

Glaucoma Detection Using Machine Learning

Sharanya S

Jr. Software Engineer

Email: [sharanyas1999\[at\]gmail.com](mailto:sharanyas1999[at]gmail.com)

Abstract: *Glaucoma is an eye disease if not detected in the early stage leads to permanent blindness. It is the second leading cause for eye blindness. The fundus camera is a type of modern imaging device that is used to examine the internal structure of the eye. Some of the methods used to detect glaucoma include the Topcon image net method, optical coherence tomography, and the retinal nerve fibre layer analyser. However due to high cost and lack of research in this field optic cup to- disc ratio is used to detect the glaucoma. To determine the glaucomatous damage, the appearance of the optic cup plays an important role. The cup enlarges with the advancement of glaucoma occupying most of the disc area. The optic cup-to disc ratio compares the diameter of the optic cup portion of the optic disc with the total diameter of the optic disc. The manual examination of optic disk and optic cup is time consuming. Hence automatic glaucoma detecting algorithm is developed. However, enlargement of optic cup itself is not indication of glaucoma because optic cupping may occur without glaucoma due to hereditary factor.*

Keywords: Machine learning, Glaucoma, kernel, MATLAB, CNN, Rim-One, ConvNets

1. Introduction

1.1 Introduction to Machine Learning

Machine learning is the idea of using algorithms to find patterns and/or make predictions based on a collection of data. There are various algorithms available, each with its own set of benefits and drawbacks, as well as levels of complexity. These algorithms are easily accessible and available across a variety of programming tools (including R and Python) with varying levels of coding requirements. They can do away with the need for comprehensive coding instructions specific to your application in favor of more general instructions.

1.2 The Basics of Image Processing

Image processing is a technique for applying operations to an image in order to improve it or extract useful information from it. It's a form of signal processing in which the input is an image and the output is that image or its characteristics/features. Image processing is one of today's fastest-evolving innovations. It is also a crucial research field in engineering and computer science. Image processing basically includes the following three steps:

- Importing the image using image acquisition software.
- Analyzing and manipulating the image.
- Output, which may be an altered image or a report based on image analysis.

2. Description

Glaucoma is nothing but the multi factorial neurodegenerative disease, which is degrades the vision over time which is responsible for loss vision of 66.8 million people in all over the world. To evaluate the potential risk of the disease is take too much time in diagnose as well as for treatment of the glaucoma disease. Intra ocular pressure (IOP) is the elevated blood pressure within the eye, and as the disease progresses, it reaches a harmful amount, causing the optic nerve axon to split. As a result, the key risk factor

for developing glaucoma is an increase in IOP pressure. This will hasten the loss of vision and eventually lead to blindness. Glaucoma is similar to intraocular hypertension, except it causes damage to the optic nerve. Since the optic nerve was damaged, the brain was unable to receive image information from the light receptors. The physiologist's complaint is the degeneration of optic nerve cells, which affects both the optic nerve head and the visual area. Even though glaucoma damage is permanent, early diagnosis and medical care recommended by ophthalmologists may help reduce the likelihood of vision loss. The diameter of a glaucoma eye varies more than the diameter of a normal eye due to fluid pressure.

Glaucoma is a condition of the optic nerve that worsens over time. It usually occurs when fluid accumulates in the front of our eyes. The fluid inside a normal eye has a pressure of 21mm Hg. If the fluid level in our eyes rises, the pressure in our eyes rises, causing damage to the optic nerves. The condition progresses rapidly and can result in vision loss in both eyes. Glaucoma-related blindness, on the other hand, can also be avoided with early detection and care. In the early stages of glaucoma, there are normally no signs, and the condition progresses within the eye.

3. Objectives

- Collecting images of human eyes using appropriate image capturing devices a large number of samples from various sources from hospitals and image database.
- To detect glaucoma in an optic image by calculating the optic cup to optic disc ratio by segmenting the optic image using CNN.

Problem Statement

To design and implement glaucoma detection to find whether patients are suffering from eye disease.

System Design

The methodology of this task is divided into three stages: pre-processing, run of GA which involves finding the simplest CNN architecture and run glaucoma classification of CNN. The proposed method's steps are represented in

Volume 10 Issue 4, April 2021

www.ijsr.net

[Licensed Under Creative Commons Attribution CC BY](#)

Figure 1.

