

# Laser Marks Classification for Retinal Images based on Convolutional Neural Network

Mustafa Ali Abuzaraida, Osama Mohamed Elrajubi

**Abstract:** Recently, deep learning approaches have been getting more attention in many research fields. Medical imaging field has been attracting widely by deep learning techniques. An example of this field categories are images segmentations, images registration, images classification and retrieval of images database. This paper is presenting a number of experiments to classify rental images using Convolutional Neural Networks (CNN). These images of retinal could contain laser marks which left by the action of the laser on the surface of the retina. (CNN) is defined as trainable multi-stages architecture composed of multiple stages. The inputs and outputs of each stage are a set of arrays which called the feature maps. For the outputs, every feature map is representing a unique feature which extracted from all the regions which located on the input. Basically, every stage is consisted of three layers which are: a filter bank, a non linearity, and a layer of feature pooling. However, the classic (CNN) is normally consisting of three or less number of layers. The results accuracy were appropriate of more than 90%. As a summary of this paper, a number of considerations are listed for possible improvements and future developments.

**Index Terms:** machine learning, image processing, deep learning, classification, rental images; laser marks.

## I. INTRODUCTION

Proliferative diabetic retinopathy (PRD) progress could be delayed by recourse to scatter laser photocoagulation treatments that destruct retinal tissue alleviating retinal hypoxia conditions and thus slowing the neovascularization processes typical of PRD [1], [2]. In some situations it is important to detect telltale signs of previous photocoagulation treatments, such as those visible in Fig. 1, using automatic image analysis. This is the case of screening studies where a general call for screening subjects can result in patients that were subject to these treatments but don't know or can't recall having undergone the treatment showing up for the study. These events have two undesirable outcomes: the first and most serious is that the patient can again be scheduled for further treatment and the second is that the scars left by the treatment can interfere negatively with the operation of further analysis of the fundus image by automatic diagnostic tools, leading for instance to false detections of exudates or other signs of eye pathology. Thus, it is important to be able to do automatic detection the

presences of scars left by photocoagulation retinal treatments. In this study an algorithmic solution to this problem is proposed, which is based on Convolutional neural networks (CNN).

The remain of this paper is structured as following; In the second Section, the related works in the field is highlighted and then theoretical background of (CNN) is explained in section 3. In section 4, the network architecture is presented while section 5 describes the details of the experiments and obtained study results. Section 6 presents the conclusion of this paper.



Fig. 1. Fundus image showing laser scars.

## II. RELATED WORKS

Even with the interests in being able for detecting laser marks which left by photocoagulation, minor number of works were published to describe the approaches for auto-detections. Dias et al [3] describe a system which is upgrading of a previously introduced an algorithm of retinal quality assessment [4] for detecting the presences of laser marks into digital images of fundus. In [5], Syed et al. introduced a method which on classification using support vector machines (SVM). The input is consisting for features of three color domain, two texture domain features and four shape features. The classifier was trained using 70% of a proprietary dataset containing 380 images and evaluation of classification performance used the remaining 30%. In [6] Tahir et al. introduced a classification approach using Minimum Distances Clustering for deciding if the image characterized using features of

Revised Manuscript Received on September 22, 2019.

Mustafa Ali Abuzaraida, School of Computing, College of Arts and Sciences, Universiti Utara Malaysia, Sintok, Kedah, Malaysia.

Osama Mohamed Elrajubi, Telecommunications and Networks Department, Faculty of Information Technology, Misurata University, Misurata, Libya.

Retrieval Number: F10300986S319/2019@BEIESP

DOI: 10.35940/ijeat.F1030.0986S319

Published By:  
Blue Eyes Intelligence Engineering  
& Sciences Publication



ten scalars has signs of laser treatments. The features used included the measuring of spatial compactness of the presumed laser mark blobs. They used another nine values to quantify the intensity and color like the maximum value for saturation and hue, maximum and mean for luminance, and the maximum and mean values of specific color channels, namely the red and green channels.

For (CNN), several studies were used this classifier recently in different research areas such as: [7] Dong et al. studied Support Vector Machine and three types of (CNN) on a group of images of malaria infected cells. The types of (CNN) were namely: LeNet, AlexNet and GoogLeNet. Findings of this study showed that all those deeply (CNN) could achieve classification accuracy rate of up to 95% better than support vector machine with about 92% accuracy rate. In [8] Du et al. proposed a Network of (CNN) for recognizing not labeled right and left images of iris using iris recognition system. The proposed network performances was evaluated using small and large scale iris datasts. Experimental result shows that the results of the approach in a classification performance above an accuracy rate 97%.

