International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 33, No. 5, October, 2019, pp. 335-340
Journal homepage: http://iieta.org/journals/ria

Classification of White Blood Cells by Deep Learning Methods for Diagnosing Disease

Muhammed Yildirim*, Ahmet Çinar

Computer Engineering Department / Fırat University, Elazığ, 23100, Turkey

Corresponding Author Email: 171129205@firat.edu.tr

https://doi.org/10.18280/ria.330502

Received: 22 June 2019
Accepted: 29 August 2019

Keywords:
classification, leukocytes, machine learning, neural networks, white blood cells

ABSTRACT

Leukocytes, also known as white blood cells, are a group of cells that protect the body against infections, which is an important part of the immune system. The classification of white blood cells is widely used to diagnose various diseases, such as AIDS, leukemia, myeloma and anemia. However, the conventional methods to classify white blood cells are time consuming and prone to errors. In this paper, one of the most popular neural networks, convolutional neural network (CNN) is selected to differentiate between different types of white blood cells, namely, eosinophil, lymphocyte, monocyte and neutrophil. The CNN was coupled with Alexnet, Resnet50, Densenet201 and GoogleNet in turn, and trained with the Kaggle Dataset. Then, Gaussian and median filters were applied separately to the images in the database. The new images were classified again by the CNN with each of the four networks. The results obtained after applying the two filters to the images were better than the results obtained with the original data. The research results make it easier to diagnose blood related diseases.

1. INTRODUCTION

Leukocytes, also known as White Blood Cells, help to protect the body against foreign substances, germs and infectious diseases, and help fight infections caused by them. Leukocytes, which make up a small part of the blood, multiply rapidly if a foreign substance enters the body. Then they leave the vessel and reach the related tissue. The number of leukocytes in the corresponding tissues is much higher than in a healthy blood sample. In addition, the presence of white blood cells in the medical world is also used as an indicator of various types of disease [1]. White blood cells become a parameter in the diagnosis of diseases such as leukemia, anemia, cancer and some other infectious diseases [2]. A blood sample of a patient with a normal blood cell and leukemia is given in Figure 1 [3].

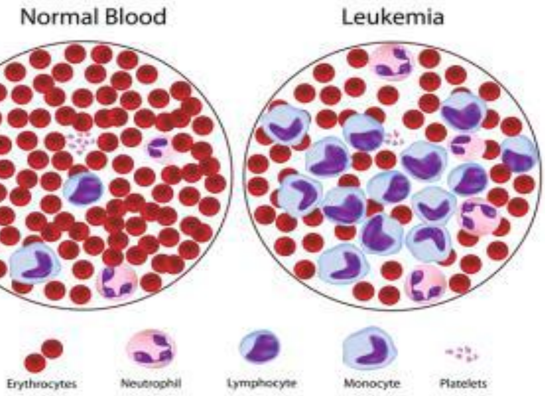


Figure 1. Normal and Leukemia blood

The aim of this study is to classify leukocytes with regard to Eosinophil, Lymphocyte, Monocyte, Neutrophil. The number of data from Kaggle dataset [4] is given in Figure 2.

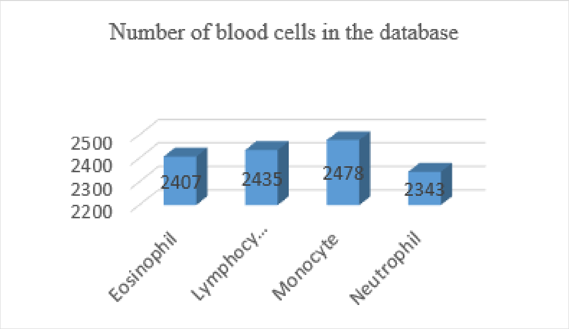


Figure 2. Number of data in each class [4]

In the use of large data sets, especially deep learning networks that are successful in the classification process are examined. Deep learning has become one of the most popular sub-branches of artificial intelligence. There are multiple architectures used in deep learning. In this paper, CNN models are used. CNN started to be preferred in many fields due to its success in Imagenet competition held in 2012 [5].

In order to achieve good performance from deep learning methods, the network needs to be trained with large amounts of data during the training phase. If the amount of data used to train the network is low, results with a high performance rate may not be obtained from this network. Working with conventional processors makes the process difficult because the amount of data used to train the network is high in deep learning architectures. Instead, the faster processors or cards should be preferred [6].

