Feynn labs ML Internship

"Early Detection of Cataracts using Deep Learning Techniques"

Vision AI

- Bringing the future of healthcare to the present

LADI JEEVAN SAI

06-02-2023

TASK - 0

Abstract:-

The project aimed to develop an AI-powered solution for early detection of cataracts in eye scans. To achieve the above mentioned goal, I used a pre-trained VGG19 model and fine-tuned it to classify normal eye scans from those affected with cataracts. I used a dataset from the kaggle which was collected from various hospitals and medical centers in China and consisted of images of over 5000 patients. The VGG19 model was trained on this dataset and was able to achieve an accuracy of 96%. The project involved the implementation of a web interface where users can input their eye scans and then the deep learning model in the backend is used to make predictions. Results are displayed on the web interface with an analysis report. This AI model can be integrated into existing healthcare systems to provide patients an efficient and accurate diagnosis for cataracts.
The project highlights the potential use of deep learning techniques in the medical field, specifically for the various early detection of eye diseases like diabetic retinopathy, cataract, glaucoma etc. The high accuracy achieved by the pre-trained VGG19 model shows that this solution has the potential to compete with traditional diagnostic methods and make a significant impact on healthcare. Further research and experimentation on the project can be conducted to improve the performance and accuracy of this AI solution.

1. Problem statement

To develop an AI-based solution for predicting the presence of cataract disease in patients using image scans of the left and right eyes, and to assess its accuracy and reliability compared to traditional diagnostic methods.
The objective of this study is to design, develop, and evaluate an AI-based solution for predicting the presence of cataract disease in patients using image scans of the left and right eyes. The system will be trained on eye scans using several deep learning algorithms and will analyze patterns and features indicative of cataracts. The accuracy and reliability of this AI solution will be compared to traditional diagnostic methods and the benefits and limitations of using AI for predicting cataracts will be analyzed.

2. Market/Customer/Business Need Assessment

(i) Market size and growth:

- The World Health Organization (WHO) estimates that cataracts are responsible for over 51% of all cases of blindness globally. Cataracts are a very common condition, especially among older adults, and their prevalence is increasing as the global population ages.

- In the United States, it is estimated that over half of all adults over the age of 80 will either have an eye cataract or have had surgery to treat it. Over 4 million cataract surgeries are performed each year in the United States.
- In India, it is estimated that over 14 million cataract surgeries are performed each year and globally, it is estimated that over 28 million cataract surgeries are performed each year.

This information demonstrates the significant size and growth of the market for cataract diagnosis and treatment, highlighting the need for effective and efficient solutions for this widespread condition.

(ii) Customer needs and demands:

- Customers always require faster and more accurate diagnoses. This requires an AI product that can quickly and accurately predict the presence of cataracts, and suggest more accurate diagnoses. The need for non-invasive diagnostic methods is also significant, as customers prefer treatments that are less intrusive. This is where the AI product comes in, as it is designed to be non-invasive and provide accurate diagnoses.
- On the other hand, healthcare providers are also interested in providing accurate and efficient diagnostic tools, and are constantly looking for innovative solutions that can improve patient outcomes and reduce costs. This drives the demand for technology that streamlines and simplifies the diagnostic process.
- In conclusion, understanding the needs and demands of both customers and healthcare providers is crucial in developing an AI product that effectively meets the needs of the market and fortunately my products satisfy all the requirements.

(iii)Market gaps and opportunities:

- Current gaps in the market: There is a gap in the market for a non-invasive, accurate, and efficient method of predicting the presence of cataract disease. The AI product will aim to fill this gap by providing a fast and accurate diagnostic tool that can be used by healthcare providers.

- Opportunities for innovation: In the 21st century there is an increasing demand for AI-based solutions, especially in medical fields, so there will be a significant opportunity for my AI product to predict the presence of cataract disease with more accurate and improved patient outcomes.
- Need for improved diagnostic tools: With the growing number of cataract patients, there is a growing need for improved diagnostic tools that can accurately predict the presence of the disease in patients. The AI product will address this gap by providing a faster, more accurate, and non-invasive diagnostic tool.
- New approaches to treatment: The AI product will provide new approaches to cataract treatment by offering a more accurate and efficient way of diagnosing the presence of the disease. This will help healthcare providers to make more informed decisions about patient treatment and improve patient outcomes.

3.Target Specifications and Characterization

- Cataracts typically begin developing in people aged 40 years and older but don't usually begin to impair vision until after age 60. However, younger people can develop cataracts so the target customers for the AI product are likely to be individuals of all ages and both genders.

- Geographically, the target customers are likely to be located in regions with an aging population and a high incidence of cataract disease, such as developed countries. The product is expected to be particularly in demand in urban areas with a high concentration of healthcare facilities and eye clinics.

- Also healthcare providers, such as ophthalmologists, optometrists, and eye clinics, can be considered as target customers for the AI product .These healthcare providers are often responsible for diagnosing and treating cataracts in their patients, and they are likely to be interested in innovative solutions that can improve the accuracy and efficiency of the diagnostic process. By incorporating the AI product into their diagnostic processes, healthcare providers can enhance their patient outcomes, reduce costs, and streamline their operations.