For other medical issues, (CNN) was used to classify different types of diseases and health care issues. As for Malaria parasite detection in thin blood smear images was classified by (CNN) by Rajaraman et al [9]. Also, Wang et al, [10] used (CNN) to propose an efficient framework for interactive 2D/3D medical image segmentation.

### III. THEORETICAL BACKGROUND OF CONVOLUTIONAL NEURAL NETWORKS

This kind of neural networks can be defined as an architecture of trainable multi-stages composed of multiple stages. The inputs and outputs for every stage are the sets of arrays which named the feature maps. For the outputs, every feature map is representing a unique feature which extracted at all areas which located on the input. Every stage is consisted from 3 layers: a filter bank, a non linearity, and a layer of feature pooling. A classic (CNN) is consisted of three or less number of layers. These three layers are followed by the classification modules. Every type of layer is described of the image recognition case as follows:

#### A. Image Input Layer

This layer is defining the input images size of any (CNN). It contains also the images raw pixels values. The image size corresponds to the weight, height, and the color channels number of input image. For instance, for grayscale images, the channels number is just one, and three for a color images. The input layer be able to normalize the data by using subtracting the mean if the image in training set from every input images.

#### B. Convolutional Layer

This layer is consisting a number of neurons those connected to sub area of the input or the output images of the previous layer. This layer can learn each feature localized using these areas while scanning through images. This layer used a number of weights sets which can applied to an area

within the image. This filter could move all along the input image horizontally and vertically. This step is repeated using the same computation for each area to do convolving the inputs. A stride is the size of this step with its moves. These local areas which the connected neurons to may overlaped depending on the filter-Size and the value of the Stride.

The weights number that used for the filter:

$$w * h * c \tag{1}$$

Where w for the width, h for the height of the size of each filter, and c is the channels number in the input image. In this point, channels are 3 for color images as an example. The filters number can determine the channels number for the convolutional layer output.

As mentioned that the filters move along the input images, it is by using the similar weights set and convolution, to form the map of features. Therefore, the feature maps number of this layer has is the same number of channels. Any feature map got different sets of bias and weights. Thus, the parameters total number of the convolutional layer as:

$$((w * h * c + 1) * Filters Number) \tag{2}$$

Where one is the bias value. Another points should be mention in this layer is Padding. It is basically counted by adding zeros columns or rows to the input image borders. It can help to control the output size of the layer.

Then, the output of width and height of this layer can computed by:

$$(Input Size - Filter Size + 2 * Padding) / Stride + 1 \tag{3}$$

To insure the whole image is fully covered, this value has to be an integer for the whole image. The software will ignore the rest parts of the image alongside the bottom and right edges in the convolution layer. This will happened if these parameters combination do not help the image to become fully covered.

The total neurons number (output size) in the feature map is the production for the outputs width and height. The neurons entire number this layer is:

$$Number of Filters * Map Size \tag{4}$$

Updating the learning rate and regularization parameters of convolutional layer is done by using the related value name pair arguments while defining this layer.

Any of (CNN) could consist of one or more convolutional layers. The quantity and complexity of data is the main reason to use multiple convolutional layers.

#### C. Batch Normalization Layer

These layers can be used between convolutional and ReLU layers to speed up the training of the network and for reducing the sensitivity of network initialization. The layer do normalization of activations of every channel by subtracting the mini batch average and divide it by the standard deviation of the mini batch. Then, this layer do shifting the input using an offset  $\beta$  and do scaling for it using a scale factor  $\gamma$ . These factors are considered as learnable parameters those are updated while training of the network.

This layer normalizes the activations and propagating of gradients in a neural networks. This to ease the optimization problem of the network training. Increasing the learning rate can take full advantage of using this layer. The network could learn faster if the parameter updates is larger to overcome the optimization problem easily.

**D. Rectified Linear Unit (ReLU) Layer**

Layers of Convolutional and batch normalization are typically followed by a nonlinear activation function like a "rectified linear unit" particular by a (ReLU) layer. This layer is used to perform threshold operations for every element, for every input value is negative is equalized to zero, as:

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

(5)

This layer does not modify the size of its inputs.