4. System Architecture

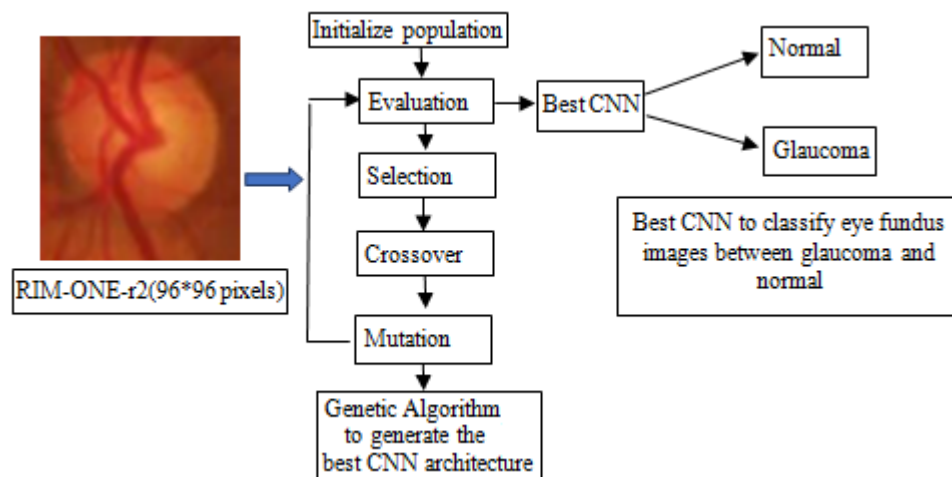


Figure 4.1: System Architecture

Rim-One, which was used in this research, is a public database that contains images of the I fundus in RGB (Red, Green, and Blue) format, as well as a guide for the creation of the optic nerve segmentation algorithm. In our experiments, we used version r2 of the Rim- One dataset, which includes images from the "glaucoma and suspect" class as well as images from the "standard" class. There are 455 images in the dataset, with 255 in the regular class and 200 in the "glaucoma and suspected" class. Pre-processing RIM-ONE-r2 with varying widths and heights yielded the images. The unique pre-processing technique used in this work is designed to fill the size of all images up to 96x96 pixels with a black pixel, with a technique known as zero-padding, with input pixels of input volume. Including zeroing the edge the format of images is thus beneficial for an accurate classification. A value of 96x96pixels was obtained from experience with poor results for another size (28x28, 32x32, 64x64 pixels). More than 96x96 pixels were also tested, but our GPU (Nvidia GeForce GTX 1060) memory could not support tests that supported output out memory errors. The entire image given by the RIM-One dataset was used as the input ROI (region of interest) for our method in this version of the dataset. This approach considered all information beyond the internal contents of the OD, such as the excavation, macula, and blood vessels. Genetic algorithms and training this work uses an approach to the development of automated CNN architectures. It is based on a neuro evolution technique known as NEAT (Neuro evolution of Augmenting Topology). The NEAT technique performs the optimization of components, topology and hyper parameters. The effect of the developed network is determined by how well they can be trained through gradient descent to perform the function of classification. The genetic algorithm is a vampirism driven by the process of natural selection. It was proposed by John Holland (1975) and commonly used to generate high-quality solutions to optimization and discovery problems by relying on bio-inspired operators such as mutations, crossovers, and selection is done.

GA was evaluated by training each CNN generated on the same dataset partition: train data with 227 images, validation data with 92 images, testing data with 136 images. Furthermore, this phase lasted more than 200 epochs, using a batch size of 16. The adapter function was used as an optimizer to compile each model and binary cross-entropy was used for loss, as our problem is a binary classification problem. To facilitate development and accelerate implementation, all processes were performed in Python with the Cairns API to initialize our GA, an initial population with 20 individuals is randomly created. This population should have 50% of batch normalization, determination, relay activation and dropout occurring with at least one layer (likely to be of other layers). Each individual (chromosome) is formed in a series of letters representing a specific layer, as described in Table 1 Example of a chromosome. B: batch normalization; C: determination; R: relay activation; d: dropout berd the evaluation function determines the fitness of an individual Due to our proposed concerns over resolving image classification tasks, classification verification loss is the best strategy to assign to their fitness (fitness=1/3loss). In this scenario, after training for this individual run with a slight validity loss, the best person from each generation will be chosen. Using the tournament selection criterion, the selection will return a random new parent, with the top performer being carried on to the next generation and mating with the rest of the population. Each child in the new population has a choice. Crossover is the technique that will choose a person's parents to mingle and merge their chromosomes to produce two new children. In our case, the parent layers must have the same length as the crossover occurs in the single point model and this point is chosen randomly, respecting the structural of the CNN. The probability of crossover in our GA is 30%. Mutation, according to allow the algorithm to diversify populations, expanding the opportunity to search for unexplained regions in the search space for fitter solutions. The mutation is implemented by giving each element in the chromosome array randomly drawn the possibility of the following scenarios: deleting or adding determination

clauses; Add or remove dropouts; Changing the order of punishment blocks; Changing the inputs of punishment blocks; Changing the output of large blocks; Change the order of large blocks. To evaluate each CNN, the training

procedure was used on the train set data, the validation set data on the test set data of the RIM-ONE-R2, and the validated procedure on the test procedure.