- At the same time the purchasing behavior of healthcare providers for the AI product will depend on several factors, including the quality and reliability of the product, its ease of use and compatibility with existing diagnostic processes, and its ability to integrate with other diagnostic tools and software.

- In terms of budget, healthcare providers may be willing to invest more in the AI product if they believe that it will result in improved patient outcomes, reduced costs, and enhanced efficiency. On the other side of the coin, they may also be hesitant to invest if they believe that the product is too expensive or if there are less expensive alternatives available in the market. So my AI product should also be cost-effective enough to attract healthcare providers.

4.External Search

- In conducting research for my project on AI-based cataract disease prediction, I utilized a variety of online information sources. To gather statistical data on the age, number of surgeries, and general information on cataracts, I consulted websites such as Pacific Eye Institute, the National Library of Medicine, and the American Academy of Ophthalmology.
- In addition to these sources, I also consulted several research papers that helped me understand the application of deep learning algorithms in predicting cataract disease. These resources were instrumental in providing a comprehensive understanding of the field and developing a well-informed approach for my project.

- I utilized a dataset from Kaggle, which was collected by Shanggong Medical Technology Co., Ltd. from various hospitals and medical centers in China. This dataset included images captured with different cameras, such as Canon, Zeiss, and Kowa, and at varying resolutions from 5000 patients of different ages.

5.Benchmarking alternate products

The data collected for the AI product was obtained from 5000 patients and it is a multimodal dataset which includes information related to the patient's age, gender, and other demographic details. The dataset was then processed to remove any irrelevant information and make sure that only relevant data will be used for training.

The model was then trained using a VGG architecture and transfer learning techniques, which involved using a pre-trained model trained on the IMAGENET dataset. The use of transfer learning ensured that the model was able to extract features related to cataract disease prediction more effectively and efficiently.

The above-mentioned steps ensure that the AI product will set a benchmark among similar products/services. The accuracy,efficiency and robustness towards new data with high performance will help it to stand out in the market.

6.Business Model:-

Few monetization ideas for the AI product that predicts the presence of cataract disease are as follows:

1. Subscription-based model: Healthcare providers can subscribe to the AI product and pay a monthly based or annual based fee to access the product.
2. Pay-per-use model: Healthcare providers can pay for each use of the product, such as for each patient's diagnosis.
3. License-based model: The AI product can be licensed to healthcare providers for a one-time fee, with an additional option for annual maintenance and support if required.
4. Integrating the product with EHR (Electronic Health Record) systems: The AI software can be integrated with the healthcare provider's EHR system and the healthcare provider can pay for this integration alone.

7.Concept Generation:-

The Concept Generation process for an AI product involves several steps, including:

1. Identifying a need or gap in the market
2. Conducting market research and gathering data on the target audience and their needs
3. Brainstorming and generating a list of potential solutions
4. Evaluating and refining the ideas to determine the best solution
5. Testing and prototyping the solution
6. Launching and commercializing the product.

It's important to constantly gather feedback and make necessary improvements to ensure the product continues to meet the needs and demands of the target audience.

8.Concept Development:-

The AI product being developed is a system that uses advanced deep learning algorithms to predict the presence of cataract disease in a patient by analyzing the images of their left and right eyes. The system will make use of large amounts of data and several deep learning algorithms to train and develop a robust model. The model will be able to accurately identify the presence of cataract and provide real-time diagnoses. This innovative solution will be beneficial to both patients and healthcare providers, as it will provide faster and more accurate diagnoses, reduce the need for invasive diagnostic methods, and ultimately improve patient outcomes and reduce costs. The system will also provide a new approach to treatment, offering new opportunities for innovation in the market. The end goal is to provide a cutting-edge, efficient and effective product that simplifies the diagnostic process, filling the current market gaps and opportunities in cataract diagnosis and treatment.

