

Visual Analysis of Prediction of Glaucoma using Machine Learning

Karanam Santoshachandra Rao , Asst Professor, Dept of CSE
Centurion University Of Technology And Management, Odhisa, India

Alugolu Avinash , Asst Professor, Dept of CSE
Centurion University Of Technology And Management, Andhra Pradesh, India

Vysyaraju Sai Sirisha, Regno: 170101120044 , Dept of CSE
Centurion University Of Technology And Management, Odhisa, India

Deverchetty Sriya, Regno: 170101120035, Dept of CSE
Centurion University Of Technology And Management, Odhisa, India

Alibilli Mahesh, Regno: 170101120032, Dept of CSE
Centurion University Of Technology And Management, Odhisa, India

Sahil Subudhi, Regno: 170101120073, Dept of CSE
Centurion University Of Technology And Management, Odhisa, India

Abstract

Glaucoma is a pathological condition of optic nerve damage and is second leading cause of vision loss. It is known as silent thief of sight^[1]. The project aimed to develop machine learning models that have strong prediction power and interpretability for diagnosis of glaucoma. This project consists of six main steps, they are data gathering, data pre-processing, feature extraction, building model, model evaluation and predicting the result. We used retinal images dataset and performed support vector machine algorithm and gained 94.4% accuracy. In this paper we have visualized prediction of glaucoma using machine learning visualization techniques like matplotlib, seaborn and plotly based on some constraints like 1) sample dataset, 2) Histogram equalization of dataset, 3) Training and testing data, 4) Classification of training data, 5) Feature extraction of retinal images using sift and surf 6) A heat map which visualizes the confusion matrix of result, 7) Comparison of test and predicted values of test images, 8) Comparing amount of glaucoma images in tested and predicted data, 9) Comparing amount of healthy images in tested and predicted data.

Keywords: Glaucoma, Machine Learning, Visualization, Support Vector Machine, Python, Plotly, Matplotlib, Packages, Numpy, Seaborn, opencv, image processing

Introduction

Glaucoma is a group of diseases that damages the eye's optic nerve and can result in vision loss and blindness. Glaucoma is a build-up of pressure within the eye that causes damage to the optic nerve. It is a disease of the eye in which fluid pressure within the eye rises, if left untreated, the patient may lose vision and even become blind. The increased pressure in your eye, called intraocular pressure, can damage your optic nerve which sends images to your brain. Glaucoma is generally described as loss of retinal ganglion cells. This loss is routinely diagnosed by the presence of thinning of optic nerve neuroretinal nerve^[2].

Data pre-processing, feature extraction, building model and model evaluation are based on python's best and top libraries like numpy, pandas, cv2 and os, machine learning techniques like histogram equalization, sift and surf feature extraction technique and machine learning algorithm like SVM. Generally, Python is high-level general-purpose programming and an interpreted language. Developed by Guido van Rossum and primarily released in 1991, Python's design intention emphasizes code readability & understanding with its easy use of significant whitespace or indentation. Its object-oriented approach targets to help developers write unambiguous, very logical code for tiny and large-scale projects.

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

Analysis of mentioned 9 points is based on data visualization techniques like matplotlib, seaborn and plotly. Data visualization is the presentation of data in graphical format. It helps people understand the significance of data by summarizing and presenting huge amount of data in a simple and easy to understand formats and helps communicate information clearly and effectively.

The main motto of this paper is to analyse above mentioned 9 points and present that information in the form that makes sense to people. Various techniques have been used for presenting data visually with the help of python libraries namely Matplotlib, Seaborn, and Plotly.

Following python library code is required to import primarily to analyse the data with the help of various function and methods.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objs as go
import plotly.express as px
import cv2
import os
from sklearn import svm
from sklearn import metrics
from sklearn.metrics import confusion_matrix
```

Understanding the dataset

Here, we have considered "retinal image dataset" for prediction. Basic idea on dataset is given by following python code.

Step1: - Reading images from the directory

