



Research Article

Application of an improved edge detection algorithm in fundus images

Gurkan KAPLAN^{1,*}, Alper ODABAS¹

¹Department of Mathematics and Computer Sciences, Eskisehir Osmangazi University, Eskişehir, 26040, Türkiye

ARTICLE INFO

Article history

Received: 30 November 2023

Revised: 15 February 2024

Accepted: 30 March 2024

Keywords:

Classification; Feature
Extraction; Filter; Image
Processing; Medical Image
Processing

ABSTRACT

One of the most important parts of artificial intelligence projects made today is edge detection. Although it is used a lot in artificial intelligence studies, it is actually a difficult process due to reasons such as bad focusing and narrow dynamic range images, etc. In order to get rid of this kind of image problem, we created a unique filter for each image. In this paper, we provide a filtering method from eigenvalues and eigenvectors of the image matrix. Fundus images were obtained publicly from Drive, Stare, Messidor databases to test the performance of this method. After a lot of visual comparisons, we tested our filter and other filters using the Convolutional Neural Networks with confusion matrix to find out which filter better classify. 2, 5, 10 epochs were used for this process and the results were compared with 5000, 10000 and 15000 steps per epoch. Our filter was the best result with %69.23 accuracy with 2 epoch, 15000 spe. This result was followed by %65.38 accuracy Roberts filter with 10 epoch, 15000 spe and our filter with 10 epoch, 5000 spe.

Cite this article as: Kaplan G, Odabas A. Application of an improved edge detection algorithm in fundus images. Sigma J Eng Nat Sci 2025;43(2):615–625.

INTRODUCTION

Feature extraction has attracted a great deal of attention, especially during the past few years [1]. Since it is necessary to diagnose from images in many fields such as medical treatment, a criminal investigation, military issues, feature extraction from images is one of the most important issues. Edge detection is the most important issue for feature extraction because an edge can be defined as a sudden change of intensity in an image. Up to now, some filters such as Canny, Sobel, Prewitt, Roberts operators, etc. have been used for edge detection from images. These filters use fixed filters and first or second derivative of the input image except for the Canny operator.

In this study, we wanted to create its filter for each picture. In this context, firstly we used the eigenvalue and eigenvector matrices of the matrix forming the image. In the literature, it has been seen that there are many studies obtained in different fields using eigenvalues and eigenvectors.

Using the associative memory to the image in RGB format, the filter for the image was obtained through eigenvalues, eigenvectors. While obtaining this filter, the image was filtered with the best 9 eigenvectors with min-max memory and the best results were obtained [2]. Harris and Stephens used the eigenvalues of the moment matrix to determine the corner and edge [3]. Eigenvectors were found among the best eigenvalues for vehicle detection and recognition

*Corresponding author.

*E-mail address: gkaplan@ogu.edu.tr

This paper was recommended for publication in revised form by
Editor-in-Chief Ahmet Selim Dalkilic



with a Multilayer perceptron (MLP) classifier using 1, 3, or 5 hidden nodes and over 90% success was achieved [4]. Eigenvalues and eigenvectors were used to find the corners using Bayes classifier when determining blur images [5]. Eigenvalues and eigenvectors were used to determine edges using vector-valued techniques in color images [6]. Eigenvectors were used for image defect detection method that is the non-model based [7]. To find the corners of the image in Parametric Feature Detection, the best 8 eigenvectors were selected from the image matrix [8]. Eigenvectors were used when using Edge Gradient and Graph Spectrum on the text. Compared with other algorithms, the proposed method improved the precision rate by 12% [9]. Eigenvectors were used to determine the corner by topographic analysis [10]. Eigenvectors were used to determine contour when contouring detection [11]. To determine the optic disc from the retinal images, operations were performed using principal component analysis with the help of eigenvectors [12]. A second directional derivative approach is proposed, using eigenvalues and eigenvectors for deriving a color edge detector [13]. Eigenvectors were used for the depth detection of targets in a monocular image sequence [14].

Recently, parameters obtained with differential equation systems have started to be used in this field. Some of these studies; Differential Equations with Non-Local Conditions [15], fractional integro-differential equations [16], The Mohand transform to solve fractional integro-differential equations [17].

Convolutional neural network was used to apply the filter we created in the study. There are many studies in the literature using CNN and edge detection filters.

The detection of brain tumors was performed using edge detection and CNN [18]. Sobel and Prewitt edge detection filters were applied to the images with CNN for medical image segmentation [1]. CNN was used for Crowd Segmentation in traffic and 95% success was achieved [19]. CNN was used to determine edges on musculoskeletal ultrasound images [20]. Retinal vessel segmentation was performed from fundus images using edge detection filters and CNN [21]. CNN and edge detection filters were used for Traffic Sign Detection and Recognition [22]. Edge detection and CNN were used for Fingertip detection [23]. CNN and edge detection filters were used for contour detection [24]. In the same way, vessel segmentation was performed with boundary detection [25]. CNN was used for self-driving cars and Sobel, Laplacian, and Canny detectors were used for road lane detector [26]. While background subtraction using encoder-decoder, CNN and edge detection filters were used [27].

In this article, an image-specific filtering method is developed by using the eigenvalues and eigenvectors of the image to filter the image appropriately. The main idea behind the emergence of this method is that the use of fixed filters does not give good results even with the slightest blur or distortion in the image. In order to compare the filters,

fundus images were selected from radiological images with relatively lower image quality as the most suitable application area. The results obtained were compared with confusion matrices.

In section 2, the 4 filters compared in the article are explained in detail with the methods they use. The method presented in section 3 is detailed. Since CNN was last applied to the method introduced in Section 4, CNN was explained and detailed.

