Classification of Leucocytes Using Deep Learning

Suganthi. N¹, Preethi. V², Swetha. K³ and Kannan. K⁴

¹CSE department, Kumaraguru College of Technology, Coimbatore, Tamilnadu, India

^{2,3,4}Information Technology, Kumaraguru College of Technology, Coimbatore, Tamilnadu, India

ABSTRACT

This work is about categorization of white blood cells (WBC) or leucocytes has become highly crucial for the analysis of anaemia, leukaemia and many other hematologic diseases. Our immune system is dependent on the WBC concentration in our blood. The aim of this paper is towards developing a computerized WBC category system using deep learning. Many models proposed for this application so far has used transfer learning by fine tuning the ResNet, Inception and VGGNet. But all these models were trained on ImageNet dataset which is completely different from the dataset used in this application. So, we have proposed a deep learning model for the white blood cells classification task from the scratch, without using transfer learning.

KEY WORDS: CONVOLUTION NEURAL NETWORK, CONVOLUTION LAYER, LEUCOCYTES, POOLING LAYER, WHITE BLOOD CELLS.

INTRODUCTION

In recently, hematology analysis has been a significant role in diagnosing many fatal diseases. There are many causes for an elevated leucocyte count like overproduction, or body issuing WBC early from the bone core. Sometime in worst case, new WBC, so-called blasts, accumulate in infected area. Some reasons of an enhanced leucocytes amount include stress, serious infections, bone marrow injury or illnesses like aplastic anemia, bone marrow. There are several kinds of leucocytes in our blood. In the blood test, the different proportion of these different leucocytes indicate several health conditions. Mostly the ratio of these leukocytes remains steady in blood, any disorder of percentage of WBC is a very consistent evidence to indicate that the blood sample is drawn from an ill patient.

MATERIAL AND METHODS

American and Japanese companies created WBC identification methods in 1970's. The image recognition issue of WBC was examined. It is used owing to more price and minimal correctness (Mazin Z et al 2017). A system (Yang X et al 1994) founded on mathematical morphology (TSMM) were created. The authors (Ushizima D M et al 2005) studied the option of using SVMs in identifying various kinds of leukocytes. Another proposal (Gaobo Liang et al 2018) suggested a novel discovery process built on cellular neural networks. This work was mostly assessed with a reduced amount of data, thus lacked good generalization capability.

A framework (Shubham Manik 2016) is proposed to increase detection and classification of leucocytes using image segmentation and nucleus improvement techniques to extract cell features. The above is fed into artificial neural networks for estimate (Alex Krizhevsky 2010). Recently, an automated leucocytes classifier method has been proposed which uses transfer learning using various state of the art models and use voting module to predict the class with maximum score in all the models (Wei Yu 2017). Another proposed prototype WBCNet (Ming Jiang 2018) that can get aspects of the microscopic WBC

ARTICLE INFORMATION

*Corresponding Author: suganthi.n.it@kct.ac.in

Received 20th Oct 2020 Accepted after revision 12th Dec 2020

Print ISSN: 0974-6455 Online ISSN: 2321-4007 CODEN: BBRCBA

Thomson Reuters ISI Web of Science Clarivate Analytics USA and Crossref Indexed Journal

NAAS Journal Score 2020 (4.31) SJIF: 2020 (7.728)

A Society of Science and Nature Publication,
Bhopal India 2020. All rights reserved.

Online Contents Available at: <http://www.bbrc.in/>

Doi: <http://dx.doi.org/10.21786/bbrc/13.11/25>

picture by merging batch normalization process, residual convolution design and enhanced activation function.

The dataset is taken from kaggle datasets (Paul Mooney 2018). The initial data set contains of 352 pictures (RGB) of WBC of resolution 640x480. It is composed of 3 Basophils, 88 Eosinophils, 33 Lymphocytes, 21 Monocytes and 207 Neutrophils shown in figure 2. Since the count of basophils is very limited in our dataset, we have ignored it to propose a generalizing model. Since the dataset is imbalanced with varying number of samples for each class, data augmentation techniques are used. Thus, the final dataset is 12444 pictures composed of 3120 Eosinophils, 3103 Lymphocytes, 3098 Monocytes, and 3123 Neutrophils. These augmented images are of resolution 320x240. The sample augmented images used in training are shown in figure.1.

