SN Computer Science (2021) 2:70
https://doi.org/10.1007/s42979-021-00458-2

ORIGINAL RESEARCH

Red Blood Cell Classification Using Image Processing and CNN

Mamata Anil Parab[2] · **Ninad Dileep Mehendale**[1]

Received: 24 May 2020 / Accepted: 8 January 2021 / Published online: 2 February 2021

Abstract

In the medical field, the analysis of the blood sample of the patient is a critical task. Abnormalities in blood cells are accountable for various health issues. Red blood cells (RBCs) are one of the major components of blood. Classifying the RBC can allow us to diagnose different diseases. The traditional, time-consuming technique of visualizing RBC manually under the microscope, is a tedious task and may lead to wrong interpretation because of the human error. The various health conditions can change the shape, texture, and size of normal RBCs. The proposed method has involved the use of image processing to classify the RBCs with the help of convolution neural networks. The algorithm can extract the feature of each segmented cell image and classify it into 9 various types. Images of blood slides were collected from the hospital. The overall accuracy was 98.5%. The system has been developed to provide accurate and fast results that can save patients' lives.

Keywords Red blood cells · Convolution neural network · Classification

Introduction

Blood is the body fluid that delivers various substances such as nutritions and oxygen to cells and takes away metabolic waste from the cells. The blood accounts for 7–8% of total body weight [1]. The human blood cells have three major components as platelets, white blood cells (WBCs), and red blood cells (RBCs). RBCs are the majority of the blood sample counts and are responsible for providing oxygen to the various vital organs of the body, determining blood type, and also carrying away the waste product. In normal human, RBCs are flexible and appear biconcave disk-shaped with 7–8 µm in diameter and 2 µm in thickness. Nearly 84% of cells in the human body are RBCs. Approximately half (i.e. 40–45%) of the blood volume is occupied by RBCs.

According to Walker et al. [2], many diseases, and health conditions may be responsible for variation in shape, size, color, or appearance of RBCs.

In the case of Pyridoxine deficiency or Thalassemia or iron deficiency, the shape of RBC shrinks compare to normal size and become almost 5 µm. This type of RBCs are termed as Microcytes. In case of severe iron deficiency, RBCs appear oval or elongated or slightly egg-shaped. Such RBCs are termed as Elliptocytes. Some liver diseases turn the shape of RBC in a coffee bean-like structure and such kinds of RBCs are termed as Stomatocytes. Macrocytic RBCs are the result of vitamin B12 or folate deficiency leads to an increase in the shape of RBC up to 9–14 µm. Round-shaped RBCs turn into sickle-shaped due to Sickle cell anemia. Disease like myelofibrosis and underlying marrow infiltrate change the normal RBC shape into teardrop-shaped and this kind of RBCs are termed as teardrop RBCs. Codocytes or target cells RBCs have the appearance of the shooting target with a bullseye. Thse target RBCs are generated because of the presence of Thalassemia, hemoglobinopathies, or liver diseases. Due to hereditary spherocytosis and anemia, RBCs become more sphere-shaped rather than usual round, biconcave shaped. Such RBCs are termed as spherocytes. Howell–Jolly RBC is the result of asplenia, in which round, dark purple to red color spot is seen in normal RBCs.

As several diseases are associated with the appearance, shape, or size of RBCs, it is necessary to analyze the RBCs carefully. Traditional inspection of RBCs under the microscope is time-consuming, expensive, and needs expert knowledge. In the last few years, new technologies are working to overcome these problems. However, this sort of image

✉ Ninad Dileep Mehendale
ninad@somaiya.edu

[1] K. J. Somaiya College of Engineering, Vidyavihar, Mumbai 400077, Maharashtra, India

[2] Ninad's Research Lab, M. G. Road, Thane 400602, Maharashtra, India

processing technologies are facing various challenges, such as (1) RBCs can overlap each other and look like cluster which hides the edges of an individual cell and make difficulties in detecting the edges. (2) The edges of RBCs may be blurry and it may be due to image capturing procedure. (3) There may be less contrast between the background and foreground. This may increase the difficulties while separating out the individual RBC. (4) The artifact may present in the captured image due to light interference. To overcome all these problems we added the help of CNN to image processing. In this manuscript, we present a microscopic RBC image analysis with the convolution neural network (CNN). CNN is a strong image classifier tool, in which image is taken as input, classify it under certain categories based on their features. In CNN, an individual unit is called a neuron. Neurons are located in a series of layers. Neurons of one layer are connected to the neurons of the next layer. Each neuron or node of one layer perform mathematical calculation and pass the results to the next node. The last layer of the neural network has increased computational power due to the accumulation of experience. In the proposed technique, the colored microscopic image was converted into a grayscaled image. The Canny edge detection algorithm is used to detect edges of cells in gray-scaled images. Area-based filters are used to filter unwanted regions from the image. In this system, a single RBC image of various types is fed as an input to the deep learning network. A testing image of a single RBC cell is given as input to this trained network and checked for the presence of the disease. In this study, we assessed the performance of multi-class classification using deep learning which is robust and modern technology on a real-life RBC image dataset. The main contributions of the proposed work are: (1) the system can identify and classify ten (one normal and 9 abnormal) different types of RBCs. In past studies, such a large number of classes has never been done with such good accuracy. Due to accurate classification, one can accurately diagnose different pathological conditions, such as, Thalassemia, iron deficiency, folate deficiency, sickle cell anemia, myelofibrosis, asplenia, etc. We believe the proposed system can start a new era of automated pathological diagnosis. (2) We have included highly optimized GPU implementation of a 2D convolution network that ensures the fast and robust network learning for a huge dataset. Since training is already done the trained model removes any need for a high-end GPU or processor and can be executed on any low configuration machine.