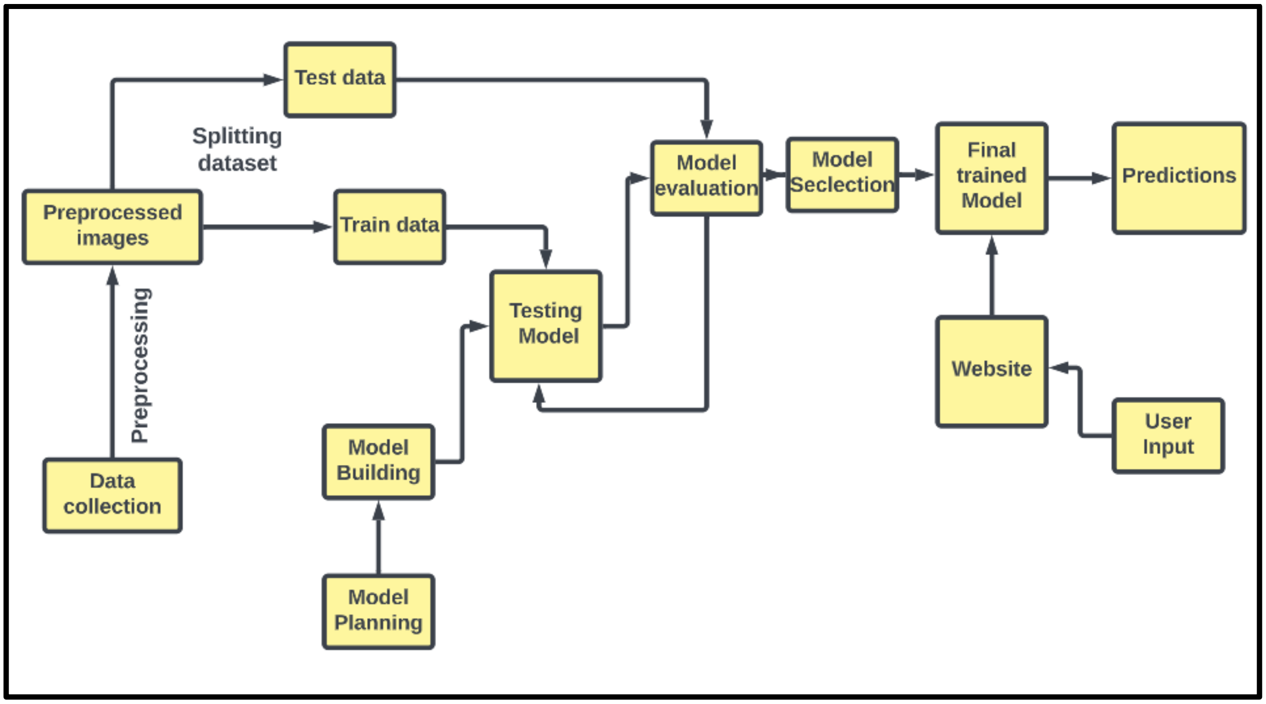
9.Final Product Prototype with Schematic Diagram:-

The Final Product Prototype of the AI product will be a software application that can predict the presence of cataract disease in a patient by analyzing the input scanned images of the left and right eyes. It will have a simple , minimal and a user-friendly interface where the user can upload the eye images. The software then uses deep learning algorithms in the backend to analyze the images and predict the presence of cataract disease. The results are displayed in a clear and concise manner, including a probability score for the prediction.

Key modules or processes-

1. Image Preprocessing - where the images are processed using several methods and prepared for analysis.
2. Model Planning and Building:- Here the model architecture will be built sequentially by adding several convolution,maxPooling layers
3. Model evaluation:- Model will be evaluated based on several performance metrics like accuracy, precision ,False-Negatives
4. Model selection:- I had used CNN and Vgg architectures and after model evaluation, the best model will be selected based on accuracy.
5. Image Upload module - here the user can upload the left and right eye images.
6. Model Prediction- where the deep learning algorithms are applied to the processed images to make predictions.
7. Results Display module - where the predictions are displayed in a clear and concise manner.

Schematic Diagram with detailed module divisions



10. Product details

(i) How does it work?

The product will take input of the scanned images and do some preprocessing on it, so that it will be easier for the model to extract required features from it, later the deep learning models will be applied on the preprocessed images, to predict the presence of the cataract disease.

(ii) Data Sources

Ocular Disease Intelligent Recognition (ODIR) is a structured ophthalmic database of 5,000 patients with age, color fundus photographs from left and right eyes and doctors' diagnostic keywords from doctors.
This dataset is meant to represent a ‘‘real-life’’ set of patient information collected by Shanggong Medical Technology Co., Ltd. from different hospitals/medical centers in China. In these institutions, fundus images are captured by various cameras in the market, such as Canon, Zeiss and Kowa, resulting in varied image resolutions.

Annotations were labeled by trained human readers with quality control management. They classify patient into eight labels including:

- Normal (N),
- Diabetes (D),
- Glaucoma (G),
- Cataract (C),
- Age related Macular Degeneration (A),
- Hypertension (H),
- Pathological Myopia (M),
- Other diseases/abnormalities (O)

For this project, a new dataset has been created using only the suparts of Normal eye (N) and Cataract infected eye (C) from the original ODIR dataset for prediction purposes .

(iii) Algorithms used

a) CNN:-

Convolutional Neural Network (CNN) is a type of deep learning algorithm that is primarily used in computer vision and image classification tasks. It is called a "convolutional" neural network because it uses mathematical operations called convolutions to process the data. The network is composed of multiple layers, including convolutional layers, activation layers, pooling layers, and fully connected layers. The convolutional layers apply filters to the input image data, while the activation layers introduce non-linearities into the network. Pooling layers reduce the spatial dimensions of the data, while fully connected layers make the final predictions. CNNs have proven to be very effective in a variety of image classification tasks, including recognizing objects, faces, and handwritten digits. As prediction of cataract disease involve image classification and feature extraction from images, CNN can be a better option for the AI product.

b) VGG:-

VGG (Visual Geometry Group) is a deep learning model that was developed by researchers at the University of Oxford. It was introduced in 2014 and is considered one of the most influential models in the history of deep learning. The VGG architecture is characterized by its use of multiple convolutional layers, followed by max pooling layers, and then dense layers for classification. The VGG model was trained on the ImageNet dataset, which is a large-scale image recognition dataset that contains over 1 million images and 1000 object categories. The VGG model is known for its high accuracy and is widely used for transfer learning, where the pre-trained model is fine-tuned on a smaller dataset for a specific task. We will be using the pre-trained VGG model to classify the presence of cataract disease in a patient.

