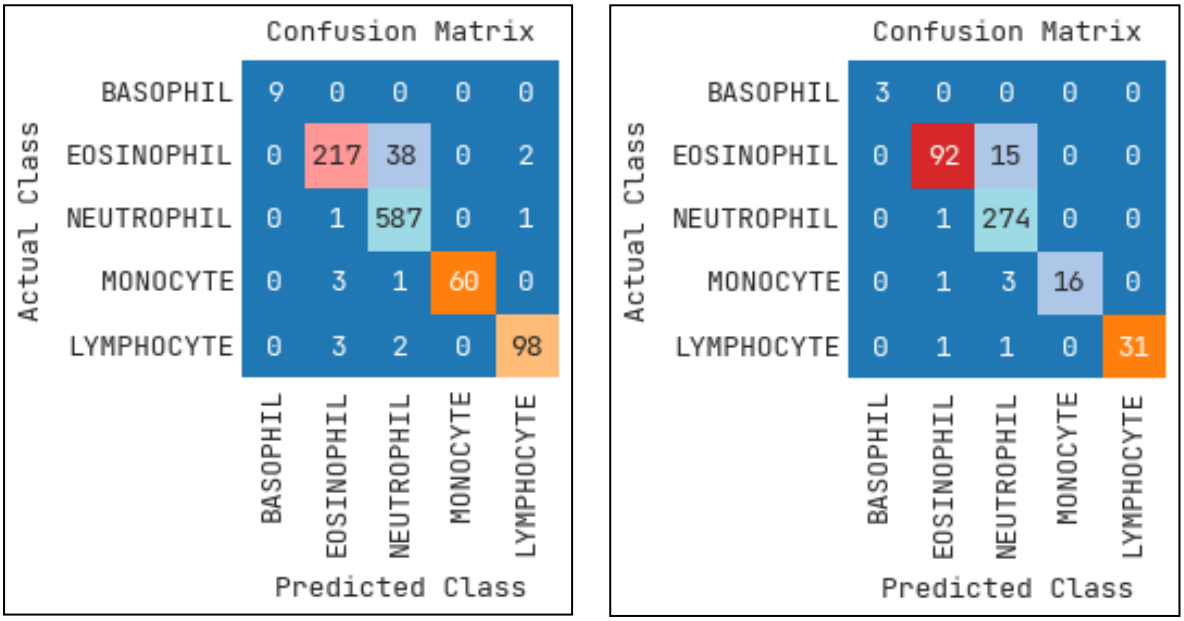

Figure 15. Confusion Matrix

The receiver operating characteristics curve of proposed model is presented in figure 16. Considering the false positive rate and true positive rate the ROC curve is obtained for all the classes. For each class, AUC score is calculated and analyzed. In general, if the the AUC score is above 0.9 then the performance is considered as excellent. In the proposed model ROC curve, it can be observed that all the classes attained above 0.9 AUC score. This indicates the excellent performance of the proposed model.