Figure 1: Microscopic image samples of four types of WBCs

Figure 2: Data distribution

The dataset is comprised of two folders –train (9957 samples) and test (2487 samples). The train data is split into two for training (80%) and validation (20%). Training data consists of 7965 samples with 1998

Eosinophils, 1986 Lymphocytes, 1982 Monocytes and 1999 Neutrophils. Validation data consists of 499 Eosinophils, 497 Lymphocytes, 496 Monocytes and 500 Neutrophils. The test data consists of 623 Eosinophils, 620 Lymphocytes, 620 Monocytes and 624 Neutrophils. The model does not require any intensive preprocessing techniques. The images are reshaped to resolution 120x120 with three channels. Then they are feature normalized by dividing each pixel value by 255 for all channels. The labels are encoded and then converted to categorical one-hot labels which are used throughout the model.

CNN most popular deep learning architectures. CNN design is highly invariant to translation, tilting, scaling etc. It also avoids the difficult preprocessing of images prior to training and it is an end to end process. Because of these reasons, CNN has highly gained the attention. There are three main characteristics of CNN are local convolution, weight sharing (same weights learnt are shared to detect features throughout the image) and multi-kernels of convolution. The different kernels in CNN help detect different features. One of the most important aspects of CNN is that a convolution layer needs to learn relatively very small number of weights compared to that learnt in a conventional fully connected feed forward neural network. This requires comparatively less memory to save the weights. Thus, the features learnt in convolution layer are fed further to traditional fully connected layers to generate output. The outcome shows the possibility that a picture be appropriate to one of multiple classes using activations like sigmoid (binary classification) or softmax (multi-class classification).

The design of this model is summarized in the Figure.3. It consists of six learned layers – three convolution layers and three connected layers. It also consists of three pooling layers stacked between convolution layers. The convolution layers use very small receptive fields of 3x3 kernels instead of using large receptive fields. These small kernels stacked up give better results compared to models using single large receptive field. Thus, we have used 3x3 kernels for all convolution layers in our model. The weights to be learnt in the model are less in convolution layers compared to those in dense layers.

We have adapted an interesting idea for stacking up convolution layers from VGG16 network (Karen Simonyan 2015). The convolution layers are stacked up in such a way that the dimension of image reduces as it passes through multiple layers and the number of channels increase with increase in number of filters used in the convolution layers. This helps model to learn more complex features in the later layers. To prevent convolution layers from overfitting, we have used small regularizers in second and third convolution layers. Throughout the model, we have used max-pooling layers with kernel size 2x2 and stride 2 to reduce the image size by half as it passes the convolution layers. This layer also allowed model to ensure the features predicted in convolution layers by taking maximum values for each pool area (2x2) across the image.

Most important features to be extracted by the convolution layers are the cell boundary and the shape of the nucleus. Thus, we have used 'same' padding in all convolution layers to ensure that the boundary information is not lost as the image passes through the layers of the model. Relu non-linearity has been used in all convolution layers and two dense layers. This activation helped the model to avoid oscillating in undesirable local optima or vanishing gradients problem.

$$f(x) = \max(0, x)$$

The last dense layer uses softmax activation to classify the leucocytes out of four different classes. The output of softmax is the probability of prediction for four classes and the predicted class is the one with maximum probability.

$$\sigma(x_j) = \frac{e^{x_j}}{\sum_i e^{x_i}}$$

The three dense layers are stacked up in such a way that the number of hidden units in each layer decrease in order. 97% of the network parameters are in dense layers. This over fitted the training data and reduced the generalization in validation and testing. Thus, we have used dropout technique in dense layers to allow them to learn less dependent on the incoming activations.

RESULTS AND DISCUSSION

We conducted this experiment in Google colab jupyter notebook on GPU(K80). We coded the model using keras library with tensor flow as backend. We trained different models with slight changes in architecture and got different results. During training, we used Data augmentation technique using keras Image Data Generator which applies simple transformations on images like shifting and flipping. We trained images on batches with batch size-32 in all models. 'Xavier uniform initializer' is used for initializing the weights in convolution and dense layers. Increasing or decreasing the learning rate did not perform well even in training. We shuffled the data randomly before training and before each epoch which prevented the model from memorizing the predictions based on the order of images trained. The evolution of the proposed model (model-6) is given in the figure 4 and 5. Images with different dimensions (160x120) where used in model-1, while the other models where trained on 120x120 images to ensure uniform padding in all convolution layers. But reducing image size reduced the accuracy of model-2 in both training and testing. Model-1 and Model-2 used 'valid' convolution which reduced image size across the convolution layers which is undesirable for this task. So, we used 'same' convolution to retain the image size in all layers for the rest of the models. Model-3 gave better accuracy using 'same' convolution.

