CATARACT DETECTION

A PROJECT REPORT

Submitted by

ARYAN SHUKLA (181B052)
KSHITIZ MISHRA (181B111)
MOHD SHARIQ KHAN (181B127)

Under the guidance of: DR. AJAY KUMAR

JUNE - 2022

Submitted in partial fulfillment for the award of the Degree

of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science & Engineering

JAYPEE UNIVERSITY OF ENGINEERING & TECHNOLOGY, AB ROAD, RAGHOGARH, DT. GUNA-473226 MP, INDIA

DECLARATION BY THE STUDENT

We hereby declare that the work reported in the B.Tech. project entitled as "**CATARACT DETECTION**", in partial fulfillment for the award of degree of B.Tech. (CSE) submitted at Jaypee University of Engineering and Technology, Guna, as per best of my knowledge and belief there is no infringement of intellectual property right and copyright. In case of any violation, we will solely be responsible.

ARYAN SHUKLA (181B052)
KSHITIZ MISHRA (181B111)
MOHD SHARIQ KHAN (181B127)

Department of Computer Science and Engineering
Jaypee University of Engineering and Technology, Guna

Date: 30TH MAY, 2022

JAYPEE UNIVERSITY OF ENGINEERING AND TECHNOLOGY
(Accredited with Grade-A by NAAC & Approved by UGC Under Section 2(f) of the UGC Act 1956)
A.B. ROAD, RAGHOGARH, DIST: GUNA (M.P.) INDIA
Phone: 07544 267051, 267310-14, Fax: 07544 267011
website: www.juet.ac.in

REG: JUET: ADM MAY 30,2022

CERTIFICATE

This is to certify that the work entitled **"CATARACT DETECTION"**, submitted by **ARYAN SHUKLA (181B052), KSHITIZ MISHRA (181B111), MOHD SHARIQ KHAN (181B127)** in partial fulfillment for the award of degree of Bachelor of Technology of Jaypee University of Engineering and Technology, Guna has been carried out under my supervision. As per best of my knowledge and belief there is no infringement of intellectual property right and copyright. Also, this work has not been submitted partially or wholly to any other Institute or University for the award of this or any other degree or diploma. In case of any violation concern student will solely be responsible.

DR. AJAY KUMAR

Asst. Professor (SG), Dept. of CSE

Date: 30th MAY, 2022

ACKNOWLEDGEMENT

We wish to thank **DR. AJAY KUMAR**, our Project Mentor, who allowed us to work in such a peaceful environment, as for collecting information about concerned topic and for providing such a wonderful project to us.

We will be thankful to him for guiding us with his valuable suggestions, which helped us a lot so as to complete our project. We enjoyed working on this project and gained so much valuable information.

Aryan Shukla (181B052)

Kshitiz Mishra (181B111)

Mohd Shariq Khan (181B127)

Date: 30th MAY, 2022

EXECUTIVE SUMMARY

Cataract is one of the most common eye disorders that causes vision distortion. The creation of a cloud on the lens of our eyes is known as a cataract. Blurred vision, faded colors, and difficulty seeing in strong light are the main symptoms of this condition. As a result, preliminary cataract detection and prevention may help to minimize the rate of blindness.

We aim towards analyzing and recognizing various eye images from a database. Database consists of various images with 300 images of normal eye and 300 eyes with cataract, with each image clicked in different conditions. With such a divergent data set, we are able to train our system to good levels and thus obtain good results.

Following the early promising results of AI systems in various eye diseases, there have also been several AI algorithms developed for automated detection and grading of cataract, based on either machine learning or deep learning approaches.

This project is aimed at classifying cataract disease using Artificial Neural Networks based on a publicly available image dataset. In this observation, we used Machine Learning Models such as Support Vector Machine (SVM), Random Forest Classifier, Logistic Regression, K-Nearest Neighbors (KNN), Naïve Based Algorithms and Deep Learning Model of Artificial Neural Network with input layer, hidden layer and output layer. This model predicts cataract disease with a training loss of 15.4574%, a training accuracy of 84.5243%. In addition, the model greatly minimizes training loss while boosting accuracy

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Cataract is a lenticular opacity clouding the transparent lens in human eyes. Typically, the lens converges the light to the retina. The presence of the cataract causes this light to be blocked and not reach the lens that results in poor visual acuity. It is a worldwide leading eye disease that develops gradually and does not affect sight early. However, after a while, it can interfere with vision and even cause vision loss in people over age 40. Cataract detection in earlier stages may avoid painful and costly surgeries and prevent blindness depending on its severity.

The world health organization (WHO) reported that about 285 million people in the world have a visual impairment. Among them, 39 million people have limited vision, and the remaining ones have impaired vision. Cataract was responsible for 33% of visual impairment, and 51% of blindness. In 2020, Flaxman *et al.* predicted that the number of people suffering from moderate to severe vision impairment (MSVI) and blindness would be 237.1 and 38.5 million, respectively. Of them, 57.1 million (24%) and 13.4 million (35%) people would be affected by cataract. The worldwide blindness will exceed 40 million by 2025. Comparing the results of these reports prove that there was only a slight improvement in the eye care system and controlling the vision loss during the last decade.

Among the leading causes of blindness such as glaucoma, corneal opacity, trachoma, and diabetic retinopathy, cataract accounts for the most significant proportion. It is considered as one of the leading causes of blindness. Cataract can be categorized into three main groups based on the location and area where it develops: Nuclear Cataract, Cortical Cataract, and Posterior Sub Capsular (PSC) Cataract.

These three types of cataracts occur due to several common factors such as aging, diabetes, and smoking. Early detection of cataracts plays a vital role in the treatment and can significantly reduce the risk of blindness. The state-of-the-art automatic cataract detection systems consist three steps: pre-processing, feature extraction, and classification. These methods are categorized into two groups based on the algorithms used in either feature extraction or classification stages: Machine Learning (ML)-based and Deep Learning (DL)-based methods.

CHAPTER 2

THEORETICAL BACKGROUND

This section introduces the theoretical backgrounds of the various aspects that are part of the thesis in detail. The topics include Machine Learning Algorithms and Deep Learning Algorithms

2.1 Machine Learning

Machine learning is a form of AI that enables a system to learn from data rather than through explicit programming. However, machine learning is not a simple process. As the algorithms ingest training data, it is then possible to produce more precise models based on that data. A machine-learning model is the output generated when you train your machine-learning algorithm with data. After training, when you provide a model with an input, you will be given an output. For example, a predictive algorithm will create a predictive model. Then, when you provide the predictive model with data, you will receive a prediction based on the data that trained the model.

Machine learning enables models to train on data sets before being deployed. Some machine- learning models are online and continuous. This iterative process of online models leads to an improvement in the types of associations made between data elements. Due to their complexity and size, these patterns and associations could have easily been overlooked by human observation. After a model has been trained, it can be used in real time to learn from data. The improvements in accuracy are a result of the training process and automation that are part of machine learning.

Machine-learning techniques are required to improve the accuracy of predictive models. Depending on the nature of the business problem being addressed, there are different approaches based on the type and volume of the data. In this section, we discuss the categories of machine learning.

2.1.1 Supervised learning

Supervised learning typically begins with an established set of data and a certain understanding of how that data is classified. Supervised learning is intended to find patterns in data that can be applied to an analytics process. This data has labeled features that define the meaning of data. For example, you can create a machine-learning application that distinguishes between millions of animals, based on images and written descriptions.

2.1.1.1 Support Vector Machine (SVM)

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:

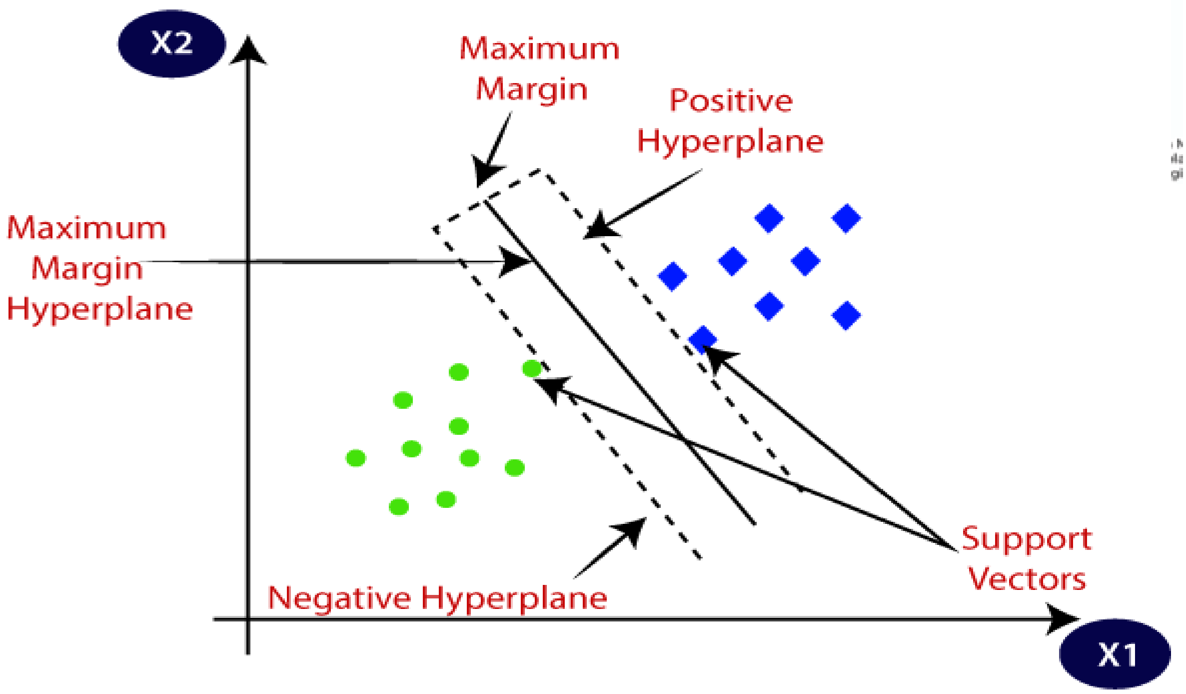

Fig 1: SVM Classifier

The working of the SVM algorithm can be understood by using an example. Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2. We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue. Consider the below image:

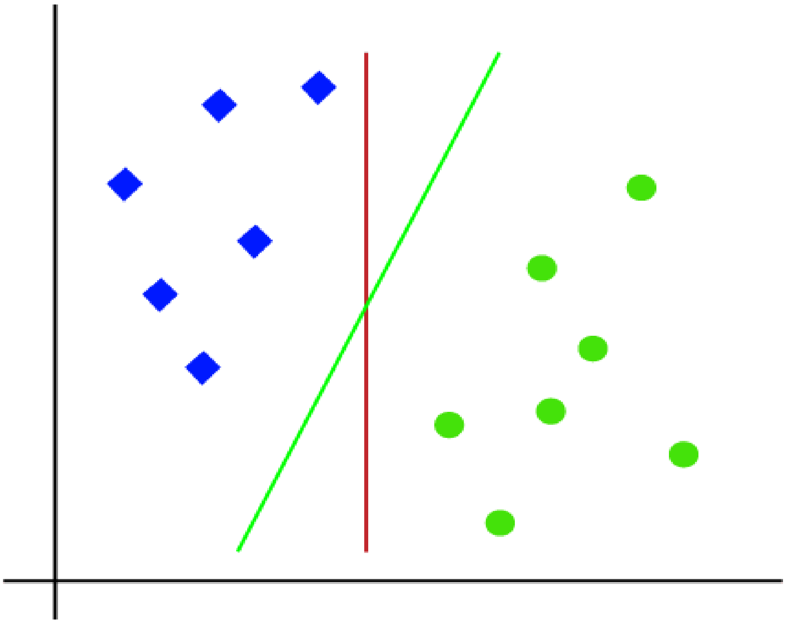

Fig 2: SVM Separation vectors

So as it is 2 d space so by just using a straight line, we can easily separate these two classes. But there can be multiple lines that can separate these classes.

Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a hyperplane. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as margin. And the goal of SVM is to maximize this margin. The hyperplane with maximum margin is called the optimal hyperplane.

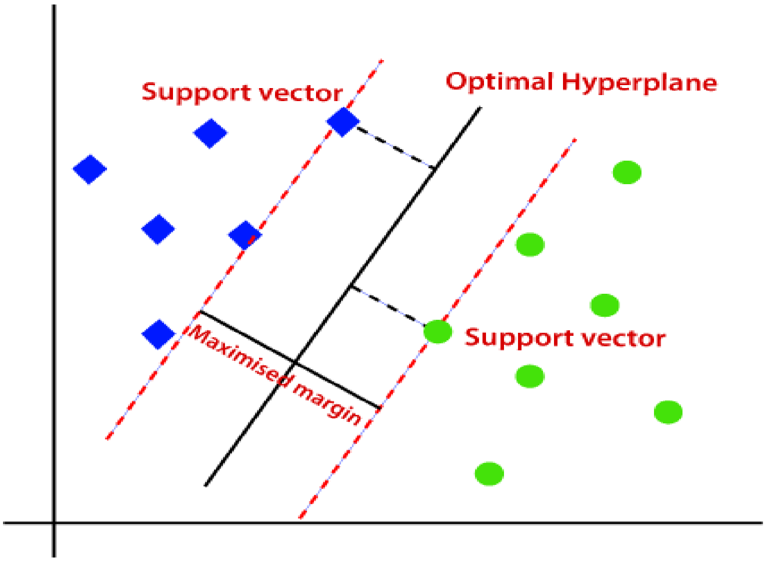

Fig 3: SVM Optimal plane

2.1.1.2 Random Forest Classifier

Random Forest Classifier is combination of n Decision Trees. Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

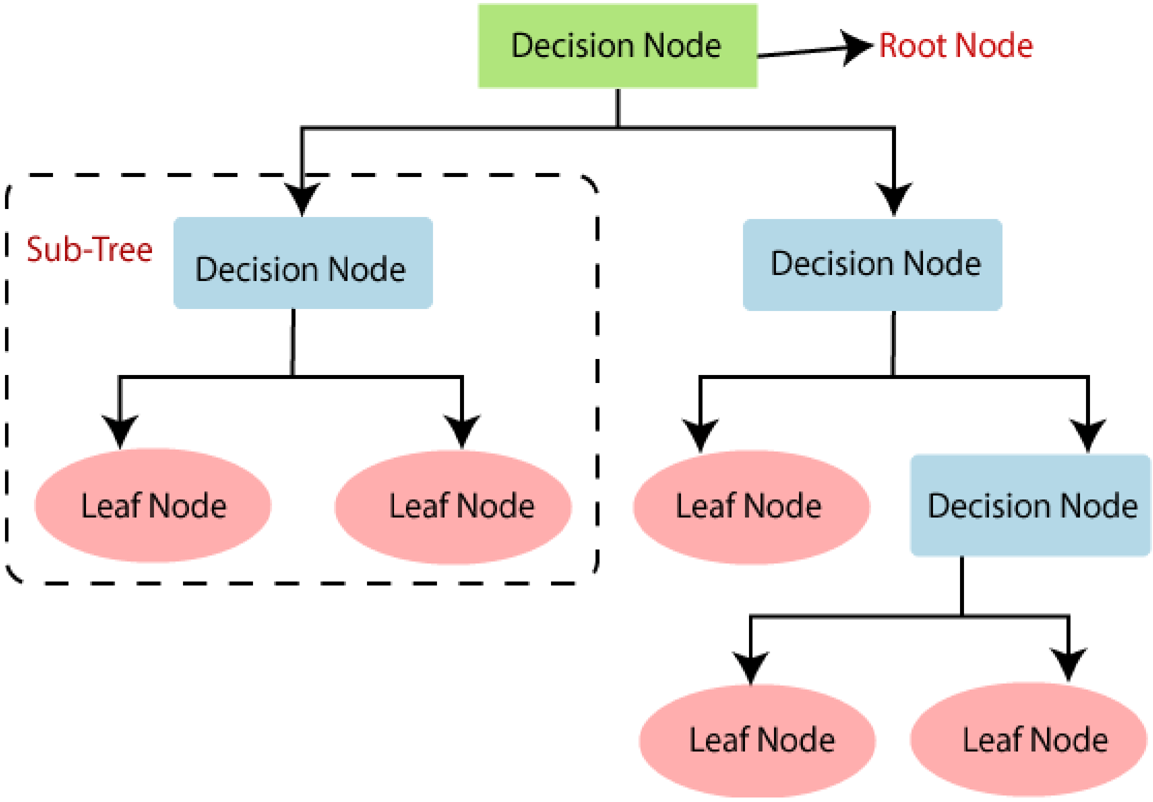

Fig 4: Random Forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting. The below diagram explains the working of the Random Forest algorithm:

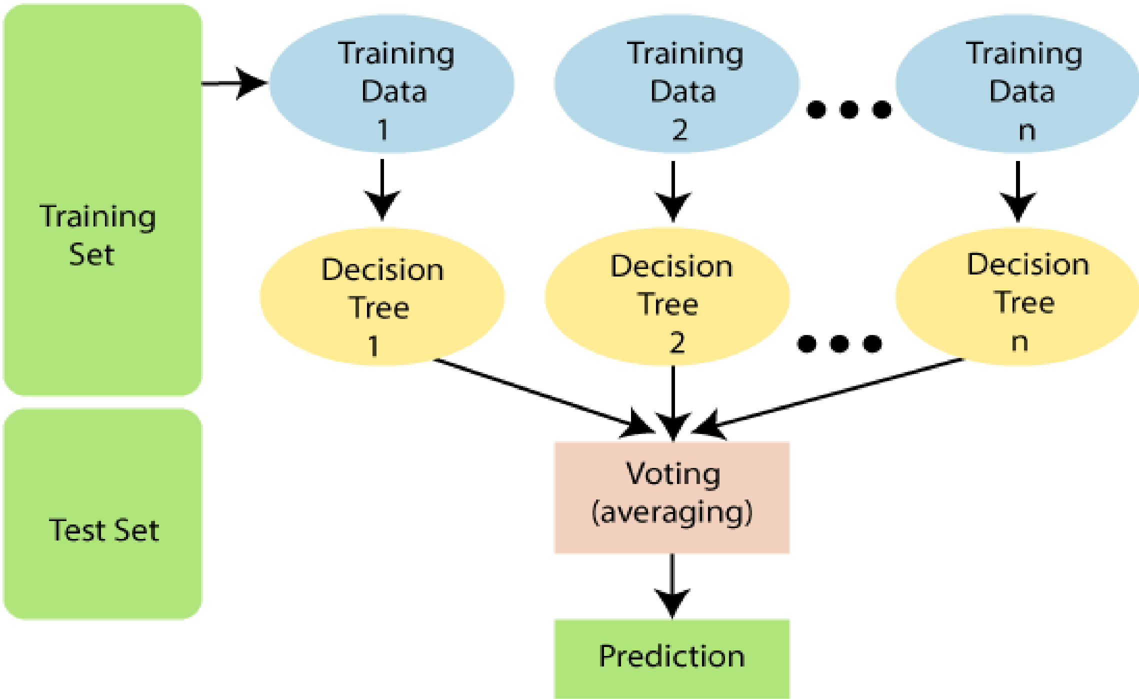

Fig 5: Working of Random Forest

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

2.1.1.3 Logistic Regression

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
- Logistic Regression is much similar to the Linear Regression except that how they are

used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
- Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.
- Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function:

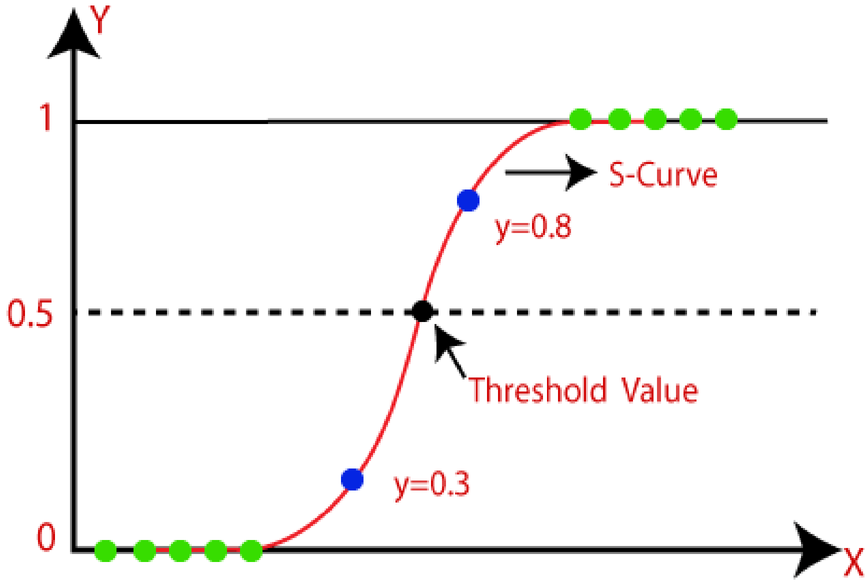

Fig 6: Logistic Function

The Logistic regression equation can be obtained from the Linear Regression equation. The mathematical steps to get Logistic Regression equations are given below:

- We know the equation of the straight line can be written as:

$$y = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

- In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y):

$$\frac{y}{1-y}\text{; 0 for y= 0, and infinity for y=1}$$

- But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:

$$\log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

The above equation is the final equation for Logistic Regression.

