INTRODUCTION

1. INTRODUCTION

1.1 OVERVIEW OF THE PROJECT

A cataract is a white cloud formation that gradually hardens and forms a yellow sheet on the normally transparent lens of the eye. This condition which leads to a decrease in the vision of a healthy eye is caused as the light does not reach the retina as the eye lens is covered by a hardened white cloud.

The World Health Organization conducted a survey on world sight day 2019 under which 36 million people were identified as lined and 217 million had moderate or severe distance vision impairment (MSVI), out of these cases of blindness and MSVI about 65 million people were affected by cataract [1]. Thus, cataract is responsible for or 48% of blindness cases.

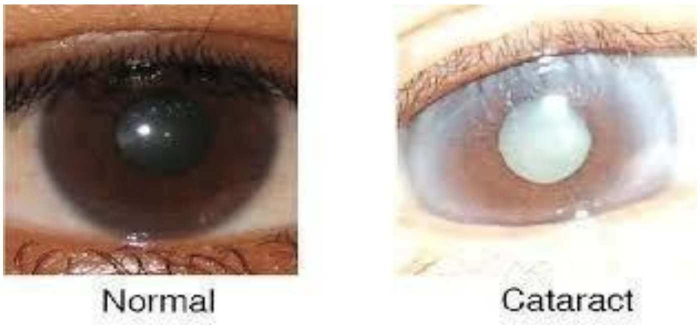


Fig 1.Normal Eye vs Cataract Eye

As per the statistics given by the National Blindness and Visual Impairment Survey India 2015-19, cataracts are responsible for 66.2% of blindness cases, 80.7% severe visual impairments, and 70.2% moderate visual problems for an age group of above 50 years of year.

Even though cataract develops very slowly it can cause long-term problems as it moves from covering a very small portion to spreading over the eye lens causing vision loss since light can no longer reach the retina due to the cloud that is formed on the lens.

SIGNS AND SYMPTOMS

1. Blurry Vision.
2. Difficulty to see things at night.
3. Eyes get sensitive to light.
4. Brighter light needed for reading,
5. Halo or circular formation is seen around light sources.

· **CAUSES AND RISK FACTORS**:

1. Ultraviolet radiation
2. Diabetes
3. Hypertension
4. Obesity
5. Smoking
6. Use of corticosteroid medications.
7. Medical history of eye injury or survey.
8. Alcohol consumption.
9. High rate of myopia
10. Previous Family history
11. Age

DIAGNOSIS OF CATARACT:

1. **Visual acuity test.** The current system of conventional practices followed to detect cataracts or other similar diseases includes a visual acuity test under which the eyesight of the patient is checked with the help of a chart or a viewing device that has letters indecreasing order of its size.

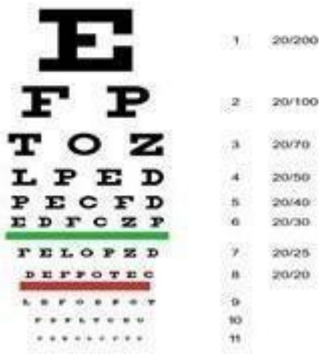


Fig 2.Chart used for acuity test

Slit-lamp examination. This detection technique helps to check the vision accuracy of the patient another method which is majorly used by the doctors and ophthalmologist is slit lamp examination to see the structure of the front of the eye with the bright illumination of light on the cornea, iris, and lens all under high magnification

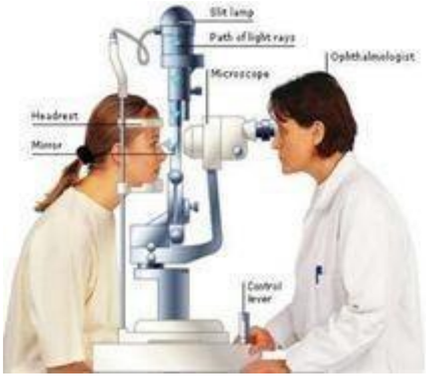


Fig 3.Slit-lamp examination

2. **Retinal exam.** Another method is a retinal exam in which using a slit lamp or an ophthalmoscope, the doctor examines the lens for signs of cataract. The pupil is made to stretch out a little wider than usual by making use of a suitable eye drop

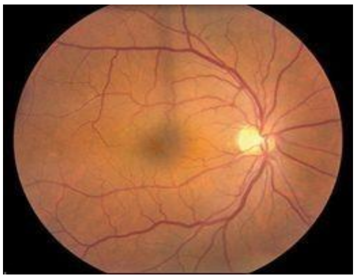


Fig 4. Fundus image

SYSTEM ANALYSIS AND STUDY

2. SYSTEM ANALYSIS AND STUDY

System analysis is a process of gathering the facts concerning the system breaking them into elements and relationship between elements. It provides a framework for visualizing the organizational and environmental factors that operates on a system. The quality of work performed by a machine is usually uniform, neat and more reliable when compared to doing the same operations manually.

- System analysis is conducted with following things in mind:
- Identify the employee needs.
- Evaluate the system concept of feasibility.
- Perform economic and technical analysis.
- Allocate functions to hardware, software, people and database.
- Create a system definition that forms the foundation for all subsequent engineering work.

System analysis is about understanding situations and not solving the problem. Effectiveness analysis, therefore emphasize investigations and questioning to learn how a system currently operates and to identify the requirements that have for a new or modified one and also it helps to handle the development and expression of the software under consideration.

System analysis is concerned with becoming aware of the problem. Identifying the relevant and most decisional variables and determining an optimal or at least a satisfactory solution. During this problem is identified, alternate system solutions are studied and recommendations are made about committing the resource used to design the system. The analyst meets the customer and the end user, obtain the information about their needs. Once the overall goals are identified, the analyst moves to an evaluation of supplementary information. In general, system analysis is the reduction of the entire system but studying the various operation performed and their relationship with the system

2.1 EXISTING SYSTEM

As we studied the existing work done on this topic a system which is based on image processing and developed on the MATLAB platform shows an accuracy of about 90% and further suggests using SVM to improve the accuracy [7]. Another project uses fundus images of the retina, the RGB image is first converted into green channel one and by applying image processing techniques, the contrast is enhanced and noise is removed. Binary SVM is implemented to classify the fundus image along with the MDA algorithm. Classification is carried out for two classes i.e. non-cataract and cataract. The grading part consists of three classes i.e. mild, moderate, and severe cataract. The dataset used consists of 261 images.

A system was developed using the following approaches 1. computer vision approach of extracting features and using them to develop machine learning models (b) automatic feature generation and classification using convolutional neural networks. A set of six hundred labeled retinal images was used to train the models[9]. A project proposed a method to detect cataracts using smartphones like Android, iOS. The camera detects the eyes, crop the required region of the eyes. When cropping the pupil they are stored in a matrix. They are analyzed with the Native Development Kit on the android platform. The two main features average or mean intensity of image and histogram are used to determine the presence of cataracts in the patient

DRAWBACKS OF EXISTING SYSTEM

At present, there has been a lot of work in the field of detection of various eye diseases and the cataract is one of the most common of them. A cataract is a white cloud formation that gradually hardens and forms a yellow sheet on the normally transparent lens of the eye. The condition of cataracts is responsible for 48% of world blindness.