In the first part of this study, brief information about the study is given and in the second part Materials and Methods are mentioned. In Chapter 3, Application and Result are examined. Finally, the conclusion section is completed.

2. MATERIALS AND METHODS

In this section, data sets, data properties, deep learning, CNN Network and CNN models will be explained. Information will also be given on the preprocessing steps and filters applied. Deep learning networks and deep learning models are utilized in the realization of this study. The images are classified using Matlab environment and the results are compared [7]. At this stage, both the deep learning models Alexnet, Googlenet, Resnet50 and DenseNet201 are utilized and the results obtained from the processed data are compared with the original results. The flowchart of the program processed with the original data is as shown in Figure 3.

The flow diagram of the application after preprocessing with Gaussian and Median filters is as in Figure 4.

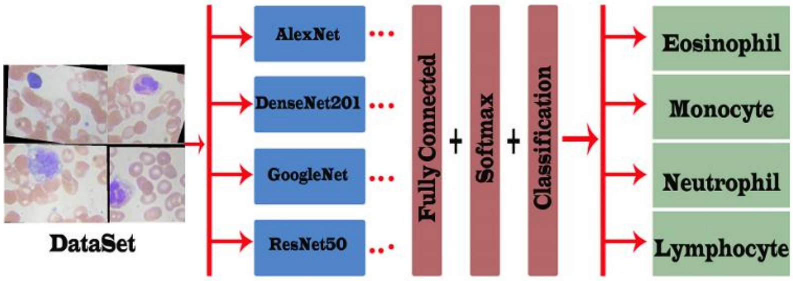


Figure 3. Classification of original data in different architectures

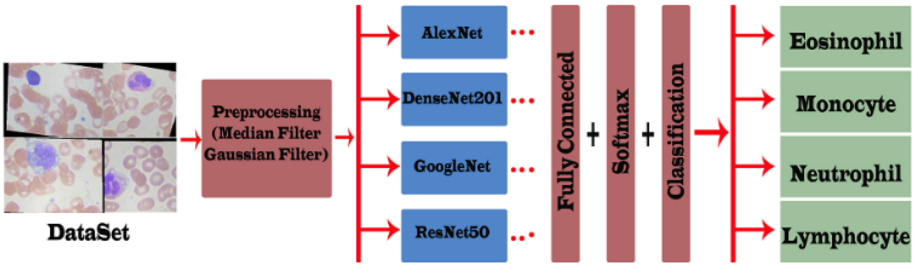


Figure 4. Classification of data with different architectures after preprocessing

2.1 Dataset

The data set used in this study are taken from the Kaggle Dataset [4]. In total, 9663 data are studied. These data are leukocyte images which is categorized into 4 different classes.

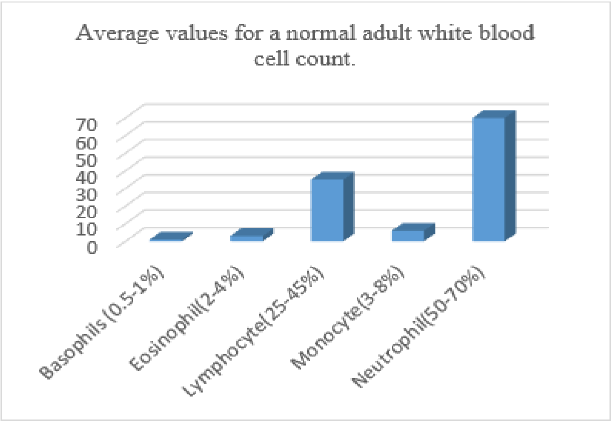


Figure 5. Average values for a normal adult white blood cell count

Leukocytes are a kind of blood cell produced in the bone marrow, found in blood and lymph tissue. Leukocytes are part of the body's immune system. They fight against to infection and other diseases. Blood images of a person have an important role in the diagnosis and control of many diseases. There are approximately 4000-10000 leukocytes in an adult human body. Having leukocyte count outside this range can be a sign of various diseases [8]. The average values n the white blood cell of a normal person is as Figure 5.

