Blood Cell Types Classification Using CNN

Ishpreet Singh, Narinder Pal Singh, Harnoor Singh, Saharsh Bawankar(✉), and Alioune Ngom

School of Computer Science, University of Windsor, 5113, Lambton Tower, Windsor, ON N9B 3P4, Canada
{singh1vb,singh1vj,singh1vp,bawanka,angom}@uwindsor.ca

Abstract. White Blood Cells also known as leukocytes plays an important role in the human body by increasing the immunity by fighting against infectious diseases. The classification of White Blood Cells, plays an important role in detection of a disease in an individual. The classification can also assist with the identification of diseases like infections, allergies, anemia, leukemia, cancer, Acquired Immune Deficiency Syndrome (AIDS), etc. that are caused due to anomalies in the immune system. This classification will assist the hematologist distinguish the type of White Blood Cells present in human body and find the root cause of diseases. Currently there are a large amount of research going on in this field. Considering a huge potential in the significance of classification of WBCs, we will be using a deep learning technique Convolution Neural Networks (CNN) which can classify the images of WBCs into its subtypes namely, Neutrophil, Eosinophil, Lymphocyte and Monocyte. In this paper, we will be reporting the results of various experiments executed on the Blood Cell Classification and Detection (BCCD) dataset using CNN.

Keywords: CNN · Blood cell classification · Basophils · Eosinophil · Monocytes · Lymphocytes · Neutrophils · Tensorflow · Keras · Softmax function · Relu function · White blood cell google colab

1 Introduction

White Blood cells are key players in the immune system of the human body. There are three broad classifications of blood cells-Red Blood Cells (RBC) that transport oxygen, White Blood Cells (WBC) the face of immune system and platelets that trigger blood clotting in damaged tissues [1]. White Blood Cells makes up 1% of the human blood in a healthy human adult. They are present throughout the body and each type of White Blood Cells have a certain functionality in the human body and serves by protecting human body against various infections and diseases. If they detect any of these in the blood, they attack them to counter any potential damage these elements can cause in the body [2,6]. The structure of the WBC, predominantly, comprises of a large lobed nucleus that can be used to distinguish a WBC from other blood cell types. Apart from a nucleus, WBC consists of cytoplasm and cell wall.

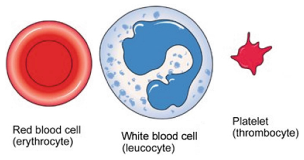


Fig. 1. Types of blood cells.

There are major five categories of WBC in human body. However due to data set constraints we have classified data into four categories: Basophils (0.4% approximately), Eosinophil (2.3% approximately), Monocytes (5.3% approximately), Lymphocytes (30% approximately) [2,7] and Neutrophils (62% approximately) (Fig. 1).

1.1 Eosinophil

The exact count of the Eosinophil in body keeps on changing during the day depending on the season and during different phases of the human body. On an average, Eosinophil amount to 2–4% of the total WBC count and can persist in the blood circulation for 8–12 days. These are found in the medulla, cortex region, lower gastrointestinal region and lymph nodes [2]. For recognizing the Eosinophil from the images, the cell is rounded in shape of skin-red color with a purple-colored bi-lobed nucleus. The lobes of the nucleus are connected by a thin strand.

1.2 Monocytes

The count of Monocytes present in a healthy human body varies between 6–9% of total WBC count. The lifetime of Monocyte varies from hours to a day. Monocytes are also responsible for presenting these pathogens to T cells so that they can be easily killed and it helps reduce the response time of antibodies in humans. Monocytes can be recognized in the BCCD image dataset using certain features, the nucleus in it is a kidney shaped-roundish cell with skin red color and some purple color in it without any lobe [2].