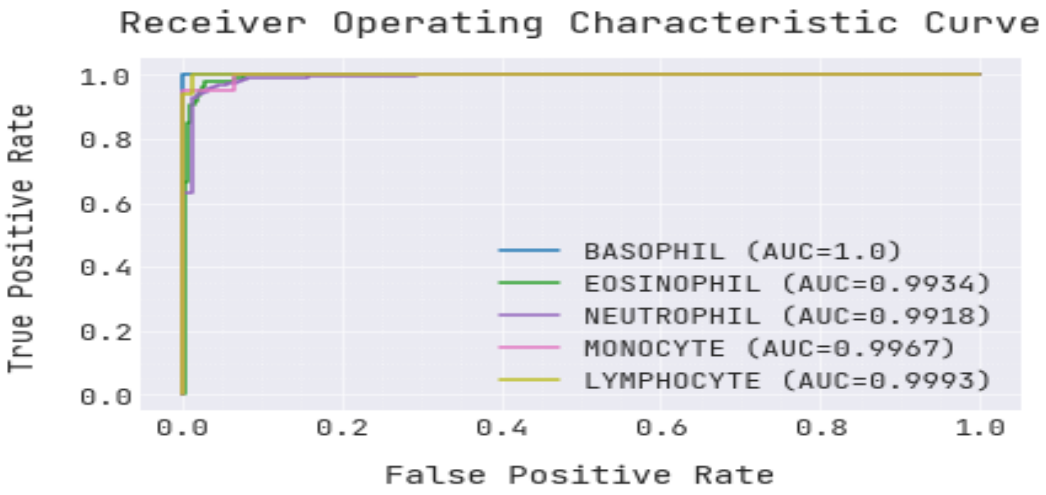

Figure 16. ROC Curve

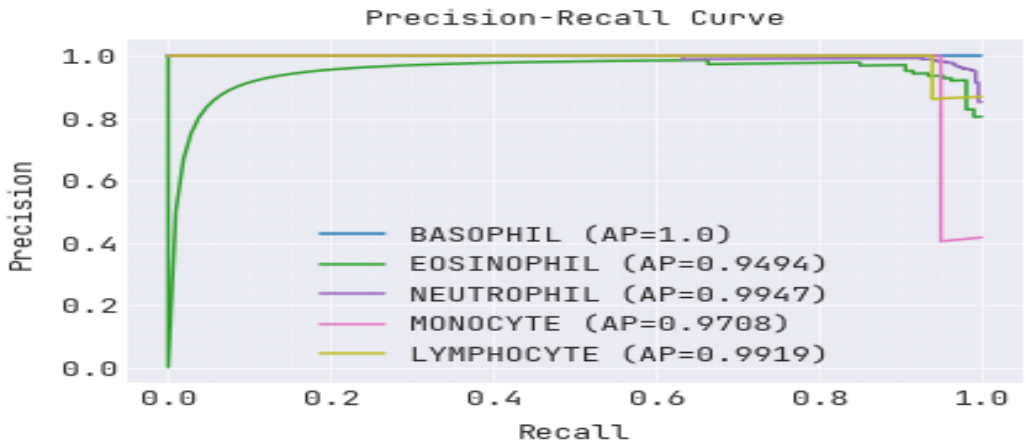

Figure 17. Precision – Recall Curve

Figure 17 depicts the precision-recall analysis of proposed model for all the classes. The average precision (AP) value obtained for all the classes are above 0.9 which indicates the better precision and recall performance of the proposed model for all the classes. The high precision in the figure indicates the low false positive rates and high recall indicates the low false negative rates.

Table 4 Proposed model Performance metrics

S.No	Metrics	Values
1	Accuracy	0.947
2	Precision	0.9807
3	Recall	0.9191
4	Specificity	0.9749
5	F1	0.9467
6	ROC	0.9962

The overall performance metrics of the proposed model is listed in table 4 for precision, specificity, Recall, F1-score, and accuracy metric. Further to validate the proposed model performances, existing research

works are considered for comparative analysis. The results of Hua Chen.et.al., 2022 [26] is considered where the proposed model combines ResNet and DenseNet algorithms as a hybrid model which is indicated as Res-DenseNet in the comparative analysis. Additionally, the traditional methods like InceptionV3, convolutional Neural Network (CNN), DenseNet and ResNet are experimented and the results are used for comparative analysis.

The comparative analysis given in figure 18, compares the proposed model performance with existing methods for precision and Recall (or) recall metrics. The results indicate the proposed model better performances over existing methods. A better precision over existing methods is attained by the proposed model which is 98.07%. Similarly, the proposed model maximum recall value of 91.91% is much better than the existing models. The lowest performance is attained by the inception model which is due to the limited feature extraction characteristics.