2.2 Deep learning

Deep learning is one of the most up-to-date and widely used areas of machine learning that allows computers to process and learn data and understand data in terms of hierarchy. Deep learning is a sub-branch of machine learning in which algorithms are modeled on the structure and the way the brain works, called artificial neural networks [9]. Deep learning is widely used in many fields such as image and video processing, biomedical image and signal processing, robotics, chemistry, advertising, search engines, finance, natural language processing, classification. One of the most important features that distinguish deep learning from traditional neural networks is that it can work with multiple layers. In conventional artificial neural networks, each cell is bound to all cells in the preceding and subsequent layers. Numerous mathematical operations are performed in each layer. Due to the increasing amount of data and the number of layers, a high amount of CPU power is needed. In recent years, GPU-based and high-memory processors have been preferred for deep learning [10].

There are multiple types of networks used for deep learning. These networks can be classified as convolutional neural networks, Long short term memory networks, Limited Boltzman Machines, deep belief networks, Recurrent Neural Networks and deep auto-encoders.

In this study, convolutional neural network is used.

2.2.1 Convolutional Neural Networks (CNN)

Convolutional neural networks, which is frequently mentioned in recent years, is one of the most widely used deep learning networks such as image classification. The origin of CNN networks is based on artificial neural networks. As the number of hidden layers in artificial neural networks increases, the complexity of the network increases and better results are obtained from the network. The number of layers used in CNN networks is quite high [11].

Convulational Layer. In the convolution layer, it is aimed to obtain attribute maps. For this purpose, the result *f* function is obtained by performing convolution between input *f* function and *g* kernel function. While the convolution layers are taught, the coefficients of the kernel function are tried to be obtained. The convolution process takes place on the image matrix by traversing a *NxN* matrix [12].

Pooling Layer. This layer is generally used after convolution, mostly using the maximum pooling layer. Pooling layer are used to reduce computational complexity. The network does not learn any parameters in this layer. The pooling layer uses an *NxN* sized matrix [13].

Full Connected Layer. In a convolutional neural network, there is a connected layer after the convolution and centering layers. The fully connected layer, as the name implies, has the previous layer of nerves. The data in the previous layer becomes a one-dimensional matrix in this layer. In this layer, class properties of objects can be obtained [14].

Rectified Linear Unit (Relu) Layer. The activation function is a function that produces an output value corresponding to the input to the artificial neuron cell. In general, the activation functions simoid, hyperbolic tangent, sinus, step, threshold value functions are used. The fact that the computational load is less than the sigmoid and hyperbolic tangent functions has made it more preferred in multilayer networks [15].

Dropout Layer. While the network is being trained in the deeper learning layers, memorizing the network called extreme learning can take place. Memorizing the network is not a desirable event. Some nodes that memorize the network must be disabled to prevent memorizing the network. In this way, memorization of the network is prevented and this improves the performance of the network [16].

Normalization. In deep artificial neural networks, it is necessary to normalize the network in order to improve the performance of the network. Therefore, the data from the layers must be in a specific order. Input data from layers may be too large or too small. For education and process, it is important to normalize these values and use them within a certain range. The normalization of input data not only improves the performance of the network, but also ensures that the data used is periodically scaled [17].

Softmax Layer. The classification layer receives input data from previous layers and uses it to classify the data. Shows the status of data for a particular class and generates value about which class is closer to it. Extracts the probability value for each class by performing the probabilistic calculation created in the deep learning network. Cross entropy is used for these procedures [18].

2.2.2 Deep learning models

There are many models widely used in deep learning. In this study, AlexNet, Resnet50, DenseNet201 and GoogleNet models which are widely used in CNN network are used. Then the results were compared with each other and the results were interpreted.

AlexNet. The AlexNet model was developed by Alex Krizhevsky, Ilya Sutskever and Geoffrey Hinton. The winner of the ILSVRC Imagenet competition in 2012 is a model that makes deep learning popular again. Thanks to this model, the error rate in pattern recognition has decreased from 26% to 15%. It is very similar to the LeNet architecture published by Yan Le Cun in 1998. Relu activation function is used for nonlinear functions because of the existence of consecutive Convulational and Pooling layers in both architectures. The AlexNet model with multiple layers is the first operation on a parallel dual GPU [19].

