



AN AUTOMATIC DETECTION OF GLAUCOMA USING DEEP LEARNING TECHNIQUES

¹Ishrat Muzafar, ²Dr.Khushboo Bansal

¹M.Tech Student, ²Assistant Professor,

¹Computer Science Engineering,

¹Desh Bhagat University, Mandi Gobindgarh, Fatehgarh Sahib, Punjab- 147301, India

Abstract: Glaucoma is generally the result of an increase in intraocular pressure, which leads to progressive damage to the optic nerve. This manifests as a gradual loss of the visual field, beginning with a decrease in lateral vision, to finally lose all vision. Since vision loss from glaucoma cannot be reversed, the diagnosis and early treatment is essential to preserve both the visual field and quality of life. Glaucoma usually develops slowly over time without pain nor acute attacks, so the lack of clear symptoms makes it difficult to detect and detect the glaucoma through regular eye exams. As glaucoma progresses, there will be a further pathological cupping of the optic disc due to changes in the optic nerve. To diagnose the same, the proposed model in this scheme algorithm is good enough to identify Glaucoma based on images by training at 40 epochs and effectively achieving a validation accuracy of 96.65%. This result is quite good and the model is successful in achieving the results for Glaucoma detection.

Index Terms - Glaucoma Detection, Deep Learning, Artificial Neural Networks, Convolutional Neural Networks.

I. INTRODUCTION

Glaucoma is a chronic, progressive neuropathy that affects the optic nerve, causing in a gradual and irreversible loss of vision, in addition, it is the second cause of blindness in the world [1]. The academy American Ophthalmology projects 111 million cases in 2040 [2]. Interventions such as medication or surgery can make a big difference in quality of life and delay the process of disease progression, but rarely change the prognosis in the long term [3]. Early and timely diagnosis of these changes can prevent blindness, since glaucoma can be treated in the initial stages [4]. Glaucoma can be distinguished into two types, Primary Open Angle Glaucoma (POAG) and closed-angle glaucoma (GAC). The first is the most common and gradually reduces peripheral vision without symptoms, and by the time symptoms are visible, damage has already occurred permanent; the destruction caused by POAG can progress to develop the so-called “tunnel vision”, in which you can only see what is directly in front of the subject [5]. The second is much less frequent, produces sudden symptoms pain in the eye, headaches, halos around the lights, dilation of the pupils, loss of vision, red eyes, nausea and vomiting [5]. On the other hand, the GAC attack can last a few hours, and then happen again, or it can be continuous or not give rest; in each attack can cause progressively greater loss of vision [6].

Among the traditional methods for diagnosis, there is tonometry, which consists of measuring the inner eye pressure (IOP), the normal pressure range is in the range of 12 to 22 mm Hg most cases are diagnosed with a pressure that exceeds 22 mm Hg; however, some people may have glaucoma within the normal range (Glaucoma Research Foundation, 2020). Ophthalmoscopy is a procedure that helps the doctor examine your optic nerve to detect damage from glaucoma by looking at the shape and the color of the optic nerve [7]. Perimetry is a field test test that generates a map of your entire visual field, this test will help your doctor

determine if glaucoma has affected your vision (Glaucoma Research Foundation, 2020), on the other hand, pachymetry is a simple test that measures the thickness of the cornea (the transparent window which is located in the front of the eye), this method can help the diagnosis, since there is the potential for corneal thickness to influence ocular pressure (IOP) readings [6]. The diagnostic methods mentioned are generally Complementary methods for the diagnosis of glaucoma. In contrast to traditional methods, to date, two tools have been developed: one mobile, Yanbao [9] and another web, Glaucoma Detection Tool [8], with in order to give an automated diagnosis of glaucoma from the fundus of the eye.

II. LITERATURE REVIEW

• General Scheme of a Convolutional Network

Convolutional network (CNN) is a special type of neural network for processing data that has a grid-like topology. The name convolutional neural network indicates that the network uses a mathematical operation called convolution. Convolution itself is a linear operation. So a convolutional network is a neural network that uses minimal convolution at one of its layers [10-14].

Convolutional neural network (ConvNets) is a special case of artificial neural network (ANN) which is currently claimed to be the best model to solve object recognition and detection problems [10-14].

Convolutional Neural Network (CNN) is included in the type of Deep Neural Network because of its high network depth and widely applied to image data. Technically, a convolutional network is an architecture that can be trained and consists of several stages. The input and output of each stage are several arrays called feature maps. The output of each stage is a feature map of the processing results from all locations on the input. Each stage consists of three layers, namely the convolution layer, activation layer and pooling layer [10-14].

Before or after subsampling, a sigmoid activation function plus bias is applied to each feature map. The above block composition can be seen in Figure 1, where a generic layout of the layers is represented.