Literature Review

From the last few decades, extensive work has been done in the field of blood cell segregation and analysis. This analysis is done by integrating various techniques like artificial intelligence, image processing, pattern recognition, and computer vision. Detection and evaluation of a disease can be done by studying various blood cell counts and their properties such as shape, color, size, etc. Filtration and ektacytometry are the standard techniques for the detection of the deformability of RBCs. Teitel [3] worked in the same field, for quantification of RBC deformability. The technique was able to measure the mean deformability of the RBC as a population. As the technique was capable of calculating the population mean deformability it overcame the problem of analysis of individual cell deformability using a rheometer. But, at the same time, the technique was very complex and time-consuming in nature. Di Ruberto et al. [4] used an image processing method for detection and recognition of malaria-infected red blood cells. They included morphological operators and a watershed algorithm for the processing of clustered cells. This technique provided noisy contours of RBCs. In most of the cases, due to color similarities and the complex nature of blood cells, it was difficult to segment RBCs from the cytoplasm. Cai et al. [5] proposed a method to overcome this problem by demonstrating a statistical model-based approach. This technique was used to detect the boundary of the cell and cytoplasm. After segregating the cells from the cytoplasm, for classification of these cells based on certain features, the Bayesian classifier is one of the popular techniques. Ghosh et al. [6] and Sinha and Ramakrishnan [7] used the Bayesian classifier technique for extracting different features of blood cells. These features included average value for color composition, area, and a number of the nuclear lobes to classify the leukocytes. The system achieved 82% accuracy for leukocyte classification. Piuri and Scotti [8] used *K*-nearest neighbor to classifies the leucocytes into basophil, neutrophil, lymphocyte, monocyte, and eosinophil. The accuracy achieved by the system was 70.6%. Tomari et al. [9] proposed a system for classification of normal and abnormal RBCs by studying artificial neural network (ANN) classifier. Their system achieved 82% accuracy.

Methodology

According to Fig. 1 the input was a raw microscopic image received from the hospital. The received images are then pre-processed and segmented to extract single RBCs images. These single RBC images are sub-images of initial raw images. Once we obtained the sub-images, the features were extracted for individual RBC images. The feature extraction technique used three features as shape (Canny edge detector), size (morphological binary operation), and color (mean RGB matrix) of each RBC. After receiving the values of features, the normalization was

