Classifying White Blood Cells in Blood Smear Images using a Convolutional Neural Network

Gulshan Sharma, Rakesh Kumar

Abstract**:** ***We have tried to automate the classification task of white blood cells by using a Convolutional Neural Network. We have divided white blood cell classification in two types of problems, a binary class problem and a 4-classification problem. In binary class problem we classify white blood cell as either mononuclear or Grenrecules. In 4-classification problem where cells are classified into their subtypes (monocytes, lymphocytes, neutrophils, basophils and eosinophils). In our experiment we were able to achieve validation accuracy of 100% in binary classification and 98.40 in multiple classifications.***

Index Terms**:** ***Convolutional Neural Network, Deep Learning, Medical Diagnosis, White Blood Cell Classification.***

I. INTRODUCTION

White Blood Cells are crucial part of our body's immune system. These cells are basically colorless, but with the use of some special stains, we can make them visible to observe under a microscope. We can categorize white blood cells in two main groups mononuclear cells and granulocytes according to the shape of their nucleus. With observing these two groups under a microscope we can further classify them into five subclasses (monocytes, lymphocytes, neutrophils, basophils and eosinophils) on the basis of their size, structure and color. These cells are present in our blood stream in a specific percentage. Whenever this count rises or falls out of the range, it might be an indication to some serious medical condition. When number of white blood cells are fewer than the specified range, condition is known as Leukopenia. In this condition our body becomes more susceptible to infections. Similarly another condition where the number of white blood cells are higher than the specified range is known as Leukocytosis. For effective diagnosis of a patient doctor might need to check patient's white blood cell count periodically.

Hence automating the task of counting and classification of white blood cells will be a very useful tool in medical diagnosis. It will make the procedure faster and less prone to human error. There are already many techniques and algorithms proposed by various researchers to automate the classification and counting task. Most of these algorithms are inspired by the digital image processing, machine learning and deep learning techniques. In this research work we are proposing a convolutional neural network architecture to classify various types of white blood cells and we will be analysing and evaluating the accuracy of our proposed model with the already existing approaches.

II. RELATED WORK

This section reviews the literature for white blood cell classification using the conventional laboratory procedures, digital image processing, machine learning and deep learning techniques.

Manually classifying white blood cells may lead to delay and to many statistical errors like sampling error which will result in poor accuracy of the results and significant level of inter-observer variation in grading [1]. Cytometry and automatic counting machines are used by some laboratories for classification of white blood cells. But these techniques can only quantitatively count and classify the white blood cells not qualitatively [1]. After the intensive research in the domain of digital image processing and machine learning, Several automated techniques have been proposed by researchers for classification of the white blood cell images.

In peripheral blood smear images white blood cells are seperated from the other components of blood, through segmentation process. Through image segmentation and feature extraction plays an important role here in accurate classification. Researchers have experimented with boundary and edge detection, morphological operators and scale space analysis , thresholding techniques with different color band. Various researches on determining the number of white blood cells through various segmentation techniques such as snake and fuzzy c means [3], by converting the color space and incorporating it with Otsu's algorithm [4], by using various machine vision system [5], and k-means clustering algorithm [6]. Manually extracting features asks for human expertise in analysis of image and its various characteristics. To find out various parameters associated with the region of interest is a complicated and tiresome process. Deep learning techniques addresses these issues very effectively.

Convolutional Neural Networks have proven themselves as an effective and efficient deep learning technique for classification of image data [10]. Many researchers have experimented with many well-known convolutional neural networks like GoogLeNet, AlexNet and LeNet5. The outcome of these and other similar studies points to the idea that convolutional network based deep learning techniques have shown good results in white blood cell classification [11-17]. Hence proposing a dedicated deep learning model which will qualitatively classify the white blood cells is an essential need of the time.

Revised Manuscript Received on June 29, 2019.

Gulshan Sharma, Computer Science & Engineering, National Institute of Technical Teachers Training & Research, Chandigarh, India.

Rakesh Kumar, Computer Science & Engineering, National Institute of Technical Teachers Training & Research, Chandigarh, India

III. METHOD

This section describes our proposed model to classify the different tyoe of white blood cells present in the blood smear images. Our proposed model is comprised of dataset acquisition, preprocessing of images followed by training our custom convolutional neural network. After this we are analysing the obtained results and theire accuracy.

A. Dataset

To implement and analyse our proposed approach, we have collected a publicly available WBC dataset [18]. This dataset contains 364 colored microscopic images of dyed white blood cells. These are 3-channeled colored images and all images are of size 640 X 480 in JPEG format. On exploring more we found that there were 207 Neutrophil, 88 Eosinophil, 33 Lymphocyte, 21 Monocyte and 3 Basophil images. Since there were only 3 of Basophil images in the available dataset. We have ignored them in our experimentation for a generalized solution.