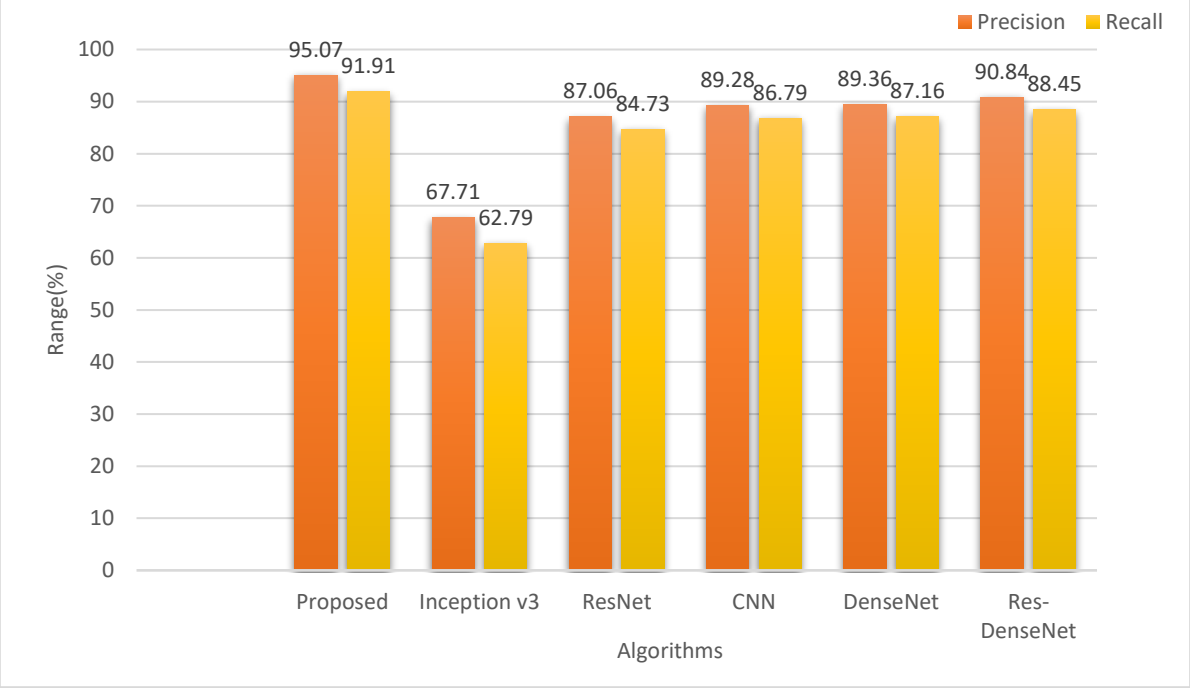

Figure 18. Precision-Recall comparative analysis

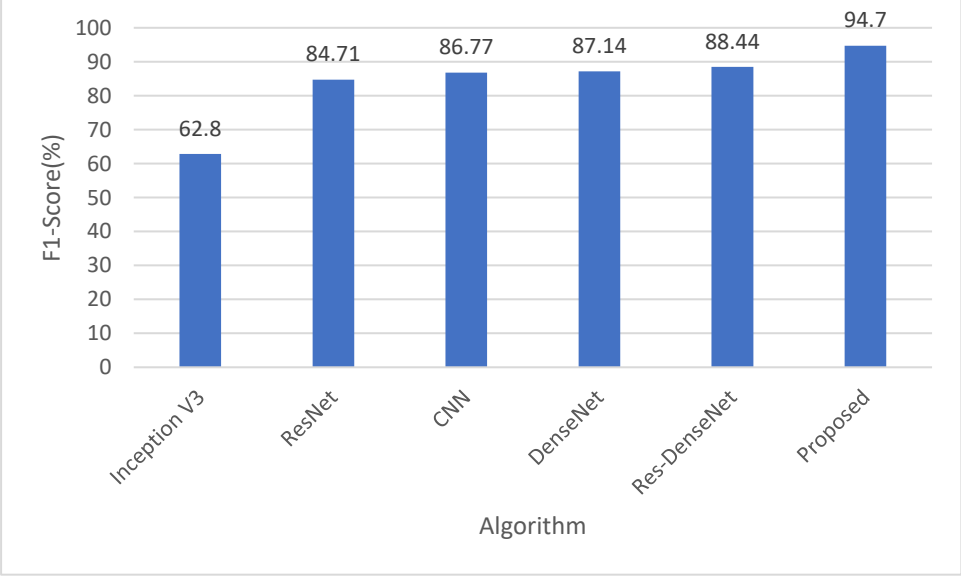

Figure 19. F1-Score comparative analysis

The proposed and existing method f1-score analysis is depicted in figure 19. Compared to existing methods, proposed model exhibited maximum f1-score. The attained 95.21% f1-score of proposed model is approximately 8% better than the Res-DenseNet model, 9% greater than the DenseNet and CNN models, 10% better than the ResNet model and 32% better than the Inception model.