2.1.1.4 K-Nearest Neighbors

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
- It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:

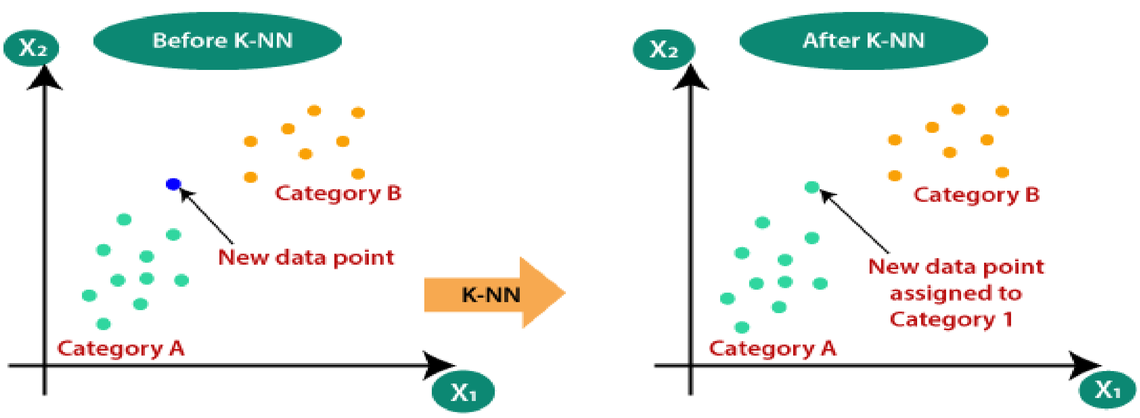

Fig 7: Working of KNN

The K-NN working can be explained on the basis of the below algorithm:

- **Step-1:** Select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of K number of neighbors
- **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
- **Step-4:** Among these k neighbors, count the number of the data points in each category.
- **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
- **Step-6:** Our model is ready.
- Suppose we have a new data point and we need to put it in the required category. Consider the below image:

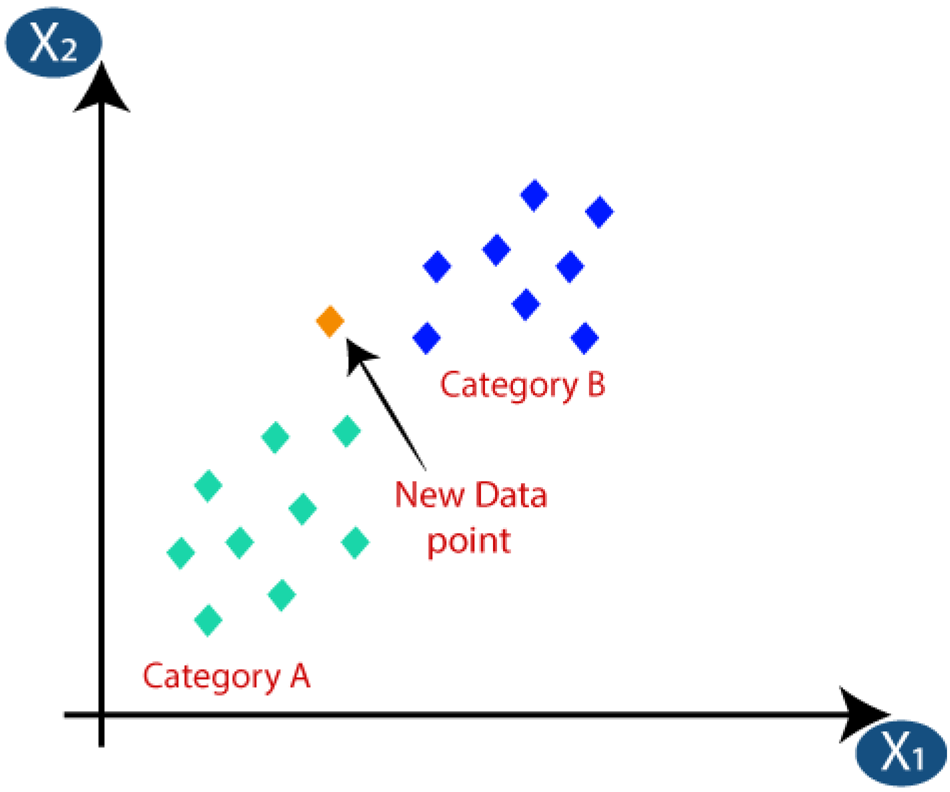

Fig 8: Neighbor Selection

- Firstly, we will choose the number of Neighbors, so we will choose the k=5.
- Next, we will calculate the **Euclidean distance** between the data points.
- The Euclidean distance is the distance between two points, which we have already studied in geometry. It can be calculated as:

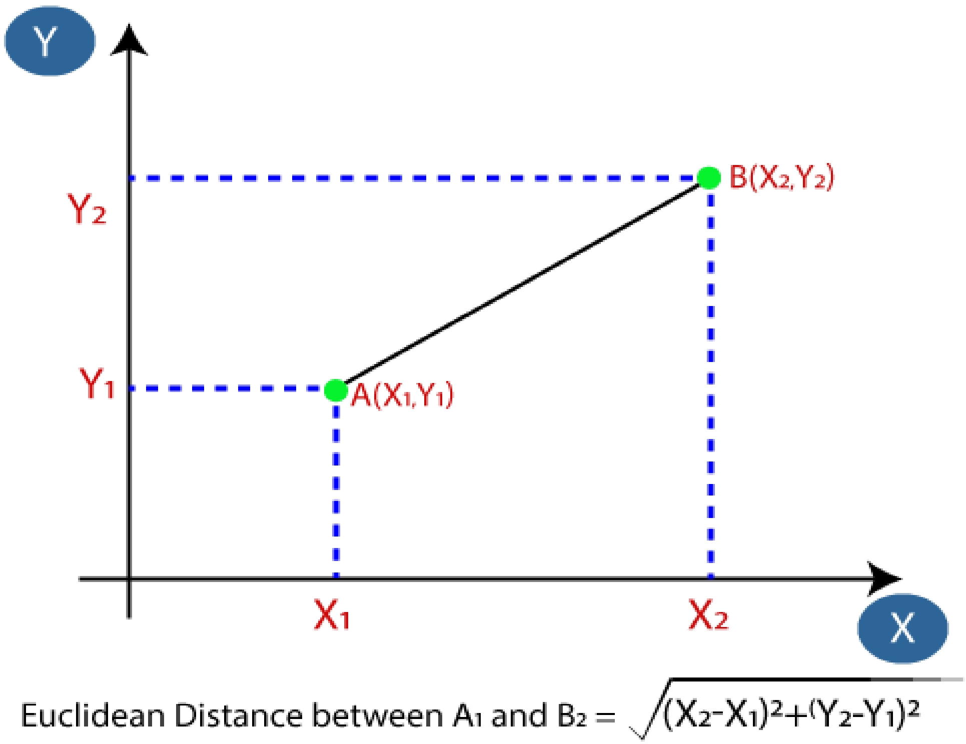

Fig 9: Euclidean Distance

- By calculating the Euclidean distance we got the nearest neighbors, as three nearest neighbors in category A and two nearest neighbors in category B. Consider the below image:

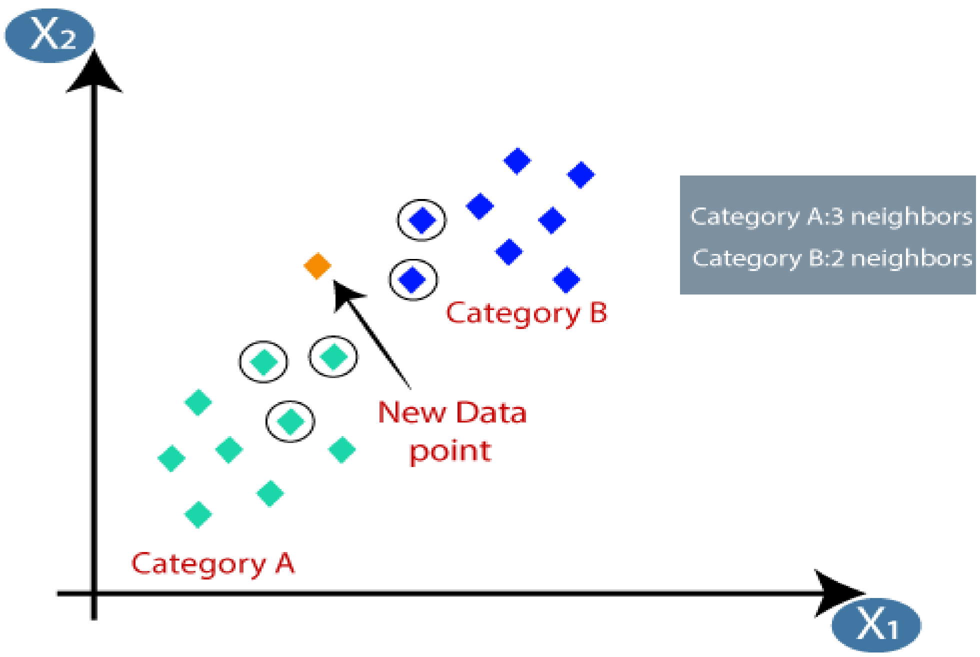

Fig 10: Neighbor Categorizing

- As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

2.1.1.5 Naïve Bayes

- Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
- It is mainly used in text classification that includes a high-dimensional training dataset.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- It is a probabilistic classifier, which means it predicts on the basis of the probability of an object.
- Some popular examples of Naïve Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles.

The Naïve Bayes algorithm is comprised of two words Naïve and Bayes, which can be described as:

- Naive: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple.
- Hence each feature individually contributes to identify that it is an apple without depending on each other.
- Bayes: It is called Bayes because it depends on the principle of Bayes' Theorem
- Bayes' theorem is also known as **Bayes' Rule** or **Bayes' law**, which is used to determine the probability of a hypothesis with prior knowledge. It depends on the conditional probability.
- The formula for Bayes' theorem is given as:

$$P(A|B)= \frac{P(B|A)P(A)}{P(B)}$$

Where,

P(A|B) is Posterior probability: Probability of hypothesis A on the observed event B.

P(B|A) is Likelihood probability: Probability of the evidence given that the probability of a hypothesis is true.

P(A) is Prior Probability: Probability of hypothesis before observing the evidence.

P(B) is Marginal Probability: Probability of Evidence.

Working of Naïve Bayes' Classifier can be understood with the help of the below example:

Suppose we have a dataset of weather conditions and corresponding target variable "Play". So, using this dataset we need to decide that whether we should play or not on a particular day according to the weather conditions. So to solve this problem, we need to follow the below steps:

- Convert the given dataset into frequency tables.
- Generate Likelihood table by finding the probabilities of given features.
- Now, use Bayes theorem to calculate the posterior probability.

2.1.2 Unsupervised learning

Unsupervised learning is used when the problem requires a massive amount of unlabeled data. For example, social media applications, such as Twitter, Instagram and Snapchat, all have large amounts of unlabeled data. Understanding the meaning behind this data requires algorithms that classify the data based on the patterns or clusters it finds. Unsupervised learning conducts an iterative process, analyzing data without human intervention. It is used with email spam-detecting technology. There are far too many variables in legitimate and spam emails for an analyst to tag unsolicited bulk email. Instead, machine-learning classifiers, based on clustering and association, are applied to identify unwanted email.

2.1.3 Reinforcement learning

Reinforcement learning is a behavioral learning model. The algorithm receives feedback from the data analysis, guiding the user to the best outcome. Reinforcement learning differs from other types of supervised learning, because the system isn't trained with the sample data set. Rather, the system learns through trial and error. Therefore, a sequence of successful decisions will result in the process being reinforced, because it best solves the problem at hand.

2.2 Deep Learning

Deep learning is based on the branch of machine learning, which is a subset of artificial intelligence. Since neural networks imitate the human brain and so deep learning will do. In deep learning, nothing is programmed explicitly. Basically, it is a machine learning class that makes use of numerous nonlinear processing units so as to perform feature extraction as well as transformation. The output from each preceding layer is taken as input by each one of the successive layers.

Deep learning models are capable enough to focus on the accurate features themselves by requiring a little guidance from the programmer and are very helpful in solving out the problem of dimensionality. Deep learning algorithms are used, especially when we have a huge no of inputs and outputs.

Since deep learning has been evolved by the machine learning, which itself is a subset of artificial intelligence and as the idea behind the artificial intelligence is to mimic the human behavior, so same is "the idea of deep learning to build such algorithm that can mimic the brain".

2.2.1 Artificial Neural Network

At earlier times, the conventional computers incorporated algorithmic approach that is the computer used to follow a set of instructions to solve a problem unless those specific steps need that the computer need to follow are known the computer cannot solve a problem. So, obviously, a person is needed in order to solve the problems or someone who can provide instructions to the computer so as to how to solve that particular problem. It actually restricted the problem-solving capacity of conventional computers to problems that we already understand and know how to solve.