B. Image Preprocessing

Images are susceptible of various types of noises. To deal with these noises preprocessing techniques are used. Following are the preprocessing techniques we have applied to our dataset.

1) **Image Resize:** To fed into the proposed architecture images are downsized to 128 X 128. Format of image is kept JPEG. Downsizing image size helped us in reducing the training and testing time significantly.
2) **Noise Reduction**: Though pepper and salt type noise is pretty common in images. This type of noise is caused by some sudden and sharp disturbances in the image signal. To avoid this noise a non-linear digital filter called 2-dimentional median filter is applied on each of the white blood cell image present in the dataset. The neighbor hood size is taken of 2x2 with zero padding.
3) **Image Augmentation:** Though the size of available dataset is not enough to take advantage of deep learning techniques. Hence we are applying image augmentation techniques(sucessive 90 degree rotaions, fliping along both axis and shearing by angle of 30 degrees) to sufficiently increase the size of the available dataset. After applying augmentation operations size of dataset is increased to 12,439 images. For 4-classification problem these augmented white blood cell images are categorized into 4 different folders according to their cell type (Monocyte, Lymphocyte, Eosinophil and Neutrophil). On an average there are approximately 3,109 augmented images in each folder. For binary classification problem we have categorized these augmented images into 2 different folders according to their cell type (Mononuclear or Granulocytes).

C. Proposed Model

Our proposed Convolutional Neural Network architecture is consists of the five 2d-convolutional layers. 128 X 128 sized coloured images are fed into it. First four convolutional layers are followed by batch normalization layer, leaky relu layer and maxpool layers. In fifth Convolutional layer, instead of maxpool we have applied fully connected layer with softmax as a classifier. We are using small sized convolutional filters at initial layers, this helps in preserving the robust features. The number of convolutional filters are also small in initial layers of our architecture and this number keeps on increasing gradually. Maxpool layer are used to downsample, which will diminish the spatial size of convolutional layer output. A dropout layer is applied before fully connected layer to avoid overfitting problem. The probability of dropout layer is set to 0.5 in our design. The comprehensive framework of our proposed model is described in Table I below.

I. PROPOSED ARCHITECTURE

Convolutiona l Layer	*Filte r Size*	*No. of Filter*	*Input Feature Size*	*Strid e*	*Dilatio n Factor*	*Paddin g*
Convo1	3x3	6	128x128	1x1	1x1	1
Convo2	5x5	16	64x64	1x1	1x1	0
Convo3	5x5	64	30x30	1x1	1x1	0
Convo4	5x5	128	13x13	1x1	1x1	0
Convo5	4x4	512	5x5	1x1	1x1	0

Convolutional layer: It is the most important layer of any convolutional neural network. This layer applies a set of convolution filters parallely on the images and each of these filters activates certain specific features. Convolutional filters of different sizes are slided in a particular sequence on the feature map obtained from the previous layer and it generates a new feature map for next layer.

Batch Normalization: Stability of a neural network is increased through this layer, Previous layer output is normalized by subtracting batch mean, and then dividing it by standard deviation of the batch. This is followed by scaling and shifting of activation outputs by some random parameters.

Leaky Relu: It's an enhanced form of a popular activation function called Rectified Linear Unit. It fixes the dying ReLU problem. The derivative of the Leaky ReLU is a small fraction in the negative part and one in the positive part.

Maxpool: It reduces the feature dimensionality and select the strong features for further processing. Max pooling also provides translation or rotation invariance to the system. Pooling can be max pool, min pool and average pool, but in our Convolutional Neural Network configuration we have used the max pooling with a stride of 2 and 0 padding.

Dropout: This layer maintains a strong resistance to overfitting and improve generalization ability in a deep learning neural networks by randomly dropping the nodes with a predefined probability. In our Convolutional Neural Network configuration we have used dropout probability of 50%.

Softmax: In a multi-class problem, softmax assigns decimal probabilities to each class. Though total sum of the probabilities will be equal to one and the target class will be having highest probability among other classes. In practice, it is implemented just before the output layer in a neural network.

D. Network Training Parameters

To train our proposed model we have randomly chosen the 80% data as the training set and 10% of the data for validation. Rest 10% data is left as the testset of our proposed model. We have executed our model on a single CPU environment having MATLAB version 2019a installed on it. The learning rate schedule is kept constant and Adam method is used as the training optimizer. Various learning parameters of our proposed architecture are shown in Table II below.