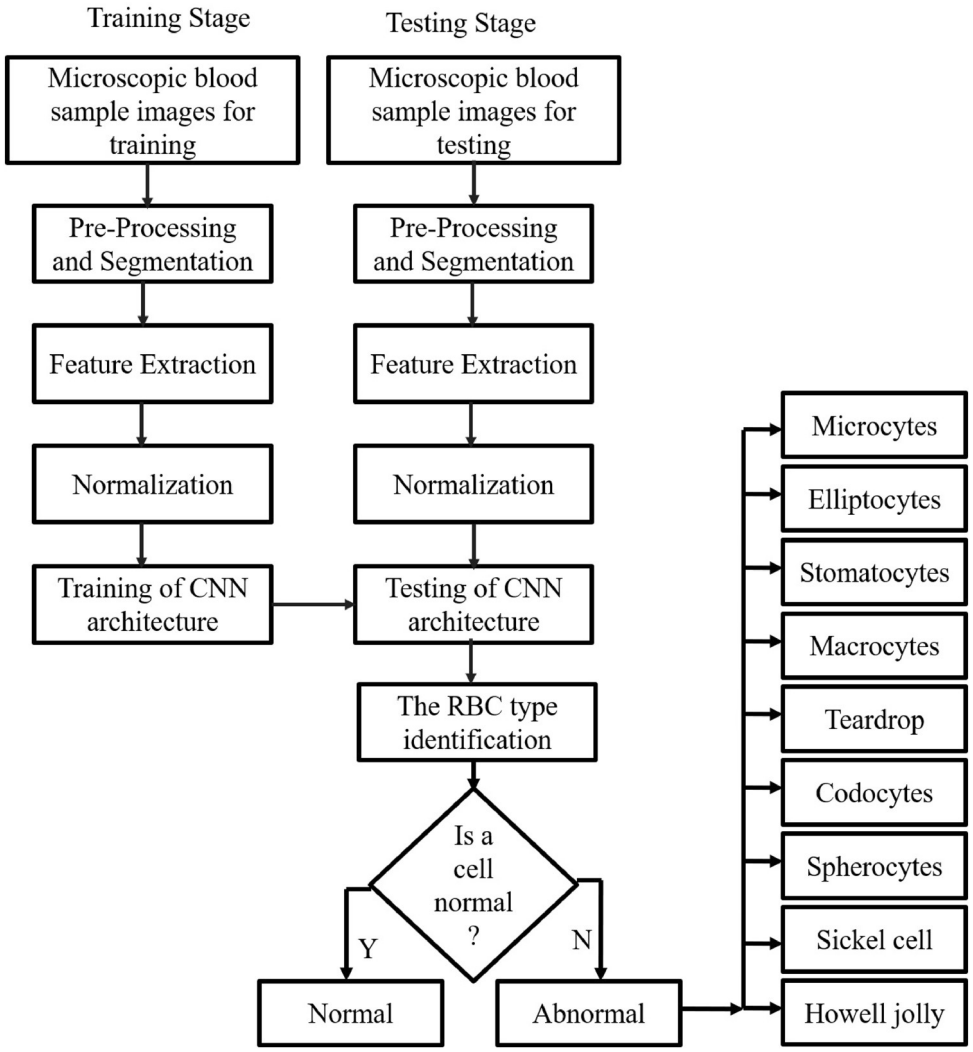

Fig. 1 The system flow diagram. The image was acquired as a raw image from a microscope. The image pre-processing, such as, RGB to gray conversion, edge detection, ROI selection, image segmentation, normalization was performed. Once the image is pre-processed then the image is normalized and features are extracted. The extracted features and the original image was fed to the convolution neural network (CNN) as a tensor input. CNN was trained in such a way that it was able to detect the type of RBC based on its color, shape, and texture features. The classification output was segregating RBCs into 10 classes

carried, and finally, image classification was done. CNN is used to classify the single-segmented images into its respective classes.

Image Pre-processing

The first and essential stage of any computer vision system is image acquisition. In our case, the raw microscopic images were taken from the Olympus IX53 microscope fitted with a USB camera. The acquired blood sample images were captured from a high-resolution of 1080 × 1080 pixels. The acquired images were affected due to small fluctuations in light sources. To improve the dynamic range, contrast stretching was performed on the images. To specify the area taken up by each cell, the image was separated into two regions as foreground and background. Rather than studying and analyzing the entire image, a small part or region of interest (ROI) around the cell was extracted and considered for further processing. Initially, for supervised learning purposes, the ROI was selected manually around the single red blood cell. Due to the extraction of ROI from the entire image, the complexity of the analysis reduced also unwanted parts of the image was removed. ROI ensures that CNN can be applied to the smallest possible size to achieve easier training. Once the ROI has been selected from images, it is important to process the image to improve the quality of it. The image intensity was enhanced by histogram equalization and also edges were determined by the canny edge detector. Canny edge detection defined optimal edge finding as a set of criteria that maximize the probability of detecting true edges while minimizing the probability of false edges. A Canny edge detector uses a Gaussian convolution technique to control the degree of smoothness.

$$g(m,n) = G(m,n) * f(m,n) \tag{1}$$

$$G\sigma = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{m^2+n^2}{2\sigma^2}\right]. \tag{2}$$

The Canny edge detection algorithm calculates the gradient magnitude and direction of each pixel. In this algorithm, the maxima and minima of the first derivative gradient are the same as the zero crossings of the second directional derivative.

$$M(m,n) = \sqrt{g_m^2(m,n) + g_n^2(m,n)}. \quad (3)$$

In this algorithm, the maxima crossings are taken into consideration because these pixels represent the areas of the sharpest intensity in an image. The zero-crossings are the ridge pixels that represent the set of possible edges. All other pixels are termed as non-ridge pixels and subsequently suppressed from an image. Finally, a two-threshold technique is used with ridge pixels to determine the final set of pixels.

Feature Extraction

Feature extraction is one of the important stages of the classification of the RBC image. The shape, color, and texture are the most important features of the RBC cell image. The method used to extract shape as the feature was invariant moments, and the method applied for extracting of the texture feature was gray-level co-occurrence matrix (GLCM). The mean, standard deviation, and skewness of pixel values are used for extracting the color feature of RBC from all three color planes.

CNN Details

CNN is one of the simplest and effective algorithms for image classification. The CNN algorithm receives an image tensor as input and converts it into an array of pixels concatenated with features at the end. The array of such features is called a feature map. The values of the feature map depend on the resolution of the image and the number of features. In the input tensor, we used the RBC image which was normalized to $128 \times 128 \times 3$, Canny edge detected image 128×128 binary image, area, GLCM matrix, and mean R, G, B values. The CNN network is a multilayer algorithm. Convolutional, activation, ReLU, and max pooling are the most common layer of CNN. In our CNN we used 13-layer architecture (Fig. 2). These 13 layers are optimized for the maximum positive prediction rate (sensitivity). The formula for the positive prediction rate is shown in Eq. 7. The first layer after the tensor was a convolutional layer, also termed as moving filters (Fig. 3c). The convolutional layer combine feature map with convolutional filter. Due to this combination, certain features from the image get activated. After convolution, back normalization was performed. After back normalization, the rectified linear units or ReLU (Fig. 3a) was used for faster training. It maps negative values to zero and maintaining positive values as it is. After first set of the convolutional layer (i.e. convolution, back normalization, ReLU), another two layers were used. Finally, max-pooling (Fig. 3b) or subsampling was used. Max-pooling uses