Resnet50. Resnet is one of the most successful CNNs developed in recent years. In addition, it introduced a new approach that would enable the training of very deep CNNs. The first CNN that exceeds the 1000 layer limit. In 2015, he won the ILSVRC ImageNet competition in various fields such as classification, segmentation and detection. ResNet's classification error rate was tested as 3.6% on average. The success of this model is clearer when the average person has a 5% error. ResNet has pioneered several successive models and has been used in a variety of applications, particularly in classification [20].

GoogleNet. The winning Googlenet architecture of the ImageNet 2014 competition is a complex combination of startup modules. Each module consists of different dimensional convolution and max-pooling processes. The model, with an error rate of around 5.7%, achieved successful results in the data sets. The GoogleNet model, which has a depth of 22 layers, has a structure of 144 layers. The initial module, which is filtered according to different dimensions, has produced a different formation from the deep learning architectures that have emerged before [21].

DenseNet201. DenseNet-201 is an evolutionary neural network that has been trained on more than one million images from the ImageNet database. Network 201 is layer-deep and has learned rich feature representations for a wide range of images. The network 224 to 224 has an image input size. Together with better parameter efficiency, a major advantage of DenseNet is the enhanced flow of information and gradient that makes it easy to train them across the network [22].

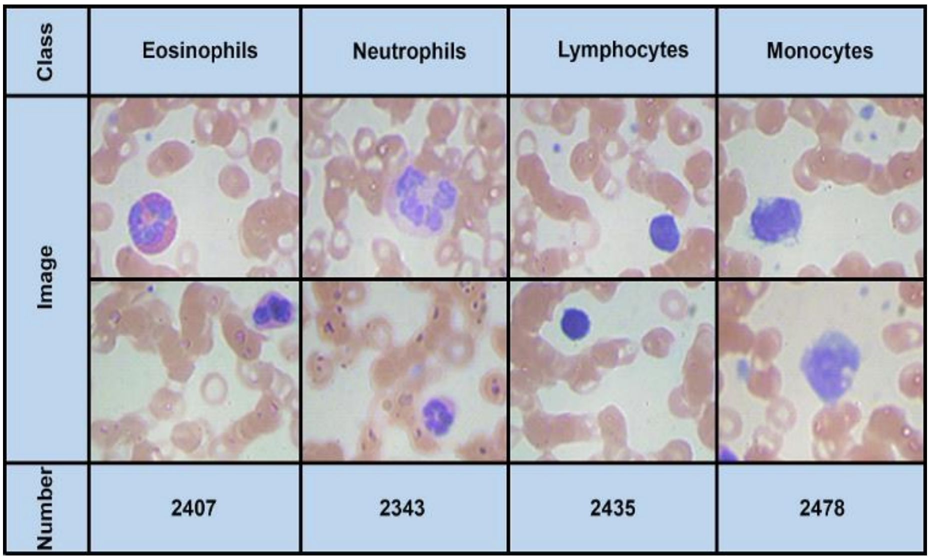
3. APPLICATION AND RESULTS

In the application, the original data was first classified with AlexNet, Resnet50, DenseNet201 and GoogleNet architectures and accuracy values and confision matrices were obtained. Then, Gaussian filter and Median filter were applied to the data in the data set and new data were obtained. Gaussian filter and Median Filter applied data were reclassified with the same 4 architectures and accuracy values and confision matrices were obtained.

The Gaussian filter is used to smooth out a given image. In other words, it removes the noise on the picture and makes the picture more meaningful. In a picture with a Gaussian filter, the neighborhood matrix is created first. The size of this matrix depends on the neighborhood value in the Gaussian filter function. In summary, the Gaussian filter reduces the hard tone changes in the picture, making the image softer [23].

Like the Gaussian filter, the aim of using a median filter is to reduce the hard tone changes in the image, making the image softer.

Used data are taken from the Kaggle Dataset. The data set consists of 4 different leukocyte cells. Each class has an average of 2300-2500 data. The data set consists of an average of 10000 data. The classes in the data set are as in Figure 6.



Class	Eosinophils	Neutrophils	Lymphocytes	Monocytes
Image				
Number	2407	2343	2435	2478

Figure 6. Class samples from the dataset

Both the original data and the pre-processing (Median Filter, Gauss Filter) results obtained in Alexnet architecture are shown in Table 1.