```
images1 = []
for img_path in glob.glob('C:/Users/user/Desktop/ml pro/dr/new/*.*'):
    images1.append(mpimg.imread(img_path,1))
```

1. Plotting the sample data set

These plotting's can be done using many machine learning techniques. With the help of above mentioned images1 list we can make a subplot of sample dataset Fig1 is its visualization^[3].

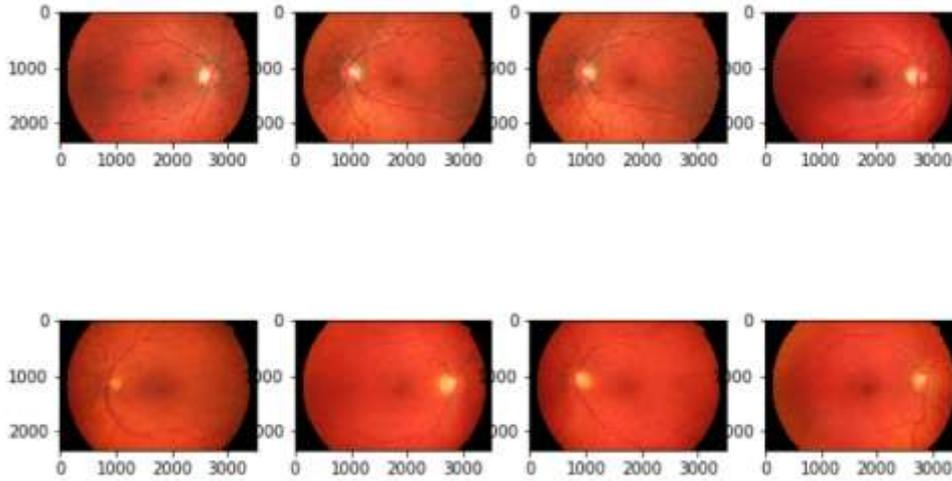


Fig 1: Sample dataset

As per the Fig 1 visualization it has been observed that in this project, we have used retinal image datasets for prediction.

Step2: - Data pre-processing using histogram equalization.

In this we pre-processed our data using histogram equalization. Here quality of image will be increased so that prediction on these images gives good result.

2. Histogram equalization of dataset

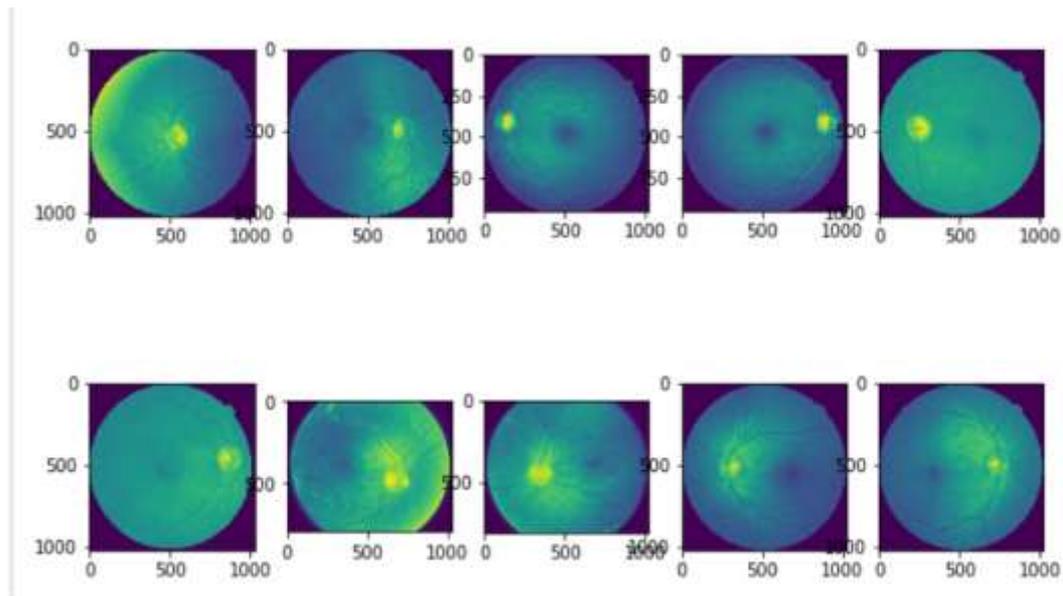


Fig 2: Histogram equalization of dataset

As per Fig 2 visualization it has been observed that quality of images increased and useful for better prediction.

Step3: -Classifying the dataset.

As we used supervised learning algorithm, we need a labelled data for prediction. In this step we have done binary classification where 0 represents healthy eye and 1 represents glaucomatous eye. These images are stored in different folders and this data was further split into training data and testing data.

3. Training and testing data

Here we splitted our data into 76:23 ratio approximately. This ratio indicated 76% trained data and 23% tested data. To plot this, we have used plotly tool which gives us interactive graph^[4].

Code: -