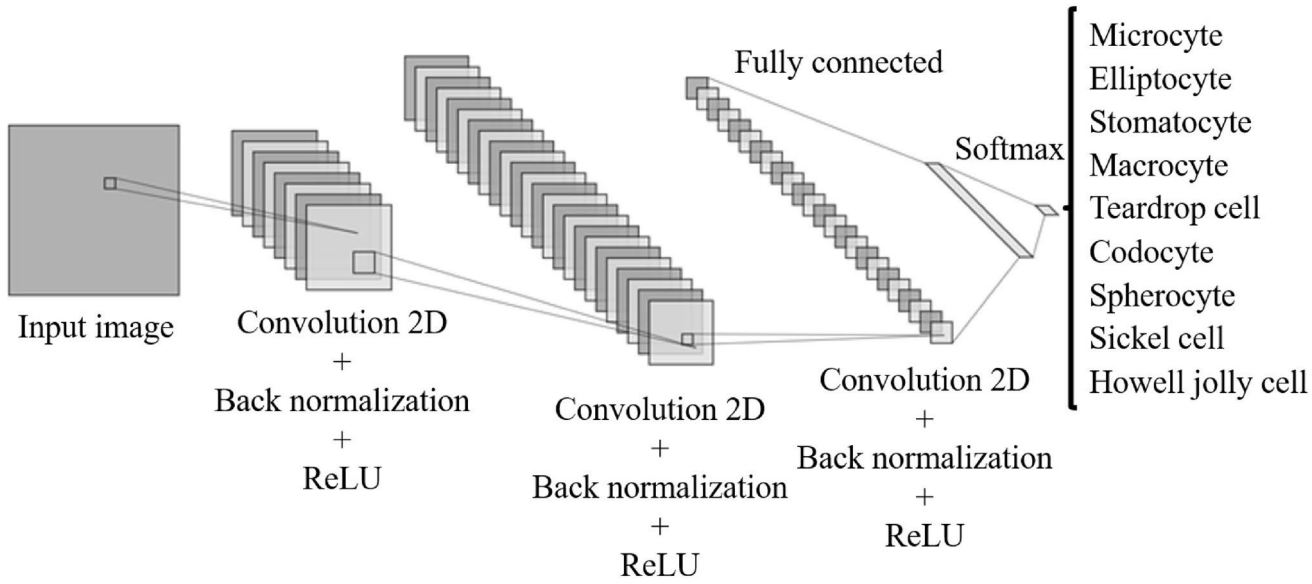

Fig. 2 The input to CNN was tensor with an RBC image and its parts. The CNN algorithm architecture is composed of three convolutional layers with max-pooling, back normalization, and ReLU. After that, there is a fully connected network layer. A final 2-ways Softmax and classification layer provides a probability of normal and abnormal (microcytes, elliptocytes, stomatocytes, macrocytes, teardrop RBCs, codocytes, spherocytes, Sickle cell RBCs, and Howell–Jolly RBCs) per RBC image