Table 1. AlexNet architecture test results

Alexnet					
	Confusion Matrix				
Original Data		1	2	3	4
	1	0.6880	0.1325	0.1368	0.0427
	2	0.1197	0.7607	0.0043	0
	3	0.0171	0.0342	0.9487	0
	4	0.1581	0.1410	0.1624	0.5385
	Accuracy:%79.27				
Data with Gauss Filter		1	2	3	4
	1	0.6923	0.0342	0.1581	0.1154
	2	0.0171	0.9274	0.0385	0.0171
	3	0	0	1	0
	4	0.1453	0.0299	0.1581	0.6667
	Accuracy: %82.16				
Data with Median Filter		1	2	3	4
	1	0.6709	0.0043	0.0641	0.2607
	2	0.1197	0.7607	0.0299	0.0897
	3	0.0085	0,0043	0.9829	0.0043
	4	0.1111	0	0.0726	0.8162
	Accuracy: %80.77				

Alexnet architecture classified the original data with an accuracy of 79.27%. After the gauss filter was applied to the images, the accuracy rate in the same architecture increased to 82.16% and when Median filter was applied, the accuracy rate increased to 80.77%. The highest accuracy of Alexnet architecture was obtained in Gaussian filtered images. The lowest accuracy was obtained in the original images. As a result, successful results were obtained at a higher rate than the data applied with gaussian filter and median filter. Accuracy rate increased by 2.89% in Gaussian filtered data and 1.5% in median filtered data.

Both the original data and the pre-processing (MedianFilter, Gauss Filter) results obtained in Resnet50 architecture are shown in Table 2.

Table 2. ResNet50 architecture test results

Resnet50					
	Confusion Matrix				
Original Data		1	2	3	4
	1	0.7479	0.0085	0.0043	0.2393
	2	0.0897	0.7735	0.0128	0.1239
	3	0.1197	0.0128	0.7650	0.1111
	4	0.1068	0.0214	0.0085	0.8632
	Accuracy: %78.74				
Data with Gauss Filter		1	2	3	4
	1	0.6838	0.1410	0.0983	0.0769
	2	0.0256	0.9402	0.0256	0.0085
	3	0.0043	0.0342	0.9829	0
	4	0.1282	0.1154	0.1624	0.5940
	Accuracy: %80.02				
Data with Median Filter		1	2	3	4
	1	0.3419	0.0513	0.0427	0.5641
	2	0.0043	0.8462	0.0043	14.53
	3	0.0043	0.0128	0.8547	0.1282
	4	0.0342	0.0085	0.0043	0.9530
	Accuracy: %74.89				

The Resnet50 architecture classified the original data with an accuracy of 78.74%. After the gauss filter was applied to the images, the accuracy rate in the same architecture increased to 80.02%. When the Median filter was applied to the same images, the accuracy rate decreased to 74.89%. In Resnet50 architecture, the highest accuracy was obtained in Gaussian filtered images. The lowest accuracy rate was obtained in median filtered images. As a result, successful results were obtained from the gaussian filtered data and the results were obtained with a lower accuracy rate than the median filtered data. Accuracy rate of Gaussian filtered data increased by 1.28%, accuracy rate of median filtered data decreased by 3.85%.

The results of the existing data obtained in Densenet201 architecture both in their original form and by preprocessing (Median Filter, Gauss Filter) are as in Table 3.

Table 3. DenseNet201 architecture test results

DenseNet201					
	Confusion Matrix				
Original Data		1	2	3	4
	1	0.4658	0.2863	0.0684	0.1795
	2	0.0043	0.9915	0.0043	0
	3	0	0.1026	0.8803	0.0171
	4	0.0299	0.1667	0.0256	0.7778
	Accuracy: %77.88				
Data with Gauss Filter		1	2	3	4
	1	0.7821	0.0256	0.0085	0.1838
	2	0.0641	0.8889	0.0128	0.0342
	3	0.0855	0.0085	0.8761	0.0299
	4	0.1709	0.0342	0.0043	0.7906
	Accuracy: %83.44				
Data with Median Filter		1	2	3	4
	1	0.6282	0	0.0855	0.2863
	2	0.1239	0.5940	0.0342	0.2479
	3	0	0	0.9658	0.0342
	4	0.0641	0	0.0171	0.9188
	Accuracy: %77.67				

The DenseNet201 Architecture classifies the original data with an accuracy of 77.88%. Accuracy rate in the same architecture increased to 83.44% after the gauss filter was applied to the images, and the accuracy rate decreased to 77.67% when the Median filter was applied. While the Gaussian filter increased the accuracy, there was not much change in the median filtered images.