But what about those problems whose answers are not clear, so that is where our traditional approach face failure and so Neural Networks came into existence. Neural Networks processes information in a similar way the human brain does, and these networks actually learn from examples, you cannot program them to perform a specific task. They will learn only from past experiences as well as examples, which is why you don't need to provide all the information regarding any specific task. So, that was the main reason why neural networks came into existence.

Neural networks are modeled in accordance with the human brain so as to imitate their functionality. The human brain can be defined as a neural network that is made up of several neurons, so is the Artificial Neural Network is made of numerous perceptron.

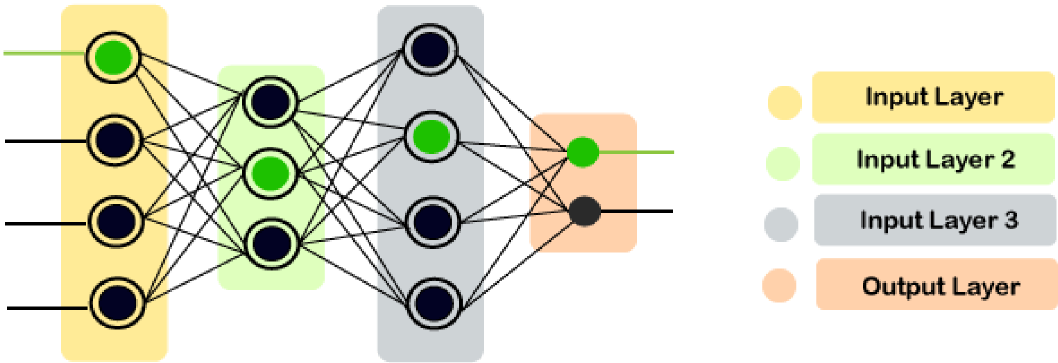

Fig 11: Neural Network

A neural network comprises of three main layers, which are as follows:

- Input layer: The input layer accepts all the inputs that are provided by the programmer.
- Hidden layer: In between the input and output layer, there is a set of hidden layers on which computations are performed that further results in the output.
- Output layer: After the input layer undergoes a series of transformations while passing through the hidden layer, it results in output that is delivered by the output layer.

Basically, the neural network is based on the neurons, which are nothing but the brain cells. A biological neuron receives input from other sources, combines them in some way, followed by performing a nonlinear operation on the result, and the output is the final result.

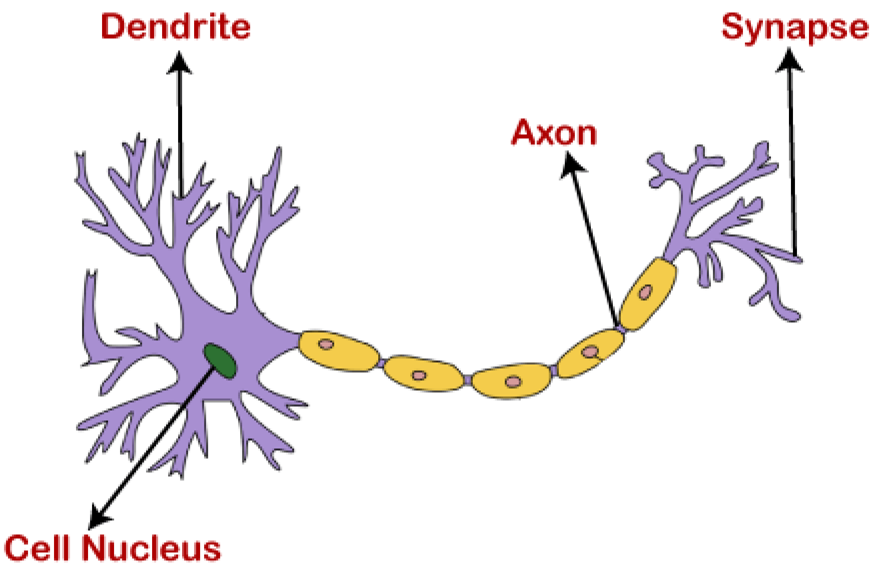

Fig 12: Human Neuron

The dendrites will act as a receiver that receives signals from other neurons, which are then passed on to the cell body. The cell body will perform some operations that can be a summation, multiplication, etc. After the operations are performed on the set of input, then they are transferred to the next neuron via axion, which is the transmitter of the signal for the neuron.

Instead of directly getting into the working of Artificial Neural Networks, lets breakdown and try to understand Neural Network's basic unit, which is called a Perceptron.

So, a perceptron can be defined as a neural network with a single layer that classifies the linear data. It further constitutes four major components, which are as follows;

- **Inputs**
- **Weights and Bias**
- **Summation Functions**
- **Activation or transformation function**

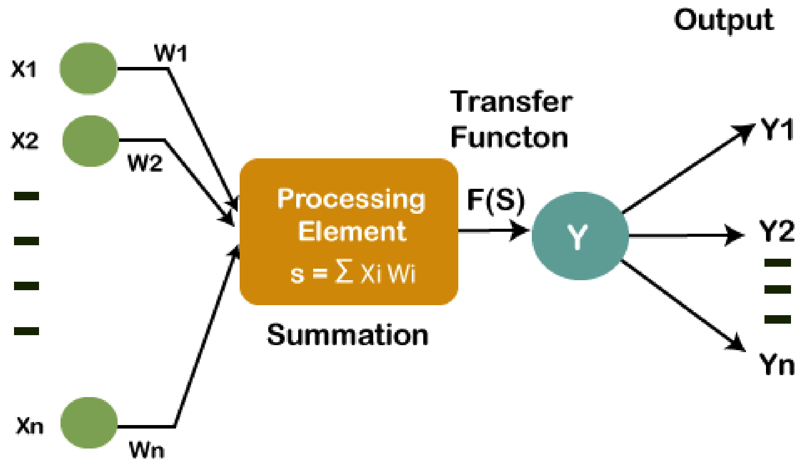

Fig 13: ANN Weight Calculation Function

The main logic behind the concept of Perceptron is as follows:

2.2.1.1 Inputs

The inputs (x) are fed into the input layer, which undergoes multiplication with the allotted weights (w) followed by experiencing addition in order to form weighted sums. Then these inputs weighted sums with their corresponding weights are executed on the pertinent activation function.

2.2.1.2 Weights and Bias

As and when the input variable is fed into the network, a random value is given as a weight of that particular input, such that each individual weight represents the importance of that input in order to make correct predictions of the result. However, bias helps in the adjustment of the curve of activation function so as to accomplish a precise output.

2.2.1.3 Summation Function

After the weights are assigned to the input, it then computes the product of each input and weights. Then the weighted sum is calculated by the summation function in which all of the products are added.

2.2.1.4 Activation Function

The main objective of the activation function is to perform a mapping of a weighted sum upon the output. The transformation function comprises of activation functions such as tanh, ReLU, sigmoid, etc.

The activation function is categorized into two main parts:

- **Sigmoid or Logistic Activation Function**

It provides a smooth gradient by preventing sudden jumps in the output values. It has an output value range between 0 and 1 that helps in the normalization of each neuron's output. For X, if it has a value above 2 or below -2, then the values of y will be much steeper. In simple language, it means that even a small change in the X can bring a lot of change in Y.

It's value ranges between 0 and 1 due to which it is highly preferred by binary classification whose result is either 0 or 1.

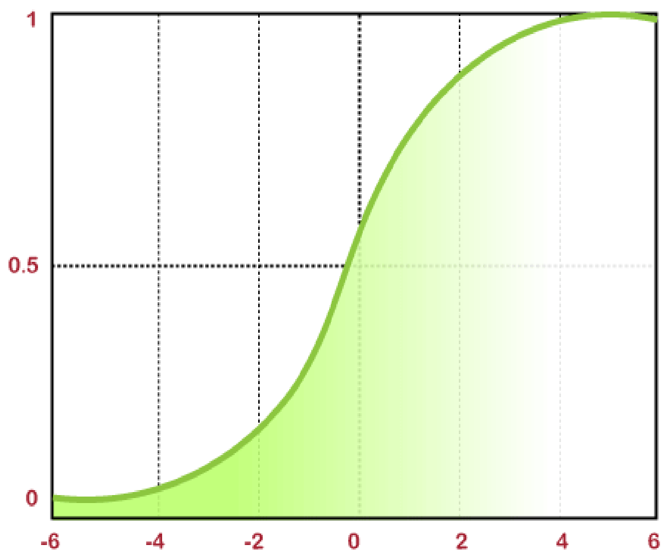

Fig 14: Sigmoid Function

- **ReLU(Rectified Linear Unit) Activation Function**

ReLU is one of the most widely used activation function by the hidden layer in the neural network. Its value ranges from 0 to infinity. It clearly helps in solving out the problem of backpropagation. It tends out to be more expensive than the sigmoid, as well as the tanh activation function. It allows only a few neurons to get activated at a particular instance that leads to effectual as well as easier computations.

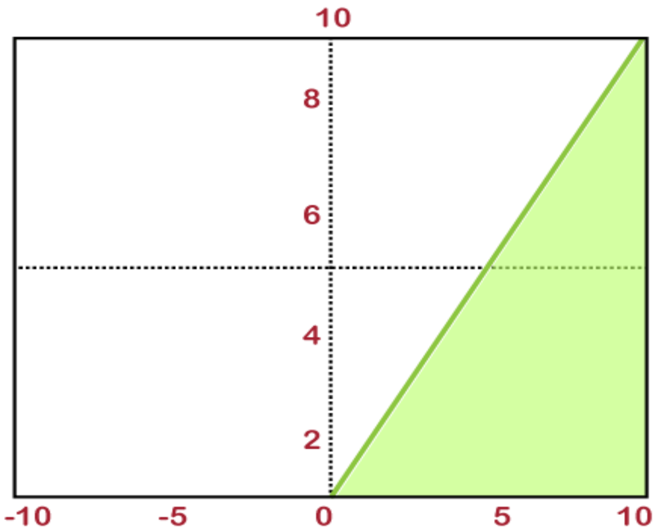

Fig 15: reLu Activation Function

- **SoftMax Function**

It is one of a kind of sigmoid function whereby solving the problems of classifications. It is mainly used to handle multiple classes for which it squeezes the output of each class between 0 and 1, followed by dividing it by the sum of outputs. This kind of function is specially used by the classifier in the output layer.

2.2.1.5 Gradient Descent Algorithm

Gradient descent is an optimization algorithm that is utilized to minimize the cost function used in various machine learning algorithms so as to update the parameters of the learning model. In linear regression, these parameters are coefficients, whereas, in the neural network, they are weights.

Procedure:

It all starts with the coefficient's initial value or function's coefficient that may be either 0.0 or any small arbitrary value. For estimating the cost of the coefficients, they are plugged into the function that helps in evaluating.

Next, the derivate will be calculated, which is one of the concepts of calculus that relates to the function's slope at any given instance. In order to know the direction in which the values of the coefficient will move, we need to calculate the slope so as to accomplish a low cost in the next iteration.

Now that we have found the downhill direction, it will further help in updating the values of coefficients. Next, we will need to specify alpha, which is a learning rate parameter, as it handles the amount of amendments made by coefficients on each update.

Until the cost of the coefficient reaches 0.0 or somewhat close enough to it, the whole process will reiterate again and again.

It can be concluded that gradient descent is a very simple as well as straightforward concept. It just requires you to know about the gradient of the cost function or simply the function that you are willing to optimize.

- **Batch Gradient Descent**

For every repetition of gradient descent, the main aim of batch gradient descent is to processes all of the training examples. In case we have a large number of training examples, then batch gradient descent tends out to be one of the most expensive and less preferable too.