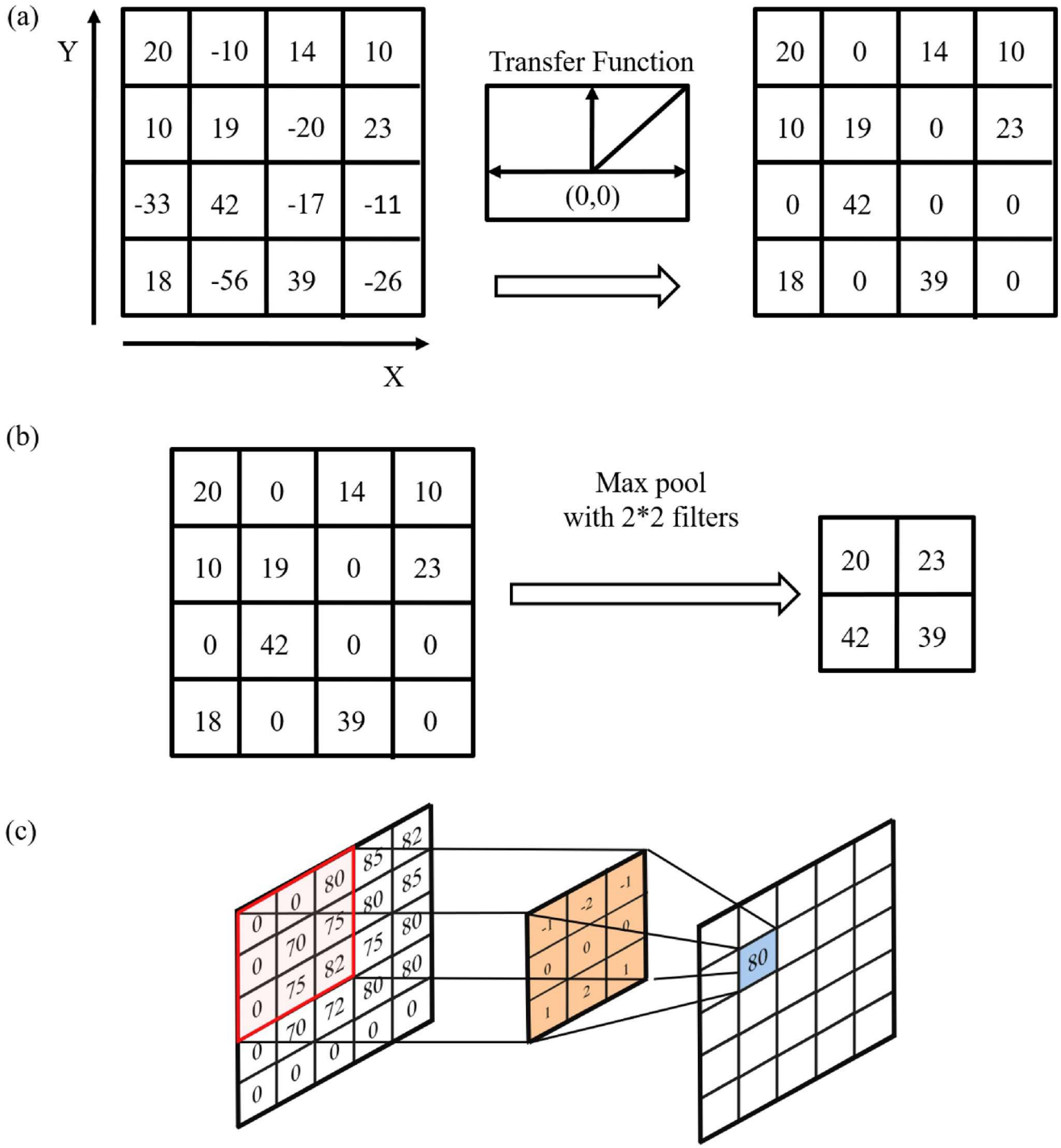

Fig. 3 **a** Rectified linear unit (ReLU) function. It maps the negative values to zero and maintaining positive values as it is. **b** Max-pooling operation. In max pooling operation maximum value from set of 2×2 matrix is selected. **c** Filtering of image. The filter kernel of 3×3 matrix has been considered. Filter convolved with part of input (3×3 matrix) to produce new output matrix

nonlinear downsampling to simplify the output. It reduces the number of the parameter that the network needs to learn. Finally fully connected CNN was used to classify the RBC image tensor into its corresponding class.

Implementation

We divided our dataset into three different parts as a training dataset, validation dataset, and testing dataset. To train CNN we marked one epoch to completed, after 40 iterations.

The input to the network was achieved by combining all the RBC images from the training dataset with a feature vector. After completing the training operation, for validation of the obtained model, we considered 500 images of each class from the validation dataset. The output data in such a case will be 10×500 i.e. we considered 5000 randomly selected labeled images for validation. We used randomly selected images from the testing dataset after validation is complete to cross-verify our results with the pathology lab technician's results.

Evaluation Criteria

We used the confusion matrix for the evaluation of the designed algorithm. Along with that precision, recall, $F1$ score, sensitivity, and specificity were also considered.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{4}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{5}$$

$$F1 = \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6}$$

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{7}$$

$$\text{Specificity} = \frac{\text{True Negative}}{\text{True Negative} + \text{False Positive}}. \tag{8}$$

Time Complexity

The time complexity or execution time is part of the computational complexity of any model and it is determined by the time taken by the model to execute the output for the given input. To predict the time complexity of the proposed system, we considered two important elements: (1) the time taken by the model to execute a single epoch and (2) predicting the number of epochs [10]:

$$\mathcal{T}_b = \sum_{i=1}^{n} B_{M_{(i)}}. \tag{9}$$

Equation 9 is used for calculating the time required for executing one epoch, we first determined the total time taken by forwarding and backward propagation in one batch. In this equation, n is the total batches required to process the complete data:

$$E = pT_b. \tag{10}$$

The total time required for executing a single epoch is determined by Eq. 10. In this equation, p is the batches required to process complete data. We also found that the time complexity of the model also depends on the hardware and software configuration of the machine on which it gets executed. Table 1 describes the hardware and software used by us to execute the proposed approach.

By calculating the time required to execute a single epoch and the number of the epoch, we found that the behavior of the algorithm is time efficient and took approximately 108 ms to run one epoch. In the proposed system, we have considered 9 epochs, so the total time complexity of the system came out as 972 ms for a given hardware configuration. As per big O notations considering the number of loops and iterative approach our time complexity was $O(n2)$.

Results and Discussion

Figure 4a shows the accuracy versus the total number of iteration plot. The total number of iterations is directly proportional to accuracy up to a certain limit. As the total number of iteration goes on increasing the accuracy of our CNN also increases. After reaching 150 iterations, accuracy saturates and becomes steady. After training the algorithm the accuracy of classification reached up to 98.5%. Figure 4b shows the plot of loss versus total numbers iteration. An inverse relationship between loss and the total number of iteration count was observed. As the accuracy and total numbers of iteration increases, the loss decreases exponentially.

As shown in Fig. 5d, the classification accuracy has reached up to overall 98.5%. There are 10 desired classes and 10 predicted classes in the confusion matrix. The total number of RBCs in each desired class was 500 and hence we were expecting all the diagonal elements (i.e. true positive) to be 500 but due to miss-classification, we got variation in the count. Figure 4c–e, depicts some of the failed cases during the classification of various cells. The boundary shown in red is the final detected cell. In Fig. 4c, the model predicted Elliptocyte to teardrop cell. This was caused because of the stress on one side of the cell made it look like a teardrop-shaped RBC. Figure 4d shows cases in which normal RBC was classified are microcyte. There were some cases when the cell does not belong to one particular class. Figure 4e shows one such case. There were 500 images selected randomly from validation image folders from all 10 types of cells. As shown in the table the max True Positive (TP) count of cell type could go up to 497 out of 500. True negative could reach up to 4499 out of 4500. The maximum value of sensitivity, specificity, and $F1$ score was 99.4%, 100%, 99.4% respectively (Table 2).