II. Network Hyper Parameter

Hyper Parameter	*Value*
Initial Learning Rate	0.001
Gradient Decay Factor	0.1
Max No of Epochs	10
Mini Batch Size used	128
Validation Frequency used	10

IV. MODEL RESULTS

The results obtained from our proposed architecture is presented over here. There are various kind of metrics available for evaluation of the performance of a given model. The choice of selection of evaluation metric is completely depends upon the type of model being used. To show the performance of our proposed architecture following parameters are used.

1) **Confusion Matrix:** It is used to delineate and measure the classifier's performance on the test data set. Confusion matrices obtained from our proposed architecture for Binary Classification and for Muti-classifiction is shown in Table III and IV respectively.

III. Confusing Matrix for Binary Classification

	Mononuclear Cells	*Granulocyte Cells*
Mononuclear Cells	**612**	2
Granulocyte Cells	1	**629**

IV. Confusing Matrix For Multi-Classification

	Eosinophils	*Lymphocyte s*	*Monocytes*	*Neutrophil s*
Eosinophils	**304**			9
Lymphocytes		**311**		
Monocytes			**309**	
Neutrophils	10	3		**304**

2) **Sensitivity:** Also known as True Positive Rate. This parameter calculates the comprehensiveness of a classifier, it tells that number of samples(images in our case) which truly belongs to a class are predicted correctly. To calculate Sensitivity we divide number of true positive predictions by the total sum of false positive and true positive predictions. Sensitivity values obtained from our proposed architecture for binary classification and for multi-classifiction is shown in table V and VI.
3) **Specificity:** This parameter tells that number of samples which don't belong to the class are predicted correctly. To calculate Specificity we divide number of true negative predictions by the total sum of false positive and true negative predictions. Specificity values obtained from our proposed architecture for binary classification and for multi-classifiction is shown in table V and VI below.

V. Sensitivity and Specificity for Binary Classification

For Binary Classification	*Sensitivity(%)*	*Specificity(%)*
Mononuclear Cells	99.8%	99.7%
Granulocyte Cells	99.7%	99.8%

VI. Sensitivity and Specificity for Multi-Classification

For Multi-Classification	*Sensitivity(%)*	*Specificity(%)*
Eosinophils	96.4%	97.1%
Lymphocytes	99.0%	100%
Monocytes	100%	100%
Neutrophils	97.1%	95.9%

4) **Test and Validation Accuracy:** For evaluation of the performance of a model, accuracy is measured on the validation set. It measures the quality of proposed model, like with what accuracy a model is able to make predictions based on unknown data. Test Accuracy and-Validation Accuracy along with Validation loss values obtained from our proposed architecture for Binary Classification and for Muti-classifiction is shown in table VII below.

VII. Accuracy for Binary and Multi-Classification

	Test Accuracy (%)	*Validation Accuracy (%)*	*Validation Loss(%)*

For Binary Classification	99.76%	100%	0.31%
For Multi-Classification	98.14%	98.40%	4.86%

5) Computational Speed: We have executed our convolutional neural network architecture on a single CPU system. The hardware configuration of our system are Fourth generation intel i7 processor CPU, installed with 8GB of RAM and 500GB of HDD. There were no GPU or other high performance computational hardware connected with our system. The average training time taken by our model for binary classification and multi-classification is 0.21 seconds approximately.

V. DISCUSSION

In this section we are comparing the results made by our approach with few existing methods proposed by various other researchers for classifying white blood cells. Comparison is shown in table VIII below.

VIII. COMPARISON WITH EXISTING MODELS

Approtch	***Accuracy(%)***
Ongun, et al. [9]	91%
W. Tai et al. [10]	95%
Liang et al. [3]	90.78%
Manik et al. [8]	98.9%
Macawile et al. [19]	96.63%
Our Proposed Model	**98.40%**

VI. CONLUSION

Our experimental studies have shown that the proposed model has achieved improved performance in terms of validation accuracy, specificity and sensitivity. Our model has scratch architecture which extracts robust features and softmax classifier efficiently classify in one of the classes present. The batch normalization layer along with adam optimization technique has improved the convergence rate with fewer computations. This method has given comparatively higher accuracy than other existing methods.

VII. FUTURE SCOPE

This method needs to be applied and tested on more challenging datasets of blood cells or other medical images datasets, which needed to be classified by using deep learning approaches.