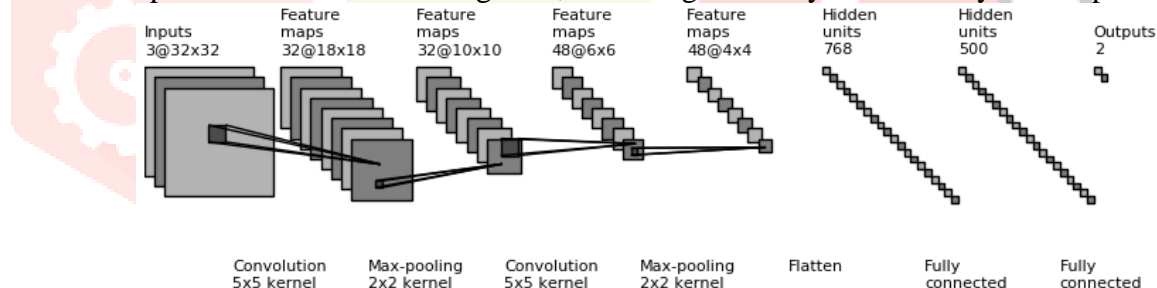


Figure 1: General Architecture of CNN

In addition, convolutional networks are designed assuming that the input to the network is an image, which allows encoding certain properties in the architecture, and allow parameters used in the network as:

$$Y_j = g \left(b_j \sum_{i=1} K_{ij} \otimes Y_i \right) \quad (eq. 1)$$

Where Y_j are the activation maps resulting from layer j Y_i are the activation maps of the previous layer i , K_{ij} are the filters or kernels corresponding to the connection between the layers i and j , b_j is the bias value associated with layer j and g is the activation function.

Furthermore, convolutional networks are designed assuming that the input to the network is an image, which allows encoding certain properties into the architecture, allowing parameters used in the network.

- **Input layer**

The first layer of the scheme of a convolutional network deals with the input to the network. As it has been seen previously, in convolutional networks you normally have the In addition, as it was also explained at the beginning of the document, in this project, the brain tumor-based database is used, so these input images will have a dimension of 3000 entries (images) will be passed to the network to carry out the training process and, once trained, its correct operation will be verified with 3000 test images [10-14].

- **Convolutional Layer**

Returning to the problem exposed in the computational system, the convolutional layer this stage of the network, a solution is carried out between the neurons of the hidden layer, and the hidden ones will only be connected with a small inspired by how the system works visual possess a series of receptive fields region or specific area. In addition, another noteworthy feature is that this means that the features or traits be the same as others that are found. On the other hand, convolution is a kernel operation) that generates a feature map. Convolutional neural networks

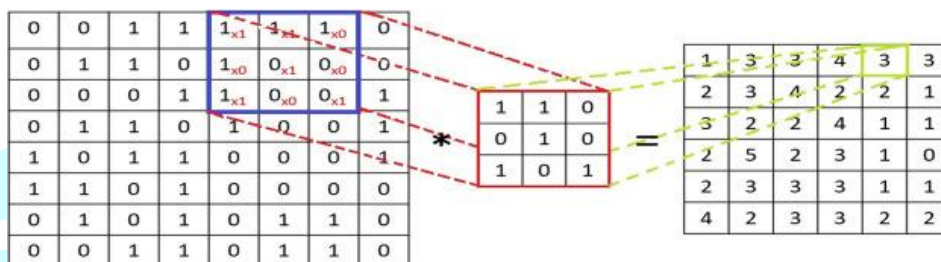


Figure 2: Convolutional Layer

- **Convolutional 2D**

$$G[i, j] = \sum_{u=-k}^k \sum_{v=-k}^k H[u, v] F[i - u, j - v] \text{ (eq. 2)}$$

- **Notation**

$$G = H \otimes F \text{ (eq. 3)}$$

In addition, convolutional networks are designed assuming that the input to the network is an image, which allows encoding certain properties in the architecture, allowing gains in efficiency and reduction. The first layer of the scheme of a convolutional network is about the input to the network. As previously seen, in convolutional networks, the input to the network is normally an image.

- **Pooling Layer**

Pooling or sub sampling is the reduction of the size of the matrix by using a pooling operation. Pooling layer is usually done after conv. Layers. There are two types of pooling that are often used, namely average pooling and max pooling. In average pooling, the value taken is the average value, while in max pooling, the value taken is the maximum value.

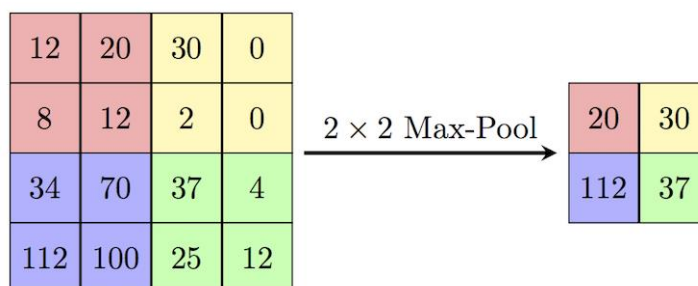


Figure 3: Max Pooling Process

Figure 2.27 shows the max-pooling operation for a 4x4 image with a 2x2 pooling mask. The output of the pooling process is a matrix with smaller dimensions than the initial matrix. The convolution and pooling processes are carried out several times to obtain a feature map of the desired size. The feature map will be the input for a fully connected neural network.