Figure 4c–e depict some of the failed cases during the classification of various cells. The boundary shown in red

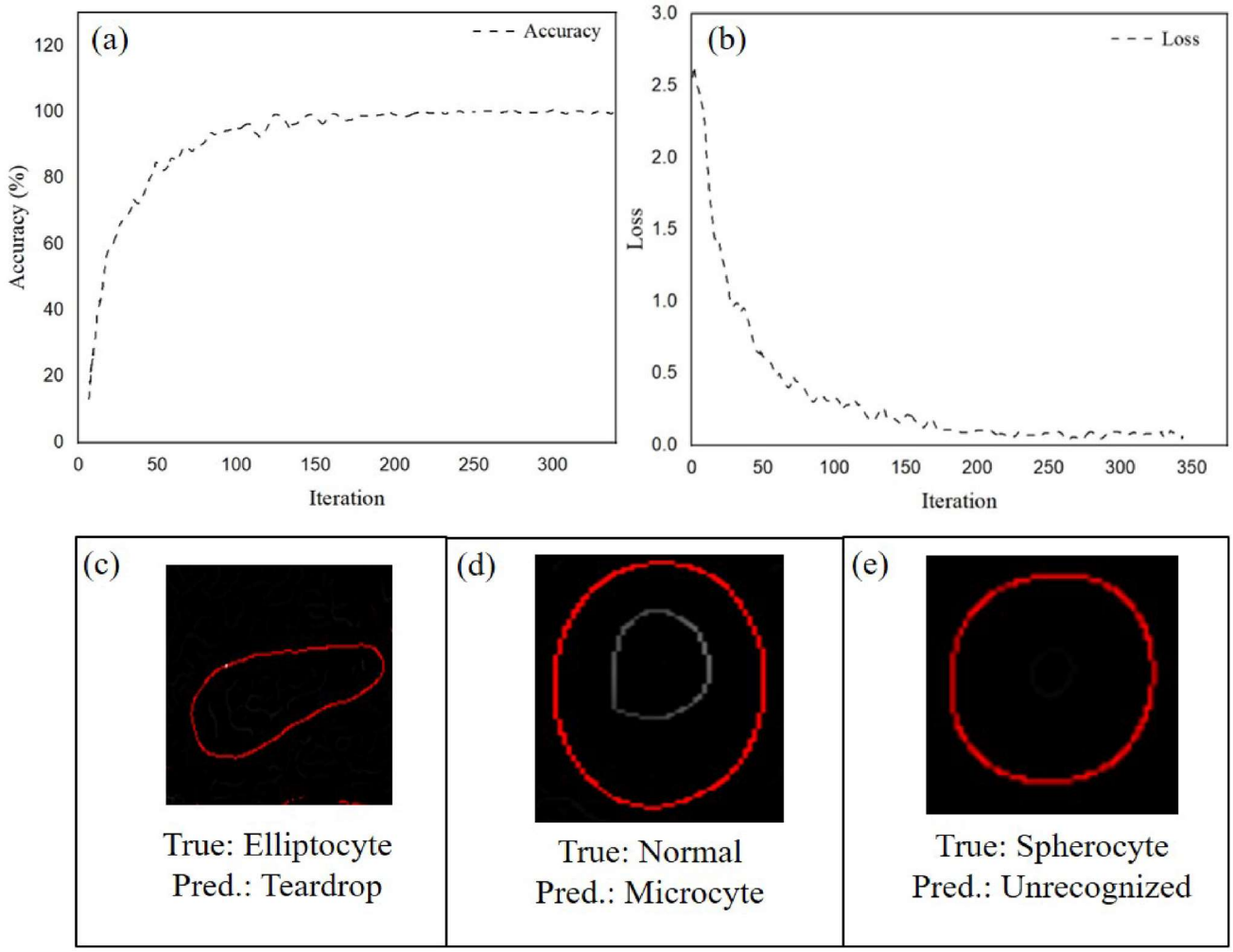

Fig. 4 **a** The accuracy versus the number of iterations plot. The number of iteration and accuracy of CNN are directly proportional. But after a certain iteration, the accuracy reaches its saturation level. We noted that after 5 epochs, the change in accuracy was saturated. **b** Shows the plot of loss versus number iterations. The loss and the total number of iterations are inversely proportional. As the number of iteration increases accuracy increases and loss decreases. **c–e** The miss classification examples of testing data

Table 1 Software and hardware requirements

Hardware and software	Characteristics
Operating system	Windows 10
RAM	24 GB DDR4
GPU	6 GB DDR5
Hard disk	512 GB SSD

is the final detected cell. In Fig. 4c, the model predicted Elliptocyte to teardrop cell. This was caused because of the stress on one side of the cell made it look like a teardrop-shaped RBC. Figure 4d shows cases in which normal RBC was classified are microcyte. There were some cases when the cell does not belong to one particular class. Figure 4c shows one such case.

As shown in Table 3, we have compared the proposed system with different blood cell identification and classification techniques reported in the literature. We found that RCNN + CNN has given the lowest prediction accuracy of about 72% for multi-class classification. Whereas, the adaptive neuro-fuzzy inference system (ANFIS) [12], CNN based you only look once (YOLO) [13], self-organized map [14] and ResNet50 [15] has similar accuracy as our proposed model. To validate and evaluate the performance of the proposed system, we have tested it on different publicly available datasets. We used more than 4k images from the blood cell count database (BCCD) [16] and 626 images from erythrocytesIDB [17]. As shown in Table 4, the accuracy of the proposed system was maximum with the BCCD dataset and minimum for erythrocytesIDB. We have also noted that the accuracy did not vary much across different datasets