4.1 Data Flow Diagram

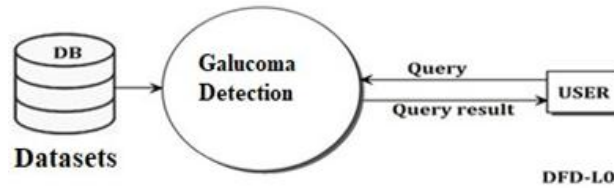


Figure 4.1: DFD LEVEL 0

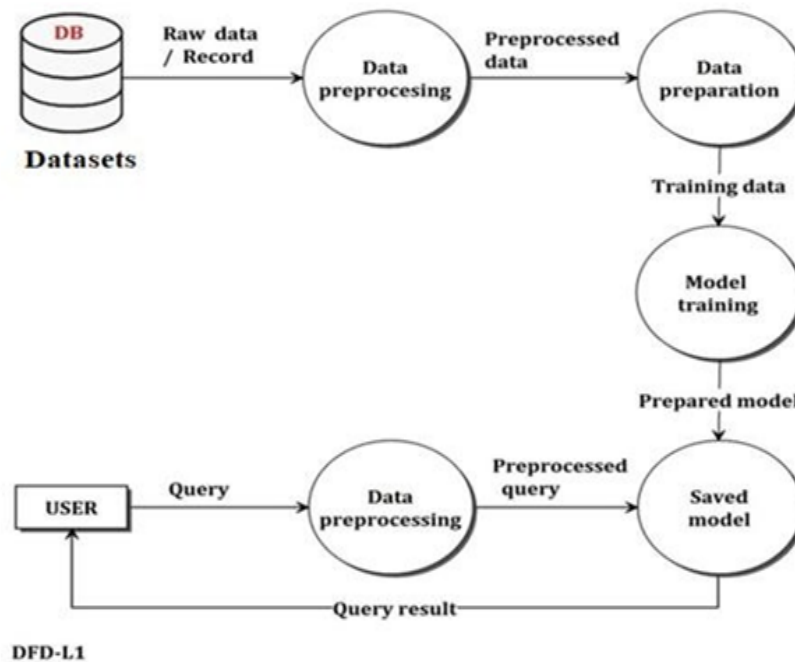
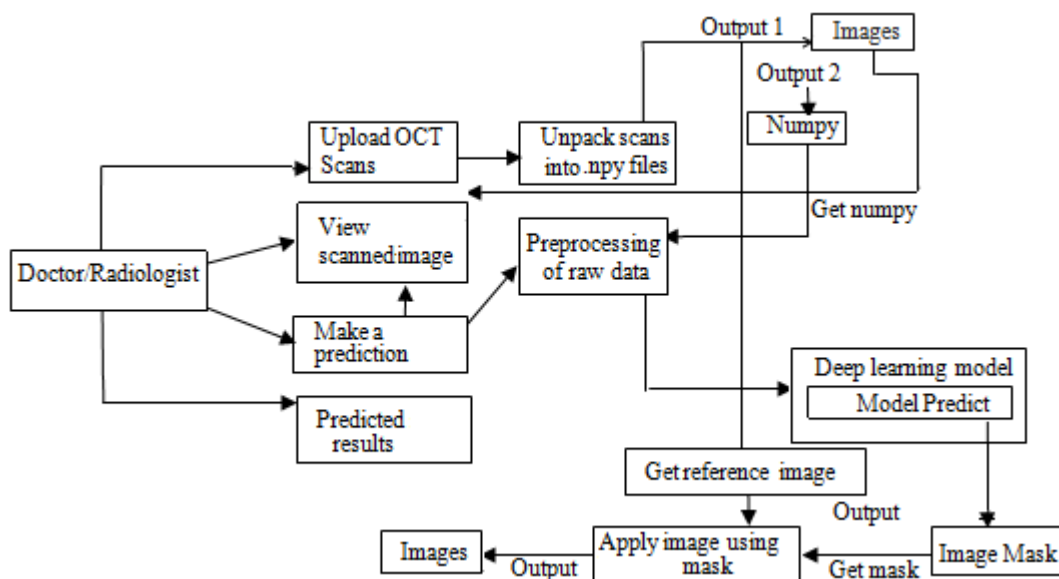


Figure 4.1.1: DFD LEVEL 1

4.2 Use caseDiagram



5. Implementation

5.1 Flow Chart

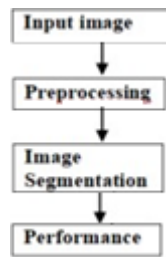


Figure 5.1: Flow chart

- The input test image is acquired and preprocessed in the next stage and it is converted into array form for comparison.
- The selected database is properly preprocessed and the renamed into proper folders.
- Though hemorrhage is difficult to detect, preprocessing is required to obtain a noise-free, clear, contrast-enhanced picture.

The steps including preprocessing to detect hemorrhages are:

- Resize the image into 512*512 pixels.
- Convert the RGB image into grey scale image.
- Use Median Filtering to remove artifacts.
- Equalize the image and enhance contrast.
 - The model is properly trained using CNN then classification takes place.
 - The comparison of the test image and trained model takes place followed by the display of the result.

We begin by extracting the green channel from the image because the affected region is visible and easy to identify during this channel. Then we add a median filter with an 8-pixel radius to create a backdrop and subtract it from the original image. This results in a blood vessel and hemorrhage picture. We detach the vessel that ends during a picture with hemorrhages suggested using a vessel mask.

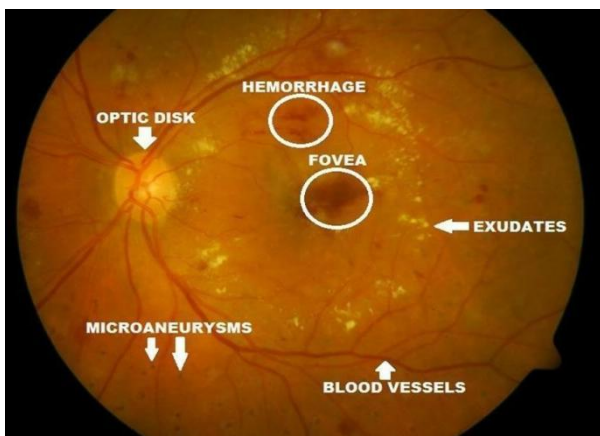


Figure 5.2: Retina with hemorrhages and exudates

5.2 Exudates detection

The method we have applied to detect exudates on human

retina is inspired by the work described in [20]. Since the data set is of completely different characteristic as we've changed in various sides. That's why we are visiting describe every step and so the rationale behind taking it. Here we wish to mention that we have got implemented some library provided by [21]. we've also used MATLAB version 2017a for this project and this detection consists of the next steps:

- Pre-processing the image.
- Detection of point and other artifacts.
- Detection of exudates in terms of optic disc and artifacts.

First, we extract intensity constituents from an image in the preprocessing phase. Here we are visiting work with gray-scale images because exudates are mostly visible in such images. We then apply Median Filtering for reducing the noise and apply Histogram Equalization to spice up contrast and brightness. The resulting image helps us to detect blind spot and accordingly exudates. This works as input image. Exudates are high intensity values likewise as optic disk. Therefore on travel for exudates detection we would like to go looking out blind spot then we'd prefer to differentiate between optic disc and exudates near and inside the blind spot area. To do and do that we consider that time is that the most important and most circular part in brightest portion of the image. We apply Gray Scale Closing is used to extract blood vessels from the retina, particularly in the blind spot region. We'll use a flat disc-shaped structure feature with an eight-radius radius. We binarize the image by thresholding it and using the resulting image as a mask. The mask is then inverted by pixels before being overlaid on the original image. On the overlaid image, we then apply reconstruction by dilation. We threshold the image and use the algorithm to find the difference between the original image and the reconstructed image. As a result, a high-intensity point is identified, and rests are eliminated.