1.3 Lymphocytes

Lymphocytes count present in a healthy human body is around 25–30% of total WBC count. These cells are present in lymphatic system than in blood. Lymphocytes consists of two types of cells, namely B-cells and T-cells. These cells responsible for directly killing virus infected cells in human [3] body and also eliminating cancer cells. Lymphocytes can be easily identified in the BCCD dataset by looking at the nucleus as it is clearly round purple colored potato and eccentric.

1.4 Neutrophils

Neutrophils are a part of the innate immune system. The WBCs consists of almost 60–70% of Neutrophils making it largest component of WBCs. The main targets of Neutrophils are bacteria and fungal pathogens. Neutrophils can be recognized as purple colored multi-lobed groundnut-shaped nucleus inside the skin-red colored cells. Generally, there are three to five lobes in the nucleus with a transparent looking cytoplasm (Figs. 2 and 3).

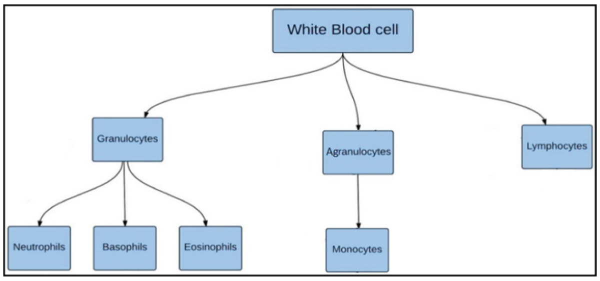


Fig. 2. Classification of WBCs.

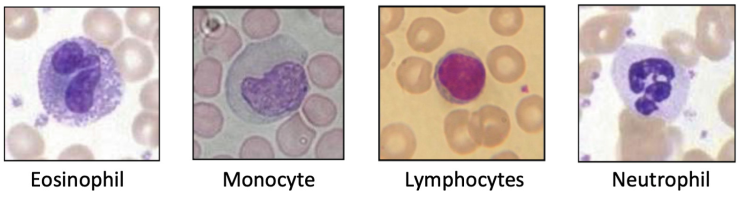


Fig. 3. Types of blood cells

2 Application of WBC Classification

Hematologist can use this classification process to diagnose the patients in an effective manner. For instance, this classification can be used to check if a person's blood sample consists of a particular type of WBC. Generally in the laboratory to classify WBC some instruments are used namely, flow cytometry.

However, this instrument is not very accurate so to overcome this disadvantage an automatic system using image processing, feature extraction and some deep learning techniques needs to be implemented for more accurate classification.

The lower level of Monocytes can be due to a lower number or lack of WBC in human body which can be due to chemotherapy, bone marrow disorder or bloodstream infection. The increase number of Monocytes classification means that the cells are increasing in response to infections, sarcoidosis and Langerhans [6]. B cells and T cells present in the WBC are also deficient which means that chances inhibiting cancer cells is very low. If the chance of classifying blood cells into Lymphocytes is more that means that the body is suffering from infection (bacterial, viral or other) or Cancer of the blood and lymphatic system. Deficiency of Neutrophils means that a person is suffering from a problem known as Neutropenia. Without these cells human body cannot resist from infections.

3 Problem Statement

In medical science, one of the challenges faced is the identification and determining the number of white blood cells. The major reason is the abundance of red blood cells. In a healthy adult, the WBCs make approximately 1% of the total blood volume. Due to this low proportion of the WBCs in blood, the recognition of the WBCs become a challenge which further toughens the job of classifying the subtypes of WBCs.

The change in number of any subtype of classification means that there is a problem in the body and the body is responding to a type of pathogen. The further prognosis of the disease can be determined by the disease type and can help in prescription of treatment for that specific ailment. The traditional method of classifying the WBC types includes studying the blood smeared slides under light and electron microscope. The blood cells are stained prior to studying them under a microscope [7]. For perfect identification, the pathologists need to look for the shape of nucleus and compare the size relative to RBCs. Since this is a manual process it is prone to error and is time consuming. The biggest drawback with the manual classification is the aspect of human error associated with mechanical scanning of glass slide and the tradeoff between the image resolution and microscopic field of view (FOV). To overcome these disadvantages, there is a need for an automated method to classify the white blood cells using an images of stained blood cells.