• The Output of the Pooling Layer

The operation of the pooling layer is as follows: first, from the upper left corner of the activation matrix, a pooling size submatrix is selected, and the following will be done: if it is mean-pooling or average-pooling, they take all the elements of the submatrix, their mean is calculated and the result is saved in the first position of the output matrix, if it is max-pooling the element with the highest value found in that submatrix is searched for and saved in the first position of the output array. Next, the next subarray that is to the right of the image is selected, the image is returned to the left, the elements are jumped down, and the image is traversed from left to right again. So, if the activation matrix created by the convolution layer is of size [10-14]. This is the last layer of the convolutional neural network scheme and it is a classifier that determines to which class the image belongs. input, ie your job in this project is to indicate what number the network gives to the input 'believes'.

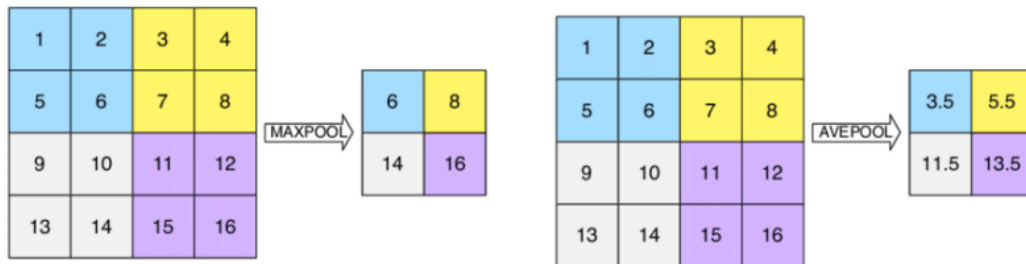


Figure 3: Output of Pooling

• The Fully Connected Layer

The Fully-Connected layer is the layer where all the activation neurons from the previous layer are all connected to the neurons in the next layer just like an ordinary artificial neural network. Each activation from the previous layer needs to be converted into one-dimensional data before it can be connected to all the neurons in the layer. The Fully-Connected layer is usually used in the Multi Layer Perceptron method to process data so that it can be classified. The difference between the Fully-Connected layer and the regular convolution layer is that the neurons in the convolution layer are connected only to certain regions of the input, while the Fully-Connected layer has neurons that are entirely connected. However, the two layers still operate pacifier products, so their functions are not so different.

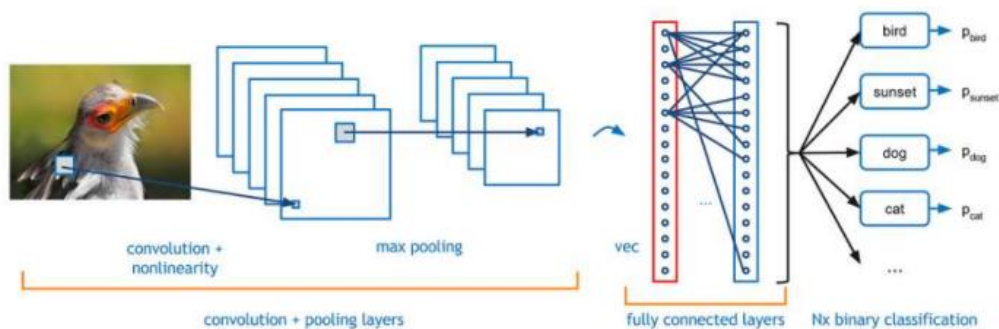


Figure 4: Fully Connected Layers for Classification

III. PROPOSED MODEL

This chapter will explain how the system design and analysis in this study. This research has several stages namely stages of system requirements to be made and solutions to the Glaucoma detection problem using CNN Algorithm based on fundus Images.