These problems with existing systems include:

1.Use of small datasets.

2.Heavy algorithms resulting in high efficiency but increased computation power.

Therefore, we are proposing a system which will help in the detection of cataract and their type by considering these above disadvantages.

2.2 PROPOSED SYSTEM

The aim of this project is to detect the presence/absence of cataracts. The system would provide quick results to the user with great accuracy and efficiency. The user can upload the image on the website and obtain the results any time and based on the results the user can take appropriate actions.

2.2.1 Advantages of proposed system

- To build a system that will help to detect the presence of cataracts and also determine the type of cataract.
- To build a system that improves the efficiency of detection
- To build a system that is accessible anywhere and at any time.
- To help save the time of the patient by producing immediate results.

2.2.2 Project Description

Our system is divided into two phases

1. First phase aims to detect the presence of cataract
2. If cataract is present then the second phase classifies the type of cataract on the severity level: normal,mild and severe.

a) First phase Our Cataract detection system involves a two stage process :

1) Detection of textural features to form the vector.

2) Input the vector to the classifier to get the results.

After forming the dataset by collecting the relevant images from a stock image website and project source project, to form a simple and better quality dataset the images were manually cropped to the region of the iris and then resized to about 128 pixels. The feature vector is formed by the combination of features extracted from SIFT and GLCM algorithms. SIFT contributed a total of 2303 feature points and GLCM gave 5 features, thus the total size of final feature vector was 2308

These features were used to train different machine learning classifiers like support vector machine, K-nearest neighbour, random forest and logistic regression. By comparing the accuracies it was concluded that Logistic Regression proved to be better for our work with a score of 96% and thus it was converted into a pickle object file that was later used in our flask web application to give the results when the user would input an image into the system.

b) Second phase

Phase 2 dataset consists of 550 images that are further into three classes namely; 220 mild, 166 normal and 164 severe images. To extract the region of interest i.e pupil of the eye , images were resized to 224 pixels and converted to hsv color space. Pupil color scale mask is applied on the images to retain the pupil part. After this Hough circle transform is applied to find the coordinates of the pupil circular area. Finally the rectangular contour is drawn around the circular area from the coordinate calculated above and the pupil part is extracted. Furthermore for the machine learning part, 2303 SIFT and 5 GLCM features were extracted from the images thus the total size of the final feature vector was 2308.

17 These preprocessed images are used for training using pre-trained deep convolutional learning models like SqueezeNet, MobileNet and VGG16. By observing the results from these models , we obtained the highest accuracy 97.66% from the SqueezeNet. On the other hand for the machine learning model , 2303 SIFT with 5 GLCM thus a total of 2308 combined features were used to train different machine learning classifiers like support vector machine, XGBoost ,random forest and AdaBoost algorithm . By comparing the results, XGBoost demonstrated the best model with accuracy of 96%.

Finally the Logistic regression model of phase 1 and SqueezeNet model of phase 2 were converted into a pickle object file that was later used in our flask web application to give the results when the user would input an image into the system

In this research paper we have made an attempt to evaluate the accuracy when different feature extraction algorithms are applied as classification parameters to the deep learning models trained with different algorithms such as VGG 16, MobileNet and SqueezeNet. The sample images from the dataset has been listed below

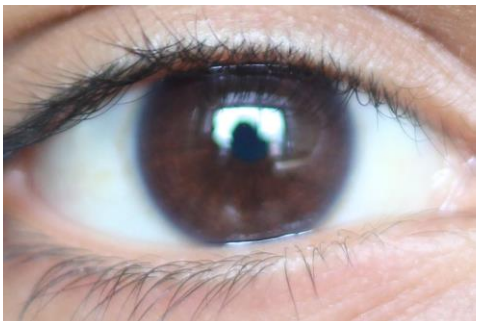


Fig 5. Healthy Eye Image from dataset

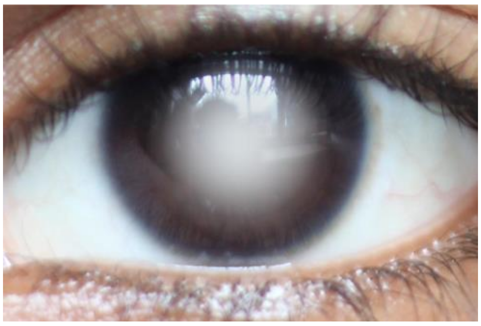


Fig 6. Mild Cataract Eye Image from dataset

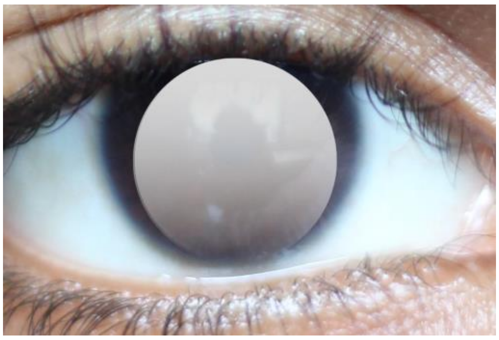


Fig 7. Severe Cataract Eye Image from dataset

SYSTEM REQUIREMENTS

3. SYSTEM REQUIREMENTS

3.1 HARDWARE REQUIREMENTS:

SYSTEM	:	Pentium IV 2.4 GHz.
HARD DISK	:	500 GB.
MONITOR	:	21 LED Colour.
MOUSE	:	Logitech
RAM	:	2 GB
KEY BOARD	:	104 Keys

3.2 SOFTWARE REQUIREMENTS:

OPERATING SYSTEM	:	Windows XP
FRONT END	:	django
BACK END	:	SQL lite

3.3 SOFTWARE DESCRIPTION:

3.3.1 DJANGO

Django is a free and open-source, Python-based web framework that follows the model–template–views architectural pattern. It is maintained by the Django Software Foundation, an independent organization established in the US as a 501 non-profit.

Django is a high-level Python web framework that enables rapid development of secure and maintainable websites. Built by experienced developers, Django takes care of much of the hassle of web development, so you can focus on writing your app without needing to reinvent the wheel.

The Django web framework is a free, open source framework that can speed up development of a web application being built in the Python programming language. Django—pronounced “Jango,” named after the famous jazz guitarist Django Reinhardt—is a free, open source framework that was first publicly released in 2005.

DJANGO ENVIRONMENT SETUP:

Django can be installed easily using pip . In the command prompt, execute the following command: pip install django . This will download and install Django. After the installation has completed, you can verify your Django installation by executing django-admin --version in the command prompt.