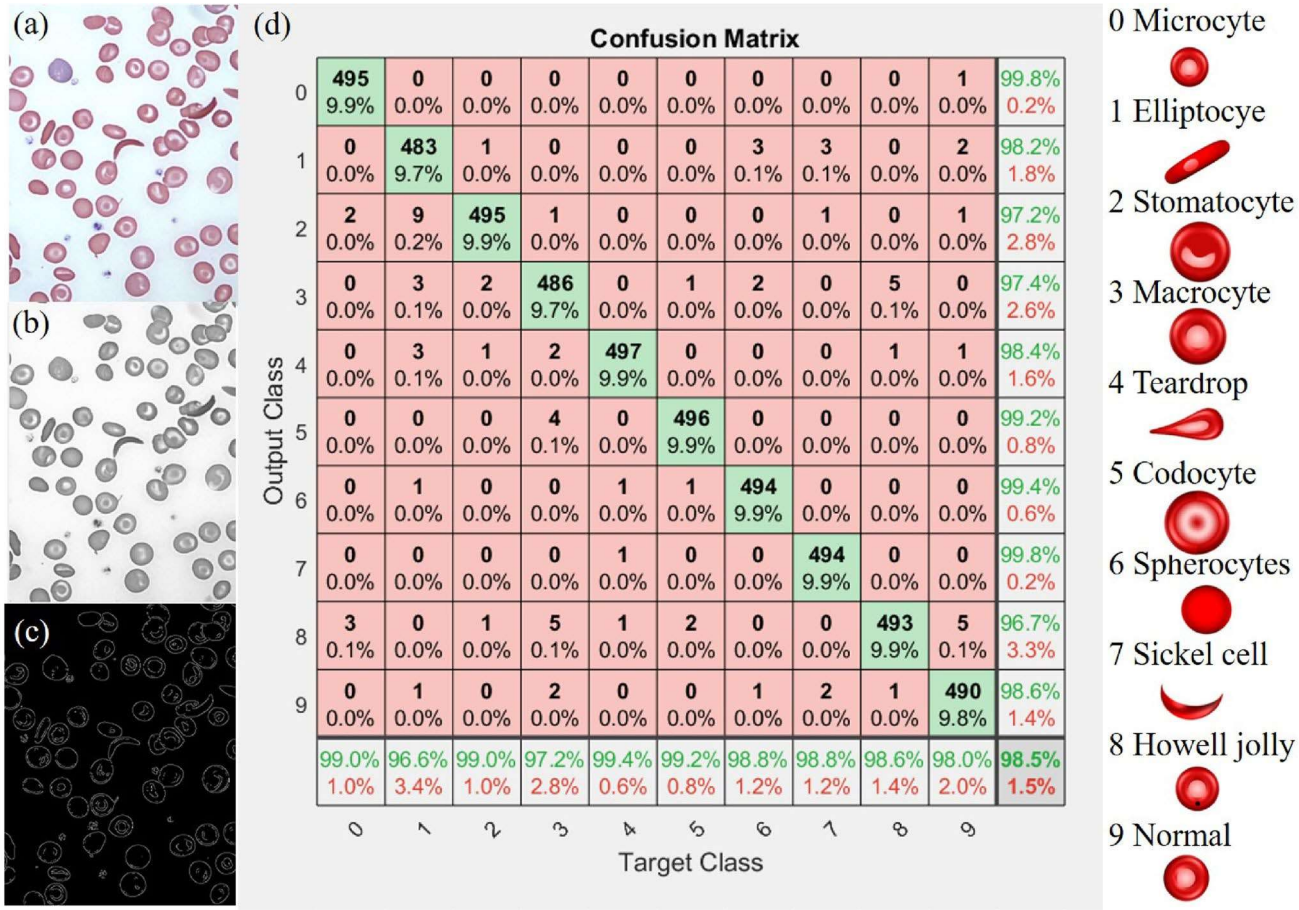

Fig. 5 The confusion matrix obtained after classification. The input blood cell is classified into 10 types including normal RBCs and abnormal RBC of 9 types

Table 2 Evaluation of CNN network

Target class	TP	TN	FP	FN	Precision (%)	Accuracy (%)	Sensitivity (%)	Specificity (%)	F1 Score (%)
Microcyte	495	4499	1	5	99.8	99.9	99.0	100.0	99.4
Elliptocyte	483	4491	9	17	98.2	99.5	96.6	99.8	97.4
Stomatocyte	495	4486	14	5	97.2	99.6	99.0	99.7	98.1
Macrocyte	486	4487	13	14	97.4	99.5	97.2	99.7	97.3
Tear drop	497	4492	8	3	98.4	99.8	99.4	99.8	98.9
Codocyte	496	4496	4	4	99.2	99.8	99.2	99.9	99.2
Spherocyte	494	4497	3	6	99.4	99.8	98.8	99.9	99.1
Sickle cell	494	4499	1	6	99.8	99.9	98.8	100.0	99.3
Howell–Jolly	493	4483	17	7	96.7	99.5	98.6	99.6	97.6
Normal	490	4493	7	10	98.6	99.7	98.0	99.8	98.3

which proves the stability and robustness of the model for unseen RBC images.