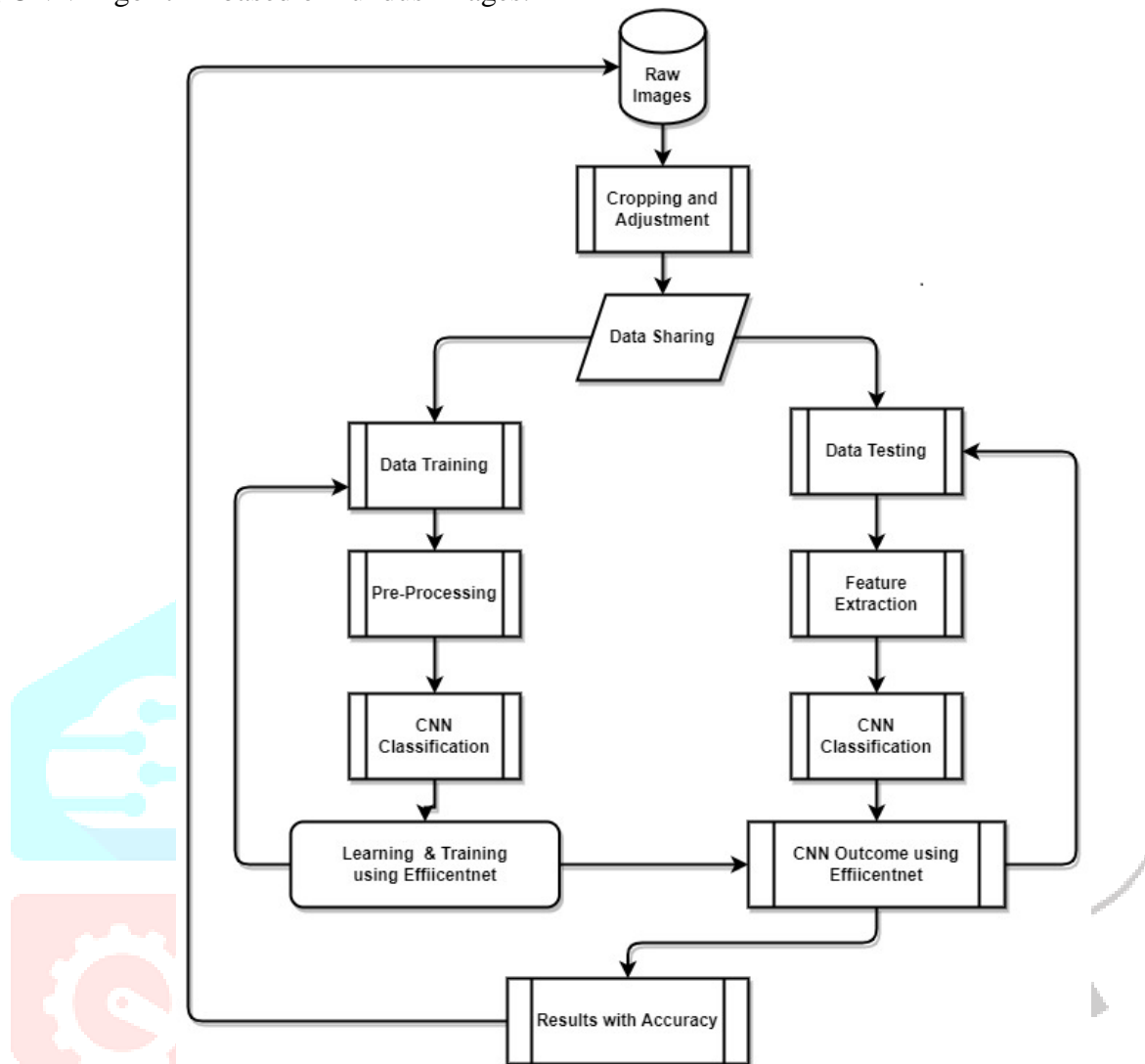


Figure 5. Proposed Scheme

- **Data Source**

Data needed for research uses primary data obtained directly from the website <https://www.kaggle.com/datasets/gunavenkatdoddi/eye-diseases-classification> which is. The data collected are two-dimensional Glaucoma images in JPG format, totalling 1000 images.

- **Pre-processing**

Pre-processing images are the steps taken to format images before use on the training model. Preprocessing image This research uses image resizing by defining image pixel size with a size of 224 x 224 at the start of the program. This thing works to reduce time training models and increase the speed of the inference model. Fully connected layers in convolutional neural network, requires all images to have the same array size. If the image size is not the same, it will affect the results of image detection. For comparison of the original image and results from preprocessing can be seen in Figure 6.

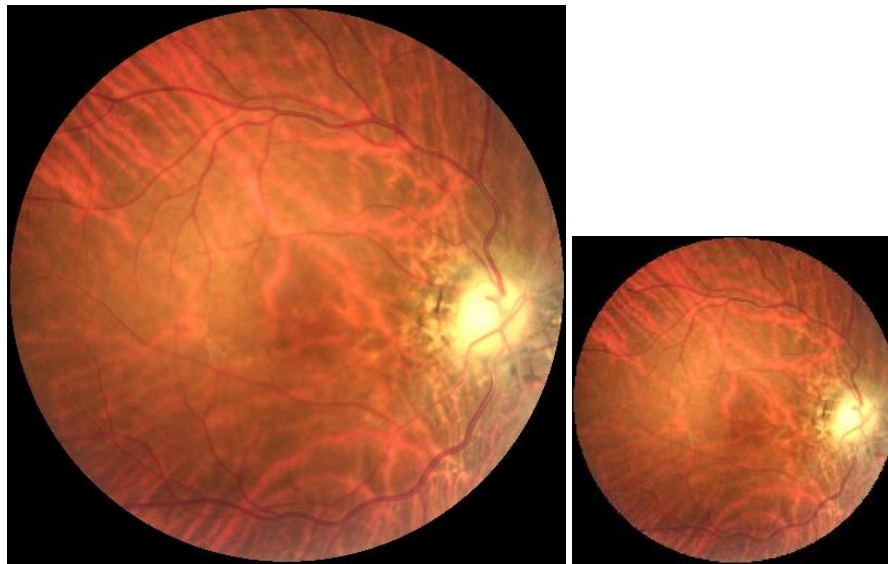


Figure 6: Original Data and Data After Resizing

- **Augmentation Image**

In this research stage after pre-processing image is an augmentation image. After the image is equalized in pixel size at the stage of pre-processing image then it will be processed with an augmentation image.