Standard ReLU layer has a number of extensions that perform different operations and could improve the performances of some applications.

**E. Cross Channel Normalization Layer**

This layer also called (Local Response Normalization) . It used to perform the normalization of "channel-wise" local response. This layer usually comes after the ReLU activation layer. It is replacing every element by a normalized value which obtained by the elements via a definite number of surrounded channels. It computes a normalized value  $x'$  of element  $x$  in the input, using:

$$x' = \frac{x}{(K + \frac{\sum_{ss} x^2}{\text{windowChannelSize}})^\beta}$$

(6)

Where  $K$ ,  $\alpha$ , and  $\beta$  are the hyper parameters in the normalization, and  $ss$  is the sum of squares of the elements in the normalization window [11].

**F. Max- and Average-Pooling Layers**

These layers are following any convolutional layer for down sampling. It is used for reducing the connection numbers to the next layers (often for fully connected layer). These layers do not perform to learn themselves. However, reducing the number of parameters to be learned can help to reduce over fitting in the following layers.

The max pooling layers return the greatest values of rectangular area of the layer input. The rectangular size area can be found out by the Max Pooling Layer poolSize argument. For instance, if Pool Size equals (3,2), then the layer is returning the greatest value in areas of height 3 and width 2.

It is the same process for the Average Pooling layers. Their outputs are the average values of rectangular are of the layer input. The rectangular area size is determined by the Pool Size argument for the average Pooling Layer. These layers of both type are used scanning through the input image horizontally and vertically and control the step sizes using the 'Stride' name value. Here, If the Pool Size was chosen less than or same to the Stride, then the pooling areas will not be overlapped.

For Stride and Pool-Size are equally which consider as

non overlapping areas. So, if the pooling layer input is  $n$ -of- $n$ , and the pooling area size is  $h$ -of- $h$ , then this layer down samples the areas by  $h$  [12]. Then, the max pooling or average pooling layers outputs for single channel of a convolutional layer as:

$$"n/h-by-n/h" \tag{7}$$

The output for overlapping areas of a pooling layer as:

$$"(Input\ Size - Pool\ Size + 2*Padding)/Stride + 1" \tag{8}$$

**G. Dropout Layers**

These layers set in random way the input elements of the layer into zeros by a specified probability. Even though, this layer output is same to its input, this operation is corresponding for temporarily dropping in random way to choose the unit and for all its connections from the network in the training duration. Here, for every new element input and train the network in random way to select a neurons subset to form a different layer architecture. The architectures of the layer use common weights. However, because of the learning process is not depending on specific neuron and connection, the layer of dropout may help to prevent over fitting [11], [13]. It is the same to the layers of max- or average-pooling where no learning takes place in these layers.

**H. Fully Connected Layer**

These layers can be one or multiple fully connected layers. These type of layers follow the convolutional layer.

All neurons in this layer are connected to all the neurons of the last layer. Fully connected layer combines all learned features by the previous layers. These features were used to identify the larger patterns and then to classify the images as for the classification problems. The last fully connected layer output size arguments to the classes numbers for the dataset. However, the output size has to be same with the response variables number for regression problems.

**I. Output Layers**

Soft Max and Classification Layers:

These layers are used for classification tasks. These both layers are following the last fully connected layer.

Here, the activation function of the output unit is the Soft Max function:

$$y_r(x) = \frac{\exp(a_r(x))}{\sum_{j=1}^k \exp(a_j(x))} \tag{9}$$

"Where  $0 \leq y_r \leq 1 = \sum_{j=1}^k y_j = 1$ "

This function presents the output unit activation function after the last fully connected layer for multi classes classification issues:

$$P(c_r|x, \theta) = \frac{P(x, \theta|c_r)P(c_r)}{\sum_{j=1}^k P(x, \theta|c_j)P(c_j)} = \frac{\exp(a_r(x, \theta))}{\sum_{j=1}^k \exp(a_j(x, \theta))}$$





$$\text{"Where } 0 \leq P(c_r|x, \theta) \leq 1 \text{ and } \sum_{j=1}^k P(c_j|x, \theta) = 1 \text{"}$$

The soft max function is also known as the normalized exponential. It is also considering the multi class generalization of the function of the logistic sigmoid [14].