In this section, we dealt with an infinite problem with this approach. The grey Scale Closing was used to delete vessels at the start of the strategy, and reconstruction was applied to the image produced from the initial image. As a result, we'll be visiting the aim area to reconstruct vessels. However, we have a problem in that there does not seem to be a single large circular blind spot. Rather, during this stage, we are detecting two or three large connected components. We used an addition dilation of the last word mask to solve this problem. As a consequence, the separate areas are bound in a circular pattern. We can see that there are already objects and other bright spots in the shot. As a consequence, if we use too much dilation, the blind spot can merge with those areas. A flat disc shaped structured feature with a radius of 4 was considered for the necessary additional dilation. Since the optic disc and also some bright artifacts are detected during this process, we've estimated for every component of the mask so on tell apart between the features some extra values. These additional values are termed as scores. Thus we have, Score = area circularity 3.

Here's a situation that warrants your attention. We decided to give circularity more significance because we have a situation where the feature rather than the point can become much larger than the blind spot. We treat elements with a

scale of less than 1100 pixels as a blind spot and treat the rest as artefacts. Tiny areas that may become exudates are not classified as objects in this case. After point extraction and artefact detection, we'll look for exudates at this level. Grey Scale Closing, as before, eliminates high-intensity blood vessels. Then we decide to create a regular deviation picture that depicts the most important features of nearly organized exudates. Grey Scale Closing, as before, eliminates high-intensity blood vessels. Then we decide to create a regular deviation picture that depicts the most

important features of nearly organized exudates.

By setting the radius to six, the resulting image is thresholded. The surface outline of the retina is then removed, and the gaps are filled with `imfill()`. Threshold is used to exclude blind spots and objects. Finally, the result is obtained by applying an A level 0.01 threshold between the primary and thus the reconstructed data. To encourage accurate vision, the created exudates mask image is overlaid onto the primary image.



Figure 5.2: Exudates detection. (a) Original image

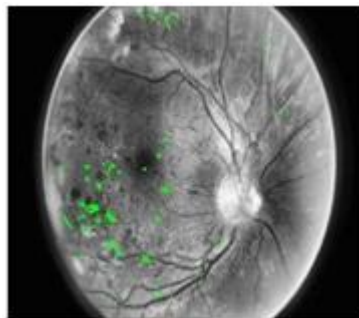


Figure 5.2: Exudates detection. (b) Exudates

After the entire feature extraction has been done, now we are visiting perform binary classification. Here we've used deep neural nets with two input layers, a whole of three layers, one representing the output. For this we've created a feed-forward back propagation network (`newff`). The phrases 'newfit' and 'newpr' stand for 'regression' and 'classification,' respectively. They together are called 'newff' the generic name which continues to be available and provides better output in our classification.

First and foremost, we have built a two-stage feed forward network. Three 'transit' neurons make up the first layer, and one 'purelin' neuron makes up the second layer. As a result, we have

```
net = newff(p,t,[3,1],);
```

Here, p denotes the input vectors matrix and t denotes the

goal vectors matrix. We used three components for the input vectors: blind spot mask, objects mask, and exudates mask.

The network is then simulated, and the results are plotted.

As a result, we have $y = \text{sim}(\text{net}, p)$;

We should notice that the network has been trained for 5000 epochs and the train-parameter target is 0.01.

After the training is completed, a .mat file is created and loaded to test our datasets. If loaded, we can tell the difference between an honest and a bad picture. Then the `corr2` library function is used to hunt out the correlation between four classes of images. The test image belongs to the category with which it correlates most.

5.3 Design of Machine Learning Model

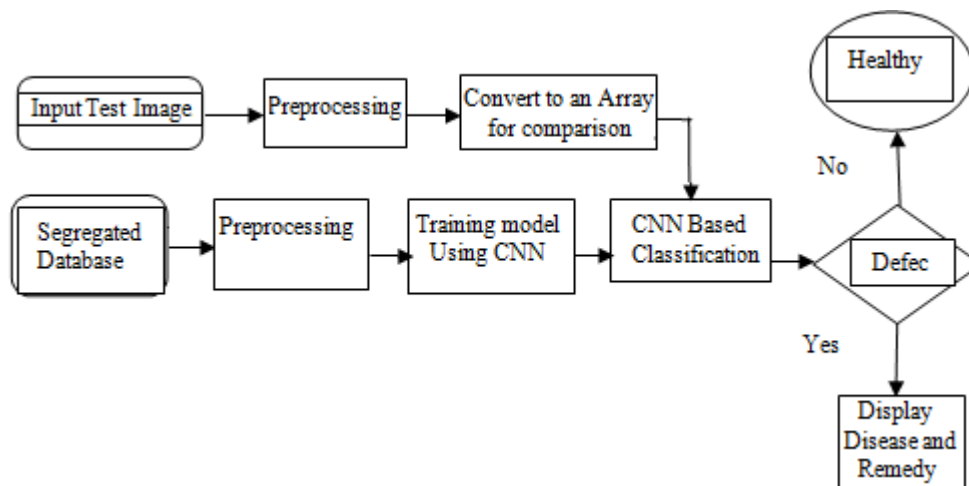


Figure 5.3: Machine Learning Model

Steps:

- 1) The input test image is acquired and pre-processed within the subsequent stage so it's converted into array form for comparison.
- 2) The selected database is correctly segregated and pre-processed so renamed into proper folders.
- 3) The model is correctly trained using CNN, Vgg16 Module then classification takes place.
- 4) The comparison of the test image and thus trained model occur followed by the display of the result.
- 5) If there is a retinal fundus it'll be classified.