(iv) Frameworks used:-

a) Tensorflow:-

It has a large community and provides several pre-trained models and tools for data processing, visualization, and training. Tensorflow can be used to build complex neural network architectures, implement various optimization techniques, and perform deep learning tasks such as image classification, object detection, and natural language processing. It was developed by the Google Brain team and released under the Apache 2.0 open source license. TensorFlow allows developers to create large-scale neural networks with many layers.

b) Keras:-

Keras is a high-level neural network library that runs on top of Tensorflow. It provides a user-friendly interface for building, training, and evaluating neural networks. Keras is designed to be simple and flexible, making it an ideal choice for fast prototyping and experimentation. With its modular architecture, developers can build, reuse, and combine different neural network components to create complex models. I used keras to build the CNN, VGG model, and train them on the processed images and ultimately evaluate them to know its accuracy and precision.

(v)Softwares and Libraries required:-

1. Deep learning frameworks: TensorFlow, Keras, etc.
2. Image processing libraries: OpenCV, scikit-image, etc.
3. Data visualization tools: Matplotlib, Seaborn, Plotly, etc.
4. Jupyter notebooks or IDLE for writing and executing code.
5. Version control software such as Git for maintaining and sharing code among team members.
6. Web Development Tools: HTML, CSS, JavaScript,ReactJS, etc
7. Responsive Web Design Framework: Bootstrap, Foundation, etc

8. Integrated Development Environment (IDE): Visual Studio Code, Sublime Text, etc
9. Server-side scripting language: PHP
10. Database Management System: MySQL

11.Steps for proposed methodology

(i) Importing the data set

The dataset is a structured ophthalmic database of 5,000 patients with age, color fundus photographs from left and right eyes and doctors' diagnostic keywords from doctors. A file is read which contains the paths to the images of left and right eye scans of 5,000 patients. Those preprocessed images are also imported from the dataset folder.

(ii) Combining the cataract data for both eyes

The dataset values are read and the patients having cataract in either of their eyes are combined to create a cataract dataset and extract those values out of the original dataset which contains scans for patients with many other diseases as well.

(iii) Combing the normal eye scan data

The dataset values are read and the patients having no detected disease as analyzed from their eye scans are combined to create a normal dataset.

(iv) Creation of dataset from extracted data

The data for patients having normal eye and cataract patients is combined along with their respective eye scan images to create the final dataset that could be used to train the model.

(v) Splitting the dataset into train and test sets

The final created dataset is then split into training and test sets for model training in the ratio 80:20.

(vi) Implementation of VGG19 model

We use the pre-trained VGG19 model to create our model to predict cataract. The final output layer after the VGG19 model has been implemented uses the sigmoid function for activation. We use the Adam optimizer and binary cross entropy as loss function for compilation of our model. Early stopping is used to stop training when it is observed that the accuracy does not improve after a maximum of 5 epochs. Model checkpoint is also used to save the model at some interval, so the model or weights can be loaded later to continue the training from the state saved.

(vii) Model evaluation

Evaluation of the model has been performed using metrics like classification report and confusion matrix.

12. Team required to develop

The project may need 4 members with expertise in deep learning, computer vision, and medical imaging and web programming would be required. Two will be focussing on the preprocessing, model building and model evaluation. And other two will be focussing on the website building and integrating it to the ML model.

13. What does it cost

There will be some expenses for the following although building the ML model and website can be done using free open source softwares and IDE's mentioned above:-

1. Web Hosting: We may need to purchase web hosting services to host your website online.
2. Server Maintenance: If we have a large website or a large number of visitors, we may need to purchase server maintenance services to keep the website running smoothly and efficiently.
3. Marketing and Advertising: We need to budget for marketing and advertising your website to attract visitors and drive traffic.
   Approximately it may cost 20000-40000 for the AI product

14. Conclusion:-

In conclusion, after comparing the CNN and VGG, I have successfully utilized the pre-trained VGG19 model to distinguish normal eye scans from cataract eye scans with an impressive accuracy of 96%. This model will definitely stand out in the market and is a better fit for the AI product as it has been trained on a large dataset from over 5000 patients. So this proves that there is a potential for using deep learning techniques in the medical field, specifically for the early detection of eye diseases such as cataracts. Further research and experimentation with other models and techniques can be conducted to further improve its accuracy and performance.

Additionally, this project has not only demonstrated the effectiveness of using transfer learning but also highlights the importance of proper data preprocessing and selection of appropriate algorithms and frameworks. The implementation of this model can significantly improve the overall healthcare system, reducing the burden on medical professionals, and providing better outcomes for patients.