EDGE DETECTION FILTERS

Sobel Operator

The Sobel operator carries out a gradient measurement on an input image and highlights high-value regions corresponding to the edges. It is often used to obtain an approximate gradient size at each point in a grayscale image. The operator uses two 3x3 convolution kernels for figure out roughly derivatives, first for horizontal changes, and second for vertical as shown in (5).

$$\begin{bmatrix} r(1,1) & r(1,2) & r(1,3) \\ r(2,1) & r(2,2) & r(2,3) \\ r(3,1) & r(3,2) & r(3,3) \end{bmatrix} \quad (1)$$

$$|G| = \left| \begin{array}{l} r(1,1) + 2r(1,2) + r(1,3) - (r(3,1) + 2r(3,2) + r(3,3)) + \\ r(1,3) + 2r(2,3) + r(3,3) - (r(1,1) + 2r(2,1) + r(3,1)) \end{array} \right| \quad (2)$$

It is envisaged that these kernels will respond highly to the edges working vertically and horizontally. Kernels can be applied to the image one by one to produce measurements of the gradient (G_x, G_y). They are calculated as follows [28]:

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (3)$$

Direction angle of the edge causing the particular gradient is given by

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right) \quad (4)$$

(G_x, G_y) as shown in (5) respectively [29],

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (5)$$

Prewitt Operator

Prewitt operator works in similar to Sobel operator. This kernel produces almost similar results to the Sobel. Prewitt operator kernels (G_x, G_y) respectively as shown in (6) [30].

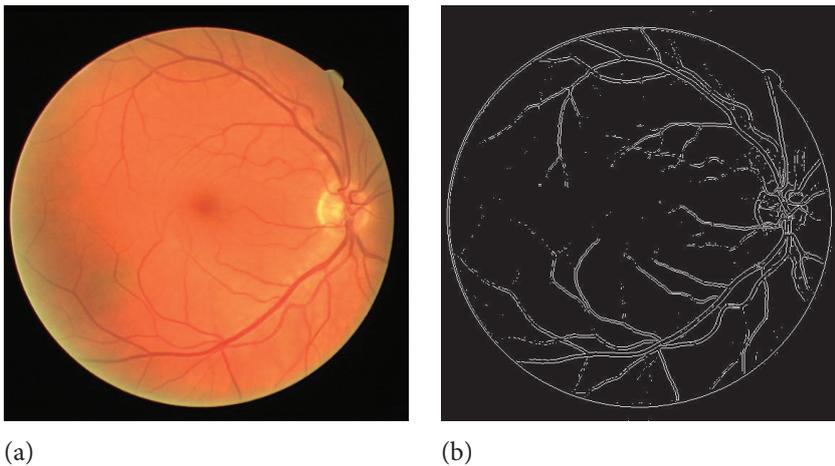


Figure 1. a) Normal image b) Sobel filtered image.

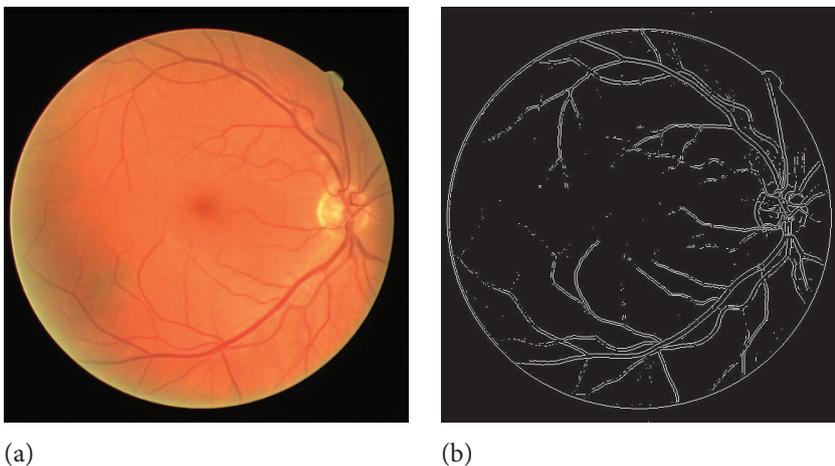


Figure 2. a) Normal image b) Prewitt filtered image.

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad (6)$$

Roberts Operator

Roberts operator is very useful for detecting diagonal of an input image. Because of the Roberts operator’s masks dimension of 2, run time is quietly diminished.

$$G = \sqrt{(r(1,1) - r(2,2))^2 + (r(2,1) - (1,2))^2} \quad (7)$$

Roberts operator masks [31],

$$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \quad (8)$$

In addition to this, because of these masks work on a small array of pixels, are as a result more sensitive to noise compared with the other filters.

Canny Operator

The Canny operator is the one of the best edge detection operator that uses multistage code block to determine a wide range of edges in input image.

The operator use in a multistage process. First, a Gaussian filter is applied to reduce noise, soften the input image. Next, the first derivative operator is performed to find the dense gradients of the image. Next, the algorithm performs non-maximal suppression to give a thin line at the edges. After this step, a double threshold was applied to identify possible edges. Finally, edges are defined by rejecting all other edges that are not related to strong edges [32].

OVERVIEW OF THE NEW FILTER

In Section 2, 4 important filters in this field were examined. It was seen that all of these filters, except the Canny filter, used fixed filters. Therefore, the filters are sensitive to the light and color intensity of the image.