5.4 Convolutional Neural Networks

The invention of the CNN in 1994 by Yann LeCun is what propelled the world of applied science and Deep learning to its former glory. The first neural network named LeNet5 had a awfully less validation accuracy of 42% since then we have got come a protracted way during this field. Nowadays almost every giant technology firms rely on CNN for more efficient performance. The concept to detect diseases in mulberry leaf incorporates the use of CNN before we dive into the “functionality and dealing of CNN” concept, we must have a basic idea on how the human brain recognizes an object in spite of its varying attributes from one another. Our brain features a fancy layer of neurons ,each layer holds some information about the article and each one the features of the article are extracted by the neurons and stored in our memory, next time after we see the identical object the brain matches the stored features to acknowledge the article, but one can easily mistake it as a simple “IF-THEN” function, yes it's to some extent but it's a further feature that has it a foot hold over other algorithms that's Self-Learning, although it cannot match a human brain but still it can provides it a difficult competition. Image is processed using the elemental CNN to detect the diseases in leaves. The data training in our CNN model must satisfy following constraints:

- 1) There should be no missing values in our dataset.
- 2) The dataset must distinctly be divided into training and testing sets, either the training or the testing set shouldn't contain any irrelevant data out of our model

domain just in case of an image dataset all the pictures must be of the identical size, one uneven distribution of image size in our dataset can decrease the efficiency of our neural network.

- 3) The pictures should be converted into black and white format before feeding it into the convolution layer because reading images in RGB would involve a 3-D numPy matrix which might reduce the execution time of our model by a considerable amount.
- 4) Any quite corrupted or blurred images should even be trimmed from the database before feeding it into the neural network. Now we've learned the data pre-processing rules, allow us to dive right into the working of the convolutional neural network.

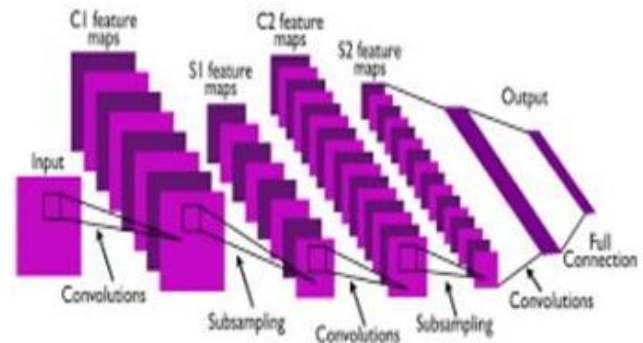


Figure 5.4: CNN layers

A. Convolution Layer

The pattern is identified by scanning the entire image and preparing it as a 3x3 matrix in this layer. The matrix kernel refers to the image's decorated function. A weight vector represents each value in the kernel.

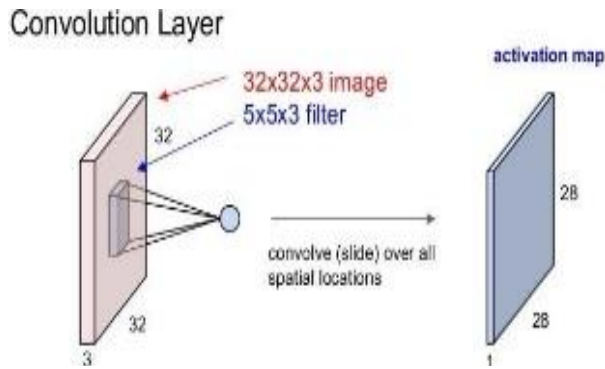


Figure 5.4.1: Convolution Layer

B. Pooling layer

When it comes to pooling, the picture matrix breaks down into a collection of four non-overlapping rectangular segments. Pooling can be divided into two types: maximum pooling and average pooling. The maximum value in the relative matrix region that is taken is given to max pooling. The average value in the relative matrix region is obtained by average pooling. The pooling layer's key benefit is that it boosts device efficiency while reducing over-fitting opportunities.

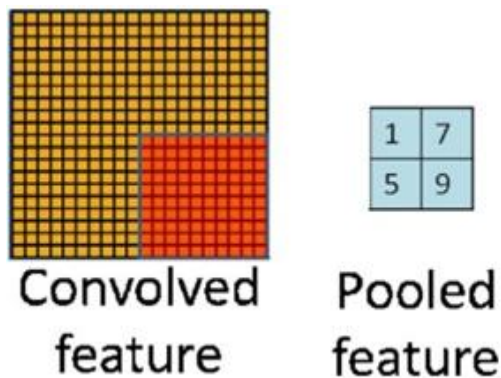
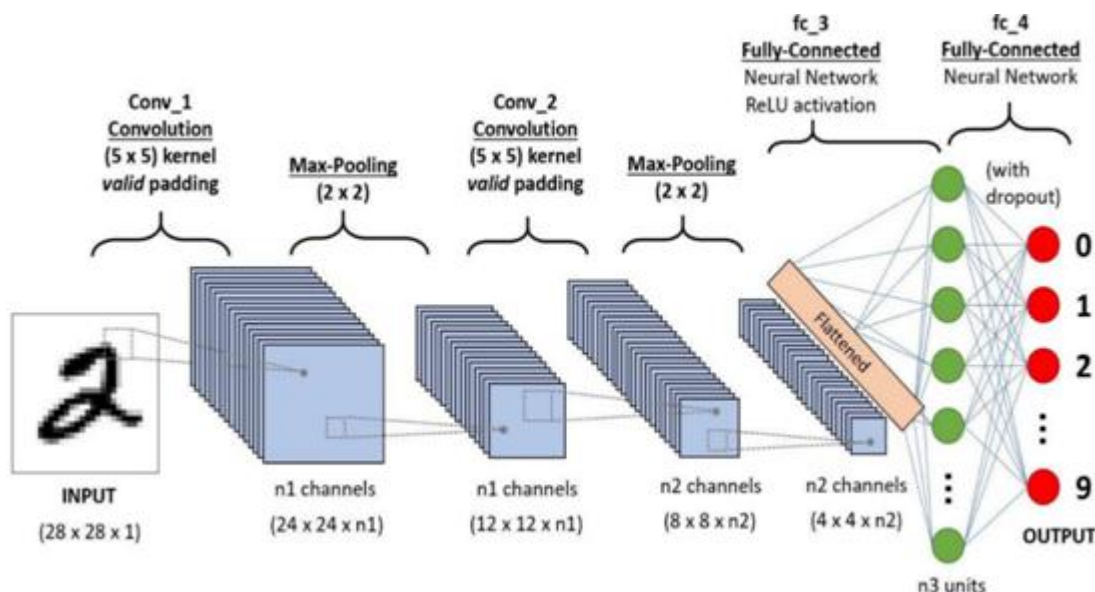