14.CODE AND RESULTS

Importing the required libraries and reading the dataset file from Drive folder

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
import cv2
import random
from tqdm import tqdm
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing.image import ImageDataGenerator

import os
for dirname, _, filenames in os.walk('/content/drive/MyDrive/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

Viewing information stored in the dataset

```
df = pd.read_csv("/content/drive/MyDrive/kaggle/input/full_df.csv")
df.head(3)
```

	ID	Patient Age	Patient Sex	Left-Fundus	Right-Fundus	Left-Diagnostic Keywords	Right-Diagnostic Keywords	N	D	G	C	A	H	M	O	filepath	labels	target	filename
0	0	69	Female	0_left.jpg	0_right.jpg	cataract	normal fundus	0	0	0	1	0	0	0	0	../input/ocular-disease-recognition-odir5k/ODI...	['N']	[1, 0, 0, 0, 0, 0, 0, 0]	0_right.jpg
1	1	57	Male	1_left.jpg	1_right.jpg	normal fundus	normal fundus	1	0	0	0	0	0	0	0	../input/ocular-disease-recognition-odir5k/ODI...	['N']	[1, 0, 0, 0, 0, 0, 0, 0]	1_right.jpg
2	2	42	Male	2_left.jpg	2_right.jpg	laser spot，moderate non proliferative retinopathy	moderate non proliferative retinopathy	0	1	0	0	0	0	0	1	../input/ocular-disease-recognition-odir5k/ODI...	['D']	[0, 1, 0, 0, 0, 0, 0, 0]	2_right.jpg

Checking if the labels indicate presence of cataract in left and right eyes separately

```
def has_cataract(text):
    if "cataract" in text:
        return 1
    else:
        return 0
```

```
df["left_cataract"] = df["Left-Diagnostic Keywords"].apply(lambda x: has_cataract(x))
df["right_cataract"] = df["Right-Diagnostic Keywords"].apply(lambda x: has_cataract(x))
```

Dividing and extracting images of left and right eye scans having cataract separately

```
left_cataract = df.loc[(df.C ==1) & (df.left_cataract == 1)]["Left-Fundus"].values
left_cataract[:15]
```

```
array(['0_left.jpg', '81_left.jpg', '103_left.jpg', '119_left.jpg',
       '254_left.jpg', '294_left.jpg', '330_left.jpg', '448_left.jpg',
       '465_left.jpg', '477_left.jpg', '553_left.jpg', '560_left.jpg',
       '594_left.jpg', '611_left.jpg', '625_left.jpg'], dtype=object)
```

```
right_cataract = df.loc[(df.C ==1) & (df.right_cataract == 1)]["Right-Fundus"].values
right_cataract[:15]
```

```
array(['24_right.jpg', '81_right.jpg', '112_right.jpg', '188_right.jpg',
       '218_right.jpg', '345_right.jpg', '354_right.jpg', '477_right.jpg',
       '553_right.jpg', '560_right.jpg', '625_right.jpg', '726_right.jpg',
       '769_right.jpg', '949_right.jpg', '955_right.jpg'], dtype=object)
```

Repeat for normal left and right eye scans

```
left_normal = df.loc[(df.C ==0) & (df["Left-Diagnostic Keywords"] == "normal fundus")]["Left-Fundus"].sample(250,random_state=42).values
right_normal = df.loc[(df.C ==0) & (df["Right-Diagnostic Keywords"] == "normal fundus")]["Right-Fundus"].sample(250,random_state=42).values
right_normal[:15]
```

```
array(['2964_right.jpg', '680_right.jpg', '500_right.jpg',
       '2368_right.jpg', '2820_right.jpg', '2769_right.jpg',
       '2696_right.jpg', '2890_right.jpg', '940_right.jpg',
       '2553_right.jpg', '3371_right.jpg', '3042_right.jpg',
       '919_right.jpg', '3427_right.jpg', '379_right.jpg'], dtype=object)
```

Combine the left and eye scans to create final cataract and normal eye arrays

```
cataract = np.concatenate((left_cataract,right_cataract),axis=0)
normal = np.concatenate((left_normal,right_normal),axis=0)
```

```
print(len(cataract),len(normal))
```

```
594 500
```

Creation of cataract and normal eye image datasets

```
from tensorflow.keras.preprocessing.image import load_img,img_to_array
dataset_dir = "/content/drive/MyDrive/kaggle/input/preprocessed_images"
image_size=224
labels = []
dataset = []
def create_dataset(image_category,label):
    for img in tqdm(image_category):
        image_path = os.path.join(dataset_dir,img)
        try:
            image = cv2.imread(image_path,cv2.IMREAD_COLOR)
            image = cv2.resize(image,(image_size,image_size))
        except:
            continue

        dataset.append([np.array(image),np.array(label)])
    random.shuffle(dataset)
    return dataset
```

```
dataset = create_dataset(cataract,1)
```

```
100%|██████████| 594/594 [01:09<00:00,  8.58it/s]
```

```
len(dataset)
```

```
588
```

```
dataset = create_dataset(normal,0)
```