- **Stochastic Gradient Descent**

At a single repetition, the stochastic gradient descent processes only one training example, which means it necessitates for all the parameters to update after the one single training example is processed per single iteration. It tends to be much faster than that of the batch gradient descent, but when we have a huge number of training examples, then also it processes a single example due to which system may undergo a large no of repetitions. To evenly train the parameters provided by each type of data, properly shuffle the dataset.

2.2.2 Backpropagation

The backpropagation consists of an input layer of neurons, an output layer, and at least one hidden layer. The neurons perform a weighted sum upon the input layer, which is then used by the activation function as an input, especially by the sigmoid activation function. It also makes use of supervised learning to teach the network. It constantly updates the weights of the network until the desired output is met by the network. It includes the following factors that are responsible for the training and performance of the network:

- Random (initial) values of weights.

- A number of training cycles.
- A number of hidden neurons.
- The training set.
- Teaching parameter values such as learning rate and momentum

Working of Backpropagation

Consider the diagram given below.

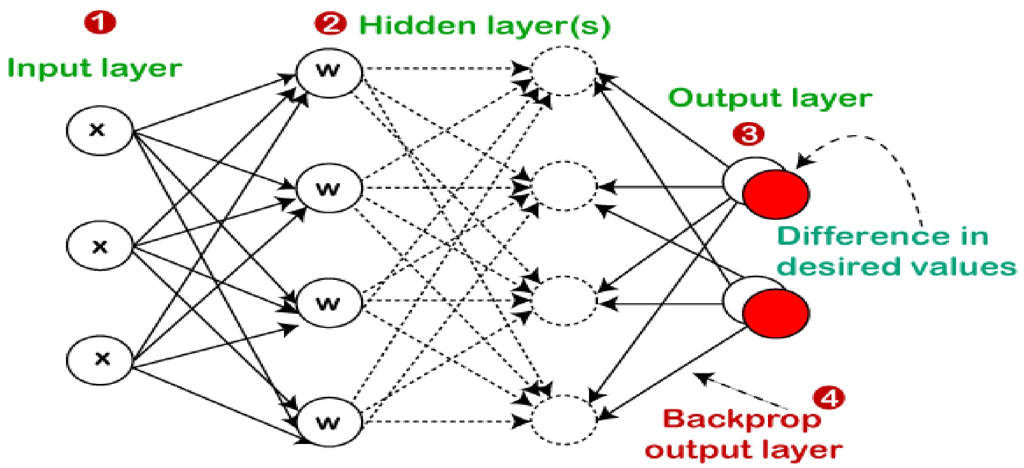

Fig 16: Backpropagation Neural network

- The preconnected paths transfer the inputs X.
- Then the weights W are randomly selected, which are used to model the input.
- After then, the output is calculated for every individual neuron that passes from the input layer to the hidden layer and then to the output layer.
- Lastly, the errors are evaluated in the outputs. Error B= Actual Output - Desired Output
- The errors are sent back to the hidden layer from the output layer for adjusting the weights to lessen the error.
- Until the desired result is achieved, keep iterating all of the processes. Need of Backpropagation
- Since it is fast as well as simple, it is very easy to implement.
- Apart from no of inputs, it does not encompass of any other parameter to perform tuning.
- As it does not necessitate any kind of prior knowledge, so it tends out to be more flexible.

2.2.3 Convolutional Neural Network

Convolutional Neural Networks are a special kind of neural network mainly used for image classification, clustering of images and object recognition. DNNs enable unsupervised construction of hierarchical image representations. To achieve the best accuracy, deep convolutional neural networks are preferred more than any other neural network.

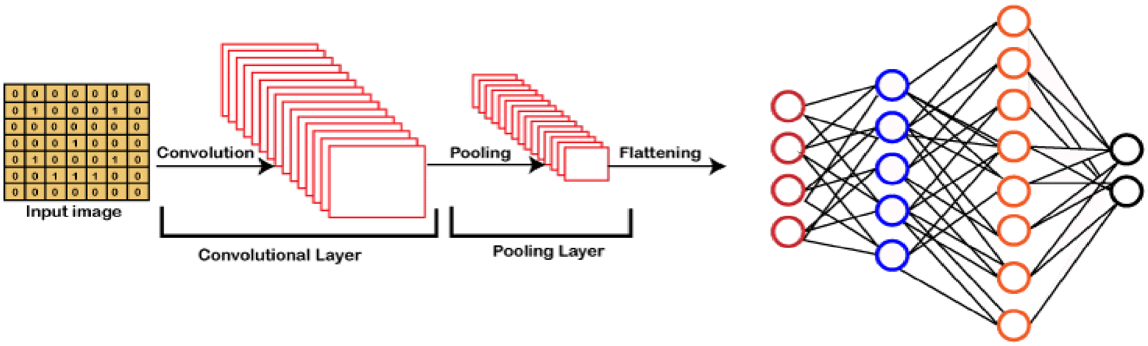

Fig 17: Convolutional Neural Network

2.2.4 Recurrent Neural Network

Recurrent neural networks are yet another variation of feed-forward networks. Here each of the neurons present in the hidden layers receives an input with a specific delay in time. The Recurrent neural network mainly accesses the preceding info of existing iterations. For example, to guess the succeeding word in any sentence, one must have knowledge about the words that were previously used. It not only processes the inputs but also shares the length as well as weights crossways time. It does not let the size of the model to increase with the increase in the input size. However, the only problem with this recurrent neural network is that it has slow computational speed as well as it does not contemplate any future input for the current state.

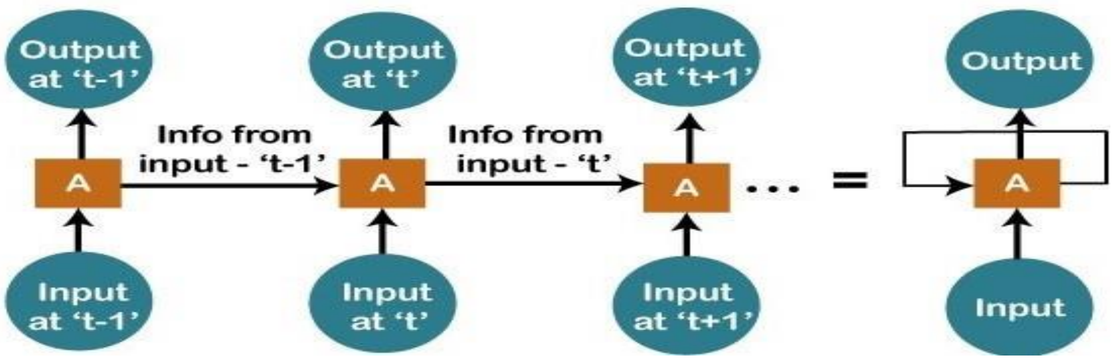

Fig 18: Recurrent Neural Network

CHAPTER 3

TECHNOLOGY

3.1 Python

Python is an OOPs (Object Oriented Programming) based, high level, interpreted programming language. It is a robust, highly useful language focused on rapid application development (RAD). Python helps in easy writing and execution of codes. Python can implement the same logic with as much as 1/5th code as compared to other OOPs languages. Python provides a huge list of benefits to all. The usage of Python is such that it cannot be limited to only one activity. Its growing popularity has allowed it to enter into some of the most popular and complex processes like Artificial Intelligence (AI), Machine Learning (ML), natural language processing, data science etc. Python has a lot of libraries for every need of this project. For our AV we had used Tkinter Library and PyGame Module Documentation.

Python is reasonably efficient. Efficiency is usually not a problem for small examples. If your Python code is not efficient enough, a general procedure to improve it is to find out what is taking most the time and implement just that part more efficiently in some lower-level language. This will result in much less programming and more efficient code (because you will have more time to optimize) than writing everything in a low-level language.

3.2 Anaconda Python Distribution

Anaconda is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free.

Package versions in Anaconda are managed by the package management system conda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for things other than Python.

Python is the most popular programming language or nothing wrong to say that it is the next-generation programming language. In every emerging field in computer science, Python makes its presence actively. Python has vast libraries for various fields such as **Machine Learning (Numpy, Pandas, Matplotlib), Artificial intelligence (Pytorch, TensorFlow).**

3.2.1 Anaconda Installation

- Download the Anaconda installer.
- RECOMMENDED: Verify data integrity with SHA-256. For more information on hashes, see What about cryptographic hash verification?
- Double-click the installer to launch.
- Click Next.
- Read the licensing terms and click I Agree.
- Select an install for Just Me unless you're installing for all users (which requires Windows Administrator privileges) and click Next.
- Select a destination folder to install Anaconda and click the Next button. See FAQ.
- Choose whether to add Anaconda to your PATH environment variable. We recommend not adding Anaconda to the PATH environment variable, since this can interfere with other software. Instead, use Anaconda software by opening Anaconda Navigator or the Anaconda Prompt from the Start Menu.
- Choose whether to register Anaconda as your default Python. Unless you plan on installing and running multiple versions of Anaconda or multiple versions of Python, accept the default and leave this box checked.
- Click Install. If you want to watch the packages Anaconda is installing, click Show Details.
- Click Next.
- After a successful installation you will see the "Thanks for installing Anaconda" dialog box.
- Click the Finish button.

3.3 Python Libraries

While The Python Language Reference describes the exact syntax and semantics of the Python language, this library reference manual describes the standard library that is distributed with Python. It also describes some of the optional components that are commonly included in Python distributions.

Python's standard library is very extensive, offering a wide range of facilities as indicated by the long table of contents listed below. The library contains built-in modules (written in C) that provide access to system functionality such as file I/O that would otherwise be inaccessible to Python programmers, as well as modules written in Python that provide standardized solutions for many problems that occur in everyday programming. Some of these modules are explicitly designed to encourage and enhance the portability of Python programs by abstracting away platform-specifics into platform-neutral APIs

There are so many Libraries in Python but here I will be mentioned only those which are used in my project:

3.3.1 Numpy

NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. It is open-source software. It contains various features including these important ones.

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined using Numpy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

Some functions of Numpy are:

- Trigonometric function
- Hyperbolic function
- Rounding
- Sum
- Products
- Differences

- Exponents
- Logarithms
- Floating point
- Rational routines
- Arithmetic's
- Handling Complex

3.3.2 Pandas

Pandas is an open-source library in Python. It provides ready to use high-performance data structures and data analysis tools. Pandas' module runs on top of NumPy and it is popularly used for data science and data analytics. NumPy is a low-level data structure that supports multi-dimensional arrays and a wide range of mathematical array operations. Pandas has a higher-level interface. It also provides streamlined alignment of tabular data and powerful time series functionality.

Data Frame is the key data structure in Pandas. It allows us to store and manipulate tabular data as a 2-D data structure. Pandas provides a rich feature-set on the Data Frame. For example, data alignment, data statistics, slicing, grouping, merging, concatenating data, etc.

Some functions of pandas are:

- read_csv ()
- head ()
- describe ()
- memory_usage ()
- astype ()
- loc [:]
- to_datetime ()
- value_counts ()
- drop_duplicates ()

- groupby ()
- merge ()
- sort_values ()
- fillna ()

3.3.3 Operating System (os)

It is possible to automatically perform many operating system tasks. The OS module in Python provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory, etc.

You first need to import the os module to interact with the underlying operating system. So, import it using the import os statement before using its functions.

Some functions of os:

- os.name ()
- os.mkdir ()
- os.error ()
- os.close ()
- os.access ()

3.3.4 Open CV

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in Numpy can be combined with OpenCV.