```
import plotly.graph_objects as go
```

```
labels = ['trained data','tested data']
pieData = [healthy+glaucoma,len(X_test)]
values = pieData
```

```
fig= go.Pie(labels=labels, values=values,hole=0.3)
layout = go.Layout(title = 'Trained and tested data',
```

```
font=dict(
family="Times New Roman",
size=18,
color="black"))
```

```
fig = go.Figure(data=fig,layout=layout)
```

```
fig.show()
```

Training and testing data

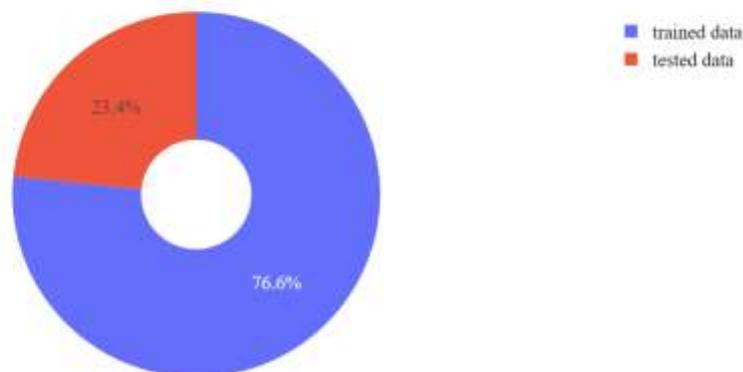


Fig 3: Training and testing data

Fig 3, shows the percentage of training and testing data. By this figure it has been observed that training data was 76.6% and testing data was 23.4%.

4. Classification of Training data

As mentioned above we have done binary classification for prediction. This graph shows the amount of healthy images and glaucomatous images in training data,

Code: -

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt;
plt.figure(figsize=(8,6))
x=['healthy images','glaucoma images']
y=[len(healthy),len(glaucoma)]
colors=['darkviolet','darkred']
rects1=plt.bar(x,y,0.75,color=colors)
def autolabel(rects):

    for rect in rects:

        height = rect.get_height()
        plt.text(rect.get_x() + rect.get_width()/2., height,
                '%d' % int(height),
                ha='center', va='bottom',fontweight='bold',fontsize=15)
autolabel(rects1)
plt.xlabel("Type of Images data",fontsize=20,fontweight='bold')
plt.ylabel("count",fontsize=20,fontweight='bold')
plt.xticks(x,color='k',fontsize=16,fontweight='bold')
plt.yticks(np.arange(0,70,10),fontsize=16)
plt.title("Classification of training data",fontsize=25,fontweight='bold',color='k')
plt.show()
```