The output of classification layer has to follow the soft max layer. For the output of classification layer, training Network take the every value from the soft max function and then assigning every input to one of the k mutually unique class by using the function of cross entropy for 1-of-k coding scheme [15]:

$$E(\theta) = -\sum_{i=1}^n \sum_{j=1}^k t_{ij} \ln y_j(x_i, \theta), \tag{11}$$

Where  $t_{ij}$  are the indicators those the  $i$ th samples belong to the  $j$ th classes.

$\theta$  is the parameter vector.

$y_j(x_i, \theta)$  is the sample output of  $i$ , which in this case, is the soft max function value. Here, the value is the probability which the network associates the  $i$ th input with class  $j$ .

**J. Regression Layer**

CNN can also be used to handle regression issues. In regression category, the output (target) variable comes in continuous formatting. In this matter, this layer has to follow the last fully connected layers. The mean squared error consider as the default loss functions of this layer as:

$$MSE = E(\theta) = \sum_{i=1}^n \frac{(t_i - y_i)^2}{n}, \tag{12}$$

Where  $t_i$  is the target variable, and  $y_i$  is the prediction of the network of the responded variable of the observation  $i$ .

The first step of training and creating a new (CNN) is by defining the architecture of the network. The (CNN) architecture can be different and it depends on the numbers and types of the layers which included. The number and type of any layer can depend on the data and the application. For instance, if handling categorical responses, it has to get a classification functions and the classification layers. However, if the responses are continuous, it has to have a regression layers at the end of the networks. To learn a small quantity of images in gray scale data, a small network size with just only less than three convolutional layers may be more sufficient. In contrast, for complexes data with billions of colored images, it may need a multiple convolutional and fully connected layers which is more complicated network.

**IV. PROPOSED NETWORK ARCHITECTURE**

In this study, CNN was used to classify laser marks left by photocoagulation on retinal images to show evidence of previous photocoagulation laser treatments.

**A. Datasets**

Throughout the development and performance testing stages of the work two sets of retinal images were used. These datasets were collected in University of Coimbra, Portugal. The full description of these datasets are explained in details in [10].

**B. Proprietary Dataset 1**

Made up of 315 retinal images annotated by optometrists as having laser marks, from an continuing DR screening program in the Portugal centre region.

**C. Pr-oprietary Dataset 2**

Composed of 415 retinal images with no laser marks were classifying with R0 grade (complete absence of lesions) by a human graders from "Association for Innovation and Biomedical Research on Light" (AIBILI).

Thirteen layers of CNN were chosen to be used in these experiments as follows:

**Layer 1:** The size of input image is 300 pixels length and 300 pixels width with RGB levels. Zero center normalization was used in this layer.

**Layer 2:** The second layer of the network is Convolution layer. In this layer, the used filters is 90 filters with a stride values of 1 and padding value of 0. The size of each scan is 9 pixels.

**Layer 3:** Standard ReLU layer was taken here to improve the performances of second layer.

**Layer 4:** Max Pooling layer was chosen to be the fourth layer of the proposed network. The size of scanning is 2 pixels. Stride and padding were used in this layer with values of 2 and 0 respectively.

**Layer 5:** Another Convolution layer is used for this layer with 5 pixels as layer size. However, 70 filters were used to process the output of the layer 4.

**Layer 6:** Another ReLU layer was chosen here to improve the performances of layer 5.

**Layer 7:** Max Pooling Layer was repeated with the same details to process the output of layer 6.

**Layer 8:** Another Convolution layer is used with 200 filters and 3 pixels as layer size were chosen.

**Layer 9:** Another ReLU layer was chosen here to improve the performances of layer 8.

**Layer 10:** Max Pooling Layer was repeated with the same details to process the output of layer 9.

**Layer 11:** In this layer, two fully connected layers were chosen. A combination of all features which learned using the previous layers across the images for identifying the larger pattern or patterns.

**Layer 12:** Here a softmax function was applied after the previous fully connected layer for multi class classification issues.

**Layer 13:** A classification output layer was following the soft max layer. For this layer, training the Network took the values from the function of the soft max and then assigned every input to one exclusive mutually class or multi-classes by using the function of the cross entropy.