Some functions of Open CV

- Reading, Writing and Displaying Images

- Changing Color Spaces
- Resizing Images
- Image Rotation
- Image Translation
- Image Thresholding
- Image Segmentation
- Edge Detection

3.3.5 Matplotlib

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack.

Some functions of matplotlib:

- Line plots
- Histogram
- Bar Plot
- Scatter Plot
- Pie Chart
- Box Plot

3.3.6 Scikit-learn

Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction.

Please note that sklearn is used to build machine learning models. It should not be used for reading the data, manipulating and summarizing it.

Some functions of Scikit-learn:

- Features Datasets
- Data Splitting
- Regression
- Classification
- Boosting
- Confusion Matrix
- Classification Report

3.3.7 TensorFlow

TensorFlow is an open-source library for fast numerical computing. It was created and is maintained by Google and released under the Apache 2.0 open-source license. The API is nominally for the Python programming language, although there is access to the underlying C++ API. Unlike other numerical libraries intended for use in Deep Learning like Theano, TensorFlow was designed for use both in research and development and in production systems. It can run on single CPU systems, GPUs as well as mobile devices and large-scale distributed systems of hundreds of machines. Some functions of TensorFlow

TensorFlow is mainly used for deep learning machine learning Classification, Perception, Understanding, Discovering, Prediction and Creation of Neural Network model with different layer of Neurons.

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 Hardware and Software Requirements

The software is designed to be light-weighted so that it doesn't be a burden on the machine running it. This system is being build keeping in mind the generally available hardware and software compatibility. Here are the minimum hardware and software requirement for Algorithm Visualizer.

Hardware:

- Intel core i5 7^{th} Gen or above.
- RAM 8GB or more.
- GPU 2GB

Software:

- Windows 7(64-bit) or above.
- Python 3.8.5
- Jupyter Notebook
- Karas
- OpenCV
- Matplot lib
- TensorFlow Library

CHAPTER 5

IMPLEMENTATION

5.1 Work flow

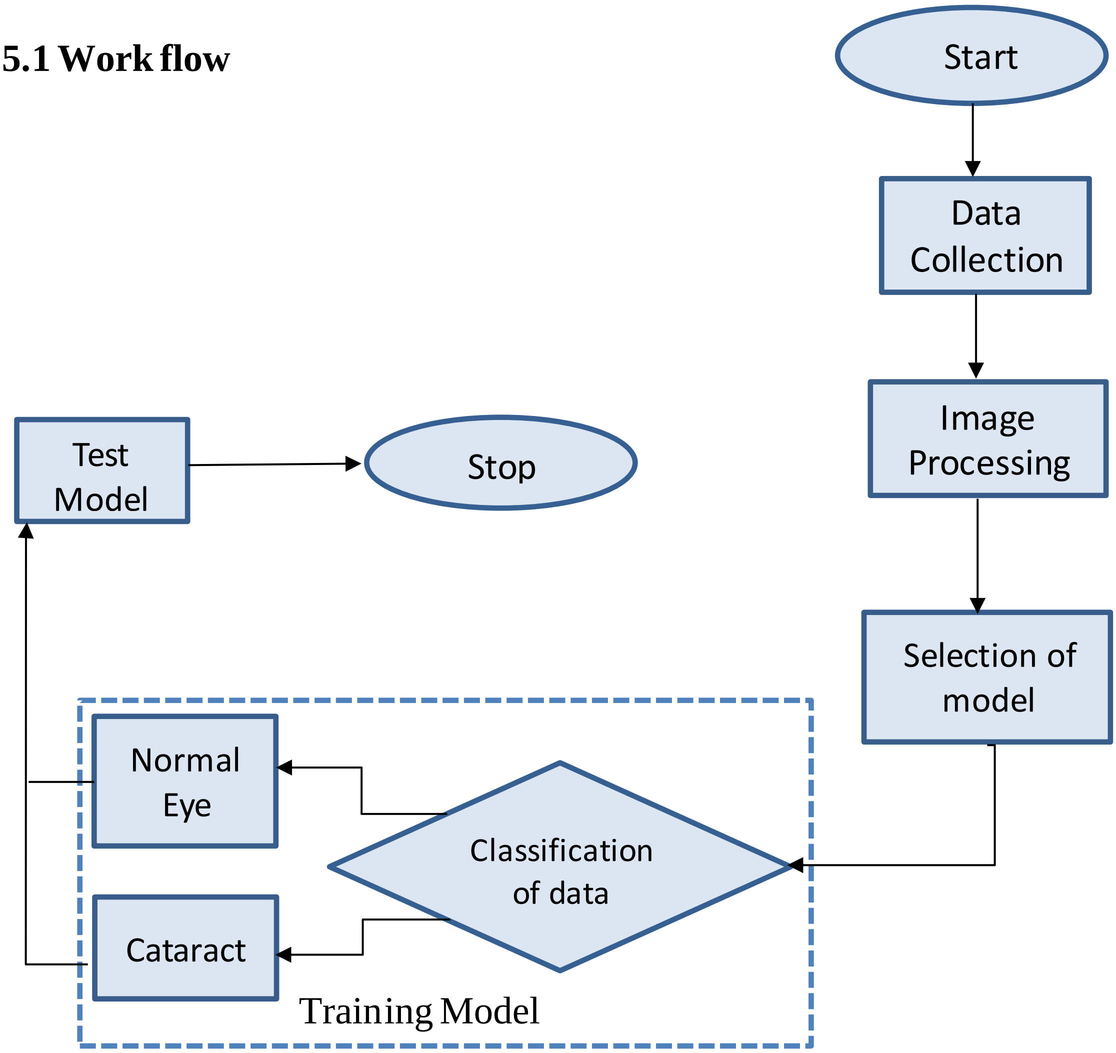

Fig 19: Flow Chart

The flow chart above in Figure describes the logic of the Model. It begins with the collection of data from open source. The image thus collected is then pre-processed (Cropping, Image Scaling etc.) after which the processed image is used in the model selected. After this the data is classified into either normal eye or cataract eye by the selected model and then we test the model for its accuracy. If the accuracy is less then we repeat this for another model.

5.2 Dataset creation

We collected images from open source

Our Dataset include images:

Two sets of images one set for Normal Eyes and another one is for Cataract Eye

- Each set contains 300 images
- Size of each image is 1600x1600
- Dot per inch (DPI) = 96 dpi
- Total images 300*2 = 600

5.3 Image Pre-processing

The aim of pre-processing is to improve the quality of the image so that we can analyses it in a better way. By preprocessing we can suppress undesired distortions and enhance some features which are necessary for the particular application we are working for. Those features might vary for different applications**.**

5.3.1 RGB to Grey Scale

- **RGB**

The most well-known color model is RGB which stands for Red-Green-Blue. As the name suggests, this model represents colors using individual values for Red, Green, and Blue. The RGB model is used in almost all digital screens throughout the world. Specifically, a color is defined using three integer values from 0 to 255 for red, green, and blue, where a zero value means dark and a value of 255 means bright. Given the values, the final color is defined when we mix these three basic colors weighted by their values.

- **Grey Scale**

Grayscale is the simplest model since it defines colors using only one component that is lightness. The amount of lightness is described using a value ranging from 0 (black) to 255 (white). On the one hand, grayscale images convey less information than RGB. However, they are common in image processing because using a grayscale image requires less available space and is faster, especially when we deal with complex computations.

- **Why is gray scale important?**

Grayscale is an important aspect of images, and it is the only portion that is not removed; otherwise, a pure black image would result no matter what color information there is. A digital image is composed of groups of three pixels with colors of red, green and blue (RGB), also called channels in digital imaging.

- **Function Used to convert RGB to Gray scale image**

We used OpenCV function "cvtColor ()" to convert image into greyscale

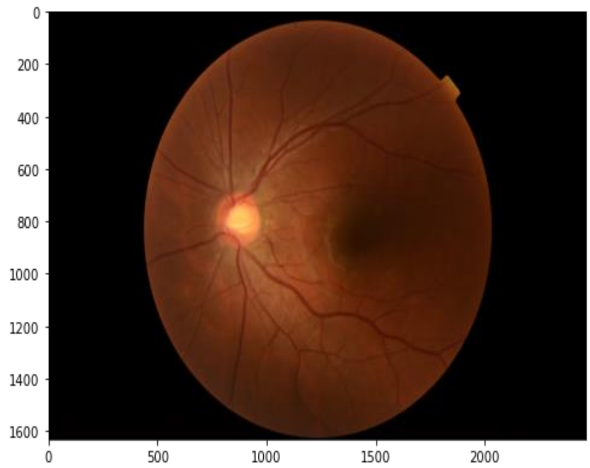

Fig 20: RGB Image

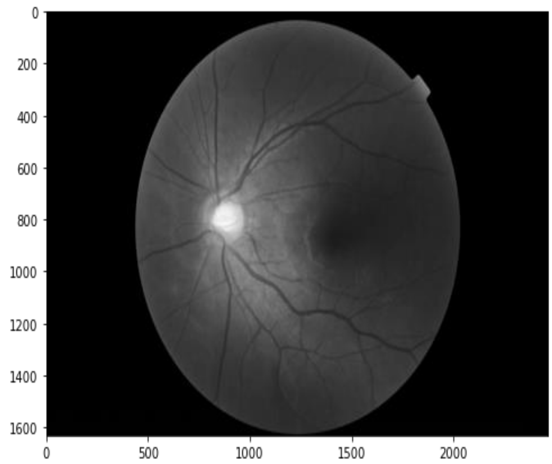

Fig 21: Grey Scale Image

5.3.2 Gray Scale Image Normalization

Image normalization is a typical process in image processing that changes the range of pixel intensity values. Its normal purpose is to convert an input image into a range of pixel values that are more familiar or normal to the senses, hence the term normalization.

In this work, we will perform a function that produces a normalization of an input image (Gray Scale). Then, we understand a representation of the range of values of the scale of the image represented between 0 and 255, in this way we get, for example, that very dark images become clearer. The linear normalization of a digital image is performed according to the formula:

$$\text{Output channel} = 255 * (\text{Input channel} - \text{min}) / (\text{max} - \text{min})$$

If we are using a grayscale image, we only need to normalize using one channel. Otherwise, we have to use 3 channels for RGB Image.

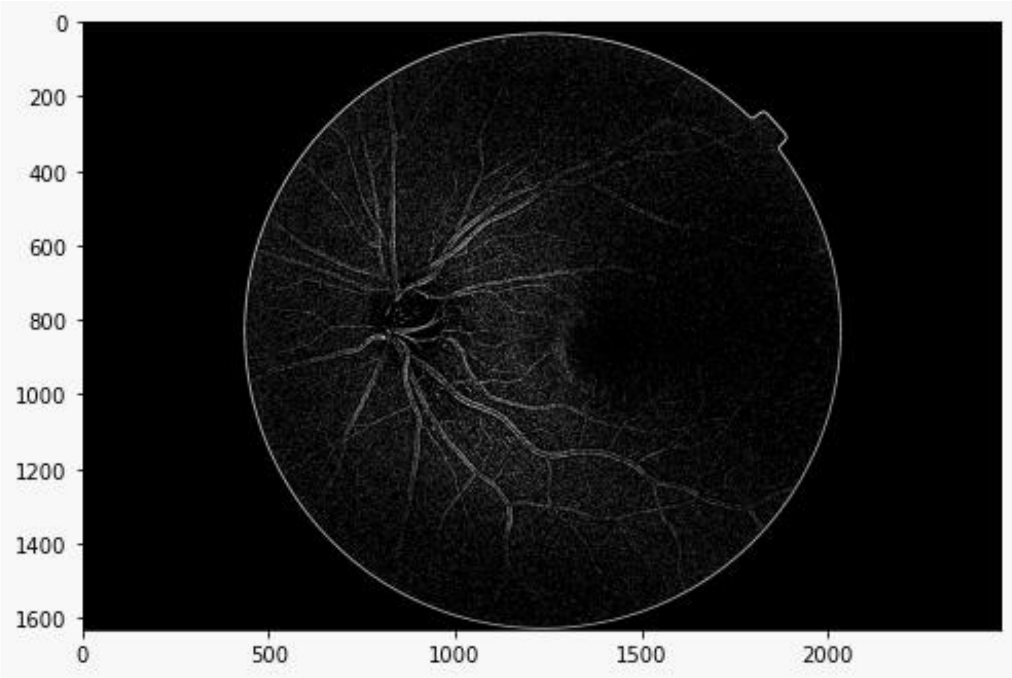

Fig 22: Normalized Image

5.3.3 Image Cropping

Cropping is done to remove all unwanted objects or areas from an image. Or even to highlight a particular feature of an image. There is no specific function for cropping using OpenCV, NumPy array slicing is what does the job. Every image that is read in, gets stored in a 2D array (for each color channel).

