

Detection and Classification of White Blood Cells Through Deep Learning Techniques

<https://doi.org/10.3991/ijoe.v16i15.15481>

M. Samir Abou El-Seoud ^(✉), Muaad Hammuda Siala, Gerard McKee
The British University, Cairo, Egypt
Samir.elseoud@bue.edu.eg

Abstract—Leukemia is one of the deadliest diseases in human life, it is a type of cancer that hits blood cells. The task of diagnosing Leukemia is time consuming and tedious for doctors; it is also challenging to determine the level and type of Leukemia. The diagnoses of Leukemia are achieved through identifying the changes on the White blood Cells (WBC). WBCs are divided into five types: Neutrophils, Eosinophils, Basophils, Monocytes, and Lymphocytes. In this paper, the authors propose a Convolutional Neural Network to detect and classify normal white blood cells. The program will learn about the shape and type of normal WBC by performing the following two tasks. The first task is identifying high level features of a normal white blood cell. The second task is classifying the normal white blood cell according to its type. Using a Convolutional Neural Network CNN, the system will be able to detect normal WBCs by comparing them with the high-level features of normal WBC. This process of identifying and classifying WBC can be vital for doctors and medical staff to make a decision. The proposed network achieves an accuracy up to 96.78% with a dataset including 10,000 blood cell images.

Keywords—Image Processing, Convolutional Neural Network, Deep learning, Histology

1 Introduction

Leukemia is one of the deadliest diseases threatening humanity [1]. It is a form of cancer that hits blood and bone marrow. It starts in the bone marrow then expands to white blood cells and some of the leukemia's start in other types of blood cells [2]. The blood cells are the main source for detecting leukemia. As is known, the human blood is divided into three main parts which are red cells, white cells, platelets [3]. The abnormal changes of number, shape, texture of the white blood cells is considered as a major symptom of having leukemia in the blood [4],[5]. Figure 1 shows the difference between healthy blood and blood Infected with leukemia.

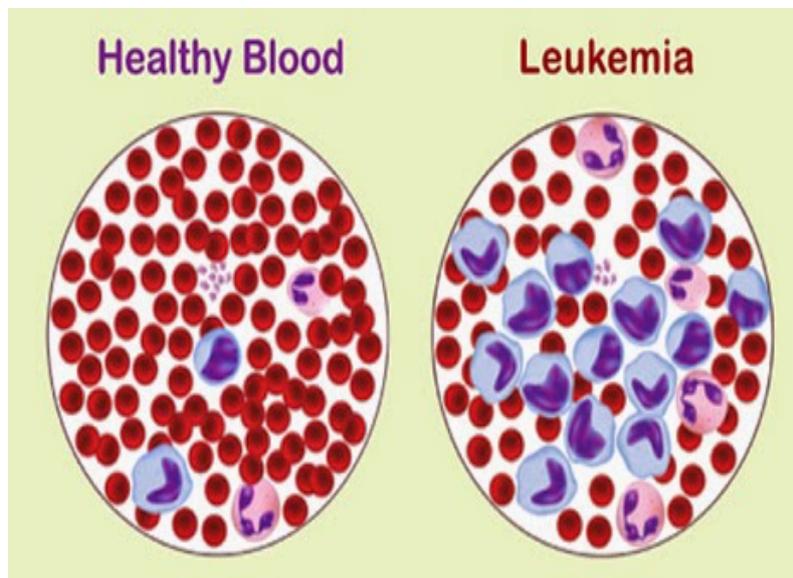


Fig. 1. The difference between healthy blood and leukemia [6].

The main focus of this study is on the classification of normal white blood cells, which is the common problem that faces most researcher in biomedical engineering. The white blood cells are divided into five types, which are Neutrophils, Eosinophils, Basophils, Monocytes, and Lymphocytes [4].