The virtual environment is an environment which is used by Django to execute an application. It is recommended to create and execute a Django application in a separate environment. Python provides a tool virtualenv to create an isolated Python environment. We will use this tool to create a virtual environment for our Django application.

Django framework development process:

To build such a complicated web system, we need three major parts for each component: database, user interface and the functions to interact in between. Django framework provides sufficient functionalities to implement these three parts. Corresponding to database, user interface and functions in between, Django has model, template and view components to deal with each part respectively. Django’s model component helps programmer to define and maintain tables in the database, while its template component helps to write html files using a combination of both html syntax and Django syntax.

Top Features of Django Framework

1. Excellent Documentation

2. python web-framework

3. SEO optimised

4. high scalability
5. versatile in nature
6. offers high security
7. thoroughly tested
8. provides rapid development

3.3.2 SQL LITE

SQLite is a database engine written in the C programming language. It is not a standalone app; rather, it is a library that software developers embed in their apps. As such, it belongs to the family of embedded databases.

SQLite is an embedded, server-less relational database management system. It is an in-memory open-source library with zero configuration and does not require any installation. Also, it is very convenient as it's less than 500kb in size, which is significantly lesser than other database management systems.

DEVELOPER (CAN SUPPORT WINDOWS XP OS)

• Easy-to-platforms to use embedded database engine that lets developers build robust Windows Desktop and mobile applications that run on all Windows platforms A free, easy Compact Business solutions limits on size or number of users SQL Server 2005 Management Studio Express includes all of the functionality of Enterprise Edition, but is licensed only for development, test, and demo use

ADMINISTRATOR’S DUTIES

• Install and configure SQL LITE

• Plan and create databases

• Back up the databases

• Restore the databases when necessary

- Set up and manage users for SQL Server
- Manage security for new users and existing users
- Import and export data
- Set up and manage tasks, alerts, and operators
- Manage the replication environment
- Tune the SQL Server system for the optimal performance
- Troubleshoot any SQL Server problems

A Simplified Installation Process

- Starting fromsetup.exe
- Click buttons other than “Cancel Cancel” in the Wizard (using most of the default setup)
  - Select components to install:
- SQL Server Database Services
- Workstation components, books online and development tools

– User “Advanced Advanced” option to setup installation path and include sample databases

– Create a default instance

– Use the built built-in System account: Local System

– User Windows Authentication Mode

Advanatage of sql lite:

One of SQLite's greatest advantages is that it can run nearly anywhere. SQLite has been ported to a wide variety of platforms: Windows, MacOS, Linux, iOS, Android, and more. Windows users in particular can use precompiled binaries for regular Win32, UWP, WinRT, and.

SYSTEM DESIGN AND DEVELOPMENT

4. SYSTEM DESIGN AND DEVELOPMENT

SYSTEM DESIGN

Design has been described as a multi step process in which represents of data structure, program, structure, interface characteristics and procedural details are synthesized from information requirements. The design phase begins when the requirements specification document for the software to be developed is available. System design is a creative and challenging phase that serves as a foundation for all software engineering and maintenance steps that follow. Design builds coherent, well-planned representation of programs. The design specification describes the features of the system, the components or elements of the system and their appearance to users. In the design phase, data flow, data-stores, processes, procedures, controls etc., are designed.

4.1 DATA FLOW DIAGRAM

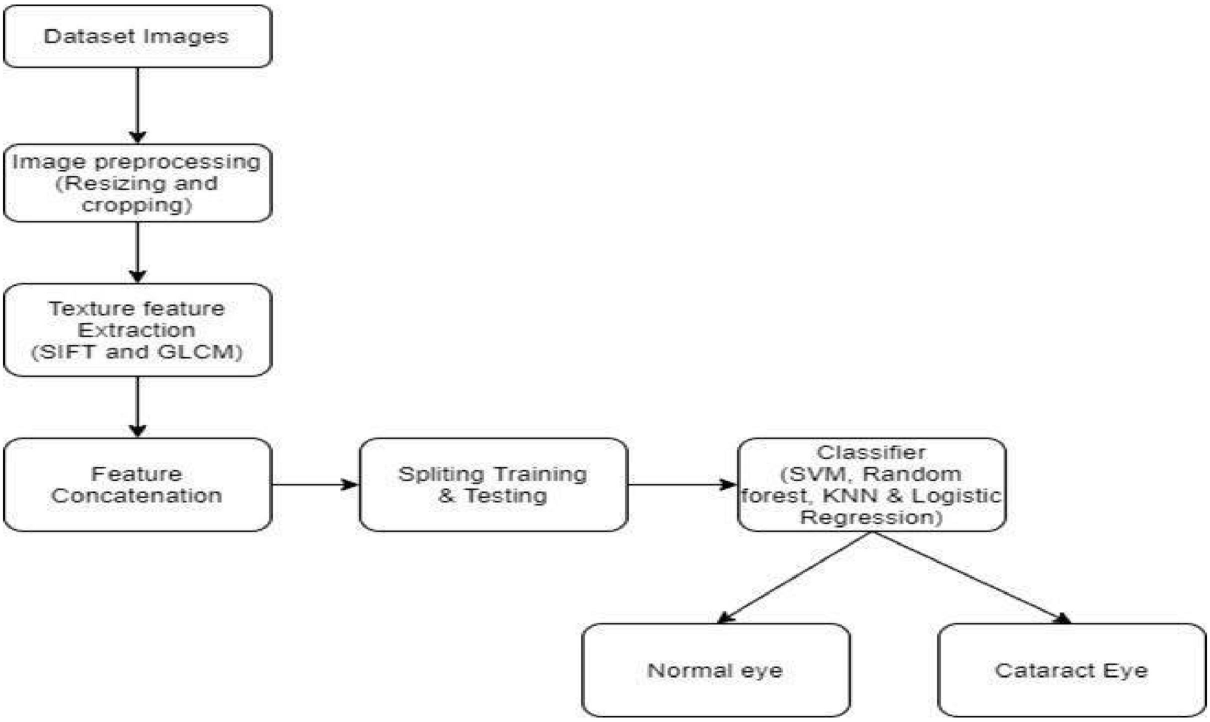

Dataset Images
Image preprocessing (Resizing and cropping)
Texture feature Extraction (SIFT and GLCM)
Feature Concatenation
Spliting Training & Testing
Classifier (SVM, Random forest, KNN & Logistic Regression)
Normal eye
Cataract Eye

Fig 8. Data Flow binary classification

For binary classification:

a) **Image Acquisition:** The two sets of images that are cataract and normal eye image used as the dataset. These images are gathered from various free stock images websites and also images used in open source projects. Total size of dataset used in 97 images.

b) **Image preprocessing:** The images in the dataset are manually cropped to extract required regions of interest as the dataset used was not very uniform and then resized to a size of 128 pixels.