Both the original data and the preprocessing (Median Filter, Gauss Filter) of the available data are obtained in the GoogleNet architecture is given in Table 4.

Table 4. GoogleNet architecture test results

GoogleNet					
	Confusion Matrix				
Original Data		1	2	3	4
	1	0.4701	0.0684	0.0085	0.4530
	2	0.0812	0.6368	0	0.2821
	3	0.1154	0.0812	0.4786	0.3248
	4	0.0342	0.0342	0	0.9316
	Accuracy: %62.93				
Gauss		1	2	3	4
	1	0.6538	0.1496	0.0598	0.1368
	2	0.0769	0.8632	0.0256	0.0342
	3	0.0726	0.0470	0.8675	0.0128
	4	0.2179	0.1111	0.0470	0.6239
	Accuracy: %75.21				
Data with Median Filter		1	2	3	4
	1	0.7308	0.0214	0.1966	0.0513
	2	0.2949	0.5128	0.1282	0.0641
	3	0.0214	0	0.9615	0.0171
	4	0.2179	0.0128	0.1966	0.5726
	Accuracy: %69.44				

GoogleNet Architecture classified the original data with an accuracy of 62.93%. After the gauss filter was applied to the images, the accuracy rate increased to 75.21% in the same architecture and to 69.44% when the Median filter was applied. Both Gaussian Filters and Median Filters have increased accuracy.

The accuracy of all results is given in Table 5.

AlexNet architecture classified the original data with the highest accuracy (79.27%). Resnet50 architecture with 78.74%, DenseNet201 architecture with 77.88% and GoogleNet architecture with 62.93% followed.

Table 5. Accuracy table of all results

	Orijinal	Gauss	Median
AlexNet	%79.27	%82.16	%80.77
ResNet50	%78.74	%80.02	%74.89
DenseNet201	%77.88	%83.44	%77.67
GoogleNet	%62.93	%75.21	%69.44

After applying Gaussian filter, DenseNet201 architecture achieved the highest accuracy with 83.44%. DenseNet201 was followed by AlexNet with 82.16%, ResNet50 with 80.02% and GoogleNet with 75.21%, respectively.

After the median filter was applied, the highest accuracy rate was obtained by AlexNet architecture with 80.77%. Alexnet architecture followed DenseNet201 with 77.67%, ResNet50 with 74.89% and GoogleNet with 69.44%, respectively.

4. CONCLUSION

In this paper, deep learning, deep learning layers and deep learning models that have made a lot of mention in recent years are examined for classification. First of all, Eosinophil, Lymphocyte, Monocyte and Neutrophil cells that make up the leukocyte blood cell were classified in Matlab application environment using Alexnet, Resnet50, Densenet201 and GoogleNet architectures. Then, gaussian and median filters were applied separately to the imagas in the database. The new images were re-classified using the same architectures. It was observed that the results obtained after applying Gaussian and Median filters to the images were better than the results obtained with the original data. Once blood cells have been classified, it is easier to make inferences about diseases by skilled persons. Because deep learning models work with large databases, it takes a long time to train and test networks. This is a challenge in Deep Learning. GPU-style cards may be preferred to overcome this speed problem. In this way, the speed problem can be minimized.

REFERENCES

[1] Fan, H., Zhang, F., Xi, L., Li, Z., Liu, G., Xu, Y. (2019). LeukocyteMask: An automated localization and segmentation method for leukocyte in blood smear images using deep neural networks. Journal of Biophotonics, 12(7): e201800488. https://doi.org/10.1002/jbio.201800488

[2] Wang, Q., Bi, S., Sun, M., Wang, Y., Wang, D., Yang, S. (2019). Deep learning approach to peripheral leukocyte recognition. PloS One, 14(6): e0218808. https://doi.org/10.1371/journal.pone.0218808

[3] Medgurus. https://www.medgurus.org/, accessed on 3 May 2019.

[4] Kaggle, https://www.kaggle.com/datasets, accessed on 3 May 2019.

[5] Sajjad, M., Khan, S., Jan, Z., Muhammad, K., Moon, H., Kwak, J.T., Rho, S., Baik, S.W., Mehmood, I. (2016). Leukocytes classification and segmentation in microscopic blood smear: A resource-aware healthcare

service in smart cities. IEEE Access, 5: 3475-3489. https://doi.org/10.1109/ACCESS.2016.2636218

[6] Rehman, A., Abbas, N., Saba, T., Rahman, S.I.U., Mehmood, Z., Kolivand, H. (2018). Classification of acute lymphoblastic leukemia using deep learning. Microscopy Research and Technique, 81(11): 1310-1317. https://doi.org/10.1002/jemt.23139

[7] Matlab, ttps://www.mathworks.com/products/matlab.html, accessed on 3 May 2019.

[8] Gautam, A., Bhadauria, H. (2014). Classification of white blood cells based on morphological features. In 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 2363-2368. https://doi.org/10.1109/ICACCI.2014.6968362

[9] Chen, H., Engkvist, O., Wang, Y., Olivecrona, M., Blaschke, T. (2018). The rise of deep learning in drug discovery. Drug Discovery Today, 23(6): 1241-1250. https://doi.org/10.1016/j.drudis.2018.01.039

[10] Kermany, D.S., Goldbaum, M., Cai, W., Valentim, C.C., Liang, H., Baxter, S.L., McKeown, A., Yang, G., Wu, X.K., Yan, F.B., Dong, J., Prasadha, M.K., Pei, J., Ting, M.Y.L., Zhu, J., Li, C., Hewett, S., Dong, J., Ziyar, I., Shi, A., Zhang, R.Z., Zheng, L.H., Hou, R., Shi, W., Fu, X., Duan, Y., Huu, V.A.N., Wen, C., Zhang, E.D., Zhang, C.L., Li, O.L., Wang, X.B., Singer, M.A., Sun, X.D., Xu, J., Tafreshi, A., Lewis, M.A., Xia, H.M., Zhang, K. (2018). Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell, 172(5): 1122-1131. https://doi.org/10.1016/j.cell.2018.02.010

[11] Molchanov, P., Tyree, S., Karras, T., Aila, T., Kautz, J. (2016). Pruning convolutional neural networks for resource efficient transfer learning. arXiv preprint arXiv:1611.06440, 3.

[12] Liu, L., Shen, C., van den Hengel, A. (2015). The treasure beneath convolutional layers: Cross-convolutional-layer pooling for image classification. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4749-4757. arXiv:1411.7466

[13] Castelluccio, M., Poggi, G., Sansone, C., Verdoliva, L. (2015). Land use classification in remote sensing images by convolutional neural networks. arXiv preprint arXiv:1508.00092.

[14] Cireşan, D.C., Giusti, A., Gambardella, L.M., Schmidhuber, J. (2013). Mitosis detection in breast cancer histology images with deep neural networks. In International Conference on Medical Image Computing and Computer-assisted Intervention, pp. 411-418. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-40763-5_51

[15] Agarap, A.F. (2018). Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375.

[16] Qi, C.R., Yi, L., Su, H., Guibas, L.J. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Advances in Neural Information Processing Systems, pp. 5099-5108.

[17] Tran, S.D., Manmatha, R. (2018). U.S. Patent No. 9,892,344. Washington, DC: U.S. Patent and Trademark Office. https://doi.org/10.2478/SJECR-2018-0063

[18] Du, S.S., Hu, W., Lee, J.D. (2018). Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In Advances in Neural Information Processing Systems, pp. 384-395. arXiv:1806.00900

[19] Krizhevsky, A., Sutskever, I., Hinton, G.E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6): 84-90. https://doi.org/10.1145/3065386

[20] He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., Li, M. (2019). Bag of tricks for image classification with convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 558-567.

[21] Szegedy, C., Liu, W., Jia, Y.Q., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA. https://doi.org/10.1109/CVPR.2015.7298594

[22] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700-4708.

[23] Izanloo, R., Fakoorian, S.A., Yazdi, H.S., Simon, D. (2016). Kalman filtering based on the maximum correntropy criterion in the presence of non-Gaussian noise. In 2016 Annual Conference on Information Science and Systems (CISS), pp. 500-505. https://doi.org/10.1109/CISS.2016.7460553