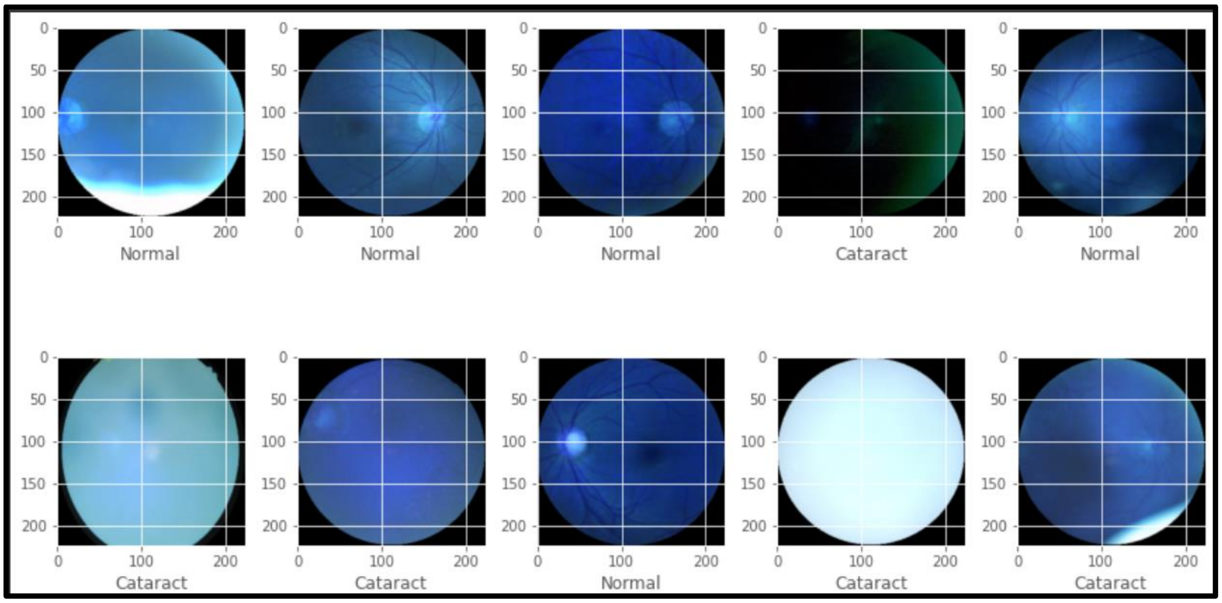
```
100%|██████████| 500/500 [00:03<00:00, 165.04it/s]
```

```
len(dataset)
```

```
1088
```

Displaying dataset images

```
plt.figure(figsize=(12,7))
for i in range(10):
    sample = random.choice(range(len(dataset)))
    image = dataset[sample][0]
    category = dataset[sample][1]
    if category== 0:
        label = "Normal"
    else:
        label = "Cataract"
    plt.subplot(2,5,i+1)
    plt.imshow(image)
    plt.xlabel(label)
plt.tight_layout()
```



Reshaping data to fit into a CNN model

```
x = np.array([i[0] for i in dataset]).reshape(-1,image_size,image_size,3)
y = np.array([i[1] for i in dataset])
```

Splitting into train and test datasets

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.2)
```

Importing VGG19 model

```
from tensorflow.keras.applications.vgg19 import VGG19
vgg = VGG19(weights="imagenet",include_top = False,input_shape=(image_size,image_size,3))
```

```
for layer in vgg.layers:
    layer.trainable = False
```

Creation of a CNN model

```
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Flatten,Dense
model = Sequential()
model.add(vgg)
model.add(Flatten())
model.add(Dense(1,activation="sigmoid"))
```

Summary of the model

```
model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 vgg19 (Functional)          (None, 7, 7, 512)         20024384  
                                                                 
 flatten_1 (Flatten)         (None, 25088)             0         
                                                                 
 dense_1 (Dense)             (None, 1)                 25089     
                                                                 
=================================================================
Total params: 20,049,473
Trainable params: 25,089
Non-trainable params: 20,024,384
_________________________________________________________________
```

Compilation of the model

```
model.compile(optimizer="adam",loss="binary_crossentropy",metrics=["accuracy"])
```

```
from tensorflow.keras.callbacks import ModelCheckpoint,EarlyStopping
checkpoint = ModelCheckpoint("vgg19.h5",monitor="val_acc",verbose=1,save_best_only=True,
                             save_weights_only=False,period=1)
earlystop = EarlyStopping(monitor="val_acc",patience=5,verbose=1)
```

Accuracy score of the model

```
history = model.fit(x_train,y_train,batch_size=32,epochs=15,validation_data=(x_test,y_test),
                    verbose=1,callbacks=[checkpoint,earlystop])
```

```
Epoch 1/15
28/28 [==============================] - ETA: 0s - loss: 1.6295 - accuracy: 0.8839WARNING:tensorflow:Can save best model only with val_acc available, sk
WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
28/28 [==============================] - 7s 217ms/step - loss: 1.6295 - accuracy: 0.8839 - val_loss: 0.9316 - val_accuracy: 0.9404
Epoch 2/15
28/28 [==============================] - ETA: 0s - loss: 0.5910 - accuracy: 0.9402WARNING:tensorflow:Can save best model only with val_acc available, sk
WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
28/28 [==============================] - 6s 210ms/step - loss: 0.5910 - accuracy: 0.9402 - val_loss: 1.0062 - val_accuracy: 0.9450
Epoch 3/15
28/28 [==============================] - ETA: 0s - loss: 0.1351 - accuracy: 0.9747WARNING:tensorflow:Can save best model only with val_acc available, sk
WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
28/28 [==============================] - 6s 212ms/step - loss: 0.1351 - accuracy: 0.9747 - val_loss: 0.8176 - val_accuracy: 0.9495
Epoch 4/15
28/28 [==============================] - ETA: 0s - loss: 0.0392 - accuracy: 0.9931WARNING:tensorflow:Can save best model only with val_acc available, sk
WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
28/28 [==============================] - 6s 214ms/step - loss: 0.0392 - accuracy: 0.9931 - val_loss: 0.2134 - val_accuracy: 0.9587
Epoch 5/15
28/28 [==============================] - ETA: 0s - loss: 0.0056 - accuracy: 0.9977WARNING:tensorflow:Can save best model only with val_acc available, sk
WARNING:tensorflow:Early stopping conditioned on metric `val_acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy
28/28 [==============================] - 6s 216ms/step - loss: 0.0056 - accuracy: 0.9977 - val_loss: 0.2687 - val_accuracy: 0.9541
Epoch 6/15
28/28 [==============================] - ETA: 0s - loss: 0.0011 - accuracy: 1.0000WARNING:tensorflow:Can save best model only with val_acc available, sk
```

Evaluation of model