Figure 5.4.2: Pooling Layer

C. Activation layer



It is that part of the conversational neural networks where the values are normalized, fitting them within a certain range. The determination used is ReLU which allows only positive values and then rejects negative values. It is a function of low computational cost.

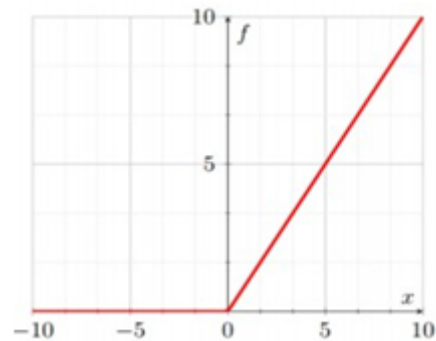


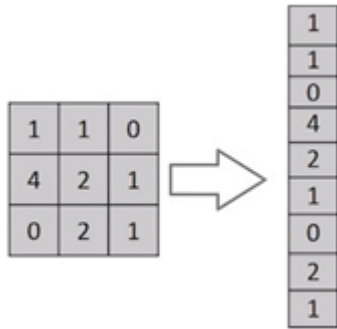
Figure 5.4.3: ReLU function

D. Fully joined layer

The features are compared to the test image's features, and the same features are applied with the defined mark. Labels are usually encoded as numbers for numerical convenience; they will be translated to their respective strings later.

Artificial intelligence has made significant progress in bridging the difference between human and computer capabilities. Researchers and enthusiasts alike work on a number of facets of the world in order to produce amazing items. The field of computer vision is one of these. The agenda for this field is to permit machines to work out the world as humans see it, to perceive it during a uniform way, and even to use knowledge for a range of tasks like image and video recognition, image analysis and classification, media recreation, recommendation systems, tongue processing, etc. Advances in deep learning computer vision are built and refined over time, primarily through a specific algorithm, a convolutional neural network.

Why ConvNet over Feed-Forward Neural Nets?



Leveling a 3x3 image matrix during a 9x1 vector

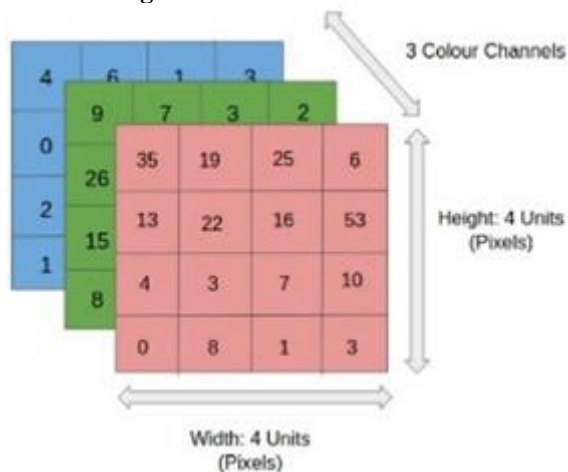
Isn't an image nothing more than a matrix of pixel values? Why not simply flatten the image (for example, a 3x3 image matrix into a 9x1 vector) and feed it to a multi-level perceptron for classification?

When it comes to extremely simple binary images, the strategy can predict classes with a mean exact score, but when it comes to complex images with pixel dependencies, there is no precision. With the use of the appropriate filters, a conveyor is capable of successfully capturing spatial and temporal dependencies during an image. Because of the reduction and reusability of the weights within the amount of parameters involved, the architecture performs much better fitting to the image dataset.

In other words, networks are often trained to better understand the complexity of a picture.

Input Image

4x4x3 RGB Image



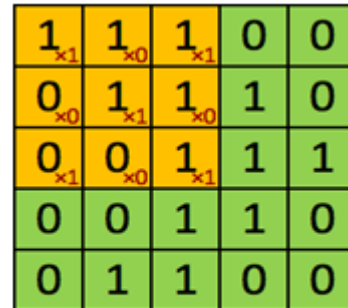
In the figure, we have an RGB image separated by three color planes-red, green and blue. There are several such color spaces in which images exist: grayscale, RGB, HSV, CMYK, etc.

You can imagine how computing resources will turn out when the images reach sizes, say, 8K (7680x4320). ConvNet's role is to transform images into a form that is easier to process without losing the functions that are critical

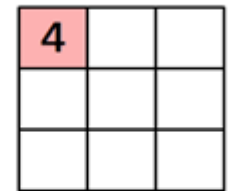
to getting a good forecast. This is important when we want to design an architecture that is not only good at learning, but also scalable for massive datasets.