4 Blood Cell Count and Detection Dataset

For training the algorithm and performing the validations and experiments, we used the Blood Cell Count and Classification (BCCD) data set. The data set comprises of augmented images of the blood cells in JPEG format. Apart from the blood cell images, it is accompanied by cell type labels in a .csv file. There are a total of 12,500 labelled images of the blood cells with approximately 3000 images for 4 classes of WBCs namely Eosinophil, Lymphocyte, Monocyte and Neutrophil [3].

There are two main folders in the data set called "dataset-master" and "dataset2-master". The "dataset-master" contains 410 subtype labelled and augmented images while the "dataset2-master" contains 2500 augmented images. In total, "dataset2-master" contains 3000 images for each sub classes while "data-master" contains 88, 33, 21 and 207 images of the subtypes. In addition to this, the data set contains xml files corresponding to each image which contains the attributes that can be used to create bounding boxes on the images to highlight the RBC and WBC in the cell image. We are not using this aspect of the data set as our research is restricted to classifying the cell image on the basis of the WBC present in the image.

5 Techniques Used

5.1 Convolution Neural Networks (CNN)

For image classification, the CNN algorithm converts the image into an array of pixels called a feature map depending on the resolution of image. The CNN network generally consists of multiple hidden layers that convolve with a multiplication or dot product of the inputs from the previous layer. The image, transformed into a feature map can be combined with a filter/kernel to perform operations over the image to deduce some meaningful output. Using these filters, operations like edge detection, sharpening the image, blurring whole/partial image and identification can be performed [14]. After the convolution procedure is complete, we perform pooling on the result for dimension reduction to reduce the number of parameters. After pooling, we have a 3D resultant matrix which we pass through a fully connected ANN architecture. In order to converge out 3D input to a 1D array, we apply flattening on the resultant matrix which arranges the 3D volume of numbers into a 1D vector [9] (Fig. 4).

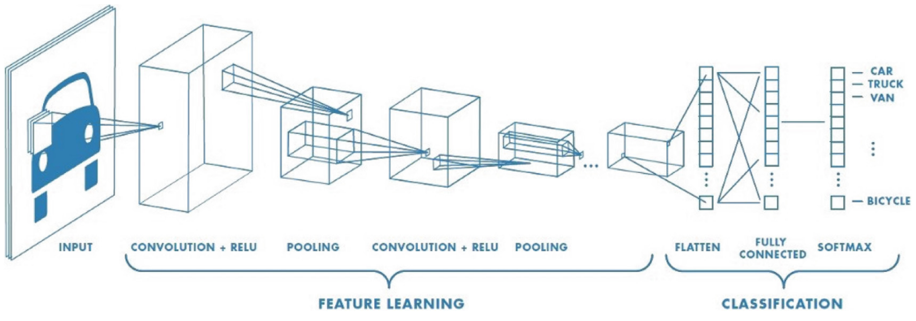


Fig. 4. A figure illustrates basic flow of CNN network.

5.2 Rectified Linear Unit (ReLU)

ReLU is a type of activation function which can be mathematically defined as y = max(0, x). It is used to extract only the positive part of the argument. ReLU is used to introduce non linearity in the training of the network. In this problem, we want our algorithm to get trained only on positive values. As compared to other activation functions, ReLU is inexpensive in terms of complexity as it does not have any complex computations.