Though the performance of the proposed model is robust, and it can be the first step toward a new era of automated histological and pathological diagnosis, there are still a few drawbacks, that we need to consider. One of the major drawbacks of the model is that, it needs domain-specific expertise for feature selection and single-cell extraction during the training phase of the model. To increase the positive prediction rate of the system, we have used the 13 hidden layers to train our model precisely. As the number of the hidden layers increases, the training

Table 3 Accuracy of different techniques for blood cell detection

Method	Total images	Accuracy (%)
RCNN + CNN [11]	313	72
ANFIS [12]	21k	96.6
CNN-YOLO [13]	364	96.09
Self-organizing map [14] neural network	28	93.78
ResNet50 [15]	3.4M	90
Proposed method	5k	98.50

Table 4 Validation results of proposed system on different datasets

Dataset	Image count	Accuracy (%)
BCCD [16]	4884	98.88
ErythrocytesIDB [17]	626	97.14
Our dataset	5000	98.50

weights also increase and it requires a huge processing speed. It also increases the processing time to reach the result during the training.

Conclusions

The RBC classification into multiclass and disease diagnostics from it is still a challenging task and it requires the expert pathologist to predict correct disease just by observation of cells. We tried to develop an image processing and CNN based system that can detect normal and abnormal RBC with an accuracy of 98% and further able to classify different types of abnormal RBCs into 9 different classes with an accuracy of 98.6%. We used RBC cell images along with other features such as shape, size, and texture of RBC. In the further stage of research, we are expecting to develop a deep learning method to classify RBCs from a microscopic image without any preprocessing and we expect to improve the accuracy of the classified results. Overall this approach can be useful not only for RBC but for any multiclass segregation problem.

Acknowledgements The authors would like to thank Dr. Madhura from Kaushalya Medical foundation hospital, Thane, and also colleagues at Ninad's Research Lab.

Compliance with Ethical Standards

Conflict of Interest Authors M. Parab and N. Mehendale declare that they have no conflict of interest.

Involvement of Human Participant and Animals This article does not contain any studies with animals or humans performed by any of the authors. And also this article contains the study on images of human blood samples. All the necessary permissions were obtained from the Institute Ethical Committee and concerned authorities.

Information About Informed Consent Informed consent was obtained from all the human participants whose blood slide images were used for image processing.

References

1. George JN, Woolf SH, Raskob GE, Wasser J, Aledort L, Ballem P, Blanchette V, Bussel J, Cines D, Kelton J, et al. Idiopathic thrombocytopenic purpura: a practice guideline developed by explicit methods for the American Society of Hematology. Blood. 1996;88(1):3.
2. Walker HK, Hall WD, Hurst JW. Peripheral blood smear-clinical methods: the history, physical, and laboratory examinations. Boston: Butterworths; 1990.
3. Teitel Pa. Basic principles of the 'Filterability test'(FT) and analysis of erythrocyte flow behavior. In: Red Cell Rheology, pp. 55–70. Springer, Berlin, Heidelberg, 1978.
4. Di Ruberto C, Dempster A, Khan S, Jarra B. Analysis of infected blood cell images using morphological operators. Image Vis Comput. 2002;20(2):133.
5. Cai R, Qingxiang W, Rui Z, Lijuan F, Chengmei R. Red blood cell segmentation using Active Appearance Model. In 2012 IEEE 11th International conference on signal processing, vol. 3, pp. 1641–1644. IEEE, 2012.
6. Ghosh M, Das D, Chakraborty C, Ray AK. Automated leukocyte recognition using fuzzy divergence. Micron. 2010;41(7):840.
7. Sinha N, Ramakrishnan AG. Automation of differential blood count. In: TENCON 2003 Conference on convergent technologies for Asia-Pacific Region, vol. 2, pp. 547–551. IEEE, 2003.
8. Piuri V, Fabio S. Morphological classification of blood leucocytes by microscope images. In: 2004 IEEE international conference on computational intelligence for measurement systems and applications, 2004. CIMSA., pp. 103–108. IEEE, 2004.
9. Tomari R, Zakaria WNW, Jamil MMA, Nor FM, Fuad NFN. Computer aided system for red blood cell classification in blood smear image. Procedia Comput Sci. 2014;42:206.
10. Justus D, John B, Stephen B, Andrew SM. Predicting the computational cost of deep learning models. In: 2018 IEEE international conference on big data (Big Data), pp. 3873–3882. IEEE, 2018.
11. Qiu, W, Jiaming G, Xiang L, Mengjia X, Mo Z, Ning G, Quanzheng L. Multi-label detection and classification of red blood cells in microscopic images. (2019). arXiv:1910.02672.
12. Khameneh NB, Hossein A, Piruz S, Saeed S. Abnormal red blood cells detection using adaptive neurofuzzy system. In: Mmvr, pp. 30–34. 2012
13. Alam MM, Islam MT. Machine learning approach of automatic identification and counting of blood cells. Healthc Technol Lett. 2019;6(4):103.

14. Rahmat RF, Wulandari FS, Faza S, Muchtar MA, Siregar I. The morphological classification of normal and abnormal red blood cell using self organizing map. In: IOP Conference Series: Mater Science Engneering vol. 308, p. 012015. 2018.
15. Doan M, Sebastian JA, Pinto RN, McQuin C, Goodman A, Wolkenhauer O, Parsons MJ et al. Label-free assessment of red blood cell storage lesions by deep learning. BioRxiv 2018;256180.
16. Singh I, Pal Singh N, Singh H, Bawankar S, Ngom A. Blood cell types classification using CNN. In: International work-conference on bioinformatics and biomedical engineering, Springer, Cham, 2020. pp. 727–738.
17. Ferreira RL, Coelho Naldi M, Fernando Mari J. Morphological analysis and classification of erythrocytes in microscopy images. In: Proceedings of the 2016 workshop de visão computacional, Campo Grande, Brazil, 2016. pp. 9-11