REFERENCES

1. S. H. Rezatofighi, H. Soltanian-Zadeh, “Automatic Recognition of Five Types of White Blood Cells in Peripheral Blood,”Computerized Medical Imaging and Graphics, vol. 35, no. 4, 2011, pp. 333-343.
2. G. Liang, H. Hong, W. Xie and L. Zheng, "Combining Convolutional Neural Network With Recursive Neural Network for Blood Cell Image Classification," in IEEE Access, vol. 6, 2018, pp. 36188-36197.
3. D.-C. Huang and K.-D. Hung, "Leukocyte Nucleus Segmentation and Recognition in Color Blood-Smear Images," in IEEE International Instrumentation and Measurement Technology, 2012.
4. Putta madegowda and Prasanna kumar, "White Blood cell segmentation using Fuzzy C means and snake," in 2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS), 2016.
5. R. Ahasan, A. U. Ratul and A. S. M. Bakibillah, "White Blood Cells Nucleus Segmentation from Microscopic Images of strained peripheral blood film during Leukemia and Normal Condition," in 5th International Conference on Informatics, Electronics and Vision (ICIEV), 2016.
6. Z. K. K. Alreza and A. Karimian, "Design a new algorithm to count white blood cells for classification Leukemic Blood Image using machine vision system," in International Conference on Computer and Knowledge Engineering (ICCKE 2016), 2016.
7. S. Manik, L.M. Saini, N. Vadera, Counting and classification of white blood cell using Artificial Neural Network (ANN), in IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), IEEE (2016).
8. G. Ongun, et al., An automated differential blood count system. Engineering in Medicine and Biology Society, in2001 Proceedings of the 23rd Annual International Conference of the IEEE,vol. 3. IEEE (2001).
9. W. Tai, R. Hu, H. C. W. Hsiao, R. Chen and J. J. P. Tsai, "Blood Cell Image Classification Based on Hierarchical SVM," 2011 IEEE International Symposium on Multimedia, Dana Point CA, 2011, pp. 129-136.
10. LeCun, Yann & Bengio, Y & Hinton, Geoffrey. (2015). “Deep Learning,” Nature. 521, 2015. pp. 436-44.
11. Mehdi Habibzadeh, Mahboobeh Jannesari, Zahra Rezaei, Hossein Baharvand, Mehdi Totonchi, "Automatic white blood cell classification using pre-trained deep learning models: ResNet and Inception," Proc. SPIE 10696, Tenth International Conference on Machine Vision (ICMV 2017)
12. H. Chang, “Skin cancer reorganization and classification with deep neural network,” arXiv preprint arXiv:1703.00534 (2017).
13. Y. Liu, K. Gadepalli, M. Norouzi, et al., “Detecting cancer metastases on gigapixel pathology images,” arXiv preprint arXiv:1703.02442 (2017).
14. M. Aubreville, C. Knipfer, N. Oetter, et al., “Automatic classification of cancerous tissue in laser endomicroscopy images of the oral cavity using deep learning,” arXiv preprint arXiv:1703.01622 (2017).
15. A. Cruz-Roa, H. Gilmore, A. Basavanhally, et al., “Accurate and reproducible invasive breast cancer detection in whole-slide images: A deep learning approach for quantifying tumor extent,” Scientific Reports 7, 46450 (2017).
16. D. Wang, A. Khosla, R. Gargeya, et al., “Deep learning for identifying metastatic breast cancer,” arXiv preprint arXiv:1606.05718 (2016).
17. K. Sirinukunwattana, S. E. A. Raza, Y. W. Tsang, et al., “Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images,” IEEE Transactions on Medical Imaging 35, 1196 (2016)
18. GitHub - Shenggan/BCCD_Dataset: BCCD Dataset is a small-scale dataset for blood cells detection. BCCD Dataset is under MIT licence. [Online]. Available: https://github.com/Shenggan/BCCD_Dataset
19. M. J. Macawile, V. V. Quiñones, A. Ballado, J. D. Cruz and M. V. Caya, "White blood cell classification and counting using convolutional neural network," 2018 3rd International Conference on Control and Robotics Engineering (ICCRE), Nagoya, 2018, pp. 259-263.

AUTHORS PROFILE

Gulshan Sharma received his Bachelor of Technology in Computer Science and Engineering from Chitkara University, Himachal Pradesh. At present, he is pursuing Master of Technology in Computer Science and Engineering from NITTTR, Chandigarh. His research interest includes Discrete Mathematics, Theory of Computation, Deep Learning, Digital Image Processing, Multimedia Systems and Music Information Retrieval.

Dr. Rakesh Kumar received his Bachelor of Technology in Computer Science and Engineering from Punjab Technical University, Jalandhar in 2004. Master of Technology in Information Technology from Guru Gobind Singh Indraprastha University, New Delhi in 2008. Ph.D in Computer Engineering from National Institute of Technology, Kurukshetra in 2015. At present, he is working as an Assistant Professor in the Department of Computer Science and Engineering at the NITTTR, Chandigarh. His research interests includes Discrete Mathematics, Cloud computing, Data Mining, Software Testing with special focus on Mobile Adhoc Networks and Wireless Sensor Networks.