5.3 Softmax Activation Function

The softmax function is another type of activation function used in training the CNN for this problem. Using the softmax function, we can perform logistic regression on the inputs and can normalize the input value to a vector of values. The vector of values are modelled as per a probability distribution where the sum of all vectors is equal to 1. The sigmoid function can be mathematically represented as:

$$\mathrm{P(y = j)} \mid \theta^{j} = \frac{e^{\theta^{(j)}}}{\sum_{j=0}^{k} e^{\theta_k^{(j)}}}$$

5.4 Keras

Keras is an open-source neural network library capable of running on top of TensorFlow. Keras is used in implementation for the blood cell training model as it provides the implementations of the convolution neural networks and filters, as required for the model construction. Using Keras, other CNN concepts are implemented like pooling, connecting to a highly connected network, dense and so on by using the inbuilt function os Keras framework.

5.5 TensorFlow

TensorFlow is also a computational network for constructing machine learning models. As Keras, it consists of end-to-end flexible ecosystem of tools which can be implemented into an algorithm in the form of libraries and build machine learning related applications. Similar to Keras, it also provides specific abstract methods which can be extended and optimised for usage in algorithms. Using TensorFlow, the architecture can also be configured on which the program has to be executed like CPU, TPU, GPU etc.

5.6 Google Colab

Due to computational limitations of the personal systems, we executed the training and testing of the algorithms on Google Colab. Colab is a Google project which provides a free Jupyter notebook environment that executes entirely on the cloud. Due to high intensity of the computations required for training the CNN network.

6 Implementation

In the implementation part, batch size is initialized as 128. Once all the images are passed through the network for one time, it marks the end of an epoch. The mean and standard image is generated using numpy by combining all the images in the training set and normalizing every pixel location and performing the mean and standard deviation functions on the images. After computation of the images, the mean image is subtracted from every image in the batch. The resultant image is then divided by the standard image (Fig. 5).

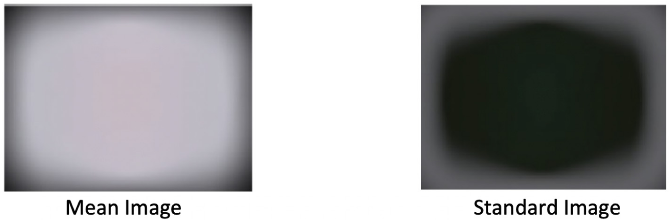


Fig. 5. Images of mean and standard deviation

After this our data set is divided into training dataset, validation dataset and test data set. The training data is of size 8961 and output data corresponding to this data is of size 8961 * 4 matrix (size of training data * number of classes). The validation set is of size 996 and output data corresponding to this data is of size 996 * 4 (size of training data * number of classes). In the CNN model, one input layer, three hidden layers and one output layer are created. For the input layer, the model is initialized using Input function and input is fed to Conv2D function of Keras, which created a convolution kernel that is convolved with the layer input to produce a tensor of outputs. In this layer features are extracted from the image using convolution, RELU, Batch Normalization, Dropout and performing pooling. After performing normalization, the Dropout is used which randomly sets a fraction rate of input units to zero at each update during training, which somehow helps prevent overfitting. Four Convolution layer are created using three different filters for feature extraction with 16, 8 and 4 layers respectively.

After performing flattening step, we create three dense hidden layers of 32, 16 and 8 dimensions using the Dense function of Keras and along with these layers an output layer of 4 dimension is created where each dimension corresponds to each class [15]. After initializing the model, the next step is to train the model. The model is trained using fit_generator() function of Keras. After feeding the training data to model summary is generated for the model that is optimized using Root mean square using RMSProp function which is an optimizer used to change the attributes of your neural network such as weights and learning rate in order to reduce losses (Fig. 6).