Model-4 is the largest model which uses third convolution layer with 128 filters followed by dense layer with 128

activation units. This model outperformed the other models on training but did not generalize well on testing which clearly shows that the number of parameters learnt over fit the available training data and did not perform well on unseen data. So, we applied batch normalization for convolution layer with 128 filters and dense layer with 128 activation units. This improved accuracy of the model but did not perform well compared to previous model. This model also produced lot of fluctuations while training and created many unwanted peaks in accuracy. We also applied decay for learning rate but it did not help to improve the accuracy after few epochs.

Figure 4: Loss Curves

Figure 5: Accuracy Curves

The model was oscillating in local optima for long time without any improvement in accuracy. Model-5 which is close to our final model has less parameters compared to model-4. It uses regularization in a convolution layer and dropouts in dense layer. This model performed like model-4 on training and it also gave good accuracy in

validation. But it gave very less accuracy on testing. Though validation data is unseen data for the model, it over fit the validation data and so it failed to perform well on test data. Model-6 which is our final model gives very good accuracy in training and generalizes well on validation and testing. It uses dropouts like model-5 and uses regularization in two convolution layers with regularization constant-0.01.

Figure 6: Comparison between different model accuracies

Figure 7: Confusion Matrix

	Eosino phils	Lympho cytes	Mono cytes	Neutro phils
Eosinophils	531	14	0	78
Lymphocytes	0	620	0	0
Monocytes	0	0	580	40
Neutrophils	53	1	12	558

The metrics measured are accuracy (Top_1 and Top_2) and loss. The graphs above show the loss and accuracy obtained during training and validation of our final model. The model gives accuracy of 97.0% on training and 96.7% on validation. The model generalizes very well on test set with correctness rate of 92.0%. The confusion matrix for prediction on the test dataset is given below in figure.7. Our model performs with 100% accuracy in predicting Lymphocytes and Monocytes with accuracy of 93.5%. This is followed by Neutrophils 89.4% and Eosinophils (85.2%). All of the misclassified monocytes and most of the eosinophils are confused by the model to be neutrophils. The model is trained for 120 epochs. Further training the model makes it to memorize the training data and reduces the generalizing capability of the model.

CONCLUSION

The proposed model can be further improved by training on more data and using large models. Mode-6 suffered bias when trying to regularize overfitting (bias –variance trade off). This can be avoided if we train very deep models. But large volume of data is required to develop such models. Thus in future, more accurate model can be proposed to avoid labor intensive manual white blood cell classification.

REFERENCES

- Alex Krizhevsky ,Sutskever, Geoffrey E. Hinton (2010) " ImageNet Classification with Deep Convolutional Neural Networks(AlexNet)", ImageNet LSVRC-2010 contest, NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1,Pages 1097-1105
- Gaobo Liang et al (2018) "Combining Convolutional Neural Network With Recursive Neural Network for Blood Cell Image Classification", Special Section On Trends, Perspectives And Prospects Of Machine Learning Applied To Biomedical Systems In Internet Of Medical Things, Vol – 6, page: 36188 – 36197
- Karen Simonyan_ & Andrew Zisserman (2015) "Very deep convolutional networks for large-scale image recognition (VGGNet)", ICLR-2015, arXiv:1409.1556
- Mazin Z. Othman,Thabit S. Mohammed,Alaa B. Ali (2017) " Neural Network Classification of White

Blood Cell using Microscopic Images”, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol.8, Issue.5

Ming Jiang et al (2018) White Blood Cells Classification with Deep Convolutional Neural Networks, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 32, No. 09, 1857006

Paul Mooney (2018) Blood Cell Images dataset-12,500 images: 4 different cell types, version-6

Shubham Manik; Lalit Mohan Saini and Nikhil Vadera (2016) "Counting and Classification of White Blood Cell using Artificial Neural Network (ANN)", 1st IEEE International Conference on Power Electronics.

Intelligent Control and Energy Systems ICPEICES-2016), doi-10.1109 /ICPEICES. 2016.7853644

Ushizima D.M, A. C. Lorena, and A. de Carvalho (2005) "Support vector machines applied to white blood cell recognition," in International Conference on Hybrid Intelligent Systems, 2005, p. 6 pp.

Wei Yu et al (2017) "Automatic Classification of Leukocytes Using Deep Neural Network", 2017 IEEE, 12th International Conference on ASIC (ASICON), pp. 1041-1044

Yang .X , L. Luo, and Y. Wei (1994) "Automatic classification system for leukocytes in human blood," Chinese Journal of Computers, vol. 17, no. 2, pp. 130-136.