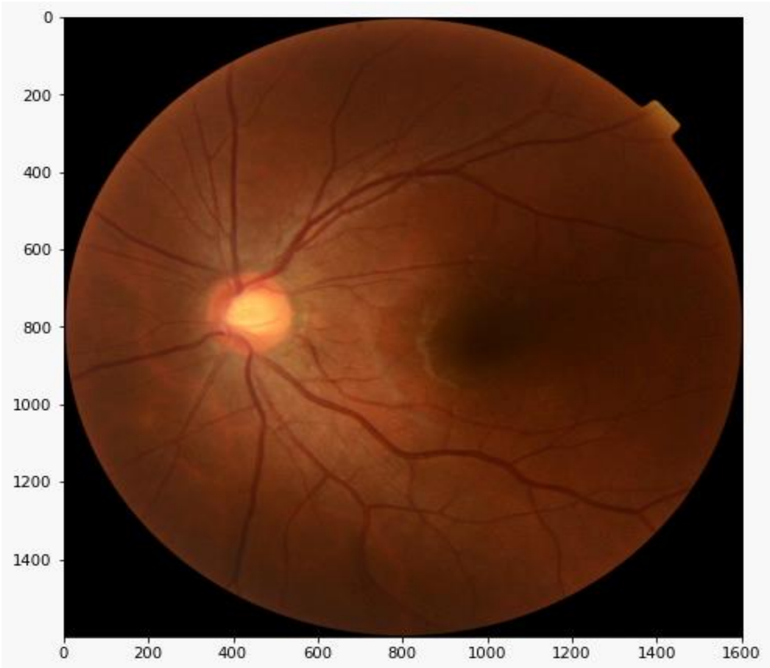

Fig 23: Cropped Image

5.4 Result of Different Machine Learning Models

5.4.1 Support Vector Machine

Classification Report after applying SVM on Cataract Image dataset:

	precision	recall	f1-score	support
0	0.94	0.64	0.76	70
1	0.65	0.94	0.77	50
accuracy			0.77	120
macro avg	0.80	0.79	0.77	120
weighted avg	0.82	0.77	0.77	120

Fig 24: Support Vector machine Classification Report

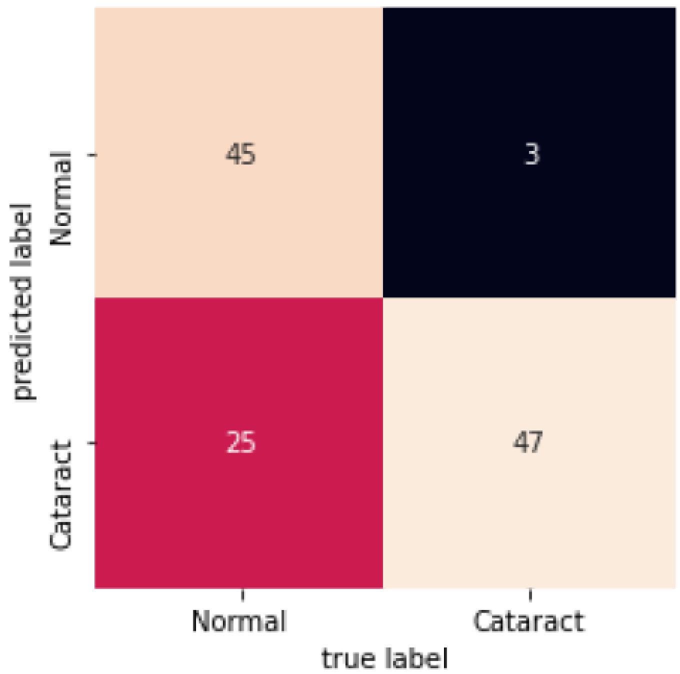

Fig 25: Support Vector machine Heatmap

5.4.2 Random Forest Classifier

Classification Report after applying Random Forest Classifier on Cataract Image dataset:

	precision	recall	f1-score	support
0	1.00	0.84	0.91	70
1	0.82	1.00	0.90	50
accuracy			0.91	120
macro avg	0.91	0.92	0.91	120
weighted avg	0.92	0.91	0.91	120

Fig 26: Random Forest Classifier Classification Report

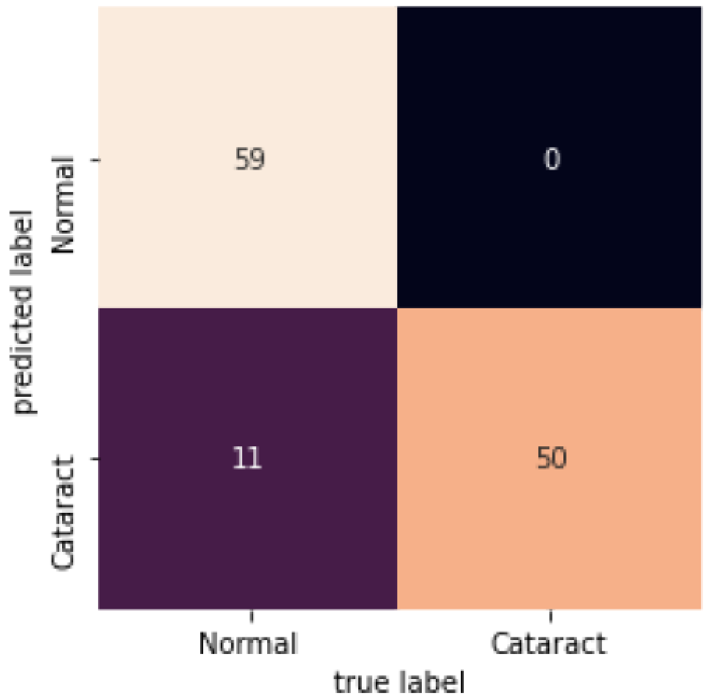

Fig 27: Random Forest Classifier Heatmap

5.4.3 Logistic Regression

Classification Report after applying Logistic Regression on Cataract Image dataset:

	precision	recall	f1-score	support
0	0.91	0.83	0.87	70
1	0.79	0.88	0.83	50
accuracy			0.85	120
macro avg	0.85	0.85	0.85	120
weighted avg	0.86	0.85	0.85	120

Fig 28: Logistic Regression Classification Report

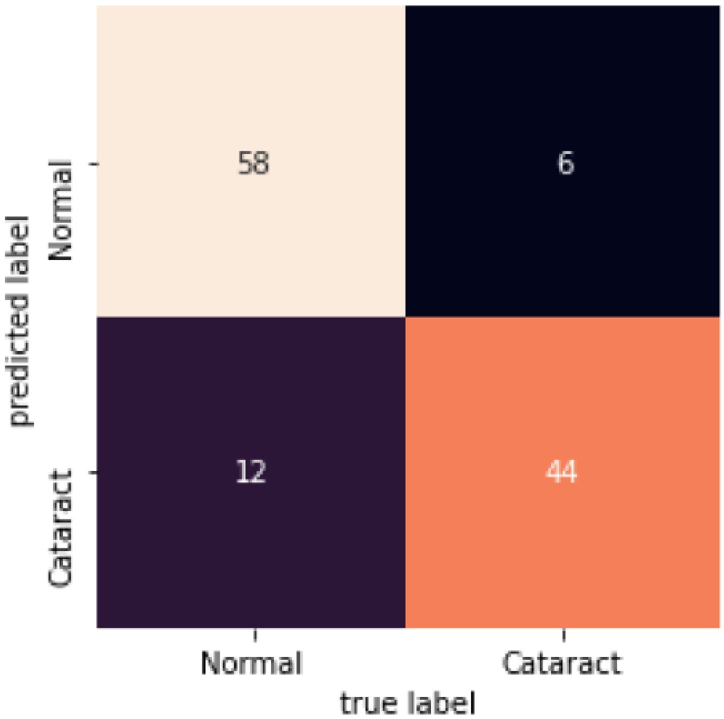

Fig 29: Logistic Regression Heatmap

5.4.4 K-Nearest Neighbors

Classification Report after applying K-Nearest Neighbors on Cataract Image dataset:

	precision	recall	f1-score	support
0	0.83	0.49	0.61	70
1	0.54	0.86	0.67	50
accuracy			0.64	120
macro avg	0.69	0.67	0.64	120
weighted avg	0.71	0.64	0.64	120

Fig 30: K-Nearest Neighbor Classification Report

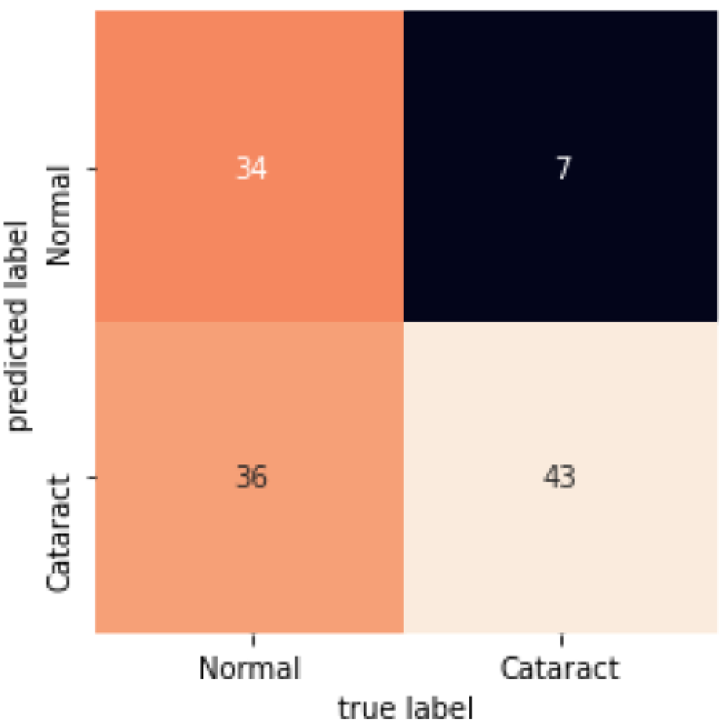

Fig 31: K-Nearest Neighbor Heatmap

5.4.5 Naive Bayes

Classification Report after applying Naïve Bayes on Cataract Image dataset:

	precision	recall	f1-score	support
0	0.91	0.83	0.87	70
1	0.79	0.88	0.83	50
accuracy			0.85	120
macro avg	0.85	0.85	0.85	120
weighted avg	0.86	0.85	0.85	120