c) **Texture feature Extraction:** The two algorithms used to extract the texture features are GLCM & SIFT. The feature vectors are then concatenated to form the combined feature vector.

d) **Splitting data for training and testing:** The data is divided in the ratio of 75% for training and 25% for testing

e) **Classifiers :** SVM, Logistic Regression, KNN, Random Forest are used to segregate the result into normal or cataract diagnosed eyes

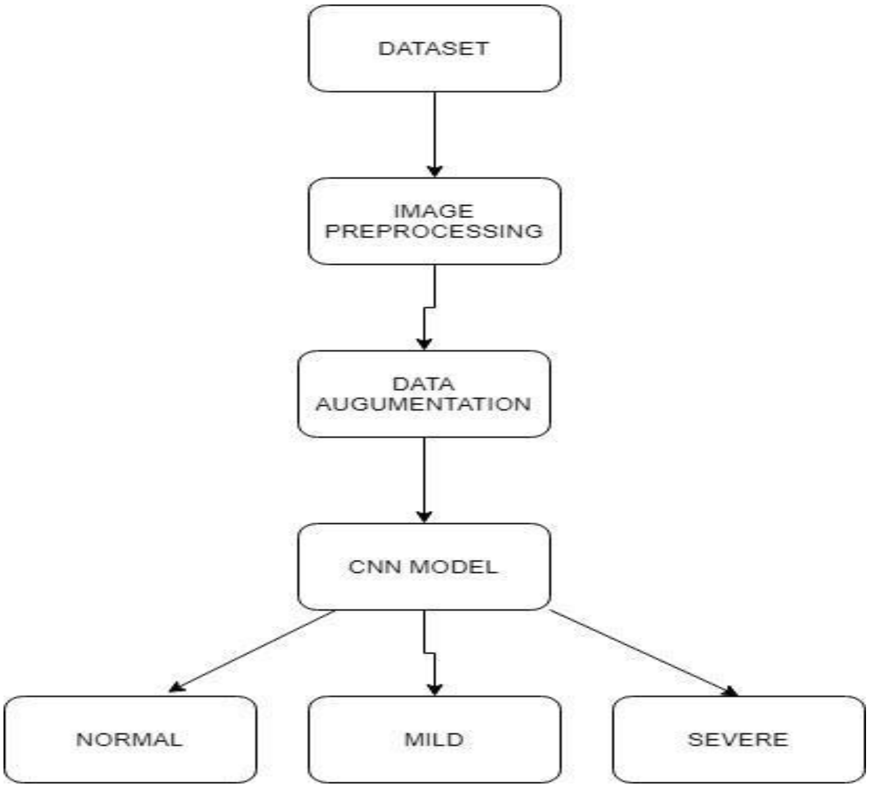


Fig 9. Data Flow multiclass classification

For Multiclass classification

a. **Image Preprocessing:**

i. First the Input Image is Resized into 224 *224.

ii. The image is then converted into HSV(Hue-Saturation-Value) format.

iii. To identify the Region of Interest(ROI) we have applied a mask to the image.

iv. After that, we threshold the image.

v. Apply Hough Circle in order to get theROI

vi. Crop the image and store it in the respective directory.

b. **Data Augmentation:** The preprocessed images then divided into train and test directories in the ratio 3:1. The corresponding data generators are created.Variations in the augmented images include changes in the horizontal alignment and zoom.

c. **Input to the CNN models:** The three pre-trained models used by us were MobileNet, VGG-16 and SqueezeNet. The Models were trained for about 20-30 epochs with the aim to maximise the validation accuracy.

4.2 SYSTEM FLOW DIAGRAM

To make our results available to the end user, we have developed a web based graphical user interface using a popular python based framework which is FLASK.

The flow of our system is as follows:

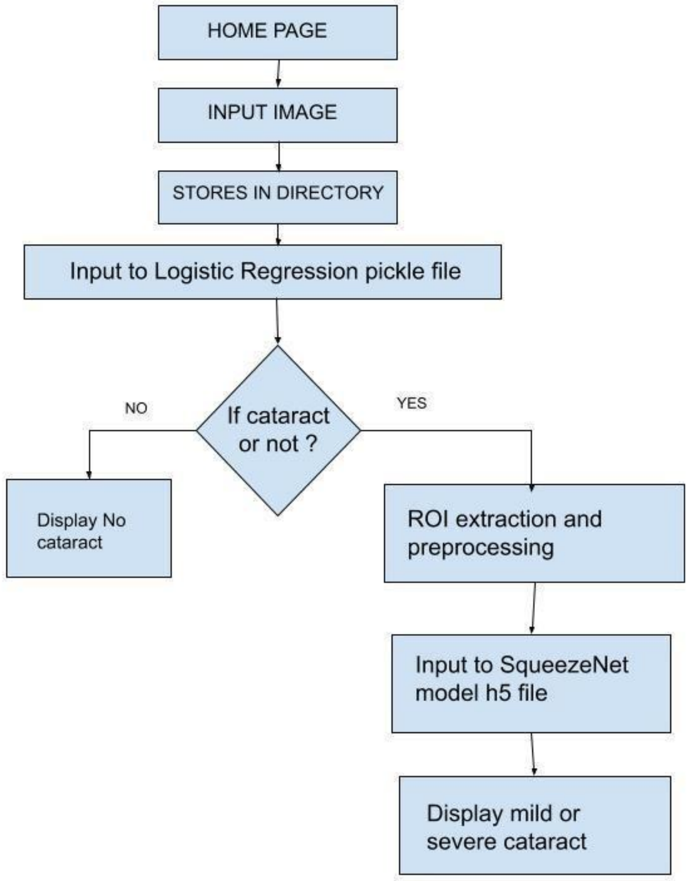

HOME PAGE
INPUT IMAGE
STORES IN DIRECTORY
Input to Logistic Regression pickle file
If cataract or not ?
NO
YES
Display No cataract
ROI extraction and preprocessing
Input to SqueezeNet model h5 file
Display mild or severe cataract

Fig 10. Flow of Flask GUI

4.3 INPUT DESIGN

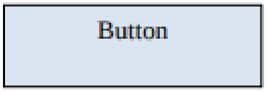
The input forms are developed in a user-friendly way so that a layman also can easily understand everything. The forms are designed in such a way that end-use can easily navigate through entire system.

The input design is a process of converting user-originated inputs to a computer-based format.

The goal of designing input data is to make entry easy, logical and free errors. In input data design, we design source document that capture the data then select the media used enter them into computer. The goal of input design is to enter to the computer details as accurately as possible

BUTTON:

Button is one of the important input designs. It helps us submit the details.