The diagnosis of Leukemia nowadays is done by taking one of the three main clinical tests. Physical test, Complete Blood Count test, Bone Marrow test. This paper focuses on the first stage of the Complete Blood Count (CBC) test. The first stage is the ability to determine the difference between normal WBC types. Doctors will look at the blood smear and focus on the normal and abnormal changes in white blood cells.

Machine learning has significantly contributed to the society's healthcare in a positive way. In the context of this paper there are two main computer technologies/tools that can help in the process of diagnosing Leukemia. The first is big data - where huge amounts of medical data are available to researchers, scientists and engineers to construct effective technologies to diagnose and treat different diseases. Secondly, there is the environments and frameworks available for researchers, scientists and engineers to manipulate the data to obtain the required results. In order to diagnose Leukemia white blood cells, need to be examined. The examiner will need to look for abnormal changes/shapes in white blood cells. By using image processing a white blood samples can be classified according to its type, later it will be checked if it contains any abnormality. The microscope image of a white blood cells sample is shown in blue in **Figure 2** [9].

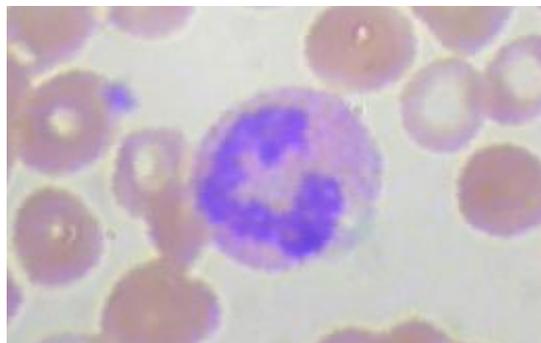


Fig. 2. Blood cells sample - **Neutrophils** cell appears in blue [9].

In previous research, the most common algorithms used to detect and classify microscopic images are the following methods: pre-processing; clustering; morphological filtering; segmentation; feature extraction, classification, and evaluation [7]. These common methods have many disadvantages including long development time, the selection of the features in order to obtain best accuracy; also, it is difficult sometimes to make an accurate decision on whether the cell is abnormal or not. On the contrary, deep learning, such as offered by **Convolutional Neural Network (CNN)** reduces the processing time by eliminating some steps of the process. It discovers high-level features and it attempts to classify images at the same time. It means the convolutional neural network (CNN) could be considered a powerful technology to build a robust image classifier [8]. This paper uses a dataset that containing 10.000 microscope images (JPEG) [9]. The blood dataset is divided into four main types of white blood cells, namely Monocytes, Lymphocytes, Neutrophils, and Eosinophils. The Basophils cell images are not included in this dataset. There are approximately 2500 images for each type [9]. This blood dataset is appropriate to construct an image classifier using several techniques and methods to classify blood images into relevant classes. In addition, the convolutional neural network has the advantage of extracting high-level features from microscope images then to classify them with less time and less code implementation. However, the main challenge of building and designing a convolutional neural network is the high complexity, in terms of number of layers required to extract useful high-level features in the microscope images and how to classify these features. Therefore, the development of the network is challenging [8]. This paper will propose an application using a convolutional neural network, that detects white blood cells from microscope images and then classifies these blood cells into one of the four classes: Class A: Monocytes, Class B: Lymphocytes, Class C: Neutrophils, Class D: Eosinophils. 1

2 Related Work

A Classification Using Convolutional Neural Network in Clinical Decision Support System"[17]. This study proposed a novel approach to classify normal and abnormal cells of blood based on convolutional neural network (CNN). The architecture of their

convolutional neural network is shown in figure 3. Convolutional Neural Network study was conducted in 2017 by Tran Thi Phuong Thanh, Jin-Hyeok Park, Suk-Hwan Lee, Kwang-Seok Moon, and Ki-Ryong Kwon named "Acute Leukaemia

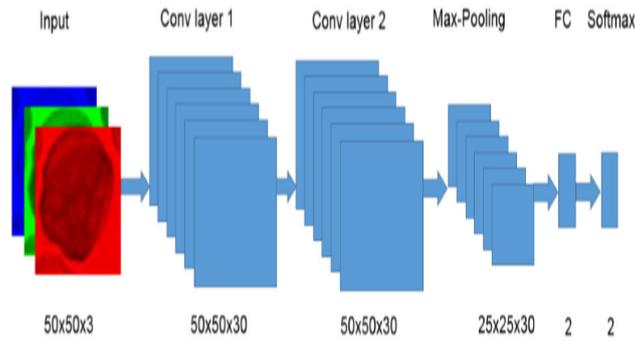


Fig. 3. The architecture of convolutional neural network study in 2017 [17].