```
loss,accuracy = model.evaluate(x_test,y_test)
print("loss:",loss)
print("Accuracy:",accuracy)
```

```
7/7 [==============================] - 1s 185ms/step - loss: 0.4058 - accuracy: 0.9633
loss: 0.4057959318161011
Accuracy: 0.963302731513977
```

Classification report

```
print(classification_report(y_test,y_pred))
```

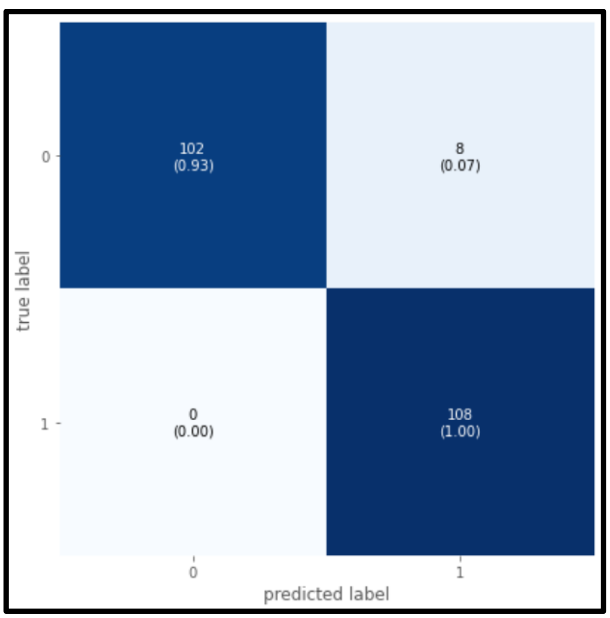
```
              precision    recall  f1-score   support

           0       1.00      0.93      0.96       110
           1       0.93      1.00      0.96       108

    accuracy                           0.96       218
   macro avg       0.97      0.96      0.96       218
weighted avg       0.97      0.96      0.96       218
```

Confusion matrix

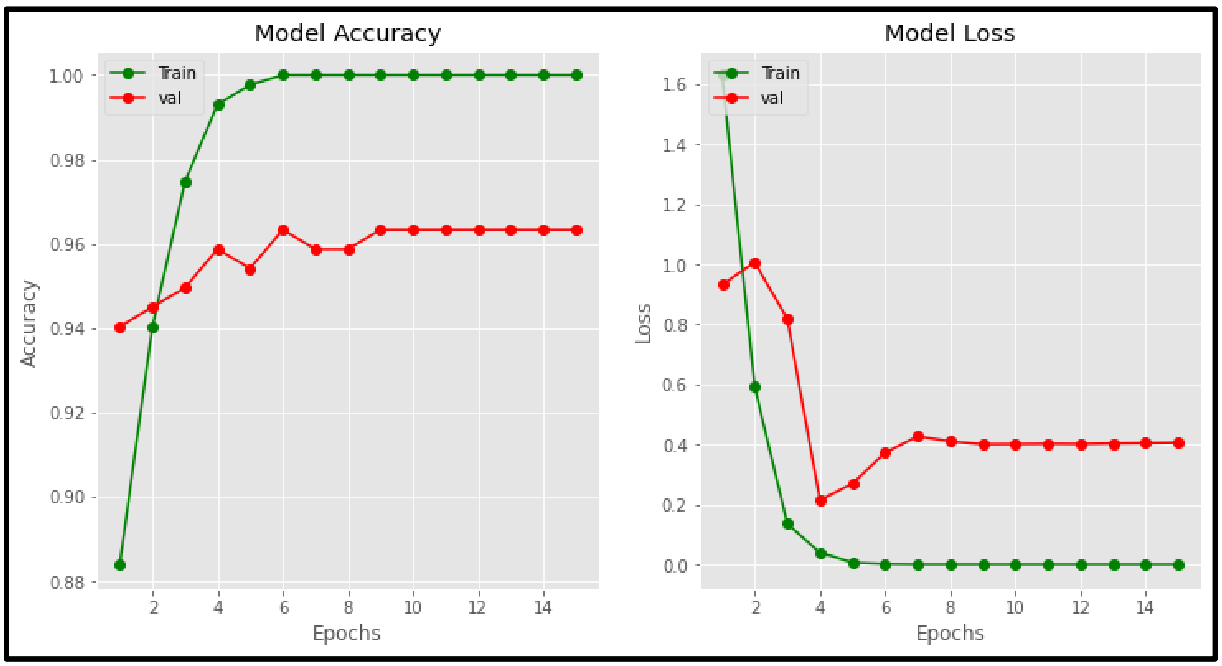
```
from mlxtend.plotting import plot_confusion_matrix
cm = confusion_matrix(y_test,y_pred)
labels = ["Normal","Cataract"]
plot_confusion_matrix( cm,labels,show_normed = True,figsize=(8,7));
```



Analyzing model accuracy and model loss