Objectives of input design:

o To achieve the highest possible level of accuracy

o To ensure that the input is acceptable and understood by the user

o Attractiveness

4.4 OUTPUT DESIGN

In the output design the emphasis is on producing a hard copy of the information requested or displaying the output on the CRT screen in a predetermined format. Two of the most output media today are printer and screen display. Computer output is the most important and direct source of information to the user and help in decision making. All the output is the most important source of information to the user. Better design should improve the systems relation and also should help in decision making.

4.5 TABLE SPECIFICATION

4.5.1 Binary classification for the presence of cataract

Table No.1 Accuracy rates of various classifiers with SIFT feature

	Classifier	Accuracy
1.	SVM-LINEAR	92%
2.	SVM-RBF	92%
3.	Random Forest	82%
4.	Logistic Regression	92%
5.	KNN	68%

Table No.2 Accuracy rates of various classifiers with GLCM features

Classifier	**Accuracy**
6. SVM-LINEAR	76%
7. SVM-RBF	56%
8. Random Forest	64%
9. Logistic Regression	80%
10. KNN	60%

Table No.3 Accuracy rates of various classifiers with SIFT and GLCM features.

Classifier	**Accuracy**
11. SVM-LINEAR	96%
12. SVM-RBF	96%
13. Random Forest	96%
14. Logistic Regression	96%
15. KNN	56%

4.5.2 Multi class classification for the detection of cataract type

Table No.4

Accuracy rates of various classifiers with preprocessed image of size 224

Sr.No	**Model**	**Validation Accuracy**
1	VGG 16	96.88
2	MobileNet	90.62
3	SqueezeNet	97.66

Table No.5 Accuracy rates of various classifiers with SIFT+ GLCM features

Classifier	**Accuracy**
1. Random Forest	87%
2. AdaBoost Classifier	89%
3. XGB Classifier	96%
4. SVM-Linear	79%

TESTING AND IMPLEMENTATION

5. SYSTEM TESTING AND IMPLEMENTATION

5.1 SYSTEM TESTING

The success of the testing process in determining the error is mostly depends upon the test case criteria, for testing any software we need to have a description of the expected behavior of the system and method of determining whether the observed behavior confirmed to the expected behavior.

All testing methods such black box, white box and integration testing have done in this system. White box testing focuses on internal structure of the component. It covers all statement, branch, path, condition and dataflow in a program.

- Black box testing
- White box testing
- Unit testing
- Integration testing
- Validation testing
- Output testing
- User acceptance testing
- Level of testing used in the project

BLACK BOX TESTING

Black box testing methods focus on the functional requirement of the software. Using the black box testing method the following errors are identified and rectified in the package.

- Incorrect or missing functions
- Interface errors
- Errors in data structure or external database access

WHITE BOX TESTING

White box testing is done with the project which drive test cases that do the following

- Guarantee that all the independent paths with in modules have been exercise at least once.
- Exercise all logical decision on the true and false side
- Execute all loops at the boundaries and within their operation bounds.
- Exercise internal data structure to ensure the validity 30

It is aimed at ensuring that the system works accurately and efficiently before live operation command.

UNIT TESTING

Unit testing mainly focused first in the smallest and low level modules, proceeding one at a time. Bottom-up testing was performed on each module. As developing a driver program, the test modules by developed or use. But for the purpose of testing modules they were used as stubs, to print verification of the actions performed. After the lower level modules were tested, the modules that the next higher level those make use of the lower modules were tested.

Each module was tested against required functionality and test cases were developed to test boundary values. The unit testing has been tested with sample data and adequate corrections were made as per the user requirement, such that login module was tested with user id and code and appropriate error message are provided for error like data entry error, id error, etc.

INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure, while at the same time conducting tests to uncover error associated with interfacing. As the system consists of the number the edge of the two modules. The software tested under this incremental bottom-up approach.

- Bottom-up approach integration strategy was implemented with the following steps.

- Low modules were combined into cluster that perform specific software sub function
- The clusters were tested

OUTPUT TESTING

After performing the validation testing, the next step is the output testing of the proposed system, since no system could be useful if it does not produce required output in the specific format. Tested asking the users about the format required by them, the output is considered into two ways: one is on the screen and other is printed format.

The output format on the screen is found to be correct as the format designed according to the user needs, for the hard copy also, the output comes as specified by the user. Hence output testing does not result in correction in the system

USER ACCEPTANCE TESTING

User acceptance testing of a system is the key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keep in touch with the prospective system user at time of developing and making changes wherever required.

The both functional and structural testing has done in this system. The functional testing focuses in the input, output behavior of the component. The structural testing focuses on the internal structure of the component. The modules in this project has been tested thoroughly and found to be accurate which can meet the needs on the user. The tested modules are finally combined together into a complete one.

5.2 SYSTEM IMPLEMENTATION

Implementation is used here to mean the process of converting a new or revised system design into operational one. System implementation is the stage of the project where the theoretical design is turned into a working system. At this stage the main work load, the greatest up level and the major impact on the existing system shifts to the user department. It the implementation is not planned and controlled it can cause chaos and confusion.

Implementation includes all those activities that take place to convert from the old system to the new one. The new system may be totally new, replacing an existing manual or automated system or it may the major modification to an existing system. Proper implementation is essential to provide a reliable system to meet the organization requirements. Successful implementation may not guarantee improvement in the organization using the new system, but improper installation will prevent it.

The process of putting the developed system in actual use is called system implementation. This includes all those activities that take place to convert from the old system to the new system. The system can be implemented only after through testing is done and if it is found to be working according to the specification. The system personnel check the feasibility of the system.

❖ The implementation stage involves following tasks.

- Careful planning
- Investigation of system and constraint
- Design of methods to achieve the changeover
- Training of the staff in the changeover phase
- Evaluation of the changeover method.

❖ There are three type of implementation,

  - Implementation of a computer system
  - Implementation of new computer system
  - Implementation of a modified application

Implementation of a computer system

It’s should be replaced a manual system the problem encountered are converting files, training users creating accurate files, and verifying printouts for integrity.

Implementation of new computer system

It's should be replace an existing one this is usually a difficult conversion. If not properly planned there can be many problems. Some large computer systems have taken even years to convert.

Implementation of a modified application

It's should be replace an existing one using the same computer. This type of conversion is relativity easy to handle provided there are no major changes to the file.

The method of implementation and the time scale to be adopted are found out initially. Next the system is tested properly and the same time users are trained in the new procedures.

SYSTEM MAINTENANCE

Software maintenance in software engineering is the modification of a software product after delivery to correct faults, to improve performance or other attributes, or to adapt the product to a modified environment.

Software maintenance process

This international standard describes the software maintenance processes as:

- The implementation processes contains software preparation and transition activities, such as the conception and creation of the maintenance plan, the preparation for handling problems identified during development, and follow-up on product configuration management.
- The problem and modification analysis process, which is executed once the application has become the responsibility of the maintenance grouped. The maintenance programmer must analyze each request, confirm it and check its validity, investigate it and propose a solution, document the request and solution proposal, and, finally, obtain all the required authorizations to apply the modifications.