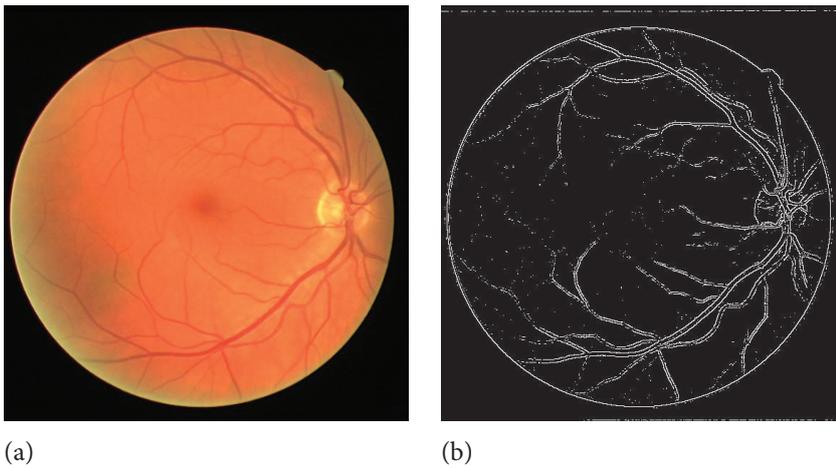


Figure 3. a) Normal image b) Roberts filtered image.

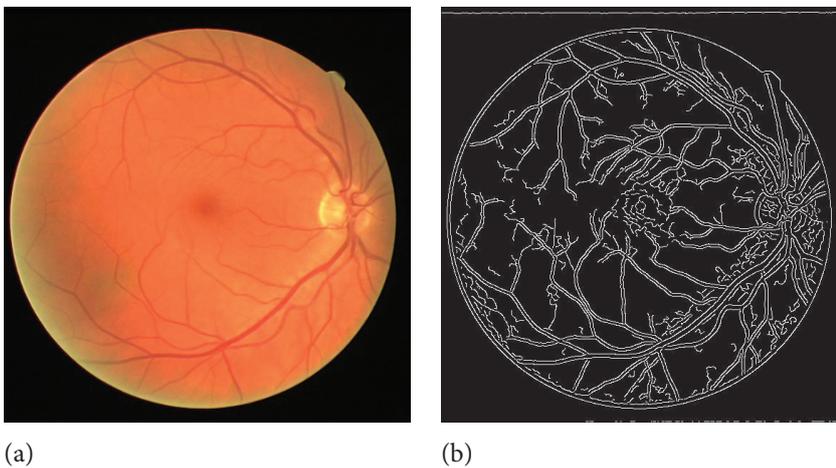


Figure 4. a) Normal image b) Canny filtered image.

In digital image processing, image can be shown as a set of pixels as shown in (9)

$$f_{xy} = \begin{bmatrix} x, y & x, y+1 & \dots & x, y+i \\ x+1, y & x+1, y+1 & \dots & x+1, y+i \\ \dots & \dots & \dots & \dots \\ x+j, y & x+j, y+1 & \dots & x+j, y+i \end{bmatrix} \quad (9)$$

In order to reduce sensitivity, we found eigenvalues ev_k and eigenvectors ec_k of 3x3 matrices that make up the image. We took the eigenvector matrix closest to additive zero. When processing the filter matrix we created, we obtained 4 different filter matrices with 90° rotations to minimize image losses. In this way, we processed each image with its own 4 filters and determined the values that are above the threshold value. Part of the code block is shown in Figure 5.

CONVOLUTIONAL NEURAL NETWORKS (CNN)

CNN is widely used in image processing. Inspired by bio-logical processes like the human brain, with the help of previous observations, it can distinguish different objects by just visualizing them [1]. The first CNN network is LeNet, an architecture introduced by Yann LeCun in 1988. In the LeNet network, the lower layers consist of consecutive layered convolution and maximum pooling layers. The subsequent upper layers correspond to the traditional Multi Layer Perceptron, which is fully connected [33].

A convolutional neural network composed of an input, an output layer and a large number of hidden layers. CNN architecture consists of 3 layers. These are convolutional layer, pooling layer and fully connected layer.

Convolutional Layer

Let's show that i th layer to understand in detail. Let N^i indicates the number of feature maps on this layer. Each feature map is indicated as $F_j^i (j = 1, \dots, N^i)$. This layer is

Algorithm 1: function [] = newfilter(A : two dimensional image array)

```

t=eigvec(A); top=sum(sum(t));s1=0;s2=0;
p is a randomly chosen positive number very close to 0
while top <> 0 do
    for i ← 1 to 3 do
        for j ← 1 to 3 do
            if top>0 then
                | t(i,j)=t(i,j)+p;
            else
                | t(i,j)=t(i,j)-p;
            end
        end
    end
end
end
Each of the y1, y2, y3 and y4 matrices is obtained by rotating the t matrix by 90°
f1=imfilter(adres,y1); f2=imfilter(adres,y2);
f3=imfilter(adres,y3); f4=imfilter(adres,y4);
[x,y]=size(adres); d=zeros(x,y); filters=zeros(1,4);
for i ← 1 to x do
    for j ← 1 to y do
        filters(1)=f1(i,j);filters(2)=f2(i,j);
        filters(3)=f3(i,j);filters(4)=f4(i,j);
        threshold = input('Threshold:');
        if max(filters)>threshold then
            | d(i,j)=1;
        end
    end
end
end
end
function [f]=eigvec(a)
[x,y]=size(a); ek=100;
for i ← 1 to x - 2 do
    for j ← 1 to y - 2 do
        k=[a(i,j),a(i,j+1),a(i,j+2);a(i+1,j),a(i+1,j+1),a(i+1,j+2);...
        a(i+2,j),a(i+2,j+1),a(i+2,j+2)];
        [w,z]=eig(double(k)); t=sum(sum(w));
        if abs(t) < ek then
            | ek=t; f=w;
        end
    end
end
end
end

```

Figure 5. Part of the code block.

parametrized by an array two-dimensional kernels W_{mj}^i associating the m th feature map f_m^{i-1} in the $(i-1)$ th layer in the bias b_j . A nonlinear activation function $\phi(A)$ is performed logically. Let * denotes the convolution operation [34]. The feature maps of the i th layer can be expressed as follows:

$$f_j^i = \phi\left(\sum_{m=1}^{N^{i-1}} f_m^{i-1} * W_{mj}^i + b_j^i\right), j = 1, 2, \dots, N^i \quad (10)$$

Each resulting matrix corresponds to a property. Extracts the properties of that region from the inputs (f_m^{i-1}) [35].

Pooling Layer

In CNN architecture, it is prevalent to add a pooling layer between consecutive CNN layers. This layer is used to reduce coefficient and calculation operations on the network. it operates every step on CNN. The most commonly used form Max and Average operator which form

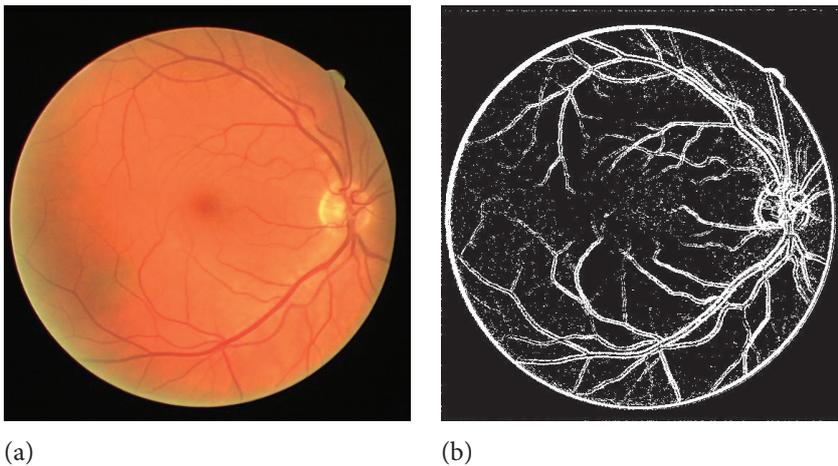


Figure 6. a) Normal image b) Filtered image with the new filter.

of non-linear down-sampling is. Max-pooling and average-pooling partition the input image into a set of non-overlapping rectangles and, for each such subregion, outputs respectively maximum, average values. They are useful for they reduces the computational complexity for upper layers and provides a form of conversion stability.

Fully Connected Layer

The final output layer is a fully connected neural network layer. This layer is a necessary component of CNN that is successfully displayed in classifying images. Many functions such as softmax can be used in this layer. This layer takes the output of the previous layers, “flattens” them and turns them into a single vector that can be an input for the next stage and takes the inputs from the feature analysis and applies the weights to predict the correct tag.

EXPERIMENTS

Dataset and Design Implementation

126 AMD and normal pictures were used for the applications within the scope of this study. The pictures obtained include an equal number of AMD and normal image. 100 of them were used for training and 26 of them were used for testing. Each of the images determined for test and train operations was filtered with Sobel, Prewitt, Roberts and Canny edge detection filters using Matlab r2018a program. Similarly, the pictures were filtered with the filter we created. As a result of these operations, 5 different images were obtained which are filtered for each picture, two of them are shown in figure 8 and figure 9.

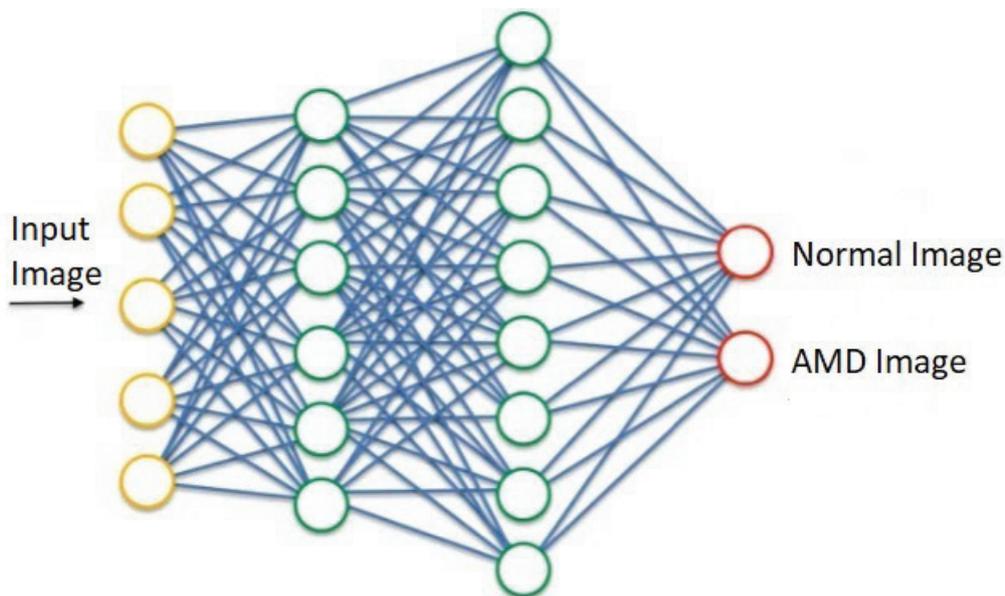


Figure 7. Fully connected layer structure.

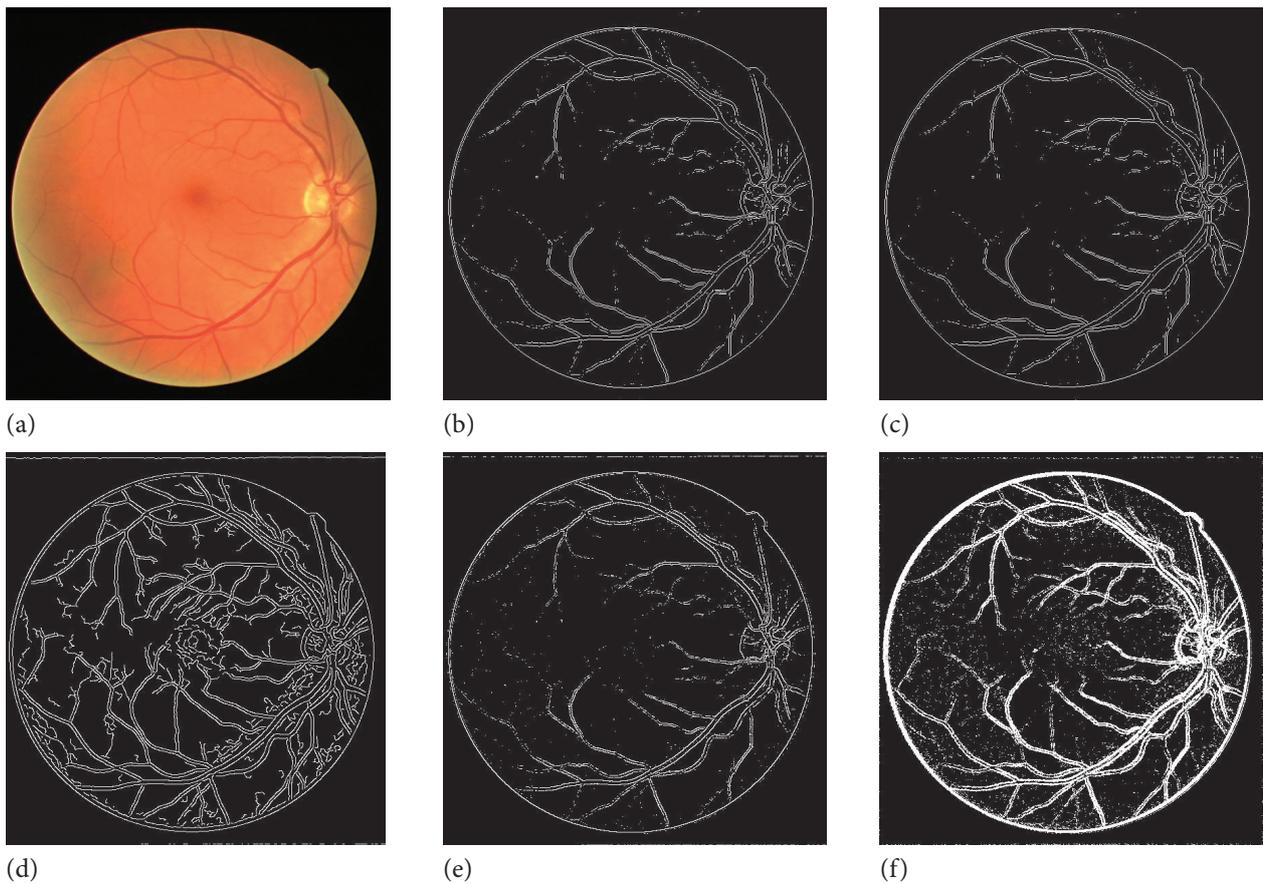


Figure 8. (a) Normal image (b) Sobel filtered image (c) Prewitt filtered image (d) Canny filtered image (e) Roberts filtered image (f) Filtered image with the new filter.

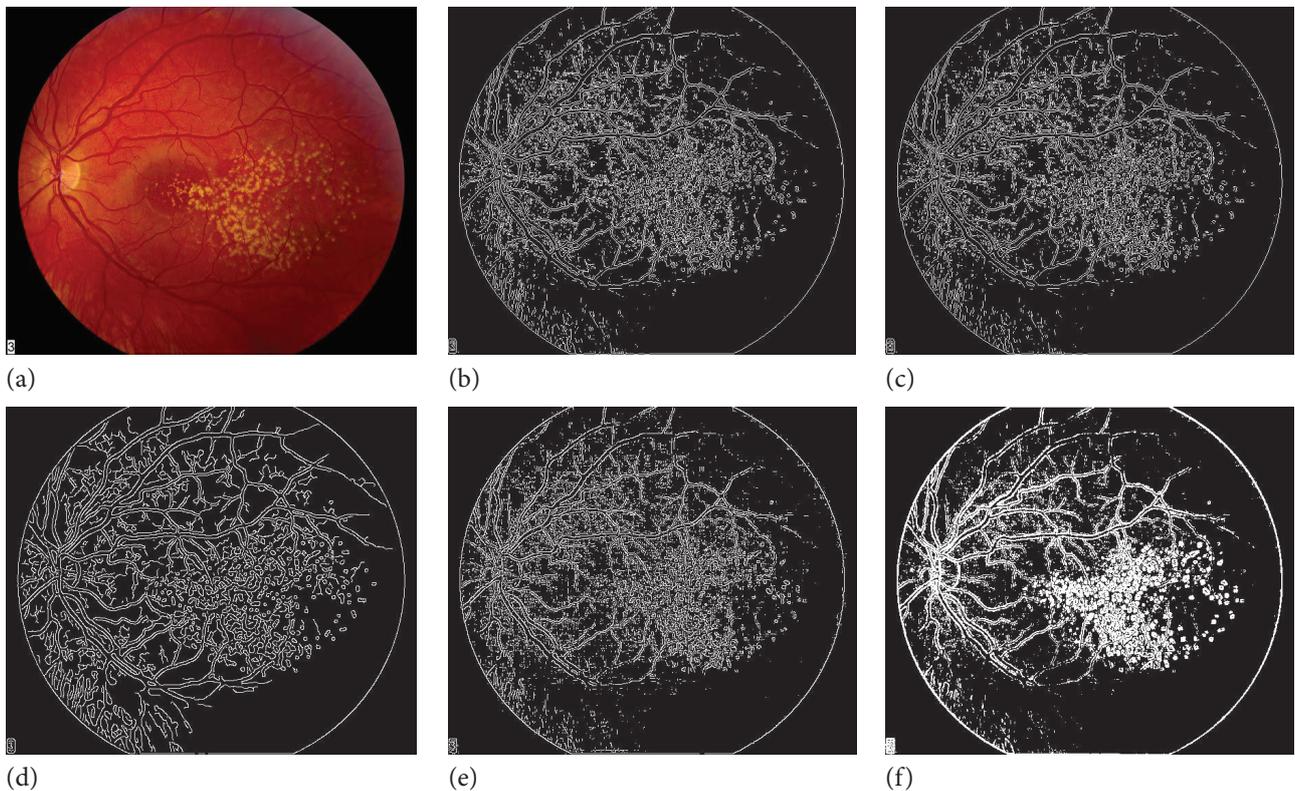


Figure 9. (a) Age related macular degenerated image (b) Sobel filtered image (c) Prewitt filtered image (d) Canny filtered image (e) Roberts filtered image (f) Filtered image with the new filter.

Codes were written using Python 3.7 to apply Convolutional Neural Network to images obtained after coding in Matlab. The dimensions of the images were equalized by making 64x64 before CNN was applied. Relu was used as activation function, binary crossentropy was used as loss function and “Adam” was used for optimizer.

For CNN, epoch number 2, 5, 10, batch size 1, validation steps 2000 values were taken. Steps per epoch(Spe) were taken as 5000, 10000, 15000 for each epoch and the filters were compared according to epoch. The confusion matrixes shown in Table 1 were used to see the accuracy values used to compare the filters.

Table 1. Confusion matrix

Data set	Actual (Amd)	Actual (normal)	Accuracy (%)
Predict (Amd)	True (T1)	False (F1)	Amd True Value (T1/(T1+F1))
Predict (Normal)	False (F2)	True (T2)	Normal True Value (T2/(T2+F2))

Table 2. Results using confusion matrix with 2 epoch

Spe	Sobel	Prewitt	Roberts	Canny	New Filter
5000	$\begin{bmatrix} 5 & 8 \\ 9 & 4 \end{bmatrix}$	$\begin{bmatrix} 6 & 7 \\ 9 & 4 \end{bmatrix}$	$\begin{bmatrix} 6 & 7 \\ 6 & 7 \end{bmatrix}$	$\begin{bmatrix} 6 & 7 \\ 6 & 7 \end{bmatrix}$	$\begin{bmatrix} 5 & 8 \\ 5 & 8 \end{bmatrix}$
10000	$\begin{bmatrix} 8 & 5 \\ 9 & 4 \end{bmatrix}$	$\begin{bmatrix} 5 & 8 \\ 9 & 4 \end{bmatrix}$	$\begin{bmatrix} 6 & 7 \\ 7 & 6 \end{bmatrix}$	$\begin{bmatrix} 5 & 8 \\ 5 & 8 \end{bmatrix}$	$\begin{bmatrix} 8 & 5 \\ 7 & 6 \end{bmatrix}$
15000	$\begin{bmatrix} 7 & 6 \\ 5 & 8 \end{bmatrix}$	$\begin{bmatrix} 9 & 4 \\ 6 & 7 \end{bmatrix}$	$\begin{bmatrix} 7 & 6 \\ 4 & 9 \end{bmatrix}$	$\begin{bmatrix} 6 & 7 \\ 7 & 6 \end{bmatrix}$	$\begin{bmatrix} 10 & 3 \\ 5 & 8 \end{bmatrix}$
Average Accuracy (%)	46.15	44.87	52.56	48.72	57.69

Table 3. Results using confusion matrix with 5 epoch

Spe	Sobel	Prewitt	Roberts	Canny	New Filter
5000	$\begin{bmatrix} 7 & 6 \\ 6 & 7 \end{bmatrix}$	$\begin{bmatrix} 8 & 5 \\ 6 & 7 \end{bmatrix}$	$\begin{bmatrix} 7 & 6 \\ 5 & 8 \end{bmatrix}$	$\begin{bmatrix} 7 & 6 \\ 6 & 7 \end{bmatrix}$	$\begin{bmatrix} 8 & 5 \\ 7 & 6 \end{bmatrix}$
10000	$\begin{bmatrix} 8 & 5 \\ 6 & 7 \end{bmatrix}$	$\begin{bmatrix} 7 & 6 \\ 8 & 5 \end{bmatrix}$	$\begin{bmatrix} 5 & 8 \\ 7 & 6 \end{bmatrix}$	$\begin{bmatrix} 5 & 8 \\ 4 & 9 \end{bmatrix}$	$\begin{bmatrix} 9 & 4 \\ 6 & 7 \end{bmatrix}$
15000	$\begin{bmatrix} 9 & 4 \\ 6 & 7 \end{bmatrix}$	$\begin{bmatrix} 6 & 7 \\ 9 & 4 \end{bmatrix}$	$\begin{bmatrix} 7 & 6 \\ 5 & 8 \end{bmatrix}$	$\begin{bmatrix} 8 & 5 \\ 7 & 6 \end{bmatrix}$	$\begin{bmatrix} 8 & 5 \\ 5 & 8 \end{bmatrix}$
Average Accuracy (%)	57.69	47.44	52.56	53.85	58.97

Table 4. Results using confusion matrix with 10 epoch

Spe	Sobel	Prewitt	Roberts	Canny	New Filter
5000	$\begin{bmatrix} 8 & 5 \\ 6 & 7 \end{bmatrix}$	$\begin{bmatrix} 8 & 5 \\ 8 & 5 \end{bmatrix}$	$\begin{bmatrix} 5 & 8 \\ 7 & 6 \end{bmatrix}$	$\begin{bmatrix} 8 & 5 \\ 8 & 5 \end{bmatrix}$	$\begin{bmatrix} 9 & 4 \\ 5 & 8 \end{bmatrix}$
10000	$\begin{bmatrix} 7 & 6 \\ 8 & 5 \end{bmatrix}$	$\begin{bmatrix} 7 & 6 \\ 8 & 5 \end{bmatrix}$	$\begin{bmatrix} 6 & 7 \\ 8 & 5 \end{bmatrix}$	$\begin{bmatrix} 4 & 9 \\ 8 & 5 \end{bmatrix}$	$\begin{bmatrix} 7 & 6 \\ 6 & 7 \end{bmatrix}$
15000	$\begin{bmatrix} 5 & 8 \\ 10 & 3 \end{bmatrix}$	$\begin{bmatrix} 8 & 5 \\ 6 & 7 \end{bmatrix}$	$\begin{bmatrix} 7 & 6 \\ 3 & 10 \end{bmatrix}$	$\begin{bmatrix} 5 & 8 \\ 4 & 9 \end{bmatrix}$	$\begin{bmatrix} 9 & 4 \\ 6 & 7 \end{bmatrix}$
Average Accuracy (%)	44.87	51.28	50	46.15	60.26

Table 5. General summary of filters

Filters	Used Filter	Best Confusion Matrix	Accuracy(%)
Sobel	$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}, \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 9 & 7 \\ 4 & 6 \end{bmatrix}$	57.69
Prewitt	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}$	$\begin{bmatrix} 8 & 5 \\ 6 & 7 \end{bmatrix}$	57.69
Roberts	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 7 & 6 \\ 3 & 10 \end{bmatrix}$	65.38
Canny	Gaussian filter, first derivative operator, etc.	$\begin{bmatrix} 5 & 8 \\ 4 & 9 \end{bmatrix}$	53.84
New Filter	It creates a special filter for each image.	$\begin{bmatrix} 10 & 3 \\ 5 & 8 \end{bmatrix}$	69.23

RESULTS AND DISCUSSION

The results of the study with a limited number of data are shown in table 2, 3 and 4. When the accuracy values measured with 3 different epoch values are compared; the best result was the new filter with %69.23 at 2 epoch and 15000 spe. When we look at the average accuracy of the results of 3 epoch values, the new filter was the most successful filter with %58.97.

The main motivation for this work was that the filter matrices used by most edge detection filters are fixed matrices. As a natural consequence of this, it was predicted that it would be far from being as successful as desired in images that were distorted or had low contrast for any reason. When the results were examined in detail, the positive performance of our filter against other filters was as expected. A general summary of the filters is given in table 5.

One of the main reasons why our filter gives successful performance at almost every epoch value is that it selects a filter matrix from the image itself specific to the image it uses. In addition, examining the image in more detail with filter rotations around itself has a positive impact on performance.

The weak point of the new filter compared to other filters is that there is a difference in time to reach the result since it filters a very detailed image. However, in computers used with today's technology, this difference is very low. For example, it was seen that the new filter we created in the study achieved results approximately 0.1-0.2 seconds slower than other filters when measured with the tic-toc function.

CONCLUSION

We have presented a two-stage study. Firstly, We propose an effective edge detection filter. We used the

eigenvalue, eigenvector matrices of the image when applying this filter and applied this filter to retinal fundus images for visual comparison. Secondly, We have applied the filter to AMD and normal fundus images using CNN to compare with the new filter with other filters. The new filter successfully improves prediction performance of neural network model.

In this study, although the new filter we created was working with a small amount of data, it was found to be successful by increasing the accuracy rate by approximately %10.

This article was obtained from my thesis. In the next stage, the filter will be detailed using the data augmentation technique from artificial intelligence technologies.

In the future, it is planned to use the filter created after this study in more socially beneficial, that is, more vital areas of medicine. One of these areas will be benign or malignant tumor detection and good or bad mole detection.

AUTHORSHIP CONTRIBUTIONS

Authors equally contributed to this work.

DATA AVAILABILITY STATEMENT

The authors confirm that the data that support the findings of this study are available within the article. Raw data that support the finding of this study are available from the corresponding author, upon reasonable request.