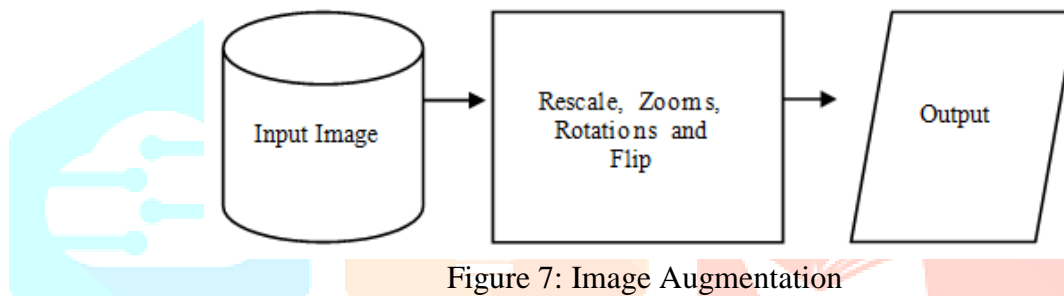


Figure 7: Image Augmentation

Image augmentation is a useful data manipulation technique for reproducing the data without losing the core data so that the algorithm can produce maximum performance

- **Algorithm CNN**

1. Input Glaucoma image with pixel size of $224 \times 224 \times 3$ where 224×224 is the size image pixels, while 3 is the number of channels which are images Red, Green, Blue (RGB), this image is then processed with two layers convolution size 2×2 and produce feature map size $224 \times 224 \times 32$, where the result of 32 is the number of channels after the image convoluted with the formula as in equation 3.1.

2. In the second process, the feature map resulting from the first process is processed with 1 layer of max-pooling measuring 2×2 and processed with a convolutional block to produce a feature map measuring $112 \times 112 \times 64$.

3. Then in the third process, the feature map resulting from the second convolution is processed with 1 layer of max-pooling measuring 2×2 and processed with a convolutional block to produce a feature map measuring $56 \times 56 \times 128$. The convolution and max-pooling formulas are found in equations 4 and 5.

$$f_l^k(p, q) = \sum_c \sum_{x, y} . i_c(x, y) e_l^k(u, v) \text{ (eq. 4)}$$

$$z_l^k = g_p(F_l^k) \text{ (eq. 5)}$$

4. Furthermore, the feature map resulting from the last process is processed with a convolutional block and the output is entered on 5 blocks, the first block is processed with 1 layer transpose and processed with 3 convolutional blocks which produce a $56 \times 56 \times 128$ feature map for image segmentation. The formula for the 2 convolutional blocks is in equation 7.

$$x_i = F(x_i, W_i), x_{i+1} = x_i + F(x_i, W_i) \text{ (eq. 7)}$$

5. Next, the second block is processed with 1 transpose layer measuring 2×2 and processed with the convolutional block which produces a $112 \times 112 \times 64$ feature map.

6. The third block is processed with 1 layer of transpose measuring 2×2 and processed which produces a feature map measuring $224 \times 224 \times 3$.

7. Transpose and convolution layers are followed by the ReLU activation function with the formula as in equation 8.

$$fReLU(h_{i,k}) = \max(0, h_{i,k}) \text{ (eq.8)}$$

8. The fully connected process uses the sigmoid activation function. The sigmoid activation function is useful for determining the results of Glaucoma detection and producing the output layer.

The use of each layer in the algorithm has an important role, where each different layer has its function. An explanation of each layer will be explained in the following paragraphs.

a. Convolutional layers: The convolutional layer consists of a set of convolutional kernels where each neuron acts as a kernel. The convolutional kernel works by dividing the image into small slices as shown in Figure 8.

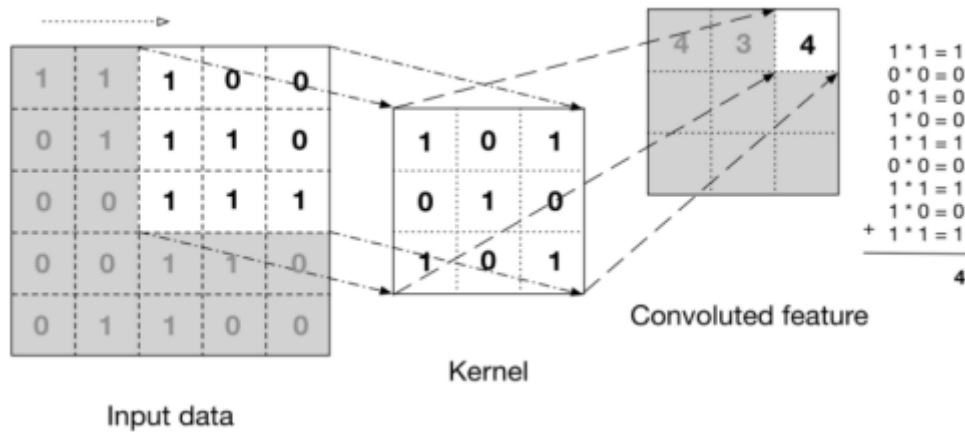


Figure 8: Process Convolutional Layer

Image Dividing: The image into small blocks helps in extracting feature. The kernel joins the images using a certain set of weights by multiplying their elements by the corresponding elements of the receptive field as $f_l^k(p, q) = \sum_c \sum_{x,y} i_c(x, y) e_l^k(u, v)$

where, $i_c(x, y)$ is the tensor element of the input image i_c , which is the wise element multiplied by $e_l^k(u, v)$ the index of the k^{th} convolution kernel of the l th layer. Meanwhile, the output map feature of the k^{th} convolution operation can be stated as $F_l^k = [f_l^k(1,1), \dots, f_l^k(p, q), \dots, f_l^k(P, Q)]$

b. The Pooling Layer: Feature Map that is generated as the output of the convolution operation, can occur at different locations in the image. Once a feature is extracted, its exact location becomes less important as long as an estimate of its position relative to others is maintained. Pooling or down-sampling is an interesting local operation. It summarizes similar information around the receptive field and displays the predominant response in this locale $Z_l^k = g_p(F_l^k)$.

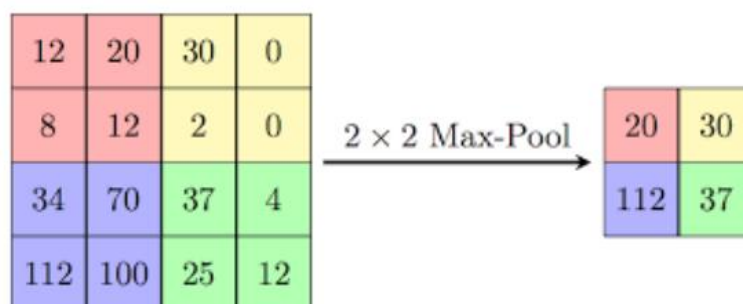


Figure 9: Max Pooling

The above equation shows a pooling operation where Z_l^k is the layer l feature map pooled l^{th} for input k features map F_l^k , while $g_p(F_l^k)$ defines the type of pooling operation.

c. Fully Connected Layer: Fully Connected Layer is a function in image detection, in this stage each neuron at the top level is interconnected with every neuron at the next level. The Fully Connected Layer receives the output from the upper layer (which represents a feature map of

higher level features) and determines the most suitable output. An overview of the fully connected layer process is shown in Figure 3.6, in this process using the ReLU and Sigmoid activation functions.

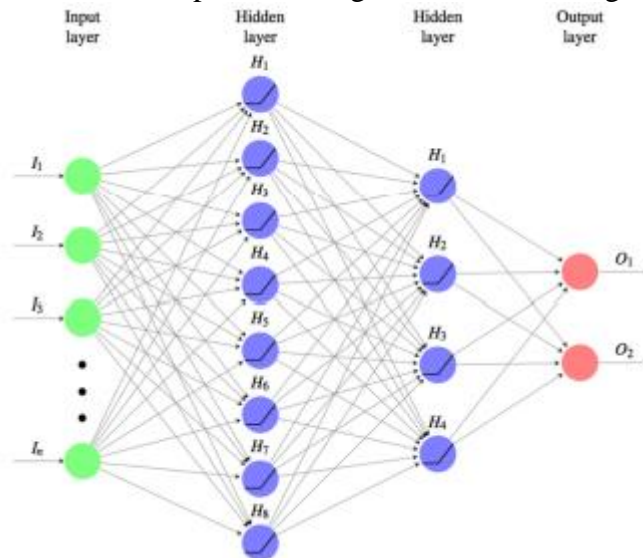


Figure 10: Fully Connected Layer

d. Activation Function The activation function serves as a decision function and helps in learning complex patterns. Choosing the right activation function can speed up the learning process. The activation function for the feature map is defined in the equation $T_l^k = g_a(F_l^k)$.

In the equation above, F_l^k is the output of the convolution, which is assigned to the activation function g_a which adds the non-linearity and returns the output T_l^k transformed for the 1th layer. Thereafter, activation function ReLU is used to embed combinations of non-linear features. However, ReLU and its variants are more widely used because they help overcome the problem of missing gradients.

ReLU - It is known that standard activation functions such as the sigmoid function or the hyperbolic tangent function are contractive almost everywhere and the gradient at large values becomes almost zero. Thus the update by the feasible stochastic gradient becomes very small as $fReLU(h_{i,k}) = \max(0, h_{i,k})$.

IV. RESULTS

• Data Augmentation

Data augmentation is a step in the processing of picture data. Augmentation is the technique of altering or modifying an image in such a manner that the computer will recognize the altered image as being different yet people can still recognize the altered image as being different. The identical image has been edited. Because the model augmentation receives additional data that can be valuable for creating a model that can generalize better, it can increase the accuracy of the CNN model that is trained. Data augmentation uses a number of techniques to process image data, including the following techniques:-