```
plt.style.use("ggplot")
fig = plt.figure(figsize=(12,6))
epochs = range(1,16)
plt.subplot(1,2,1)
plt.plot(epochs,history.history["accuracy"],"go-")
plt.plot(epochs,history.history["val_accuracy"],"ro-")
plt.title("Model Accuracy")
plt.xlabel("Epochs")
plt.ylabel("Accuracy")
plt.legend(["Train","val"],loc = "upper left")

plt.subplot(1,2,2)
plt.plot(epochs,history.history["loss"],"go-")
plt.plot(epochs,history.history["val_loss"],"ro-")
plt.title("Model Loss")
plt.xlabel("Epochs")
plt.ylabel("Loss")
plt.legend(["Train","val"],loc = "upper left")
plt.show()
```



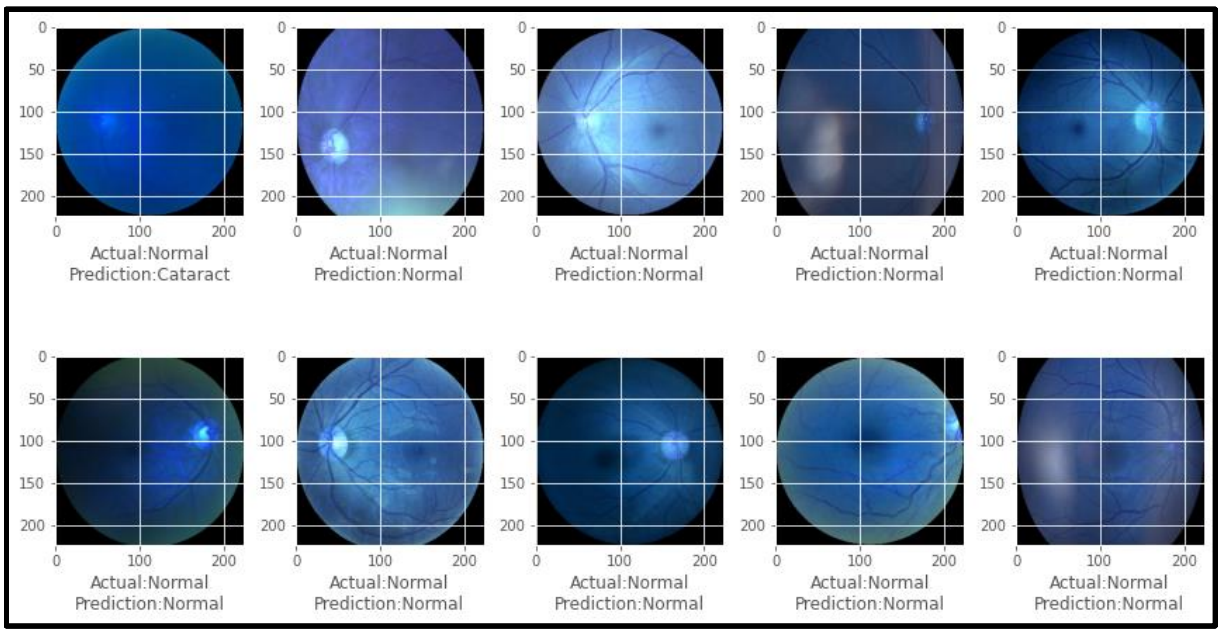
Displaying images with predicted labels

```
plt.figure(figsize=(12,7))
for i in range(10):
    sample = random.choice(range(len(x_test)))
    image = x_test[sample]
    category = y_test[sample]
    pred_category = y_pred[sample]

    if category== 0:
        label = "Normal"
    else:
        label = "Cataract"

    if pred_category== 0:
        pred_label = "Normal"
    else:
        pred_label = "Cataract"

    plt.subplot(2,5,i+1)
    plt.imshow(image)
    plt.xlabel("Actual:{}\nPrediction:{}".format(label,pred_label))
plt.tight_layout()
```



Reference links:-

Average Age for Cataract Surgery | Pacific Eye Institute | Upland.

Outcome and number of cataract surgeries in India: policy issues for blindness control.

Cataracts | National Eye Institute

Cataract Disease Detection by Using Transfer Learning-Based Intelligent Methods

The dataset was acquired from Kaggle:

https://www.kaggle.com/datasets/andrewmvd/ocular-disease-recognition-odir5k

Github link for the code implementation:-

https://github.com/ladijeevansai/Catract-disease-Prediction-using-Deep-learning