CONFLICT OF INTEREST

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

REFERENCES

- [1] Vardhana M, Arunkumar N, Lasrado S, Abdulhay E, Ramirez-Gonzalez G. Convolutional neural network for bio-medical image segmentation with hardware acceleration. *Cogn Syst Res* 2018;50:10–14. [\[CrossRef\]](#)
- [2] Acevedo E, Acevedo A, Martínez F, Chávez A, Velasco P. Associative approach for edge detection. In: 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC); 2014 Oct; p. 152–157. IEEE. [\[CrossRef\]](#)
- [3] Harris C, Stephens M. A combined corner and edge detector. In: *Alvey Vision Conference*; 1988 Aug; p. 10–5244.
- [4] Matthews ND, An PE, Charnley D, Harris CJ. Vehicle detection and recognition in greyscale imagery. *Control Eng Pract* 1996;4:473–479. [\[CrossRef\]](#)
- [5] Liu R, Li Z, Jia J. Image partial blur detection and classification. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition; 2008 Jun; p. 1–8. IEEE. [\[CrossRef\]](#)
- [6] Koschan A, Abidi M. Detection and classification of edges in color images. *IEEE Signal Process Mag* 2005;22:64–73. [\[CrossRef\]](#)
- [7] Amano T. Correlation based image defect detection. In: 18th International Conference on Pattern Recognition (ICPR'06); 2006 Aug; p. 163–166. IEEE. [\[CrossRef\]](#)
- [8] Baker S, Nayar SK, Murase H. Parametric feature detection. *Int J Comput Vis* 1998;27:27–50. [\[CrossRef\]](#)
- [9] Zhang J, Kasturi R. Text detection using edge gradient and graph spectrum. In: 2010 20th International Conference on Pattern Recognition; 2010 Aug; p. 3979–3982. IEEE. [\[CrossRef\]](#)
- [10] Luo B, Cross AD, Hancock ER. Corner detection via topographic analysis of vector-potential. *Pattern Recognit Lett* 1999;20:635–650. [\[CrossRef\]](#)
- [11] Arbelaez P, Maire M, Fowlkes C, Malik J. Contour detection and hierarchical image segmentation. *IEEE Trans Pattern Anal Mach Intell* 2010;33:898–916. [\[CrossRef\]](#)
- [12] Malek J, Abdallah MB, Mansour A, Tourki R. Automated optic disc detection in retinal images by applying region-based active contour model in a variational level set formulation. In: 2012 International Conference on Computer Vision in Remote Sensing; 2012 Dec; p. 39–44. IEEE. [\[CrossRef\]](#)
- [13] Alshatti W, Lambert P. Using eigenvectors of a vector field for deriving a second directional derivative operator for color images. In: 5th International Conference on Computer Analysis of Images and Patterns (CAIP'93); 1993 Sep; p. 149–156. Springer Berlin Heidelberg. [\[CrossRef\]](#)
- [14] Guo H, Lu Y, Sarka S. Depth detection of targets in a monocular image sequence. In: 18th Digital Avionics Systems Conference; 1999 Oct; p. 8-A. IEEE.
- [15] Gunasekar T, Thiravidarani J, Mahdal M, Raghavendran P, Venkatesan A, Elangovan M. Study of non-linear impulsive neutral fuzzy delay differential equations with non-local conditions. *Mathematics* 2023;11:3734. [\[CrossRef\]](#)
- [16] Raghavendran P, Gunasekar T, Balasundaram H, Santra SS, Majumder D, Baleanu D. Solving fractional integro-differential equations by Aboodh transform. *J Math Comput Sci* 2023;32:229–240. [\[CrossRef\]](#)
- [17] Gunasekar T, Raghavendran P. The Mohand transform approach to fractional integro-differential equations. *J Comput Anal Appl* 2024;358. [\[CrossRef\]](#)
- [18] Pan Y, Huang W, Lin Z, Zhu W, Zhou J, Wong J, Ding Z. Brain tumor grading based on neural networks and convolutional neural networks. In: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC); 2015 Aug; p. 699–702. IEEE. [\[CrossRef\]](#)
- [19] Kang K, Wang X. Fully convolutional neural networks for crowd segmentation. *arXiv preprint arXiv:1411.4464*. 2014.
- [20] Jabbar SI, Day CR, Heinz N, Chadwick EK. Using convolutional neural network for edge detection in musculoskeletal ultrasound images. In: 2016 International Joint Conference on Neural Networks (IJCNN); 2016 Jul; p. 4619–4626. IEEE. [\[CrossRef\]](#)
- [21] Hu K, Zhang Z, Niu X, Zhang Y, Cao C, Xiao F, Gao X. Retinal vessel segmentation of color fundus images using multiscale convolutional neural network with an improved cross-entropy loss function. *Neurocomputing* 2018;309:179–191. [\[CrossRef\]](#)
- [22] Qian R, Zhang B, Yue Y, Wang Z, Coenen F. Robust Chinese traffic sign detection and recognition with deep convolutional neural network. In: 2015 11th International Conference on Natural Computation (ICNC); 2015 Aug; p. 791–796. IEEE. [\[CrossRef\]](#)
- [23] Guo H, Wang G, Chen X. Two-stream convolutional neural network for accurate RGB-D fingertip detection using depth and edge information. *arXiv preprint arXiv:1612.07978*. 2016. [\[CrossRef\]](#)
- [24] Bertasius G, Shi J, Torresani L. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*; 2015; p. 4380–4389. [\[CrossRef\]](#)
- [25] Fu H, Xu Y, Wong DWK, Liu J. Retinal vessel segmentation via deep learning network and fully-connected conditional random fields. In: 2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI); 2016 Apr; p. 698–701. IEEE. [\[CrossRef\]](#)
- [26] Nugraha BT, Su SF. Towards self-driving car using convolutional neural network and road lane detector. In: 2017 2nd International Conference on Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT); 2017; p. 65–69. IEEE. [\[CrossRef\]](#)

- [27] Lim K, Jang WD, Kim CS. Background subtraction using encoder-decoder structured convolutional neural network. In: 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS); 2017; p. 1–6. IEEE. [\[CrossRef\]](#)
- [28] Joshi D, Patel MS, Ami N, Gohel J. Shape extraction using edge detection techniques. In: Proceedings of the 2014 International Conference on Information and Communication Technology for Competitive Strategies; 2014; p. 1–5. [\[CrossRef\]](#)
- [29] Peng-o T, Chaikan P. High performance and energy efficient Sobel edge detection. *Microprocess Microsyst* 2021;87:104368. [\[CrossRef\]](#)
- [30] Ismail SM, Said LA, Radwan AG, Madian AH, Abu-ElYazeed MF. A novel image encryption system merging fractional-order edge detection and generalized chaotic maps. *Signal Process* 2020;167:107280. [\[CrossRef\]](#)
- [31] Hoang ND, Nguyen QL. Metaheuristic optimized edge detection for recognition of concrete wall cracks: a comparative study on the performances of Roberts, Prewitt, Canny, and Sobel algorithms. *Adv Civ Eng* 2018;2018:7163580. [\[CrossRef\]](#)
- [32] Sundani D, Widiyanto S, Karyanti Y, Wardani DT. Identification of image edge using quantum Canny edge detection algorithm. *J ICT Res Appl* 2019;13:133–144. [\[CrossRef\]](#)
- [33] Şeker A, Diri B, Balık HH. Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme. *Gazi Mühendis Bilim Derg* 2017;3:47–64.
- [34] Gao Z, Wang L, Zhou L, Zhang J. HEp-2 cell image classification with deep convolutional neural networks. *IEEE J Biomed Health Inform* 2016;21:416–428. [\[CrossRef\]](#)
- [35] Li Q, Cai W, Wang X, Zhou Y, Feng DD, Chen M. Medical image classification with convolutional neural network. In: 2014 13th International Conference on Control Automation Robotics & Vision (ICARCV); 2014 Dec; p. 844–848. IEEE. [\[CrossRef\]](#)