The proposed CNN structure contains 5 layers. The first 3 layers are for features extraction, and the other 2 layers are for classifying the features.

The image used as an input was [50x50x3], while the size of the receptive field (also called filter) is 5x5. The stride is 1 and the filter is moved one pixel in each iteration. To control the spatial size of the output image zero-padding equal 2 is used. The filter of the pooling layer is 2x2 with 2 strides in each iteration.

The last 2 layers are: a fully connected layer with 2 neurons and a soft-max layer used for the classification [17]. The dataset used in this study was 108 cell images divided into 49 abnormal cells and 59 normal cells. Table 1 below shows the numbers for the training set and the testing set.

Table 1. The number of training set and testing set [13].

	Training set	Testing set
Normal Blood cell	40	19
Abnormal Blood cell	40	9
Total	80	28

According to the experiment conducted on Matlab in [17], The proposed model resulted in accuracy rate up to 96.7%.

3 Proposed Solution

The proposed solution section describes the algorithms and methods that are used to build the proposed convolutional neural network model. The first stage is Image Pre-processing which contains five methods: Images Read, Convert RGB images to

Grayscale, Image Resize, Image Labeling, Image Pickling. The second stage is a Convolutional Neural Network Model. Figure 4 shows diagram of study stages.

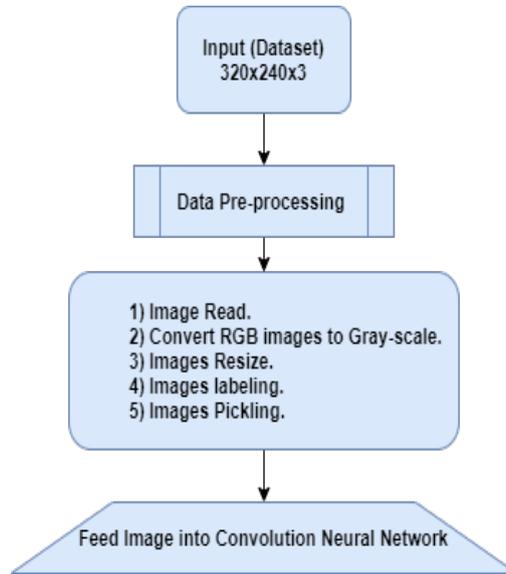


Fig. 4. Proposed solution stages.

3.1 Data pre-processing:

a) Images Read

In the Images Read step the images are read using the code shown in Figure 5, which is self-explanatory.

```
# Path & categorizing|
DATADIR="drive/My Drive/dataset"
CATEGORIES = ["EOSINOPHIL", "LYMPHOCYTE", "MONOCYTE", "NEUTROPHIL"]

#Reading All Images
for category in CATEGORIES:
    path = os.path.join(DATADIR, category)
    for img in os.listdir(path):
        img_array = cv2.imread(os.path.join(path, img), cv2.IMREAD_GRAYSCALE)
        plt.imshow(img_array, cmap="gray")
        plt.show()
        break
    break
```

Fig. 5. Reading images from folder and categorized [10][11].

b) Convert RGB images to Grayscale

Gray-scaling is simply reducing the complexity of the image dimensions from 3D pixel values (Red, Green, Blue) to 2D pixel values (Black, White). Thus, by applying

grayscale conversion on the image it will narrow pixel intensity, which will provide an easier way to deal with the image in the form of 1 for white color and 0 for black color [12]. The grayscale conversion is applied on image as shown in the **Figure 6**.