Fig 32: Naïve Bayes Classification Report

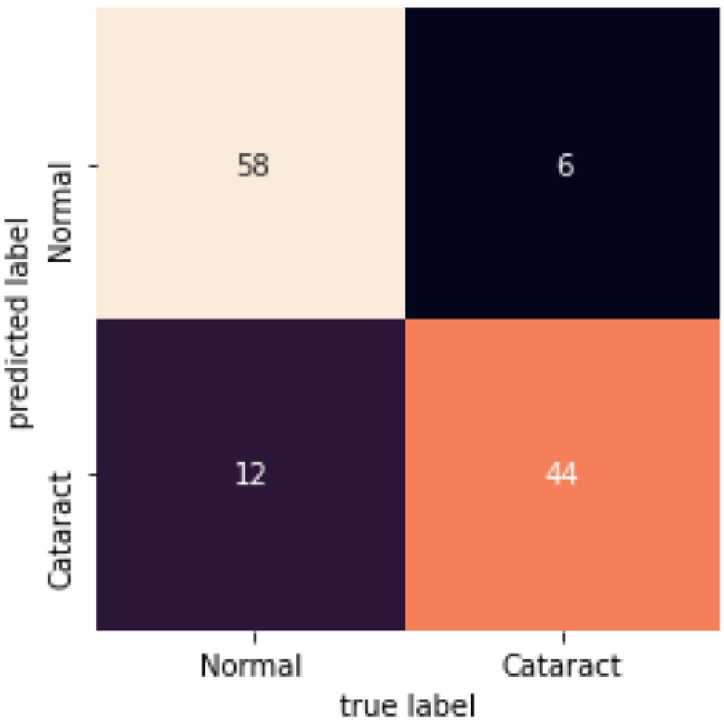

Fig 33: Naïve Bayes Heatmap

5.5 Result of Deep Learning Models

5.5.1 ANN with Input and Output Layer

Artificial Neural Network consist of Neurons Multiple of 100 in Input Layer and output layer consist of 1 Neuron, Activation function used is reLu and Sigmoid with 10 epochs, Number of Neurons were multiple of 100, They starts from 100 to 1000 with 9 possible cases, result of 9 Models with similar conditions are:

Accuracy report of models

Model	1st_Layer	Accuracy
1	100	67.5000011920929
2	200	43.33333373069763
3	300	85.00000238418579
4	400	60.83333492279053
5	500	66.66666865348816
6	600	64.99999761581421
7	700	82.4999988079071
8	800	58.33333134651184
9	900	59.16666388511658

Fig 34: Accuracy of 9 Models with 1 Input Layer

Classifiation Report of 3 Model Because of its maximum Accuracy

	precision	recall	f1-score	support
0	0.83	0.93	0.88	70
1	0.88	0.74	0.80	50
accuracy			0.85	120
macro avg	0.86	0.83	0.84	120
weighted avg	0.85	0.85	0.85	120

Fig 35: Classification report of 3rd Model

5.5.2 ANN with Input Layer, 1 Hidden Layer and Output Layer

Artificial Neural Network consist of Neurons Multiple of 100 in Input Layer and 2 times Neurons in Hidden Layer and output layer consist of 1 Neuron, Activation function used is reLu and Sigmoid with 10 epochs, Number of Neurons were multiple of 100, They starts from 100 to 1000 with 9 possible cases, result of 9 Models with similar conditions are:

Accuracy report of models

Model	1st_Layer	Hidden	Accuracy
1	100	200	68.33333373069763
2	200	400	50.0
3	300	600	72.50000238418579
4	400	800	41.66666567325592
5	500	1000	55.83333373069763
6	600	1200	57.499998807907104
7	700	1400	70.83333134651184
8	800	1600	42.500001192092896
9	900	1800	44.1666626930237

Fig 36: Accuracy of 9 Models with 1 Input Layer and 1 Hidden Layer

Classifiation Report of 3 Model Because of its maximum Accuracy

	precision	recall	f1-score	support
0	0.97	0.54	0.70	70
1	0.60	0.98	0.75	50
accuracy			0.73	120
macro avg	0.79	0.76	0.72	120
weighted avg	0.82	0.72	0.72	120

Fig 37: Classification report of 3rd Model

5.5.3 ANN with Input Layer, 2 Hidden Layer and Output Layer

Artificial Neural Network consist of Neurons Multiple of 100 in Input Layer and 2 times Neurons in 1st Hidden Layer and 3 times Neurons in 2nd Hidden Layer and output layer consist of 1 Neuron, Activation function used is reLu and Sigmoid with 10 epochs, Number of Neurons were multiple of 100, They starts from 100 to 1000 with 9 possible cases, result of 9 Models with similar conditions are:

Accuracy report of models

Model	1st_Layer	Hidden	2nd Hidden	Accuracy
1	100	200	300	41.66666567325592
2	200	400	600	41.66666567325592
3	300	600	900	41.66666567325592
4	400	800	1200	41.66666567325592
5	500	1000	1500	41.66666567325592
6	600	1200	1800	44.16666626930237
7	700	1400	2100	41.66666567325592
8	800	1600	2400	42.500001192092896
9	900	1800	2700	43.33333373069763

Fig 38: Accuracy of 9 Models with 1 Input Layer 2 Hidden Layer

Classifiation Report of 6 Model Because of its maximum Accuracy

	precision	recall	f1-score	support
0	1.00	0.04	0.08	70
1	0.43	1.00	0.60	50
accuracy			0.44	120
macro avg	0.71	0.52	0.34	120
weighted avg	0.76	0.44	0.30	120

Fig 39: Classification report of 6th Model

5.5.4 ANN with Input Layer, 1 Hidden & Output Layer (Half Neurons)

Artificial Neural Network consist of Neurons Multiple of 100 in Input Layer and 1/2 times Neurons in 1st Hidden Layer and output layer consist of 1 Neuron, Activation function used is reLu and Sigmoid with 10 epochs, Number of Neurons were multiple of 100, They starts from 100 to 1000 with 9 possible cases, result of 9 Models with similar conditions are:

Accuracy report of models

Model	1st_Layer	Hidden	Accuracy
1	100	50.0	80.83333373069763
2	200	100.0	64.99999761581421
3	300	150.0	80.0000011920929
4	400	200.0	58.3333134651184
5	500	250.0	80.0000011920929
6	600	300.0	77.49999761581421
7	700	350.0	58.3333134651184
8	800	400.0	72.50000238418579
9	900	450.0	82.4999988079071

Fig 40: Accuracy of 9 Models with 1 Input Layer 1 Hidden Layer with Half Neurons

Classifiation Report of 9 Model Because of its maximum Accuracy

	precision	recall	f1-score	support
0	0.87	0.83	0.85	70
1	0.77	0.82	0.80	50
accuracy			0.82	120
macro avg	0.82	0.82	0.82	120
weighted avg	0.83	0.82	0.83	120

Fig 41: Classification report of 9th Model

CONCLUSION

Our project aims to make work of cataract detection easier by introducing concept of machine learning in Ophthalmology.

An automated cataract diagnostic system would be highly useful in poor countries with insufficient numbers of qualified ophthalmologists to treat patients. Such approaches would make healthcare more accessible, reduce time and screening costs for both the patient and the ophthalmologist, and enable early diagnosis.

Initially, a cataract dataset of fundus images was rearranged, pre-processed, and augmented to improve the dataset to feed the deep network. After that we trained our different models with present dataset and analysed their results. In Machine Learning Model, Random Forest Classifier gave best accuracy with accuracy score of 91% and In Deep Learning Model of ANN with 900 Neurons in Input Layer and 450 Neurons in Hidden Layer and 1 Neuron in Output layer, Activation function used is reLu and Sigmoid with Adam optimizer for 10 Epochs for Cataract Detection gave accuracy of 85.5% . So the best performing model is Random Forest Classifier and will be used for further action.

However, our method cannot discriminate the three types of age-related cataracts (nuclear cataracts, cortical cataracts, and PSCs). Besides, it was only proposed for cataract detection and not for grading or finding its exact location, which can be helpful for ophthalmologists. These issues need further investigation in the future.

In the future, we can focus on improving the accuracy of the model by using a larger and more complex dataset. We can also try to apply various image processing methods so that the model can learn the image pattern more accurately and give better accuracy more efficiently. We can also build a website for easy access by all people worldwide.

REFERENCES

1. Brill R. 1986. The Conference of Educational Administrators Serving the Eye: A History. Washington, DC: Gallaudet University Press.

2. Munib Q., Habeeb M., Takruri B. and Al-Malik H. A. 2007. Cataract Detection (CD) based on Hough transform and neural networks. Expert Systems with Applications. 32: 24-37.

3. Zeshan U., Vasishta M. M. and Sethna M. 2005. Implementation of Cataract Detection in Medical Settings. Asia Pacific Journal. (1): 16-40.

4. Banerji J. N. 1928. India International Reports of Schools for the Medical. Washington City: Volta Bureau. pp. 18-19.

5. Vasishta M., Woodward J. and Wilson K. 1978. Cataract Detection in India: regional variation within the cataract population. Indian Journal of Cataract. 4(2): 66-74.

6. https://docs.opencv.org/4.x/index.html

7. https://medium.com/@aybukeyalcinerr/cataract -db9500331b2f67

8. https://kushalvyas.github.io/BOV.

9. Greg C. Lee & Fu-Hao Yeh & Yi-Han Hsiao. Kinect-based Taiwanese image recognition system. Multimed Tools Appl. 2014 October.

10. Zhang LG, Chen Y, Fang G, Chen X, Gao W. A Vision-Based Image Recognition System. In Proceedings of the 6th International Conference on Multimodal Interfaces; 2004; Pennyslavia: ACM. p. 198-204.

11. Lokhande P. Data Gloves for Image Recognition System. International Journal of Computer Applications. 2015: p. 11-14.

12. https://github.com/imRishabhGupta/Indian-Image -Recognition

13. Huang J, Zhou W, Li H, Li W. Image recognition using 3D convolutional neural networks. In Multimedia and Expo (ICME), 2015 IEEE International Conference on; 2015: IEEE. p. 1-6.

14. Chai X, Li G, Lin Y, Xu Z, Tang Y, Chen X. Image Recognition and Translation The 10th IEEE International Conference on Automatic Eye and Image Recognition. 2013; p. 22-26.

15. Pigou L, Dieleman S, Kindermans PJ, Schrauwen B. Sign Language Recognition using Convolutional Neural Networks. In Workshop at the European Conference on Computer Vision; 2014; Belgium. p. 572-578

PERSONAL DETAILS

Name: Aryan Shukla
Enrollment No.: 181B052
Branch: Computer Science and Engineering
Email ID: aryananiket123@gmail.com
Address: H No. 115, Eldeco Shaurya, Bijnor, Lucknow, Uttar Pradesh - 226002
University: Jaypee University of Engineering and Technology, Guna
Phone No.: +91-6306980974

Name: Kshitiz Mishra
Enrollment No.: 181B111
Branch: Computer Science and Engineering
Email ID: kshitizmishra3@gmail.com
Address: H No. 1632 Lavkushpuram Kalyanpur Kanpur, Uttar Pradesh - 208017
University: Jaypee University of Engineering and Technology, Guna
Phone No.: +91-7905414260

Name- Mohd Shariq Khan
Enrollment No: 181B127
Branch: Computer Science and Engineering
Email ID: shariqk071@gmail.com
Address: F-32,Satyam Apartment, Civil Lines, Prayagraj, Uttar Pradesh, 211001
University: Jaypee University of Engineering and Technology, Guna
Phone No.: +91-9140339707