1. **Resize:** Resize is a method for changing the size of an image to size which are desired. Resize aims to decrease the size of the image to 224 *224 512 *512 as the size of the image is at 512 *512 of resolution which is very high for processing over GPU.
2. **Grayscale:** Grayscale: The process of changing the image channel to grayscale aims to save image size because it only uses one channel that only displays the intensity value or the degree of gray using $gray_{ij} = \frac{f(i,j)_R + f(i,j)_G + f(i,j)_B}{3}$ as under.


```

# define model parameters
img_size = (224, 224)
channels = 3 # either BGR or Grayscale
color = 'rgb'
img_shape = (img_size[0], img_size[1], channels)

tr_gen = ImageDataGenerator(preprocessing_function= scalar, horizontal_flip= True)
ts_gen = ImageDataGenerator(preprocessing_function= scalar)

```

Figure 11: Data Augmentation

• Model Training

The learning parameter that is initialized in the code above is the learning rate parameter with the Stochastic Gradient Descent (SGD) optimizer. SGD is a machine learning optimization method that is useful for updating the weights for each parameter being trained. Due to the large number of parameters that must be trained on the FC Layer network, the SGD optimizer is considered suitable for use. From the CNN architecture above, the dataset will be trained according to the training model that has been created previously. The process for training this model is often referred to as model fitting. The model fitting process can be done by calling the `model.fit()` method, then entering the data variables and iteration parameters that must be executed during the model fitting process.

```

1 batch_size = 40 # set batch size for training
2 epochs = 40 # number of all epochs in training
3 patience = 1 #number of epochs to wait to adjust lr if monitored value does not improve
4 stop_patience = 3 # number of epochs to wait before stopping training if monitored value does not improve
5 threshold = 0.9 # if train accuracy is < threshold adjust monitor accuracy, else monitor validation loss
6 factor = 0.5 # factor to reduce lr by
7 ask_epoch = 5 # number of epochs to run before asking if you want to halt training
8 batches = int(np.ceil(len(train_gen.labels) / batch_size)) # number of training batch to run per epoch
9

```

Figure 12: Model Training

When this fitting model is run, the CNN architecture that has been formed will immediately work to execute the data that has been prepared previously. The training process is carried out in a number of predetermined epochs. Every time you run iteration, the system will automatically display the accuracy value and loss value from the training data. Consequently, the model achieves the accuracy value of 99.91% at the time of training.

13 /40	0.346	98.786	0.45360	94.313	0.00100	0.00100	val_loss	12.29	70.27
14 /40	0.298	98.904	0.39866	94.787	0.00100	0.00100	val_loss	12.11	68.92
15 /40	0.272	98.963	0.40819	93.602	0.00100	0.00050	val_loss	-2.39	69.39
16 /40	0.232	99.615	0.37696	93.839	0.00050	0.00050	val_loss	5.44	70.14
17 /40	0.217	99.763	0.37321	94.550	0.00050	0.00050	val_loss	0.99	70.50
18 /40	0.203	99.763	0.35401	94.550	0.00050	0.00050	val_loss	5.15	69.38
19 /40	0.198	99.585	0.36696	94.313	0.00050	0.00025	val_loss	-3.66	69.90
20 /40	0.187	99.704	0.33505	95.261	0.00025	0.00025	val_loss	5.36	70.11
21 /40	0.186	99.556	0.34651	93.839	0.00025	0.00013	val_loss	-3.42	69.47
22 /40	0.177	99.793	0.32151	95.735	0.00013	0.00013	val_loss	4.04	69.49
23 /40	0.174	99.822	0.31998	95.261	0.00013	0.00013	val_loss	0.48	70.61
24 /40	0.170	99.882	0.31657	95.024	0.00013	0.00013	val_loss	1.06	70.62
25 /40	0.169	99.793	0.31110	95.261	0.00013	0.00013	val_loss	1.73	70.73
26 /40	0.165	99.941	0.31213	95.498	0.00013	0.00006	val_loss	-0.33	70.38
27 /40	0.166	99.911	0.31727	95.024	0.00006	0.00003	val_loss	-1.98	71.58