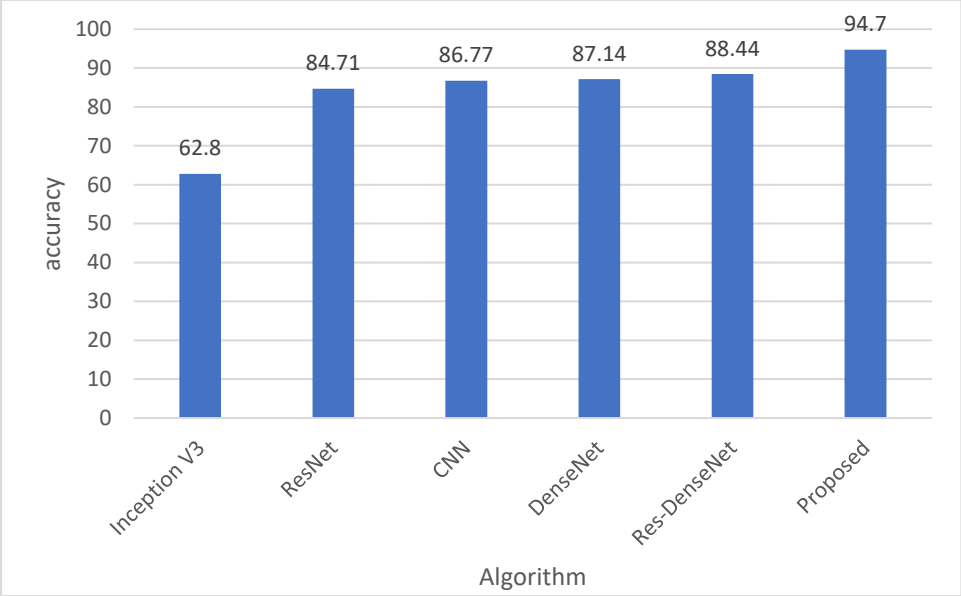

Figure 20. Accuracy comparative analysis

Figure 20 depicts the accuracy comparative analysis of proposed and existing methods. The proposed model attained maximum classification accuracy compared to existing methods. The maximum accuracy attained by the proposed model is 95.42% where as 88.44% is attained by the Res-DenseNet model which is approximately 8% lesser than the proposed model accuracy. 87.14 % of accuracy was attained by the DenseNet model. However, it is 9% lesser than the proposed model. The accuracy attained by the CNN model is 86.77% which is approximately 10% lesser than the proposed model. The accuracy attained by the ResNet algorithm is 84.71% which is 11% lesser than the proposed model. The minimum accuracy attained by the inception v3 model is 62.8% which is 33% lesser than the proposed model.

Table 5. Comparative analysis with existing algorithms

S.No	Algorithms	Metrics			
		Accuracy	Precision	Recall	F1-Score
1	Proposed	94.7	98.07	91.91	94.67
2	Inception v3	62.8	67.71	62.79	63.52
3	ResNet	84.71	87.06	84.73	85.15
4	CNN	86.77	89.28	86.79	87.1
5	DenseNet	87.14	89.36	87.16	87.48
6	Res-DenseNet	88.44	90.84	88.45	88.73

From the comparative analysis given in table 5, it can be observed that proposed model attained better classification performances over existing methods. The maximum accuracy and f1-score of proposed model validates that the classification performance is accurate than existing deep learning models.