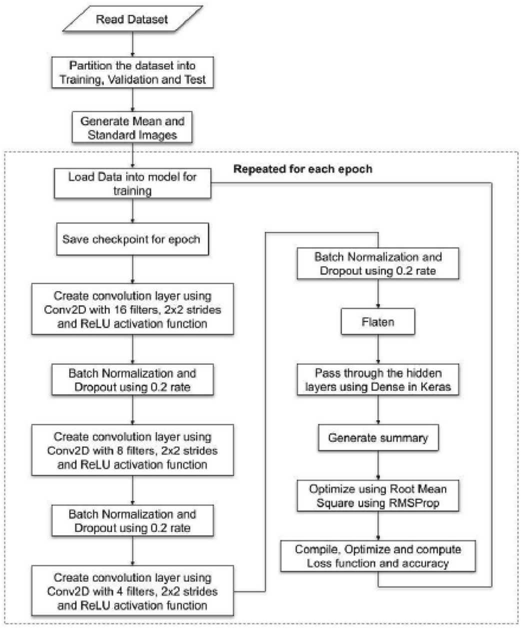
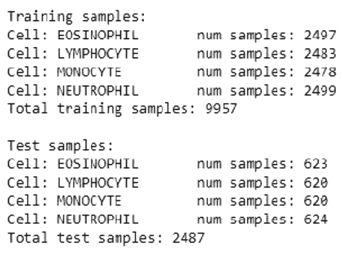

Read Dataset
Partition the dataset into Training, Validation and Test
Generate Mean and Standard Images
Repeated for each epoch
Load Data into model for training
Save checkpoint for epoch
Create convolution layer using Conv2D with 16 filters, 2x2 strides and ReLU activation function
Batch Normalization and Dropout using 0.2 rate
Create convolution layer using Conv2D with 8 filters, 2x2 strides and ReLU activation function
Batch Normalization and Dropout using 0.2 rate
Create convolution layer using Conv2D with 4 filters, 2x2 strides and ReLU activation function
Batch Normalization and Dropout using 0.2 rate
Flaten
Pass through the hidden layers using Dense in Keras
Generate summary
Optimize using Root Mean Square using RMSProp
Compile, Optimize and compute Loss function and accuracy

Fig. 6. Flowchart for training the algorithm.

7 Results and Discussions

7.1 Experiment

For performing the experiments, we trained and tested the model on three different epoch values i.e. 10, 100 and 200 on Google Cola using the GPU processor. The test was carried out on approximately 16000 parameters and trained on 9957 images and tested on 2487 images. The images are 240×320 pixel size.


```
Training samples:
Cell: EOSINOPHIL       num samples: 2497
Cell: LYMPHOCYTE       num samples: 2483
Cell: MONOCYTE         num samples: 2478
Cell: NEUTROPHIL       num samples: 2499
Total training samples: 9957

Test samples:
Cell: EOSINOPHIL       num samples: 623
Cell: LYMPHOCYTE       num samples: 620
Cell: MONOCYTE         num samples: 620
Cell: NEUTROPHIL       num samples: 624
Total test samples: 2487
```

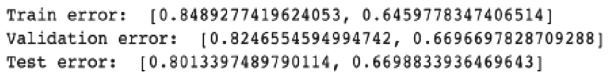
Fig. 7. Summary of dataset

7.2 Selection of Metrics

For evaluating the performance of the model, we recorded three major performance metric namely train performance, validation performance and test performance from every time we trained the model and tested the trained model. As being a multi-class problem with multiple labels. The loss and accuracy of whole training and testing of the program is calculated using an in-built Keras function called categorical crossentrophy. The output value for loss and accuracy value obtained is in the range of 0 to 1. For each performance metric recorded, we analysed loss and accuracy and is represented as a tuple in the following format: Performance: [Loss, Accuracy] (Fig. 7).