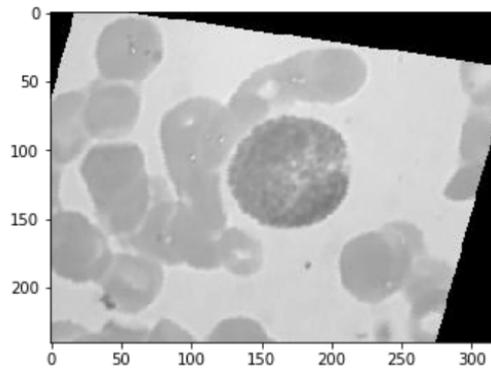


Fig. 6. Blood image sample after been converted to grayscale.

c) Image Resize

In this step we decreased the dimension of the image from [340x240x3] to [50x50x1]. Image resizing is important when we need to decrease the total number of pixels in image.

d) Image labeling

In this step the images are labelled and appended in X(feature) and Y(label) arrays, shown in Figure 7.

```
for features,label in training_data:  
    X.append(features)  
    y.append(label)]  
X = np.array(X).reshape(-1,IMG_SIZE,IMG_SIZE,1)
```

Fig. 7. Labeling the training set using the append function.

e) Image Pickling

After image labeled and appending, the data is pickled and saved to an external file (using the Pickle Library). The main reason of pickling data is to convert a python object into byte stream to be stored in external file or external database. Furthermore, pickling images is helpful for transporting data over the network, since we use a TensorFlow as backend engine. **Figure 8** shows the pickling of the training dataset [13].

```

pickle_out = open("X.pickle","wb")
pickle.dump(X,pickle_out)
pickle_out.close()

pickle_out = open("y.pickle","wb")
pickle.dump(y,pickle_out)
pickle_out.close()

pickle_in = open("X.pickle","rb")
X = pickle.load(pickle_in)
    
```

Fig. 8. The pickling of the training dataset [13].

3.2 Convolutional neural network architecture

After preprocessing has been applied on input images. The images would be entering in a convolutional neural network. The network contains **five** layers. **Four** layers for extracting features from an input image. Last layer (Output Layer) to classify the features gained from the previous four layers. The dimensions of the input image are [50x50x1]. The dimensions of the convolutional filter are 3x3. The Max-Pooling filter is 3x3. The stride is 1. Then moving filter one pixel at a time with zero paddings [8]. The processes of this network are divided into two parts: The **first part** is the layers for extracting features of the input image resulted from applying three operations on each layer (1-Convolutional Operation, 2- Activation, 3- Max-Pooling). The **second part** is the output layer contains (Flatten / Dense / softmax) for image classification as shown in **Figure 9**.

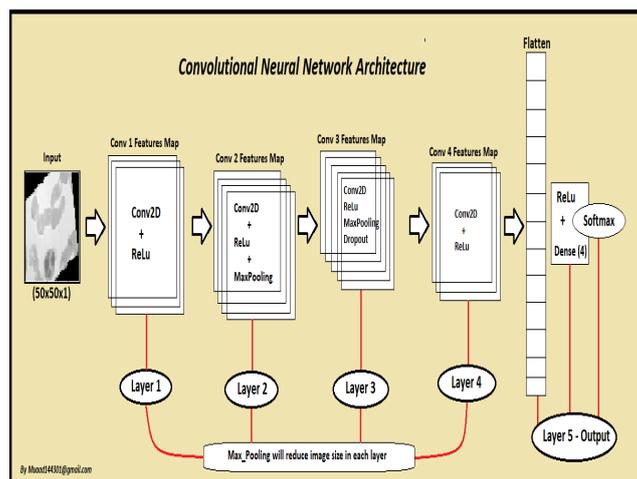


Fig. 9. The architecture of convolutional neural network model.