5. CONCLUSION

This research work presents an efficient deep learning hybrid model for improving the classification accuracy of blood cell images. The presented approach includes wiener filter and CLAHE to pre-process the input image. From the pre-processed image, segmentation is performed by k-means clustering. Further SqueezeNet model is incorporated for feature extraction and classified through support vector machine. Due to effective pre-processing and optimal feature extraction, the classification accuracy is greatly improved in the proposed model. Experimental results are compared with existing deep learning models like Inception v3, ResNet, CNN, DenseNet, Res-DenseNet algorithms to validate the proposed model better performances. Further this research work can be extended by incorporating hybrid deep learning algorithms and optimization algorithms for better classification performances.

REFERENCES

1. Lei Cai, Jingyang Gao, Di Zhao. (2020). A review of the application of deep learning in medical image classification and segmentation. Annals of Translational Medicine. Vol. 8, no.11. pp. 1-15.

2. Joshi, S., Kumar, R. and Dwivedi, A. (2020). Hybrid DSSCS and convolutional neural network for peripheral blood cell recognition system. IET Image Process. Vol.14, pp. 4450-4460.

3. S. H. Kassani, P. H. Kassani, M. J. Wesolowski, K. A. Schneider and R. Deters. (2019). A Hybrid Deep Learning Architecture for Leukemic B-lymphoblast Classification. International Conference on Information and Communication Technology Convergence. pp. 271-276.

4. S. Ramaneswaran, Kathiravan Srinivasan, P. M. Durai Raj Vincent and Chuan-Yu Chang (2021). Hybrid Inception v3 XGBoost Model for Acute Lymphoblastic Leukemia Classification. Computational and Mathematical Methods in Medicine. Vol. 2021, pp. 1-10.

5. A. Malkawi, R. Al-Assi, T. Salameh, B. Sheyab, H. Alquran and A. M. Alqudah. (2020). White Blood Cells Classification Using Convolutional Neural Network Hybrid System IEEE 5th Middle East and Africa Conference on Biomedical Engineering (MECBME), pp. 1-5.

6. Mingyu Kim, Jihye Yun, Yongwon Cho, Keewon Shin, Ryoungwoo Jang, Hyun-jin Bae, Namkug Kim. (2019). Deep Learning in Medical Imaging. Neurospine. Vol. 16, no. 4, pp. 657-668.

7. Mohamed, Ensaf & El-Behaidy, Wessam & Khoriba, Ghada & Li, Jie. (2020). Improved White Blood Cells Classification based on Pre-trained Deep Learning Models. Journal of Communications Software and Systems. Vol. 16, No. 1, pp. 1-9.

8. Zolfaghari, M., Sajedi, H. (2022). A survey on automated detection and classification of acute leukemia and WBCs in microscopic blood cells. Multimed Tools Appl. Vol. 81, pp. 6723–6753.

9. Alzubaidi, Laith, Mohammed A. Fadhel, Omran Al-Shamma, Jinglan Zhang, and Ye Duan. (2020). Deep Learning Models for Classification of Red Blood Cells in Microscopy Images to Aid in Sickle Cell Anemia Diagnosis. Electronic. Vol. 9, no. 3. pp. 1-18.

10. Isabella Castiglioni, Leonardo Rundo, Marina Codari, Giovanni Di Leo, Christian Salvatore, Matteo Interlenghi, Francesca Gallivanone, Andrea Cozzi, Natascha Claudia D'Amico, Francesco Sardanelli. (2021). AI applications to medical images: From machine learning to deep learning. Physica Medica. Vol. 83, pp. 9–24.

11. A. Batool and Y. -C. Byun. (2023). Lightweight EfficientNetB3 Model Based on Depthwise Separable Convolutions for Enhancing Classification of Leukemia White Blood Cell Images. IEEE Access. vol. 11, pp. 37203-37215.

12. Mustafa Ghaderzadeh , Farkhondeh Asadi , Azamossadat Hosseini , Davood Bashash , Hassan Abolghasemi and Arash Roshanpour. (2021). Machine Learning in Detection and Classification of Leukemia Using Smear Blood Images: A Systematic Review. Scientific Programming. Vol. 2021, pp. 1-14.