7.3 Results

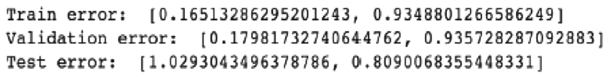
10 Epochs: In training analysis for 10 epochs, we can see that the loss function gives a value of 0.85 which signifies a high loss. A perfect model would have a log loss entropy of 0. An accuracy of 64.5% is observed. Similarly, a loss of 0.82 and 0.80 was observed for validation and test loss, respectively. The validation and test accuracy was evaluated as 66.9% and 66.98%. On a general, we can see that we are getting a very high value for the loss function and a low accuracy. The main reason is the low amount of training provided to the model since the number of iterations is only 10 (Figs. 8 and 9).



```
Train error:  [0.8489277419624053, 0.6459778347406514]
Validation error:  [0.8246554594994742, 0.6696697828709288]
Test error:  [0.8013397489790114, 0.6698833936469643]
```

Fig. 8. Result of 10 epochs

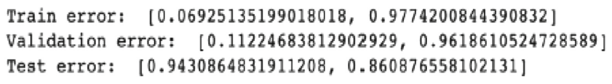
100 Epochs: We can see that the metrics improved when we increased the number of epochs to 100 from 10. The loss function for training dropped down to 0.165. The accuracy increased significantly to 93.48% for training. Similarly, a significant improvement can be seen in validation and test part as the loss function value dropped to 0.17 and 1.02, respectively. The accuracy is evaluated as 93.5% for validation and 80.9% for test. The test performance is relatively poor due to low number of epochs used for training (Fig. 10).



```
Train error:  [0.16513286295201243, 0.9348801266586249]
Validation error:  [0.17981732740644762, 0.935728287092883]
Test error:  [1.0293043496378786, 0.8090068355448331]
```

Fig. 9. Result of 100 epochs

200 Epochs: On further training till 200 epochs, the loss function value decreased that shows that with more number of epochs the quality of training improved significantly. For test performance, we can see that the value of loss function is 0.06 which is very close to 0 which shows that with further training the model is coming close to being a perfect model. The accuracy also increased to 97.7% for testing. Similarly, the validation performance also got better with the loss function value reaching to 0.11 and the accuracy elevated to 96.1%. However, we do not see much improvement in test performance as the value of loss function decreased slightly to 0.94 and accuracy increased by 6% to 86% as compared to 100 epochs.



```
Train error:  [0.06925135199018018, 0.9774200844390832]
Validation error:  [0.11224683812902929, 0.9618610524728589]
Test error:  [0.9430864831911208, 0.860876558102131]
```

Fig. 10. Result of 200 epochs

7.4 Comparison of Recorded Metrics

On analyzing the above graph plotted for loss value it can be deduced that when the value of epochs was increased, training and validation loss decreases as we increase the number of epochs. However, while using the model for testing on new images, we can see that the value of loss function increased as compared to lower number of iterations. However, due to high computation complexity, there is a need of better GPU processor to experiment using more number of epochs. On the analysis of the accuracy metric for all the count of epochs, it can be observed from the graph that on increasing the number of epochs, the accuracy improves. The accuracy improved significantly when the epochs were increased from 10 to 100. A small improvement in accuracy can be observed for train, test and validation when the epochs were elevated from 100 to 200. From this trend, it can be ascertained that the accuracy increases when the number of epochs are increased (Fig. 11).

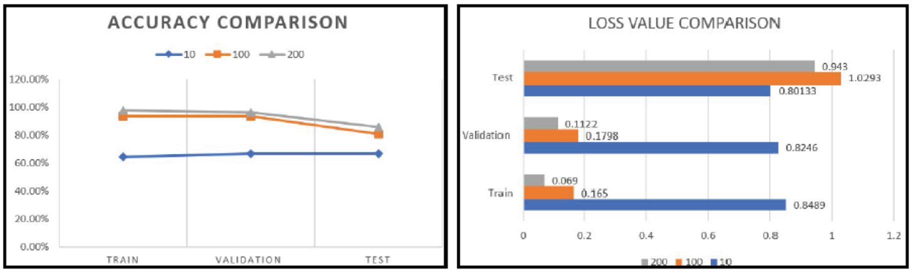


Fig. 11. Comparison of accuracy and loss function value for 10, 100 and 200 epochs

8 Challenges Faced

The biggest concern was that while training and testing the model the time taken by the system was very high due to shortage of RAM in the system. At first the model was executed for around 1000 epochs which would have taken around 8 to 10 h just to train the model and the testing would have taken even more time. So to reduce the time the number of epochs was decreased to around 200 and Google Colab platform was used, which provides Jupyter platform for python development completely on cloud.