**V. DETAILS OF THE EXPERIMENTS AND RESULTS**

Several experiments were conducted in this study. However, these experiments are conducting to improve the results of the previous study [16]. In this study, 13 layers were used instead of 8 layers of the previous study. The



experiments of this study are explained as following:

**A. Finding the Best MaxEpoch Value**

"MaxEpoch" value is the number of training iterations. For example, if MaxEpoch is 2 means that the network will get the trained data twice and so forth. In this study, the system uses 80% of 730 input images for each iteration to

train the network and the rest was used for testing it. The highest accuracy was determined by choosing different value of "MaxEpoch" which are (1,2,5,10,20,30,40,50). The best value was selected of the average after conducting five experiments for each values as shown in Table I.

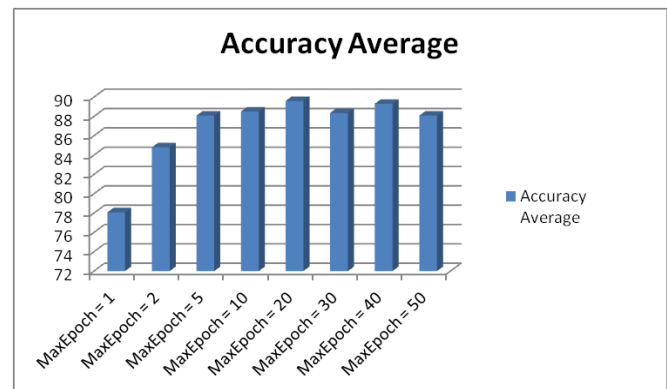
**Table- I.** Calculating the Best Maxepoch Value

Max Epoch	1	2	3	4	5	Accuracy Average
1	78.08	78.08	78.08	78.08	78.08	78.08
2	85.62	84.93	84.93	84.25	84.25	84.80
5	88.36	87.67	88.36	87.67	88.36	88.08
10	88.36	88.36	88.36	89.04	88.36	88.50
20	90.41	89.73	89.04	89.04	89.73	89.60
30	88.36	88.36	88.36	88.36	88.36	88.36
40	89.04	90.41	89.04	87.67	90.41	89.31
50	89.04	87.67	88.36	87.67	87.67	88.08

As seen from Table I, 40 experiments were conducted in this study. For each value of MaxEpoch, five experiments were conducted to get the average of the output layer accuracy. However, the accuracies range of each MaxEpoch experiments have a gap by different values especially for MaxEpoch 1, and 2 values. This gap could be happened because of changing of dividing the training and testing images for less number of experiments like MaxEpoch values equal to 1.

It is noticeable that the gap was smaller for MaxEpoch values which are greater than 5. Here, the number of trained images was more than the first two experiments so this make an important reason that the network could learn better from many samples rather than less samples.

However, it is clear from Table I that the accuracy is improved by MaxEpoch value of 20 by an average accuracy of 89.60 %. Then, the accuracy starts declining by using value greater than 20. It is recommended that using values greater than 20 will cost time of training process without any improvement. Fig. 2 illustrates the output of this experiment.



**Fig. 2.** Finding the best MaxEpoch value

**A. Finding the Best Accuracy Rate**

After getting the best "MaxEpoch", the network was tested by different number of input images. The system has shown high accuracy rates with less number of images which used as training dataset. However, using more images gave better results as shown in Table II.

**Table- II.** Experiments accuracy rates with MaxEpoch = 20

Ratio of training Dataset	Number of images used for training	Accuracy when MaxEpoch = 20
0.1	73	76.52
0.2	146	78.42
0.3	219	80.78
0.4	292	84.25
0.5	365	85.44
0.6	438	85.96
0.7	511	86.24
0.8	584	89.04
0.9	657	87.67

**VI. CONCLUSION**

In clinical practice, minor operations using laser technology on retinal surfaces are used to overcome the some lack in vision causes. In the other hand, these operations can

cause misbehavior of automated retinal diagnostic systems. In this paper, Some images of retinal contain laser marks left by the action of the laser on the surface of the retina were used to train and test a classifier based on (CNN).

Several experiments were conducted using different number of trained images. The results were promising of almost accuracy rate average up to 89.60 %. comparing the results with the previous study [16], the average of the accuracy was slightly dropped. The previous study came out with an accuracy average of 91.65 % with just eight layers only. For future plan, more experiments can be conducted using extra number of layers to improve and enhance the rate of the system accuracy rate.

## REFERENCES

1. Early Treatment Diabetic Retinopathy Study Research Group, "Treatment techniques and clinical guidelines for photocoagulation of diabetic macular edema: Early Treatment Diabetic Retinopathy Study report number 2," *Ophthalmology*, vol. 94, no. 7, 1987, pp. 761-774.
2. J. Cai, and M. Boulton, "The pathogenesis of diabetic retinopathy: old concepts and new questions," *Eye*, vol. 16, 2002, pp. 242-260.
3. J. M. P. Dias, C. M. Oliveira, and L. A. da Silva Cruz, "Detection of laser marks in retinal images," In *Proceeding of the IEEE 26th International Symposium on Computer-Based Medical Systems (CBMS)*, 2013, pp. 532-533.
4. J. M. P. Dias, C. M. Oliveira, and L. A. da Silva Cruz, "Retinal image quality assessment using generic image quality indicators," *Information Fusion*, vol. 19, 2014, pp. 73-90.
5. A. M. Syed, M. U. Akbar, M. U. Akram, and J. Fatima, "Automated laser mark segmentation from colored retinal images," In *Proceeding of the Multi-Topic Conference (INMIC), 2014 IEEE 17th International*, 2014, pp. 282-286.
6. F. Tahir, M. U. Akram, M. Abbass, and A. A. Khan, "Laser marks detection from fundus images," In *Proceeding of 14th International Conference on the Hybrid Intelligent Systems (HIS)*, 2014, pp. 147-151.
7. Y. Dong, et al., "Evaluations of deep convolutional neural networks for automatic identification of malaria infected cells," In *Proceeding of the IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, 2017, pp. 101-104.
8. Y. Du, T. Bourlai, and J. Dawson, "Automated classification of mislabeled near-infrared left and right iris images using convolutional neural networks," In *Proceeding of the IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, 2016, pp. 1-6.
9. S. Rajaraman, S. K. Antani, M. Poostchi, K. Silamut, M. A. Hossain, R. J. Maude, S. Jaeger, and G. R. Thoma, "Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images," *PeerJ*, vol. 6, 2018, e4568.
10. G. Wang, W. Li, M. A. Zuluaga, R. Pratt, P. A. Patel, M. Aertsen, and T. Vercauteren, "Interactive medical image segmentation using deep learning with image-specific fine tuning," *IEEE transactions on medical imaging*, vol. 37, no. 7, 2018, pp. 1562-1573.
11. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," In *Proceeding of the Advances in neural information processing systems*, 2012, pp. 1097-1105.
12. J. Nagi, et al., "Max-pooling convolutional neural networks for vision-based hand gesture recognition," In *Proceeding of the 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, 2011, pp. 342-347.
13. N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of machine learning research*, vol. 15, 2014, pp. 1929-1958.
14. C. M. Bishop, "Periodic Variables," *Pattern recognition and machine learning*, vol. 1, 2006.
15. J. G. R. A. e Sousa, C. M. Oliveira, and L. A. da Silva Cruz, "Automatic detection of laser marks in retinal digital fundus images," In *Proceeding of the 24th European Signal Processing Conference (EUSIPCO)*, 2016, pp. 1313-1317.
16. O. M. Elrajubi, M. A. Abuzaraida, A. M. Zeki, "Retinal Image Laser Marks Detection using a Convolutional Neural Network.," *Paper presented at 2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT2018)*, Bahrain, University of Bahrain, November 18-19, 2018.

## AUTHORS PROFILE



**Mustafa Ali Abuzaraida** received the bachelor's degree in (Computer Science) from Misurata University, Misurata, Libya in 2001. He obtained his master degree in (Intelligent System) from University Utara Malaysia (UUM) in 2006 and a Ph.D. in (Computer Science) from International Islamic University Malaysia (IIUM) in 2015.

His research interest is on Image processing, Data science, Data mining, Artificial intelligence application, and Natural Language Processing (NLP) mainly on text normalization and sentiment analysis and recognizing online handwritten text. Currently, he joined School of Computing, College of Arts and Sciences, University Utara Malaysia as visiting senior lecturer.



**Osama Mohamed Elrajubi** received the BSc de-gree in Electronic Engineering – Computer division from the College of Industrial Technology, Libya in 2002, and MSc degree in Computer Sciences from the Academy of Postgraduate Studies -Tripoli, Libya in 2009. He also hold an MSc degree in Computer Engineering from Nottingham

Trent University, UK in 2011. He is currently a Lecturer at faculty of Information Technology, at Misurata University, Libya. His research interests include Digital Image Processing, Natural Language Processing, and Computer Networks.