13. Shekoofeh Azizi, Basil Mustafa, Fiona Ryan∗, Zachary Beaver, Jan Freyberg, Jonathan Deaton, Aaron Loh, Alan Karthikesalingam, Simon Kornblith, Ting Chen, Vivek Natarajan, Mohammad Norouzi. (2021). IEEE/CVF International Conference on Computer Vision (ICCV)

14. Manthouri M, Aghajari Z, Safary S. (2022). Computational Intelligence Method for Detection of White Blood Cells Using Hybrid of Convolutional Deep Learning and SIFT. Comput Math Methods Med. vol 2022, pp. 1-8.

15. Alharbi, A. H., Aravinda, C. V., Lin, M., Venugopala, P. S., Reddicherla, P., & Shah, M. A. (2022). Segmentation and Classification of White Blood Cells Using the UNet. Contrast media & molecular imaging. Vol. 2022. pp. 1-8.

16. Almezhghwi K, Serte S.(2020). Improved Classification of White Blood Cells with the Generative Adversarial Network and Deep Convolutional Neural Network. Computational Intelligence and Neuroscience. Vol. 2020. pp. 1-12.

17. Ma L, Shuai R, Ran X, Liu W, Ye C.(2020). Combining DC-GAN with ResNet for blood cell image classification. Med Biol Eng Comput. Vol. 58, no.6, pp. 1251-1264.

18. Basnet, J., Alsadoon, A., Prasad, P.W.C. et al. (2020). A Novel Solution of Using Deep Learning for White Blood Cells Classification: Enhanced Loss Function with Regularization and Weighted Loss (ELFRWL). Neural Process Lett. Vol. 52, pp. 1517–1553.

19. Sakthiraj, F.S.K. (2022). Autonomous Leukemia Detection Scheme Based on Hybrid Convolutional Neural Network Model Using Learning Algorithm. Wireless Pers Commun. Vol. 126, pp. 2191–2206.

20. Sahlol, A.T., Kollmannsberger, P. & Ewees. (2020). A.A. Efficient Classification of White Blood Cell Leukemia with Improved Swarm Optimization of Deep Features. Sci Rep. vol. 10, no. 2536. pp. 1-11.

21. Çınar, A., Tuncer, S.A. (2021). Classification of lymphocytes, monocytes, eosinophils, and neutrophils on white blood cells using hybrid Alexnet-GoogleNet-SVM. SN Appl. Sci. vol. 3, no. 503. pp. 1-11.

22. Chola, C., Muaad, A. Y., Bin Heyat, M. B., Benifa, J. V. B., Naji, W. R., Hemachandran, K., Mahmoud, N. F., Samee, N. A., Al-Antari, M. A., Kadah, Y. M., & Kim, T. S. (2022). BCNet: A Deep Learning Computer-Aided Diagnosis Framework for Human Peripheral Blood Cell Identification. Diagnostics, vol. 12, no. 11. pp. 1-17.

23. Khan, Siraj & Sajjad, Muhammad & Abbas, Naveed & Rehman, Amjad. (2022). A Review on Machine Learning-Based WBCs Analysis in Blood Smear Images: Key Challenges, Datasets, and Future Directions. Prognostic Models in Healthcare: AI and Statistical Approaches. vol. 109, pp 293–314.

24. A.M. Patil, M.D. Patil, G.K. Birajdar. (2021). White Blood Cells Image Classification Using Deep Learning with Canonical Correlation Analysis, IRBM,Vol. 42, no. 5, pp. 378-389.

25. Al-Qudah, R., & Suen, C. Y. (2021). Improving blood cells classification in peripheral blood smears using enhanced incremental training. Computers in biology and medicine. Vol. 131. pp. 1-16.

26. Hua Chen, Juan Liu, Chunbing Hua, Jing Feng, Baochuan Pang, Dehua Cao, Cheng Li.(2022). Accurate classification of white blood cells by coupling pre-trained ResNet and DenseNet with SCAM mechanism, BMC Bioinformatics, Vol.23, no. 282, pp. 1-20.