There are a number of processes, activities and practices that are unique to maintainers, for example:

- Transition: a controlled and coordinated sequence of activities during which a system is transferred progressively from the developer to the maintainer.
- Service level agreements (SLAs) and specialized (domain-specific) maintenance contracts negotiated maintainers.
- Modification request and problem report help desk: a problem-handling process used by maintainers to prioritize documents and route the requests they receive.

CONCLUSION

6. CONCLUSION

In this project, various machine learning and deep learning algorithms have been proposed for the Cataract detection and Classification.

For Cataract Detection , we have used texture features. We have used a new set of features by concatenating two sets of features: SIFT (Scale Invariant Feature Transform) and GLCM (Gray Level Cooccurrence Matrix). Thus, the new set of features have been obtained and used for Cataract prediction. The various machine learning algorithms proposed in our project are SVM(Linear), SVM(RBF), KNN, Logistic Regression, Random Forest. Out of which the highest accuracy achieved is 96%. Therefore, the combined set of features have obtained better results than the individual features.

For Cataract Classification, we have extracted the region of interest i.e. pupil and trained our Deep Learning models for classifying the eye image as mild cataract, severe cataract or normal eye image. The various Deep Learning models used are VGG 16, MobileNet and SqueezeNet. Out of these SqueezeNet has been able to achieve the highest accuracy of 97.66%. We have also evaluated the accuracies for various Machine learning Algorithms such as SVM-linear, XGBoost, Random Forest and Adaboost Classifier on the texture features obtained from the SIFT and GLCM algorithms. However the accuracies obtained are less when compared to the Deep Learning Models. In conclusion, Deep Learning models have achieved better accuracies when compared to the machine learning models.

SCOPE FOR FUTURE ENHANCEMENT

7. SCOPE FOR FUTURE ENHANCEMENT

The accuracy of the system can be increased by working on a large dataset with a different variety of pupil color space. Developing mobile applications along with existing websites will help the patients enabling easy access . A communication channel can be established between patients and ophthalmology clinics. Live video conference between patient and doctor to save time.

BIBILOGRAPHY

8. BIBLIOGRAPHY

[1] “Recent Approaches for Automatic Cataract Detection Analysis Using Image Processing” Journal of Network Communications and Emerging Technologies (JNCET), October (2017)

[2] “Remote Automated Cataract Detection System Based on Fundus Images” International Journal of Innovative Research in Science, Engineering, and Technology (June 2016)

[3] ” Automatic Retinal Disease Classification Using Machine Learning and AI”

[4] ” Cataract Detection using Smartphone” Journal of Network Communications and Emerging Technologies (JNCET), December (2017)

[5] “Cataracts,” Mayo Clinic, 23-Jun-2018.[Online].Available: https://www.mayoclinic.org/diseasesconditions/cataracts/symptoms-causes/syc20353790.

[6] Chart used for acuity test[Online] Available: https://www.zazzle.com/health+mousepads?rf=238177452029091674&tc=2280303

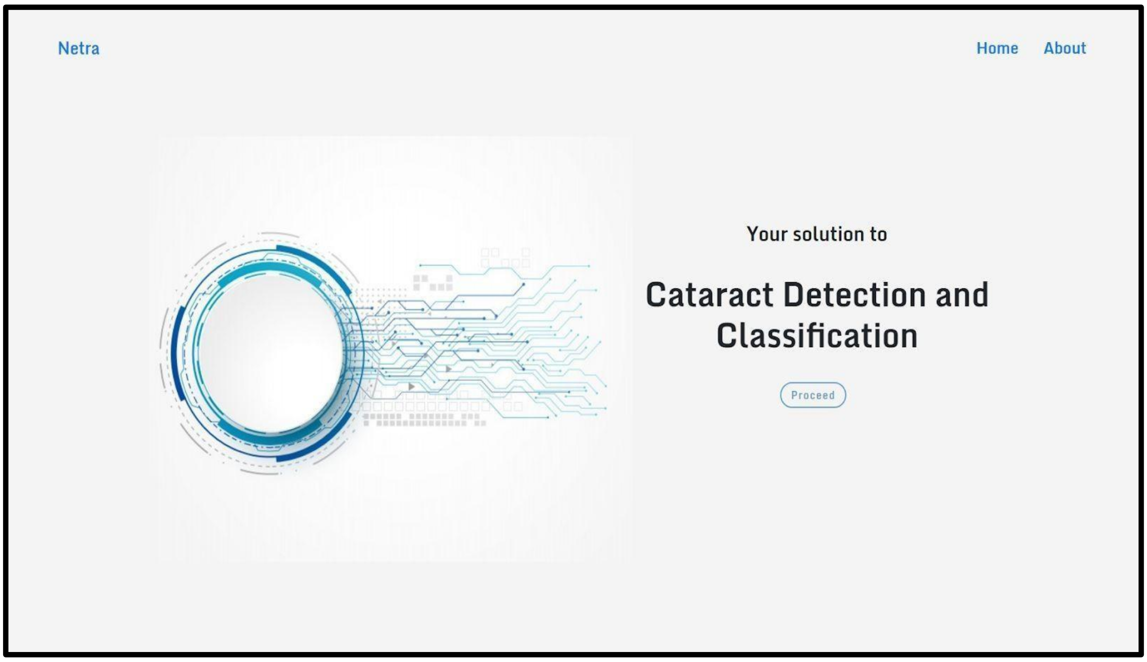
[7] Slit-lamp examination[Online] Available: http://www.webdicine.com/drooping-eyelid-part-2.html

ANNEXURE

9. ANNEXURE

SCREEN SHOT

MAIN PAGE:



Netra

Home About

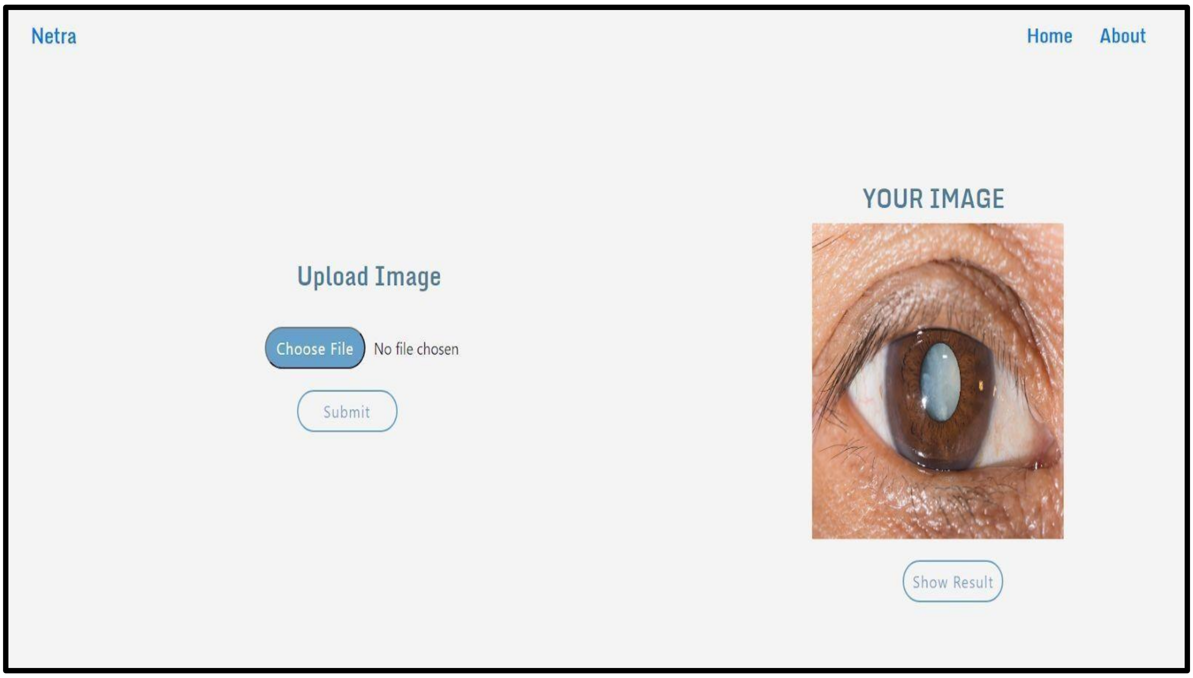
Your solution to

Cataract Detection and Classification

Proceed

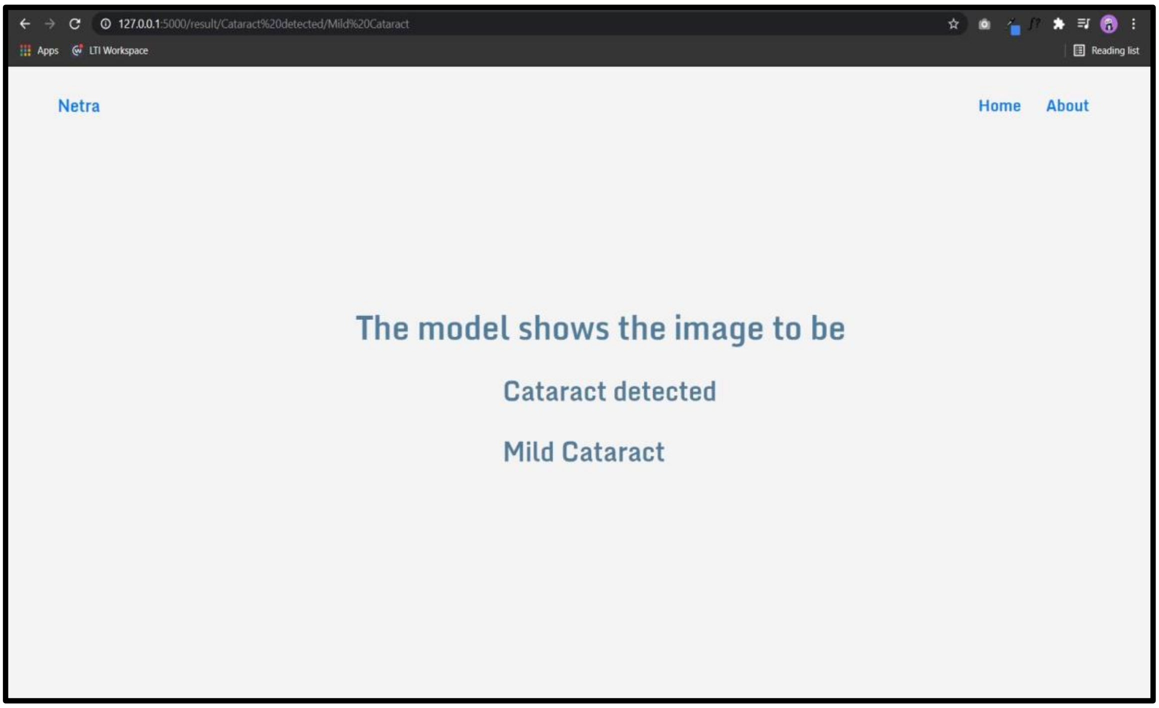
Main screen that the user would visit first:

ADD IMAGE:



Uploading the Image

RESULT:



The model shows the image to be

Cataract detected

Mild Cataract

Prediction by the model

SAMPLE CODE:

```
from flask import Flask, render_template, request
import cv2
import numpy as np
from tensorflow import keras

app = Flask(__name__)

# Load the trained model
model = keras.models.load_model('cataract_model.h5')

# Define the labels for the output classes
labels = ['No cataract', 'Mild cataract', 'Moderate cataract', 'Severe cataract']

@app.route('/')
def index():
   return render_template('index.html')

@app.route('/detect', methods=['POST'])
def detect():
   # Get the uploaded image
   img = request.files['image']

   # Read the image
   img = cv2.imdecode(np.fromstring(img.read(), np.uint8), cv2.IMREAD_COLOR)

   # Convert the image to grayscale
   gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

   # Apply Canny edge detection
   edges = cv2.Canny(gray, 100, 200)

   # Apply Hough transform to detect circles
   circles = cv2.HoughCircles(edges, cv2.HOUGH_GRADIENT, 1, 20, param1=50, param2=30,
minRadius=0, maxRadius=0)

   # If a circle is detected, classify the cataract
   if circles is not None:
        # Crop the image to the circle
        x, y, r = circles[0][0].astype(np.int)
        crop_img = img[y-r:y+r, x-r:x+r]

        # Resize the image to the input size of the model
        resize_img = cv2.resize(crop_img, (224, 224))

        # Normalize the image pixel values
        norm_img = resize_img.astype(np.float32) / 255.0

        # Add a batch dimension to the image
        batch_img = np.expand_dims(norm_img, axis=0)

        # Use the model to predict the class of the image
        pred = model.predict(batch_img)

        # Get the index of the predicted class with the highest probability
        class_idx = np.argmax(pred)

        # Get the label for the predicted class
        label = labels[class_idx]

        return f'Cataract detected: {label}'
    else:
        return 'No cataract detected.'

if __name__ == '__main__':
    app.run(debug=True)
```