Table 2. The analysis of each layer in convolutional neural network

Network Layers	Operations in the Layers	Part
Layer 1	Convolutional Operation / ReLu	Part 1: Features Extraction
Layer 2	Convolutional Operation / ReLu/ Max Pooling	Part 1: Features Extraction
Layer 3	Convolutional Operation / ReLu / Max Pooling / Dropout	Part 1: Features Extraction
Layer 4	Convolutional operation / ReLu	Part 1: Features Extraction
Layer 5	Flatten/ Dense/ Softmax	Part 2: Classification

Part 1 of the network: which are the features learning through 4 convolutional layers. The convolutional layer contains Convolutional Operation, Activation and Max Pooling respectively.

Convolutional operation is a dot operation on the receptive fields of the image which is a multiplication of the 3x3 kernel (filter) on the image and slide it across the image iteratively (Eq3.2) [14][15].

$$Eq.2 \rightarrow h(x) = f * g = \sum_{-\infty}^{\infty} f(x - a) g(a) Da = F^{-1}(\sqrt{2\pi} F|f|F|g|)$$

The figures below are an explanation of the convolutional operation with kernel 3x3, subsequently multiply 3x3 this kernel by the 5x5 green matrix in figures 10, 11 and 12. In this example there are nine iteration to reach the final result (3x3 matrix) of the convolutional operation with stride = 1, moving up and down. The result of each iteration is a summation of the multiplication of the respective field (yellow matrix). In the figures below:

1. Figure 10 shows the receptive field movements in the first iteration.
2. Figure 11 shows the receptive field movements in second iteration.
3. Figure 12 shows the receptive field movements in the ninth iteration.

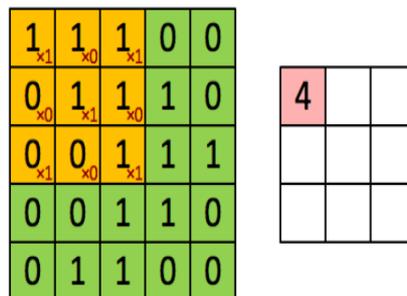


Fig. 10. Convolutional multiplication in iteration =1 [15].

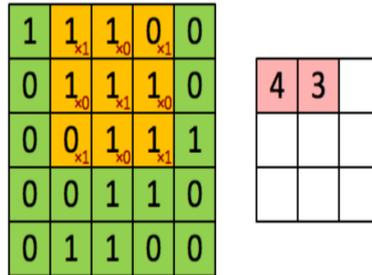


Fig. 11.Convolutional multiplication in iteration =2 [15].

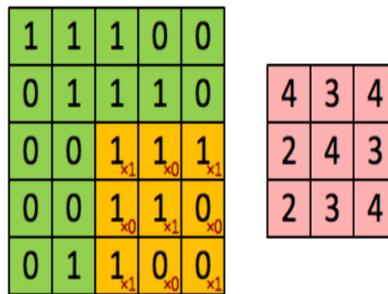


Fig. 12.Convolutional multiplication in iteration =9 [15].

Activation or normalization (ReLU): ReLu is a specific kind of activation function. The convolutional operation may result some negative value in the matrix image. The activation function (ReLU) will turn all negative value of the image to zero. The purpose of this kind of normalization is to make our model as a nonlinear model and able to learn [15].

Max pooling: The max pooling function is used for reducing the dimension of the image by taking the maximum value of the receptive fields of image which is 2x2. **Figure 13** shows that in the first iteration of pooling it takes a 35 as maximum value from the set of (35,19,13,22). And the second iteration It takes a 25 from the set of (19,25,22,16), and so on till reach the end of the input file [15].

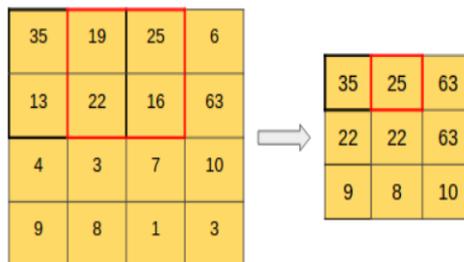


Fig. 13.Taking max value of the 2x2 respective fields [15].