Convolution Layer the Kernel

Conversion of 5x5x1 image with a 3x3x1 kernel to request a 3x3x1 decorated feature Image dimensions = 5 (height) x 5 (width) x 1 (number of channels, eg. RGB)



Image



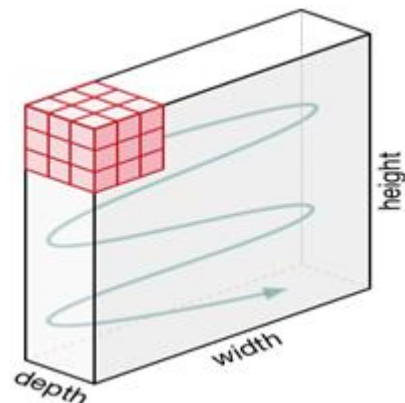
Convolved Feature

In the above demonstration, the green segment corresponds to our 5x5x1 input image, I. The first element involved in completing the convolution operation within a part of a conventional layer named the kernel / filter, represented within the color yellow has gone. We have chosen K as the 3x3x1 matrix.

Kernel/Filter, K=

1 0 1
0 1 0
1 0 1

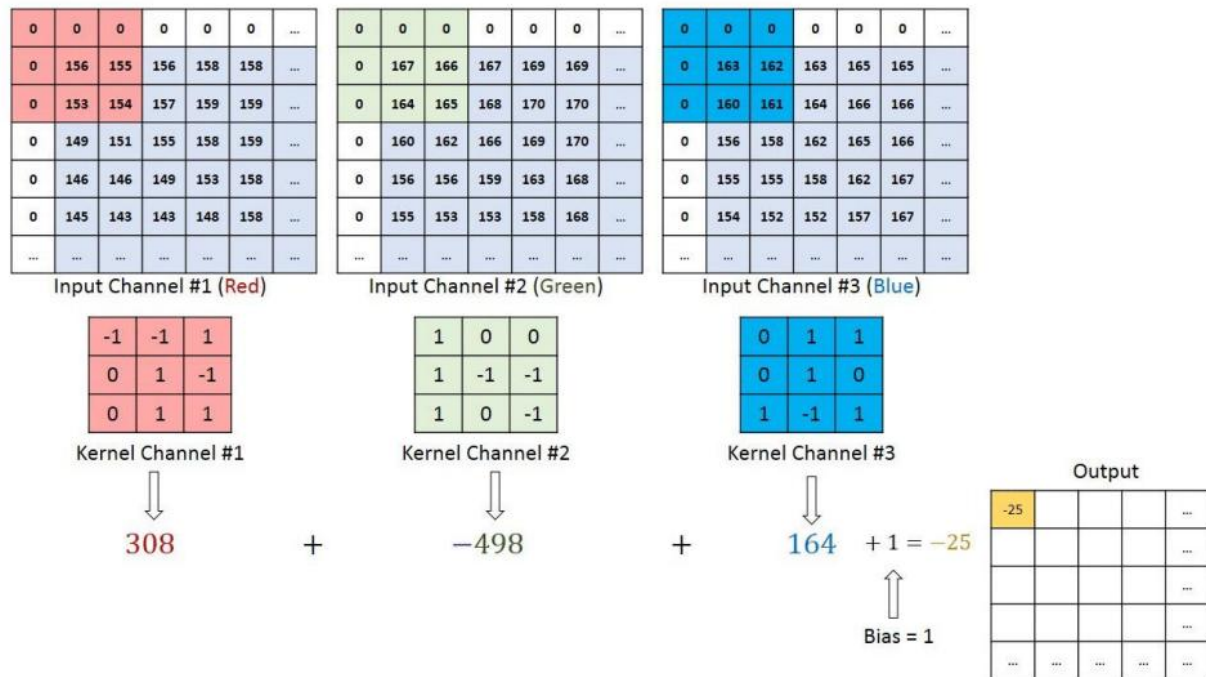
The core shifts 9 times due to the step length=1 (without sharpness), each time performing the matrix multiplication operation between K and part P of the image over which the core hangs.



Kernel movement

The filter moves to the right with a fixed stride value until it crosses the entire width. Moving forward, it moves down to the beginning (left) of the image with the same stride

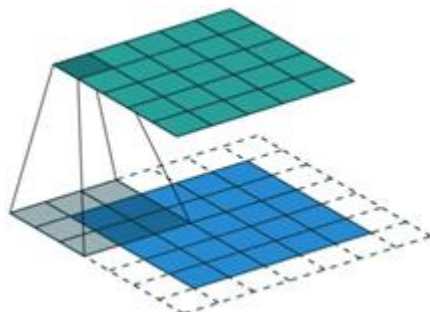
value and repeats this process until the entire image traverses.



Convention operation on $M \times N \times 3$ image matrix with $3 \times 3 \times 3$ kernel

The kernel has the same depth as the input image in the case of images with multiple channels (such as RGB). Matrix multiplication is performed between the kernel and the stack ($[K1, I1]; [K2, I2]; [K3, I3]$) and all results are expressed with bias such that the squash-depth channel's function performance is obtained.

The convention operation is used to detach high-level curvets from the input image as edges. There is no need to restrict ConvNets to only one Convolutional Layer. The first converter is usually in charge of capturing low-level features like edges, colour, gradient orientation, and so on. The design adopts high-level features with additional layers, giving us a network that makes sense. Similar representations of how we'd be in a dataset. There are two types of results of the operation - one in which the deflected attribute is reduced in dimensionality compared to the input, and the other in which the dimensionality is either increased or the same. This is done by applying a valid padding in the former case, or asingle padding in the latter case



Same Filling: $5 \times 5 \times 1$ image is supplemented with 0 s to

create a $6 \times 6 \times 1$ image

The convoluted matrix is $5 \times 5 \times 1$ in size when we expand the $5 \times 5 \times 1$ image to a $6 \times 6 \times 1$ image and then add the $3 \times 3 \times 1$ core to it. As a result, the gasket has the same name. On the other hand, if we perform the same operation without filling, we get a matrix that has the dimensions of the core ($3 \times 3 \times 1$) Valid Padding.