9 Future Work

On analyzing the dataset, there are attribute files attached with the dataset corresponding to each image. In that xml file, the attributes specify the dimensions of bounding boxes. Based on that, every cell present in the image can be classified as a WBC, RBC or platelet. The cell image count helps in diagnosis of various diseases as an elevation or depreciation in the count of WBC, RBC and platelets can be a pointer to identify an ailment. The main reason for this diagnosis is that during a dengue infection, the number of platelets start decreasing.

10 Conclusion

This research helps the hematologist to classify White Blood Cells into their subtypes with the help of microscopic images of cell using Convolutional Neural Network techniques. This classification helps to distinguish the cells and check what type of disease a patient is suffering from. The results obtained from this experiment helps identify images in a robust way as compared to the orthodox lab methods. The good level of accuracy above 90 for the test set. Hence, when the model is trained with high computational abilities present, a perfect model can be trained and can be applied in the medical analysis and applications dealing with the number of white blood cells and sub types of white blood cells.

References

1. Maton, A.: Human Biology and Health. Prentice Hall, Englewood Cliffs (1993)
2. LaFleur-Brooks, M.: Exploring Medical Language: A Student-Directed Approach, 7th edn, p. 398. Mosby Elsevier, St. Louis Missouri (2008). ISBN 978-0-323-04950-4
3. Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., Walter, P.: Molecular Biology of the Cell, p. 1367. Garland Science, New York (2002)
4. Kampbell, N.A.: Biology. Benjamin Cummings, San Francisco (n.d.)
5. NCI Dictionary of Cancer Terms (n.d.). https://www.cancer.gov/publications/dictionaries/cancer-terms/
6. Macawile, M.J., Quinones, V.V., Ballado, A., Cruz, J.D., Caya, M.V.: White blood cell classification and counting using convolutional neural network. In: 2018 3rd International Conference on Control and Robotics Engineering (ICCRE) (2018)

7. Al-Dulaimi, K., Chandran, V., Banks, J., Tomeo-Reyes, I., Nguyen, K.: Classification of white blood cells using bispectral invariant features of nuclei shape. In: 2018 Digital Image Computing: Techniques and Applications (DICTA) (2018)
8. Dertat, A.: Applied Deep Learning - Part 4: Convolutional Neural Networks, Medium, 13 November 2017. https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2
9. Prabhu: Understanding of Convolutional Neural Network (CNN) - Deep Learning, Medium, 21 November 2019. https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148
10. Dernoncourt, F.: What is batch size in neural network? Cross Validated, 01 June 1965. https://stats.stackexchange.com/questions/153531/what-is-batchsize-in-neural-network
11. Daniel: What's is the difference between train, validation and test set, in neural networks? Stack Overflow, 01 July 1960. https://stackoverflow.com/questions/2976452/whats-is-the-difference-between-train-validation-and-test-set-in-neural-netwo
12. LNCS Home Page. http://www.springer.com/lncs. Accessed 4 Oct 2017
13. A Comprehensive Guide to Convolutional Neural Networks - the ELI5 way. https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. Accessed 17 Dec 2018
14. ReLU. https://www.tinymind.com/learn/terms/relu. Accessed 30 Oct 2017
15. Di Ruberto, C., Putzu, L.: Accurate blood cells segmentation through intuitionistic fuzzy set threshold. In: 2014 Tenth International Conference on Signal-Image Technology and Internet-Based Systems, pp. 57–64, November 2014. https://doi.org/10.1109/SITIS.2014.43
16. Blood Cell Images. https://www.kaggle.com/paultimothymooney/blood-cells. Accessed 21 Apr 2018
17. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)