Create a new HTML file called **`index.html`** in a new folder called **`templates`**. Add the following code:

```
<!DOCTYPE html>
<html>
        <head>
          <title>Cataract Eye Detection and Classification</title>
        </head>
        <body>
```

```
    <h1>Cataract Eye Detection and Classification</h1>
    <form method="POST" action="/detect" enctype="multipart/form-data">
      <input type="file" name="image">
      <br>
      <button type="submit">Detect and Classify Cataract</button>
    </form>
  </body>
</html>
```

```
"source": [
 "import cv2\n",
 "import numpy as np\n",
 "from skimage.feature import  graycomatrix, graycoprops\n",
 "images_sift = []\n",
 "glcm=[]\n",
 "labels = []\n",
 "size = 128\n",
 "sift = cv2.SIFT_create()\n",
 "cataract=0\n",
 "normal=0\n",
 "for i, file in enumerate(file_names):\n",
 "       image = cv2.imread(mypath+\"/\"+file,0)\n",
 "       h,w=image.shape\n",
 "       if(h>128 and w>128):\n",
 "        image = cv2.resize(image,(size, size),interpolation = cv2.INTER_AREA)\n",

 "        img_arr = np.array(image)\n",

 "        gCoMat = graycomatrix(img_arr,[1],[0],256,symmetric=True, normed=True)
 # Co-occurance matrix\n",

 "        contrast = graycoprops(gCoMat, prop='contrast')[0][0]\n",

 "        dissimilarity = graycoprops(gCoMat, prop='dissimilarity')[0][0]\n",

 "        homogeneity = graycoprops(gCoMat, prop='homogeneity')[0][0]\n",

 "        energy = graycoprops(gCoMat, prop='energy')[0][0]\n",

 "        correlation = graycoprops(gCoMat, prop='correlation')[0][0]\n",

 "        keypoints, descriptors = sift.detectAndCompute(image,None)\n",

 "        descriptors=np.array(descriptors)\n",

 "        descriptors=descriptors.flatten()\n",

 "        glcm.append([contrast,dissimilarity,homogeneity,energy,correlation])\n",

 "        images_sift.append(descriptors[:2304])\n",

 "\n",

 "        #print(descriptors.shape)\n",

 "        if file_names[i][0] == \"c\":\n",

 "            cataract+=1\n",

 "            labels.append(1)\n",

 "        if file_names[i][0] == \"n\":\n",

 "            normal+=1\n",
     "                labels.append(0)\n",
     "        \n",
     "print(\"Testing and validation split done!\")\n",
     "print(cataract)\n",
     "print(normal)\n",
     "print(images_sift)\n",
     "print(glcm)"
    ]
   },
   {
    "cell_type": "code",
    "execution_count": 27,
    "metadata": {},
    "outputs": [
     {
      "data": {
       "text/plain": [
        "53"
       ]
      },
    "execution_count": 27,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "len(labels)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": 29,
  "metadata": {},
  "outputs": [
   {
    "data": {
     "text/plain": [
      "(53, 2304)"
     ]
    },
      "execution_count": 29,
      "metadata": {},
      "output_type": "execute_result"
     }
    ],
    "source": [
     "images_sift=np.array(images_sift)\n",
     "images_sift.shape\n",
     "\n",
     "\n",
     "\n"
    ]
   },
   {
    "cell_type": "code",
    "execution_count": 125,
    "metadata": {},
    "outputs": [
     {
      "data": {
```

```
   "text/plain": [
    "array([[3.00000000e+01, 5.00000000e+00, 1.00000000e+00, ...,\n",
    "        2.10072013e-01, 1.98859587e-02, 9.70085948e-01],\n",
    "       [0.00000000e+00, 1.00000000e+00, 1.70000000e+01, ...,\n",
    "        2.75864084e-01, 2.94907227e-02, 9.82885226e-01],\n",
    "       [3.00000000e+01, 4.00000000e+01, 7.00000000e+00, ...,\n",
    "        1.70442415e-01, 1.66696327e-02, 9.46103064e-01],\n",
    "       ...,\n",
    "       [1.38000000e+02, 1.60000000e+01, 0.00000000e+00, ...,\n",
    "        1.70217030e-01, 3.61030365e-02, 9.61894831e-01],\n",
    "       [0.00000000e+00, 0.00000000e+00, 0.00000000e+00, ...,\n",
    "        2.73063781e-01, 2.23627724e-02, 9.94805514e-01],\n",
    "       [1.40000000e+01, 1.30000000e+01, 1.00000000e+00, ...,\n",
    "        2.86541824e-01, 3.24069491e-02, 9.79961578e-01]])"
   ]
  },
  "execution_count": 125,
  "metadata": {},
  "output_type": "execute_result"
 }
```

```
   ],
   "source": [
    "glcm=np.array(glcm)\n",
    "images_sift_glcm=np.concatenate((images_sift,glcm),axis=1)\n",
    "images_sift_glcm"
   ]
  },
  {
   "cell_type": "markdown",
   "metadata": {},
   "source": [
    "## Step 4: Building various  ML models"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": 31,
   "metadata": {},
   "outputs": [],
   "source": [
    "from sklearn import preprocessing\n",
    "import pandas as pd\n",
    "from sklearn.model_selection import KFold\n",
    "from sklearn.model_selection import cross_val_score\n",
    "\n",
    "from sklearn.linear_model import LogisticRegression\n",
    "\n",
    "from sklearn.metrics import confusion_matrix  \n",
    "from sklearn.model_selection import train_test_split\n",
    "from sklearn.metrics import accuracy_score\n",
    "\n",
    "\n",
    "\n",
    "logModel = LogisticRegression(solver='liblinear')\n",
    "\n",
    "model_names={\"logistic regression\":logModel}\n",
    "    \n",
    "\n"
   ]
  },
{
 "cell_type": "code",
 "execution_count": 48,
 "metadata": {},
 "outputs": [],
 "source": [
  "#images_sift\n",
  " #panda dataframe \n",
  "df=pd.DataFrame(images_sift_glcm)\n",
  "df['label']=labels\n",
  "df=df.sample(frac=1)\n",
  "X=df.drop(['label'], axis = 1)\n",
  "y=df['label']    \n",
  "    "
 ]
},
{
 "cell_type": "code",
 "execution_count": 126,
 "metadata": {},
"outputs": [
 {
  "data": {
   "text/html": [
    "<div>\n",
    "<style scoped>\n",
    "    .dataframe tbody tr th:only-of-type {\n",
    "        vertical-align: middle;\n",
    "    }\n",
    "\n",
    "    .dataframe tbody tr th {\n",
    "        vertical-align: top;\n",
    "    }\n",
    "\n",
    "    .dataframe thead th {\n",
    "        text-align: right;\n",
    "    }\n",
    "</style>\n",
    "<table border=\"1\" class=\"dataframe\">\n",
    "  <thead>\n",
     "    <tr style=\"text-align: right;\">\n",
     {
   "data": {
    "text/plain": [
     "'Predicted: Healthy Eye'"
    ]
   },
   "execution_count": 7,
   "metadata": {},
   "output_type": "execute_result"
  }
 ],
 "source": [
  "\n",
  "predict_new('healthyeye.jpeg')"
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
   "metadata": {},
   "outputs": [],
   "source": []
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "Python 3 (ipykernel)",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.8.5"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 4
}
```