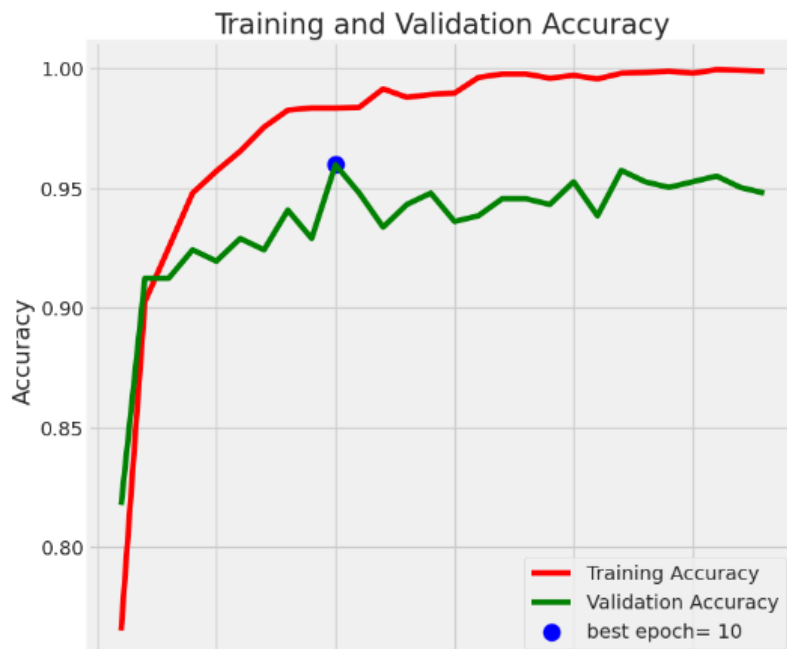


Figure 13: Accuracy Evaluation of Proposed Model

After the CNN model goes through the training process, the model will be tested to test how well the performance of the previously trained model. This testing process is done by testing a new image that has never been trained before by the model. That way, it will be seen how well the performance of the model that has been made to identify. The first step is to call the model that was previously formed during the training. The next process is to bring up the prediction results from the tested image. To bring up the prediction results, the system will compare the weight between the image to be predicted with the output weight on the model that has been called. After the image weight is validated, the image will be classified to the output with the closest weight. The results of this test will be in the form of classification results and values probability of the output weights closest to the image being tested. Therefore, able to achieve the validation accuracy of 96.65% effectively.

V. CONCLUSION

From the results of research carried out to identify Glaucoma using the Convolutional Neural Network (CNN) method, the following conclusions can be drawn:

1. Basically the CNN algorithm is an algorithm capable of classifying images without requiring additional feature extraction. Because the algorithm already has a feature learning stage or process. However, in the course of this research, too little amount of data is owned and the poor quality of the data obtained makes researchers add additional feature extraction to make it easier for the algorithm to perform feature extraction on the image to be processed by rescaling the images.

2. The CNN algorithm is good enough to identify Glaucoma based on this images by training at 40 epochs and the validation accuracy of 96.65% effectively. This result is quite good and model is successfully to achieve the results for Glaucoma detection.

REFERENCES

- [1] Allison K, Patel D, Alabi O. Epidemiology of Glaucoma: The Past, Present, and Predictions for the Future. *Cureus*. 2020 Nov 24;12(11):e11686. doi: 10.7759/cureus.11686. PMID: 33391921; PMCID: PMC7769798.
- [2] Tham, Yih-Chung & Li, Xiang & Wong, T-Y & Quigley, Harry & Aung, Tin & Cheng, Ching-yu. (2014). Global Prevalence of Glaucoma and Projections of Glaucoma Burden through 2040 A Systematic Review and Meta-Analysis. *Ophthalmology*. 121. 10.1016/j.ophtha.2014.05.013.
- [3] Diaz-Pinto, Andres & Morales, Sandra & Naranjo, Valery & Köhler, Thomas & Mossi, José & Navea, Amparo. (2019). CNNs for automatic glaucoma assessment using fundus images: An extensive validation. *BioMedical Engineering OnLine*. 18. 10.1186/s12938-019-0649-y.

- [4] Bua, Silvia & Supuran, Claudiu. (2019). Diagnostic markers for glaucoma: a patent and literature review (2013-2019). *Expert Opinion on Therapeutic Patents*. 29. 10.1080/13543776.2019.1667336.
- [5] Abbasi, Ashkan & Antony, Bhavna & Gowrisankaran, Sowjanya & Wollstein, Gadi & Schuman, Joel & Ishikawa, Hiroshi. (2023). Can Glaucoma Suspect Data Help to Improve the Performance of Glaucoma Diagnosis?. *Translational vision science & technology*. 12. 6. 10.1167/tvst.12.8.6.
- [6] Rieger-Reyes, Cristina & Rubio-Galán, F.J.. (2013). [Glaucoma: Pharmacological implications].. *Semergen / Sociedad Española de Medicina Rural y Generalista*. 39. 26-33. 10.1016/j.semerg.2012.05.003.
- [7] Alkhatib, Ali. (2023). Glaucoma: Types, Risk Factors, Detection, and Management. *Scholars Academic Journal of Biosciences*. 11. 207-211. 10.36347/sajb.2023.v11i06.003.
- [8] Soorya, Megha & Issac, Ashish & Dutta, Malay. (2019). Automated Framework for Screening of Glaucoma Through Cloud Computing. *Journal of Medical Systems*. 43. 136. 10.1007/s10916-019-1260-2.
- [9] F. Guo et al., "Yanbao: A Mobile App Using the Measurement of Clinical Parameters for Glaucoma Screening," in *IEEE Access*, vol. 6, pp. 77414-77428, 2018, doi: 10.1109/ACCESS.2018.2882946.
- [10] Teoh, Teik & Rong, Zheng. (2022). Convolutional Neural Networks. 10.1007/978-981-16-8615-3_16.
- [11] Ye, Jong Chul. (2022). Convolutional Neural Networks. 10.1007/978-981-16-6046-7_7.
- [12] Ronneberger, Olaf & Fischer, Philipp & Brox, Thomas. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation.
- [13] Sewak, M., Karim, M. R., & Pujari, P. (2018). *Practical Convolutional Neural Networks: Implement advanced deep learning models using Python*. Packt Publishing.
- [14] Khan, S., Rahmani, H., & Shah, S. A. A. (2018). *Guide to Convolutional Neural Networks for Computer Vision (Synthesis Lectures on Computer Vision)*. Morgan & Claypool.