After the three operations has been applied in the first three layers, the output of layer 3 will be the input in the layer 4 which is Probability Conversion layer.

Part 2 of the network: The second part of the convolutional neural network which is a classification image into their related class (**Neutrophils, Eosinophils, Monocytes, Lymphocytes**). In this paper, we used the flatten function followed by Dense (fully connected layer) and end up with SoftMax function respectively. The main purpose of using these three functions in the convolutional neural network is classifying the extracted features from fourth layer. Specifically, the flatten function is used to get a copy of a given array collapsed into one-dimension array. Dense layer is the regular deeply connected neural network layer. Softmax function is used just to map the non-normalized output of a network to a probability over predicted output classes [16].

4 Results

The dataset used in this paper is provided by Paul Mooney, Developer Advocate at Kaggle [9]. During the experiments, the hyperparameter of the convolutional neural network was in continual change between each experiment. Hyperparameters such as **training set quantity, testing set quantity, input size, data label, kernel size in each layer, pooling size in each layer, epoch, activation type and its order in the network, number of layers**. Therefore, the dataset quality, labels, quantity, and size are considered as the main factors that have an impact on the accuracy of white blood cells classification. The dataset is divided into four classes which are the four types of white blood cells (Neutrophils, Eosinophils, Monocytes, and Lymphocytes) [9]. The quantity and dimension of the training set and testing set is in **Table 3**.

Table 3. The dataset quantity and dimension [9]

Image Class	Training set	Testing set	Image dimension
Neutrophils	2,499	48	320 x 240
Eosinophils	2,497	38	320 x 240
Monocytes	2,478	33	320 x 240
Lymphocytes	2,483	6	320 x 240

The experiments were conducted on Google Colaboratory as notebook, Anaconda as python distributor, Keras library, and TensorFlow as backend engine. After dozens of experiments, the accuracy rate of the recognition of types of white blood cells by our proposed Convolutional Neural Network model achieved 96.78%.

5 Conclusion

Due to its efficiency in diagnosing Leukemia at early stages, the convolutional neural network would be of a great value to the medical diagnostic system used to detect Leukemia in the blood cells. The availability of medical data is the main factor used to improve the complete diagnostic system. The Convolutional Neural Network is an

efficient method of image classification because it eliminates most of the steps used in the traditional image classification techniques. The detection and classification of white blood cells is dependent on two factors. The first factor is the provided dataset by the medical unit, the second is the development of the convolutional neural network to handle and classify the dataset. The accuracy of proposed program is 96.78% which is highly reliable in white blood cells classification. In the future work, abnormal white blood cells will be used to train and optimize the proposed CNN model.

6 Future Work

The network could be optimized to accept better data sets (also with more/accurate blood cells). In future work, we would like to use new dataset include abnormal white blood cells to complete the diagnostic system.