6. Results

```

Epoch 1/3 [=====] - ETA: 0s - loss: 0.3199 - accuracy: 0.4950 - val_loss: 0.7525 - val_accuracy: 0.5938
Epoch 2/3 [=====] - ETA: 0s - loss: 0.4209 - accuracy: 0.4564 - val_loss: 0.4927 - val_accuracy: 0.5938
Epoch 3/3 [=====] - ETA: 0s - loss: 0.4774 - accuracy: 0.7050 - val_loss: 0.4927 - val_accuracy: 0.5938
Epoch 4/3 [=====] - ETA: 0s - loss: 0.4825 - accuracy: 0.7171 - val_loss: 0.5043 - val_accuracy: 0.5625
Epoch 5/3 [=====] - ETA: 0s - loss: 0.3772 - accuracy: 0.8021 - val_loss: 0.5312 - val_accuracy: 0.5625
Epoch 6/3 [=====] - ETA: 0s - loss: 0.4205 - accuracy: 0.7500 - val_loss: 0.5646 - val_accuracy: 0.6075
Epoch 7/3 [=====] - ETA: 0s - loss: 0.3567 - accuracy: 0.7969 - val_loss: 0.6119 - val_accuracy: 0.6075
Epoch 8/3 [=====] - ETA: 0s - loss: 0.3712 - accuracy: 0.8113 - val_loss: 0.5035 - val_accuracy: 0.5625
Epoch 9/3 [=====] - ETA: 0s - loss: 0.4253 - accuracy: 0.4547 - val_loss: 0.7184 - val_accuracy: 0.7188
Epoch 10/3 [=====] - ETA: 0s - loss: 0.4240 - accuracy: 0.4711 - val_loss: 0.5201 - val_accuracy: 0.7188
Epoch 11/3 [=====] - ETA: 0s - loss: 0.4130 - accuracy: 0.6975 - val_loss: 0.5035 - val_accuracy: 0.5625
Epoch 12/3 [=====] - ETA: 0s - loss: 0.3993 - accuracy: 0.7184 - val_loss: 0.4470 - val_accuracy: 0.4091
Total: 320
loss: 0.351675791121608 accuracy: 0.554630221847086
conv_engine_sequential.Sequential object at 0x00000204007798c0
model loaded
Not Olanoma
Olanoma
Olanoma
Olanoma
Traceback (most recent call last):

```

Figure 6.1: Error detection

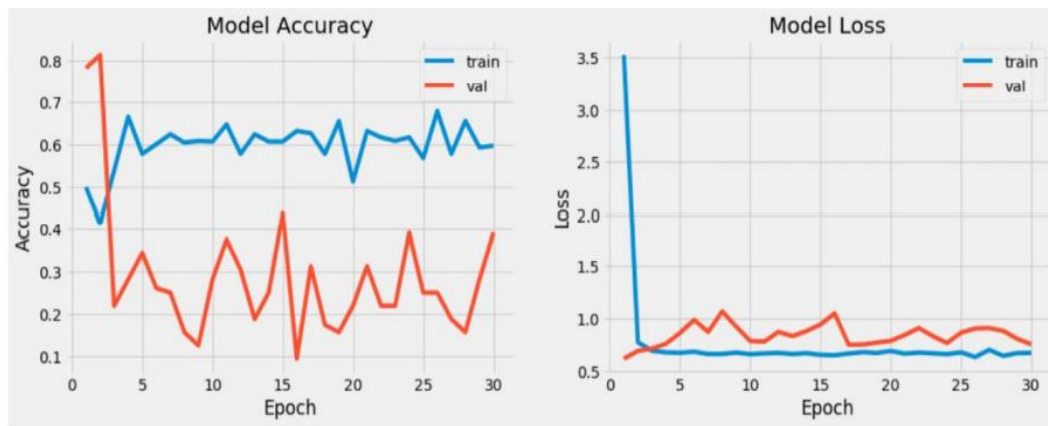


Figure 6.2: Accuracy detection

7. Conclusion

We have proposed a new deep learning method, named CNN, for automatic glaucoma detection and pathological area localization upon fundus images. The attention prediction, pathological region localization, and glaucoma classification subnets make up our CNN model. We developed the LAG database with 5,824 fundus images labelled with either positive or negative glaucoma, as well as their focus maps on glaucoma detection, to train the CNN model.

References

- [1] Amira Soltani, Tahar Battikh, Imed Jabri, Yosra Mlouhi, Med Najeh Lakhroua, "Study of contour detection methods as applied on optic nerve's images for glaucoma diagnosis", CoDIT'16 - April 6-8, 2016, Malta.
- [2] Sharanagouda Nawaldgi, "Review of Automated Glaucoma Detection Techniques", This full-text paper was peer-reviewed and accepted to be presented at the IEEE WiSPNET 2016 conference.
- [3] Shwetali M. Nikam, Dr. C. Y. Patil, "Glaucoma Detection from Fundus Images Using MATLAB GUI", 2017 IEEE
- [4] Shivangi C. Patel, Manish I. Patel, "Analysis of CDR of Fundus Images for Glaucoma Detection", Proceedings of the 2nd International Conference on Trends in Electronics and Informatics (ICOEI 2018)
- [5] Juan Carrillo, Lola Bautista, Jorge Villamizar, Juan Rueda, Mary Sanchez and Daniel Arueda, "GLAUCOMADETECTIONUSINGFUNDUSIMAGES OFTHEEYE", 2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA)