

Improving Laser Mark Detection for Retinal Images based on the AlexNet Model



Mustafa Ali Abuzaraida¹, Osama Mohamed Elrajubi²

¹School of Computing, College of Arts and Sciences,

Universiti Utara Malaysia, Sintok, Kedah, Malaysia, abuzaraida@uum.edu.my

²Telecommunications and Networks Department, Faculty of Information Technology,

Misurata University, Misurata, Libya, osrjubi@yahoo.com

ABSTRACT

Recently, the field of deep learning has received increased attention due to its high accuracy. A common deep learning technique is Convolutional Neural Networks (CNN), which is as a construction of trainable multi-stages using multiple phases. In this paper, we use a type of CNN called AlexNet to classify human retinal images into 'normal' or 'have been treated using photocoagulation laser treatments' classes. Indeed, this classification technique will help experts to examine any case and make the examination process faster and more efficient. The study was conducted through several experiments using 730 images of human retina that were either treated by laser or not treated. An average accuracy rate of more than 97% was obtained. Additionally, possible improvements and destiny traits are suggested to summarize this study.

Key words: AlexNet Model, Classification, Deep Learning, Health care, laser, Retinal image

1. INTRODUCTION

Recently, laser technology has become an important component of industrial, commercial, medical, and many other fields. In medical applications, laser eye injury report numbers are increasing, like those occurring in Figure 1. The main reasons for that are the eye optical transparency and also the focusing power into the retina [1].

It is recommended to continue doing research in laser technologies while, as mentioned, the laser technologies will play an important role over the world. Research and studies have to be handled to minimize and recognize laser scars on the retina that will help to treat the injures appropriately.

Retina laser operations can cause two unwanted consequences. The most dangerous one is that patients could be scheduled for unnecessary repeated treatments. The other consequence is the scars that remain after the treatments could negatively conflict with other fundus image analysis

using a computerized diagnosis technique, for example leading to incorrect discoveries of secretions or other eye disease signs.

Therefore, it becomes imperative to automatically recognize the existence of scars that were left by previous photocoagulation treatments of the retina. In this study, a computerized solution for minimizing the mentioned problems is introduced. This solution is based on a deep learning approach called AlexNet. AlexNet is considered to be an enhanced type of (CNN) Convolutional Neural Networks. AlexNet is widely used in many applications to as powerful deep learning technique. This study will use this network as part of a research to compare with other deep learning techniques using the same datasets.

This paper is structured as follows. Section 2 presents regards studies while section 3 illustrates the theoretical background of (CNN). The architecture of the system is presented in section 4 while section 5 describes the experiment details and obtained results of this study. Finally, section 6 presents the conclusion of the study.



Figure 1: Laser scars on retinal fundus

2. RELATED STUDIES

Many studies were conducted to recognize diseases or eye illnesses using different types of Machine Learning methods such as Naive Bayes, Random Forest, Random Tree, Support Vector Machines, Multilayer Perceptron, RBF Network [2], logistic regression, and random forests [3]. The prospect of

being able to deal with eye disease is getting encouraging. The number of published studies has rapidly increased to construct approaches for auto recognition. Deep learning techniques were recently used to recognize different types of eye disease. The results were promising in general. However, more research is recommended to overcome the complexity and increase the recognition accuracy rates.

Recently, Syed et al. [4] presented a method to classify retinal scars based on Support Vector Machine (SVM). The method uses three colors as input features. These features are two textures and four shape features. The classification used 70% of the dataset for training from a proprietary dataset containing 380 images. The testing step used 30% to evaluate the classifier performance. Tahir et al. [5] presented a study to classify images for the detection of laser treatment signs. The system used Minimum Distance Clustering by determining if the images characterized using 10 scalar features. The used features include the spatial measurements and compactness of presumed blobs of laser marks. They used an additional 9 features to measure the intensity and colors like the maximum values of mean and maximum of luminance, hue and saturation, and the mean and maximum values for specific channels of colors, namely the green and red channels.

Deep learning techniques have received more interest recently due to high performance. Many studies were conducted: [6] Dong et al. compared three types of (CNN) with Support Vector Machine. The study used a dataset of malaria-infected cell images. (CNN) types were: LeNet, GoogLeNet and AlexNet. The study results showed that the version of (CNN) achieved a rate of accuracy with 95.0% higher than (SVM) by just a 3%. Later, Du et al. [7] presented their study to recognize unlabeled left and right iris images based on (CNN) network. The performances of the proposed (CNN) were evaluated by large and small scale iris datasets. The output results showed that the classification performance accuracy rate was above 97%.

AlexNet is a division of (CNN) that uses different numbers of layers. It has been used for classifying different types of patterns like face recognition [8], fingerprint classification [9], Internet of Things (IOT) [10] and in the medical filed [18, 19].

3. ARCHITECTURE OF CONVOLUTIONAL NEURAL NETWORKS

(CNN) is a multi-stage trainable architecture comprising multiple stages. Each input and output for every stage is a number of matrix of the features map. However, each feature map is represented by an exclusive feature that is extracted by all areas given in the input. Each stage consists of three layers namely: “a filter bank, a non-linearity, and a layer of feature pooling”.

The regular (CNN) consists of a maximum of three layers. Then, the classification modules follow these three layers.

The network of (CNN) can be used to deal with images, voice, or any signal. However, each layer will be described to deal with images only as follows:

3.1. Input Layers

This layer is used to define the input size for any network. This layer also contains the images' raw pixel values. The input image's size corresponds to the weight, height and the color channels numbers. As an example, the channel number is 1 for grayscale images, and 3 for colored images.

3.2. Convolutional Layers

This layer consists of connected neurons that are connecting to sub-area of the output or the input images for the last layer. This layer is trained by every localized feature using scanned regions of the image. The convolutional layer uses a set of weight numbers that can be performed on each region in the input images. The scanner moves over the input image in horizontal and vertical directions. The step repeats using similar computations for all regions for convolving the input image. The size of each region is called a stride, which moves and covers all the image regions. These regions may be overlapped depending by the Filter-Size and the stride values of this layer.

The weights values that are used by this filter can be calculated using:

$$h \times c \times w \quad (1)$$

Where h is the region height, c is the number of channels of the input image, and w is the region width. However, 3 channels are used for color images. The filters number determines the channels number for the convolutional layers output.

As stated above, the filters move along the image. It can be done using the same set of weights and convolutions, to formalize features map. Thus, the features number map of the convolutional layer comes with the same channel number. Each feature map has different bias values and weights. The parameter numbers for the convolutional layer are calculated by:

$$(Number\ of\ Filters \times (h \times c \times w + 1)) \quad (2)$$

Where the value 1 is a bias value.

This layer uses a Padding value which is normally done by adding values of zero to some columns and/or rows to the borders of input images. This method helps for controlling the output size of this layer.

The height and width outputs for convolutional layers are calculated using equation 3.

$$(Input\ Size - Filter\ Size + 2 \times Padding) / Stride + 1 \quad (3)$$

The values have to be integers for the entire image; to be sure that the entire image is covered. The network then ignores the remaining image locations combined with the right and bottom edges for this layer. This happens if the parameter combinations are not fully covering the image.

The output size in the map of features (number of total neurons) is the product of the height and width. The output size is calculated by:

$$Map\ Size \times Filters\ Number \tag{4}$$

This type of deep networks may contain single or multi convolutional layers based on the system type, where the data quantity and complexity are the main reason for using multi convolutional layers.

3.3. Batch Normalization Layer

This layer is set before ReLU and after the convolutional layers for speeding up the network training and to reduce initialization sensitivity of the network. It normalizes every channel activation using the subtraction of the average of mini batches and then dividing it by the standard deviation of the mini batch. Then, the input is shifted via an offset β and they scaled via a scale factor γ . The mentioned factors are used to update the learning parameters during the network training.

These layers normalize the propagating gradients and activations of the network for the simplicity the optimization problems of the network training. This layer could be highly advantageous in increasing the rate of learning. The network will learn quicker if the parameters can be updated with large values for overcoming the optimization problems effortlessly.

3.4. (ReLU) Layers

Rectified Linear Unit (ReLU) layer is following batch normalization and convolutional layers which defined as a nonlinear activation functions. The (ReLU) can be used for performing threshold procedures to every element, into each input value which is negative or greater or equalized to 0, by:

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \tag{5}$$

The regular (ReLU) layers have 0 extension numbers those are performing dissimilar operations and could enhance several application performances.

3.5. Cross Channel Normalization Layers

This layer, also known as (Local Response Normalization), is used for performing the "channel wise" normalization of local responses. This layer usually sits next to the ReLU layers. It is replacing every element using normalizing values that were gained from the elements using definite channels bordered number. The normalizing of x' value from element x in the input, can be computed a by:

$$x' = \frac{x}{\left(k + \frac{a * ss}{windowChannelSize}\right)^{\beta}} \tag{6}$$

Where ss represents the squares summation for the elements of the normalization window β , a , and K represent the parameters of hyper of the normalization process [11].

3.6. Max- and Average-Pooling Layers

This kind of layer comes after the convolutional layers. This layer is designed to reduce the connection values into the following layer. The Max Pooling layers cannot perform for learning themselves. However, reducing the learned parameter numbers helps reduce over-fitting in the next layer. This layer returns the maximum value in the input rectangular area of the layer. The size of this area is defined via Max Pooling layer arguments by the pooling size. For example, if the size of pool is (5,4), then the output will return the highest value in the area of height and width of 5 and 4. Also, for the Average Pooling layer, it follows the same process where the output is extracted using average values for the rectangular area for the input of the layer. Same as Max pooling layer, the rectangular area size will be determined via the size of pool arguments. Both layers use the scanning step concluded on the input image in horizontal and vertical directions. Here, stride value is used to control the size of this step. Where, the pooling area would not be overlapped if the size of the pool was taken less than or similar to the stride value.

In the case where the strides and pool size values are equal, it is by default considered as a non-overlapping area. Thus, while the pooling layers inputs are represented using $n \times n$, and the pooling area size is represented using $h \times h$ [12]. In this case, the output of a single channel for a convolutional layer will use the max pooling and the average pooling layers as n/h -by- n/h .

The overlapping areas of the pooling layer output are:

$$(IS - PS + 2 \times Padding) / Stride + 1 \tag{7}$$

Where IS is the size of input and PS is the size of pooling.

3.7. Dropout Layers

The dropout layers are randomly fixed in the input essentials for the layer into zero using a specified probability. Although the output of the dropout layer is similar to the layer input, the process is identical to briefly dropped randomly for selecting the unit and its connection units from the network during the process of the training. So, for each new input, the training of the network is done randomly for choosing the subset for the neurons by a different layer's structure using regular weights. However, since the learning process is not depending on exact neurons that can depend on any connection, the dropout layer might help to avoid over-fitting [11][13]. It is the same to the max pooling and average pooling layers where learning is not taking any place.

3.8. Fully Connected Layer

This layer follows the convolutional layers and comes in single or multiple fully connected layers. The fully connected layer is a combination of all features learned from the earlier layers. The feature set is used for recognizing the greater patterns and to categorize the images for the problem of the classification. However, the output sizes have to be similar to the response variables number of the regression problem.

3.9. Output Layers

These layers come after the fully connected layers. It is considered that the soft max and the classification layers are used for classification usage. The function of activation for the outputs is known as the soft-max function that presented as:

$$y_r(x) = \frac{\exp(a_r(x))}{\sum_{j=1}^k \exp(a_j(x))} \quad (8)$$

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

Equation 8 presented the output unit activation function. For classifying multi class issues:

$$P(c_r|x, \theta) = \frac{P(x, \theta|c_r)P(c_r)}{\sum_{j=1}^k P(x, \theta|c_j)P(c_j)} = \frac{\exp(a_r(x, \theta))}{\sum_{j=1}^k \exp(a_j(x, \theta))} \quad (9)$$

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

Function 9 is also called the normalized exponential. This function is considered to be a multi class generalization for the logistic sigmoid functions [14].

The classification layer output follows the layer of soft max. The layer outputs are determined by training the network which takes each value from the function of the soft max and then assigns each input to one of the k mutually exclusive classes. This process is done via the cross entropy function to code the schemes of 1-of-k [14]:

$$E(\theta) = - \sum_{i=1}^n \sum_{j=1}^k t_{ij} \ln y_j(x_i, \theta) \quad (10)$$

Where t_{ij} is the indicators of the sample i which is belonging to the class j th and θ represents the vector parameter.

$y_j(x_i, \theta)$ represents the i output sample, that presents the value of soft max function. In this case, this value is considered as the network probability associating the i th inputs with class j .

3.10. Regression Layers

Convolutional Neural Networks are also can handle the regression concerns. Here, the target variables come in continuous structuring and the regression layer is following the previous fully connected layer. The standard loss function for this layer is calculated by the mean squared error is considered to be by:

$$MSE = E(\theta) = \sum_{i=1}^n \frac{(t_i - y_i)^2}{n} \quad (11)$$

Where t_i represents the target variables, while y_i is considered as the network predictions for the observation responded variables i .

4. PROPOSED NETWORK ARCHITECTURE

The basic step to train and create any new (CNN) network is by defining the architecture of the network. The architecture of (CNN) could be diverse and the network depends on the layer's number and the type of each layer that is included to build the network. Also, the type and number of used layers may depend on the data type and also on the application style. As an example, if the network is designed to handle categorical dilemma, it is necessary to use classification functions and to use a number of classification layers. Nevertheless, for continuous style responses, it is necessary to use a regression layer to end the proposed network.

For training a network using a small number of grayscale dataset images, then the size of the network should be small and using no more than 3 convolutional layers, which might be more appropriate. In contrast, to train huge datasets with millions of colored images, the size of the network should be larger with multiple convolutional layers that are fully connected.

For this study, (AlexNet) network is used for classifying laser marks that remain by photocoagulation operations on retrieval eyes. The retinal images are used to analyze and show evidence of occurrences of any previous photocoagulation laser operations. This study is conducted as a continuation of previous studies using (CNN) networks.

In this section, the used dataset and architecture of the used (AlexNet) network is presented in detail.

In this experimental study, two datasets of retinal images were used throughout the training and testing phases of the experiments. Each dataset was collected by researchers at the University of Coimbra, Portugal [15]. The descriptions for each dataset will be explained as follows:

4.1 Dataset 1

This dataset consists of 315 retinal images marked via optometrists as polluted with marks of laser. This dataset was collected from DR screening continuing program which is located in Portugal's center region.

4.2 Dataset 2

This dataset is consists of 415 retinal images. This dataset has no marks of laser and is classified of grade R0 (completely no lesions). The classification was done by optometrist's graders from "Association for Innovation and Biomedical Research on Light" (AIBILI)[15].

In this study, a network of (AlexNet) is constructed using 25 layers and this network was used for a number of experiments as listed:

- Layer 1: As the same used dataset of previous studies [16][17], the images were sized at 227×227 pixels with level of RGB. Here, the normalization was 0 center for layer 1.
- Layer 2: the convolution layer is the second layer for the proposed network. Here, 96 filters were used with 4×4 stride values and zero padding value. 11×11 pixels were chosen in this layer to be scanned.
- Layer 3: the ReLU layer is considered as a third layer. It is used to enhance the performances for the last layer. However, this layer will be repeated several times as a separate layer.
- Layer 4: here, a layer of cross channel normalization was used for performing the normalization. Here, 5 channels were selected to perform the step of normalization.
- Layer 5: in this layer, a layer of Max pooling is performed. The scanning area dimension was 3×3 pixels with 2 strides and 2 paddings.
- Layer 6: In this layer, another convolution layer is used to start a new stage. Here, 256 filters were used with 1×1 strides value and a padding value of 2. 5×5 pixels were chosen in this layer to be scanned with 48 layers.
- Layer 7: In this layer, another ReLU layer is performed for improving layer 6 output.
- Layer 8: the second cross channel normalization layer is used for performing the normalization. The structure is similar to layer 4.
- Layer 9: In this layer, the Max Pooling layer is performed as in layer 5.
- Layer 10: The third convolution layer is repeated with 384 filters with 1×1 stride values and padding for 3×3 pixels of the scanned area.
- Layer 11: In this layer, ReLU is again listed.
- Layer 12: Repeated convolution layer is performed.
- Layer 13: Here, the ReLU layer is listed.
- Layer 14: The fourth convolution layer is repeated with 256 filters with 1×1 stride values and padding for 3×3 pixels of scanned area.
- Layer 15: Here, the ReLU layer is listed.
- Layer 16: In this layer, the Max Pooling layer is performed as in layers 5 and 9.
- Layer 17: here, 2 fully connected layers were performed. All the features accompanied those that were learned by the last layers.

- Layer 18: the ReLU layer is listed in this layer.
- Layer 19: In this layer, the dropout layer is used on 50% of samples.
- Layer 20: In this layer, another fully connected layer is performed as in layer 17.
- Layer 21: the ReLU layer.
- Layer 22: It is a dropout layer as in layer 19.
- Layer 23: a fully connected layer with 2 fully connected layers.
- Layer 24: for this layer, after the fully connected layer, a softmax function is performed for minimizing multi class classification issues.
- Layer 25: after the layer of soft max a classification output layer is ended the network. In the classification layer, the training of the network takes the values of the soft max function and then assigns each input into one exclusive mutual class or multi classes by the cross entropy function.

For training and testing the proposed network, the used datasets were divided randomly into 80% for training and 20% for testing. This structure was followed to be the same as the last two studies on the same datasets [16, 17].

5.EXPERIMENTS AND RESULTS

In this study, a number of experiments were accomplished using different MaxEpoch values. The experiments are presented profusely in this section.

MaxEpoch is a value that shows the training iteration number. For instance, when MaxEpoch is equal to 2, the network is trained twice. In this study, the network will use 730 input images, which is 80% of the total, for training and the rest will be used to test the network. The accuracy rate is determined by selecting variety values of MaxEpoch. Here, MaxEpoch values of 1, 2, 5, 10, 20, 30, 40, and 50 are chosen. The suitable value is extracted after doing 5 experiments for each MaxEpoch value and getting the average of the accuracy rate of these experiments. Table 1 illustrates the output of this experiment.

Table 1: Best MaxEpoch values calculations

Experiment.No	1	2	3	4	5	Accuracy Average
MaxEpoch = 1	88.36	84.25	86.99	92.47	86.30	87.67
MaxEpoch = 2	91.10	86.99	87.67	89.73	90.41	89.18
MaxEpoch = 5	93.84	95.21	84.93	90.41	92.47	91.37
MaxEpoch= 10	94.52	94.52	93.84	91.78	92.47	93.43
MaxEpoch= 20	93.84	95.89	87.67	93.15	95.21	93.15
MaxEpoch= 30	94.52	97.95	95.89	97.26	93.15	95.75
MaxEpoch= 40	92.47	94.52	95.21	91.10	93.15	93.29
MaxEpoch= 50	92.47	93.15	94.52	97.26	92.47	93.97

In Table 1, 40 experiments were conducted to find the best MaxEpoch value. In particular, 5 experiments were accomplished to catch the output layer accuracy average. However, the range of accuracy rates has a slight gap. Just 1 and 2 MaxEpoch values gave an average accuracy rate of less than 90%. However, the highest accuracy rate was obtained using a value of MaxEpoch of 30 with an average of 95.75%. The gap of differences might have resulted due to the image distribution in the training and testing sets. This becomes clear when conducting fewer experiments, such that MaxEpoch is 1 and 2.

It is noticeable that the accuracy rate drops for MaxEpoch values for 40 and 50 due to overtraining the network. In this case, it is not recommended to choose values that are greater than 30 because this leads to an increase in the time cost of the modeling process with no improvements. Figure 2 presents the accuracy average rates of the study experiments.

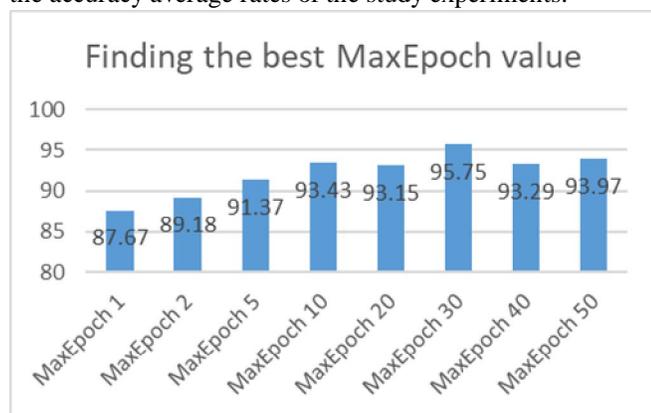


Figure 2: Finding the best MaxEpoch value

However, it is clear from Table I that the accuracy rate of the five experiments reached 97.95% in the second experiment of MaxEpoch 30. This rate shows a high accuracy rate compared to previous studies [16][17].

6. CONCLUSION

In recent years, minor laser operations are widely used on retinal surfaces. In these clinical mini operations, laser technologies are used to overcome some causes of lack in vision. Laser treatments might leave some marks that can affect the eye’s vision. In this paper, a set of retinal images containing laser signs that were left by laser operations on the surface of the retina were the domain images for training and testing of a deep learning network based on (AlexNet).

In this study, a number of experiments have been performed. The trained images were selected in different numbers to reach the best training set. The experiment results were auspicious with an average accuracy rates of 95.75% and reached to 97.95% as a top rate. This rate average is higher than the previous studies [16,17] with an average accuracy rate of 91.65% and 89.60% as shown in Table 2. However, both studies used a CNN network with 8 and 13 layers respectively and were tested on the same dataset that was used

in this study for training and testing. Figure 3 illustrates the comparison between the three studies’ outputs.

Table 2: Comparison of the study output with last studies

Average of accuracy / MaxEpochs	Network with 8 layer	Network with 13 layer	AlexNet
1	60.14	78.08	87.67
2	69.18	84.80	89.18
5	72.33	88.08	91.37
10	87.67	88.50	93.43
20	91.65	89.60	93.15
30	90.69	88.36	95.75
40	88.90	89.31	93.29
50	89.32	88.08	93.97
60	--	--	95.35

Figure 3 compares the model used in this study with two other studies based on CNN and used the same dataset for training and testing.

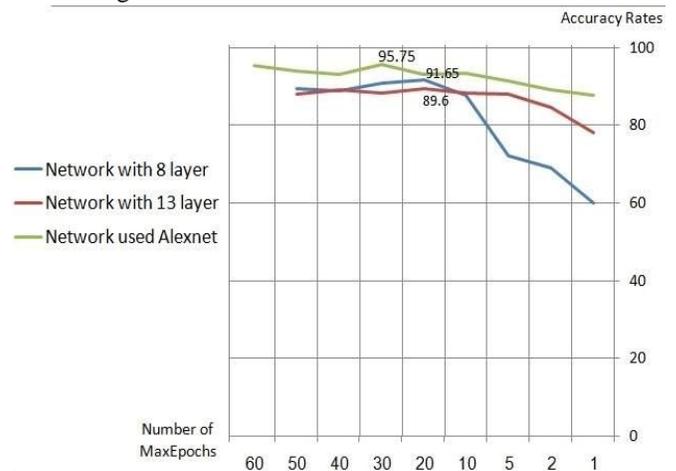


Figure 3: The three studies outputs

From Figure 3, it can be noted that the best rate average of the AlexNet model was 95.75% which was at value of MaxEpochs 30 while the best rate average of the previous studies [16,17] was at MaxEpochs value of 20.

For future studies, updating experiments should be performed using different architectures like ResNet for enhancing the rate of the network accuracy and by including extra images.

REFERENCES

1. P. W. Commiskey, C. J. Heisel, and Y. M. Paulus, “Non-Therapeutic Laser Retinal Injury” *Int. J. Ophthalmic Res.*, vol. 5, no. 1, pp. 321–335, 2019.
2. L. B. da Cruz et al., “Interferometer eye image classification for dry eye categorization using phylogenetic diversity indexes for texture analysis” *Comput. Methods Programs Biomed.*, vol. 188, p. 105269, May 2020, doi: 10.1016/J.CMPB.2019.105269.

3. S. L. Baxter, C. Marks, T.-T. Kuo, L. Ohno-Machado, and R. N. Weinreb, “**Machine Learning-Based Predictive Modeling of Surgical Intervention in Glaucoma Using Systemic Data From Electronic Health Records**” *Am. J. Ophthalmol.*, vol. 208, pp. 30–40, Dec. 2019, doi: 10.1016/J.AJO.2019.07.005.
4. A. M. Syed, M. U. Akbar, M. U. Akram, and J. Fatima, “**Automated laser mark segmentation from colored retinal images**” in 17th IEEE International Multi Topic Conference 2014, 2014, pp. 282–286, doi: 10.1109/INMIC.2014.7097352.
5. F. Tahir, M. U. Akram, M. Abbass, and A. A. Khan, “**Laser marks detection from fundus images**” in 2014 14th International Conference on Hybrid Intelligent Systems, 2014, pp. 147–151, doi: 10.1109/HIS.2014.7086188.
6. Y. Dong et al., “**Evaluations of deep convolutional neural networks for automatic identification of malaria infected cells**” in 2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI), 2017, pp. 101–104, doi: 10.1109/BHI.2017.7897215.
7. Y. Du, T. Bourlai, and J. Dawson, “**Automated classification of mislabeled near-infrared left and right iris images using convolutional neural networks**” in 2016 IEEE 8th International Conference on Biometrics Theory, Applications and Systems (BTAS), 2016, pp. 1–6, doi: 10.1109/BTAS.2016.7791162.
8. H. B. Alwan and K. R. Ku-Mahamud, “**Cancellable Face Biometrics Template Using AlexNet**” in International Conference on Applied Computing to Support Industry: Innovation and Technology, 2020, pp. 336–348.
9. F. Luo, F. Wu, H. Li, T. Li, J. He, and Q. Chen, “**Extraction and Recognition of Fingerprint Characteristic of Mobile Terminal’s Transient Signal**” in IOP Conference Series: Materials Science and Engineering, 2020, vol. 719, no. 1, p. 012034, doi: 10.1088/1757-899X/719/1/012034.
10. R. Hadidi, J. Cao, M. Woodward, M. S. Ryoo, and H. Kim, “**Real-time image recognition using collaborative IoT devices**” in Proceedings of the 1st Reproducible Quality-Efficient Systems Tournament on Co-Designing Pareto-Efficient Deep Learning, ReQuEST 2018 - Co-located with ACM ASPLOS 2018, 2018, doi: 10.1145/3229762.3229765.
11. A. Krizhevsky, I. Sutskever, and G. E. Hinton, “**ImageNet Classification with Deep Convolutional Neural Networks**” in Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105.
12. J. Nagi et al., “**Max-pooling convolutional neural networks for vision-based hand gesture recognition**” in 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), 2011, pp. 342–347, doi: 10.1109/ICSIPA.2011.6144164.
13. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “**Dropout: A Simple Way to Prevent Neural Networks from Overfitting**” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jan. 2014.
14. C. M. Bishop, “**Periodic Variables**” *Pattern Recognit. Mach. Learn.*, vol. 1, no. day 365, pp. 181–198, 2006.
15. J. G. R. A. e Sousa, C. M. Oliveira, and L. A. da S. Cruz, “**Automatic detection of laser marks in retinal digital fundus images**” in 2016 24th European Signal Processing Conference (EUSIPCO), 2016, pp. 1313–1317, doi: 10.1109/EUSIPCO.2016.7760461.
16. O. M. Elrajubi, M. A. Abuzaraida, and A. M. Zeki, “**Retinal image laser marks detection using a convolutional neural network**” in 2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies, 3ICT 2018, 2018, pp. 1–5, doi: 10.1109/3ICT.2018.8855784.
17. M. A. Abuzaraida and O. M. Elrajubi, “**Laser marks classification for retinal images based on convolutional neural network**” *Int. J. Eng. Adv. Technol.*, vol. 8, no. 6 Special Issue 3, pp. 188–193, 2019, doi: 10.35940/ijeat.F1030.0986S319.
18. Jessica Velasco, Cherry Pascion, Jean Wilmar Alberio, Jonathan Apuang, John Stephen Cruz, Mark Angelo Gomez, Benjamin Jr. Molina, Lyndon Tuala, August Thio-ac, Romeo Jr. Jorda, **A Smartphone-Based Skin Disease Classification Using MobileNet CNN**, *International Journal of Advanced Trends in Computer Science and Engineering*, pp. 2632 – 2637. DOI: 10.30534/ijatcse/2019/116852019.
19. Bharat, Vinod & Malik, Dr. (2019). “**Study of Detection of Various types of Cancers by using Deep Learning: A Survey**”. *International Journal of Advanced Trends in Computer Science and Engineering*. 8. 1228-1233. 10.30534/ijatcse/2019/31842019.