7 References

- [1] Amanda Chan, LiveScience > Health, September 10, 2010 07:44am, "The 10 Deadliest Cancers and Why There's No Cure". <https://www.livescience.com/11041-10-deadliest-cancers-cure.html>.
- [2] Leukemia & Lymphoma Society, "Leukemia Definition and types". <https://www.lls.org/leukemia>.
- [3] Blood Cells Types, Wikipedia. https://en.wikipedia.org/wiki/Blood_cell.
- [4] Health Encyclopedia, University of ROCHESTER Medical Center 2019. What are White Blood Cells: Types of white blood cells. <https://www.urmc.rochester.edu/encyclopedia/content.aspx?ContentID=35&ContentTypeID=160>
- [5] By Mayo Clinic Staff, Mayo Clinic, "Leukemia Diagnosis". <https://www.mayoclinic.org/diseases-conditions/leukemia/diagnosis-treatment/drc-20374378>
- [6] Sabtu, March 2015. "Leukemia of Th Blood Cells". <http://informaticsofhealth.blogspot.com/2015/03/leukemia-cancer-of-blood-cells.html>.
- [7] Himali P. Vaghela, Hardik Modi, Manoj Pandya, M. B. Patdar PhD. "Leukemia Detection using Digital Image Processing Techniques: International Journal of Applied Information System (IJ AIS) – ISSN: 2249-0868. Foundation of Computer Science FCS, New York, USA. Volume 10 – No.1, November 2015 – www.ijais.org. <https://doi.org/10.5120/ijais2015451461>
- [8] T.T.P Thanh, Caleb Vununu, Sukhrob Atoev, Suk-Hwan Lee, and Ki-Ryong Kwon "Leukemia Blood Cell Image Classification Using Convolutional Neural Network: International Journal of Computer Theory and Engineering, Vol. 10, No. 2, April 2018. <https://doi.org/10.7763/ijcte.2018.v10.1198>
- [9] Paul Mooney, Developer Advocate at Kaggle. Boulder, CO, USA. [Online] Available <https://www.kaggle.com/paultimothymooney/blood-cells>.
- [10] The Python Software Foundation. 2019. Miscellaneous operating system interfaces: The Python Standard Library >> Generic Operating System Services.
- [11] Alexander Mordvintsev & Abid K. 2013. OpenCV-Python Tutorials: Gui Features in OpenCV >> Getting Started with Images.
- [12] Arthur Coste, University of Utah. Gray scale digital image. http://www.sci.utah.edu/~acoste/uou/Image/project1/Arthur_COSTE_Project_1_report.html.

- [13] By Ashutosh Agrawal on Tuesday, November 18th, 2014 "Understanding Python pickling and how to use it securely". <https://www.synopsys.com/blogs/software-security/python-pickling/>
- [14] Tim Dettmers, 2015 "Understanding Convolution in Deep Learning". <https://timdettmers.com/2015/03/26/convolution-deep-learning/>
- [15] Siraj Raval, San Francisco, CA, 2019 "Convolutional Neural Network. ipynb". https://github.com/IIISource/Convolutional_neural_network/blob/master/convolutional_network_tutorial.ipynb. https://doi.org/10.1007/978-1-4842-5572-8_6
- [16] Keras Documentation >> Activation <https://keras.io/activations/>
- [17] Thanh.TTP1, Giao N. Pham1, Jin-Hyeok Park1, Kwang-Seok Moon2, Suk-Hwan Lee3, and Ki-Ryong Kwon1. "Acute Leukemia Classification Using Convolutional Neural Network in Clinical Decision Support System".

8 Authors

Professor Samir Abou El-Seoud is a German computer scientist, received his BSc degree in Physics, Electronics and Mathematics from Cairo University in 1967, his Higher Diploma in Computing from Technical University of Darmstadt (TUD) /Germany in 1975 and his Doctor of Science from the same University (TUD) in 1979.

Field of study: Scientific Computations and Parallel Algorithms. Research interests: Computer Aided Learning, Parallel Algorithms, Mobile Applications, Augmented Reality, Cloud Computing, IoT, Numerical Scientific Computations and Computational Fluid Mechanics. Email: Samir.elseoud@bue.edu.eg

Maid Hammuda Siala is a BUE graduate student and currently works as a system analyst at the United Covers Industrial, Cairo, Egypt. Email: Muaad144301@bue.edu.eg

Gerard McKee is a graduate of the University of Manchester, UK, receiving both his BSc in Electronics and PhD from the same. He is a member of the IEEE, IET, ACM and AAAI and a Fellow of the British Computer Society. His research interest is in the general areas of robotics and artificial intelligence, with particular focus on networked robotics, space robotics and human-robot interaction. His recent research has focused on swarm robotics systems and related aspects in high performance computing. Email: Gerard.McKee@bue.edu.eg

Article submitted 2020-05-09. Resubmitted 2020-06-14. Final acceptance 2020-06-14. Final version published as submitted by the authors.