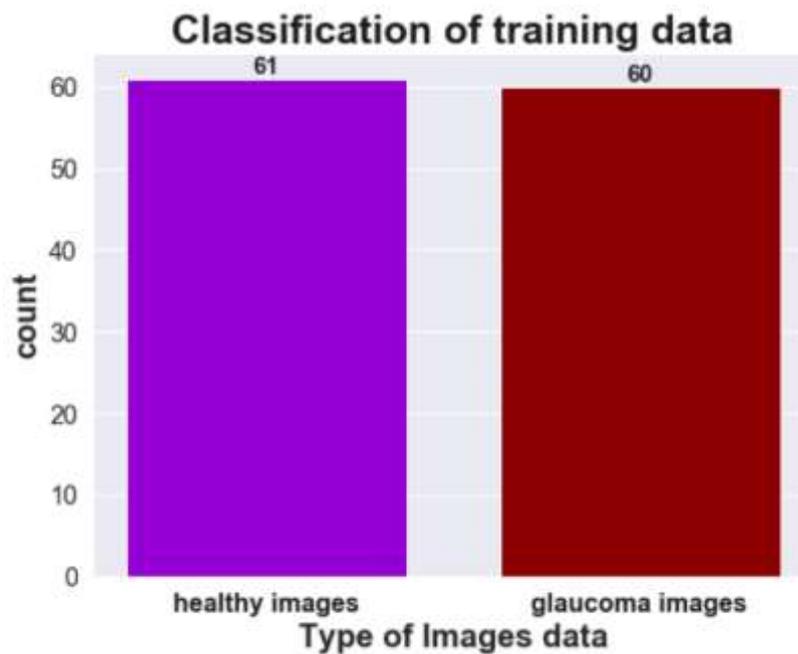


Fig 4: Classification of training data

In Fig 4, it has been observed that training data contains 61 healthy images and 60 glaucoma images.

Step 4: - Feature extraction

In this step we have to extract features of each training data image to train the machine. To extract feature from image we used opencv techniques called sift and surf.

The scale-invariant feature transform (SIFT) is a feature detection algorithm in computer vision to detect and describe local features in images. Speeded up robust features (SURF) is a patented local feature detector and descriptor. It can be used for tasks such as object recognition, image registration, classification, or 3D reconstruction. It is partly inspired by the scale-invariant feature transform (SIFT) descriptor^[5].

5. Feature Extraction using sift and surf

To plot this graph, we have imported opencv and performed some image processing techniques like sift and surf to extract features of image.

Code: -

```
import numpy as np
import cv2
from matplotlib import pyplot as plt
img = cv2.imread('image.jpg',0)
surf = cv2.xfeatures2d.SURF_create()
keypoints_surf, descriptors_surf = surf.detectAndCompute(img, None)

sift = cv2.xfeatures2d.SIFT_create()
```

```
keypoints_sift, descriptors_sift = sift.detectAndCompute(img, None)
kp=np.hstack([keypoints_surf, keypoints_sift])
img2 = cv2.drawKeypoints(img,kp,img, flags=0)
cv2.imwrite("abcd.jpg",img2)
plt.imshow(img2),plt.show()
```

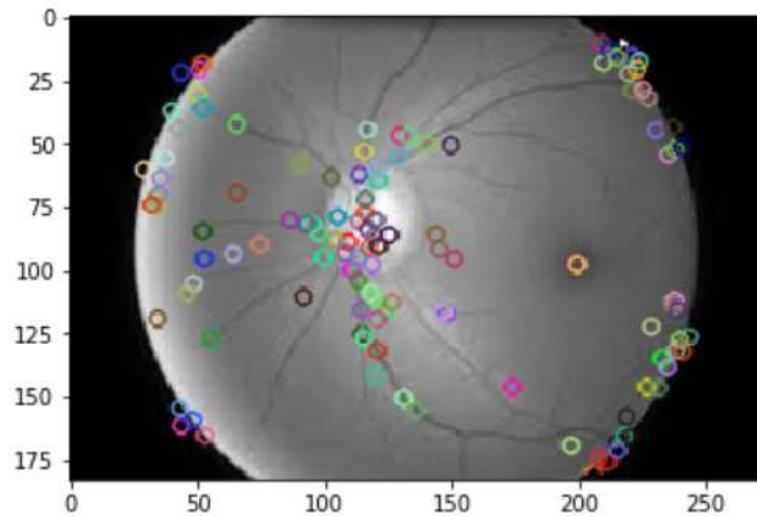


Fig 5: Feature Extraction using sift and surf

In Fig 5, it has been observed that the circles are the features extracted from the retinal image.

In this way we extracted features of all training dataset images and stored. Using these features, we trained the machine and predicted result for test images.

Step 5: - Building the model

The training data will be used to build and analyse the model. The logic of the model is based on the support vector machine algorithm. This algorithm is trained using feature extracted from training dataset images^[6].

Step 6: - Evaluating the model

After building a model by using the training data set, it is finally time to put the model to a test. The testing data set is used to check the efficiency of the model and how accurately it can predict the outcome. The prediction is evaluated for accuracy and if the accuracy is acceptable, the Machine Learning algorithm is deployed. If the accuracy is not acceptable, the Machine Learning algorithm is trained again and again with an augmented training data set. Methods like parameter tuning and cross-validation can be used to improve the performance of the model.

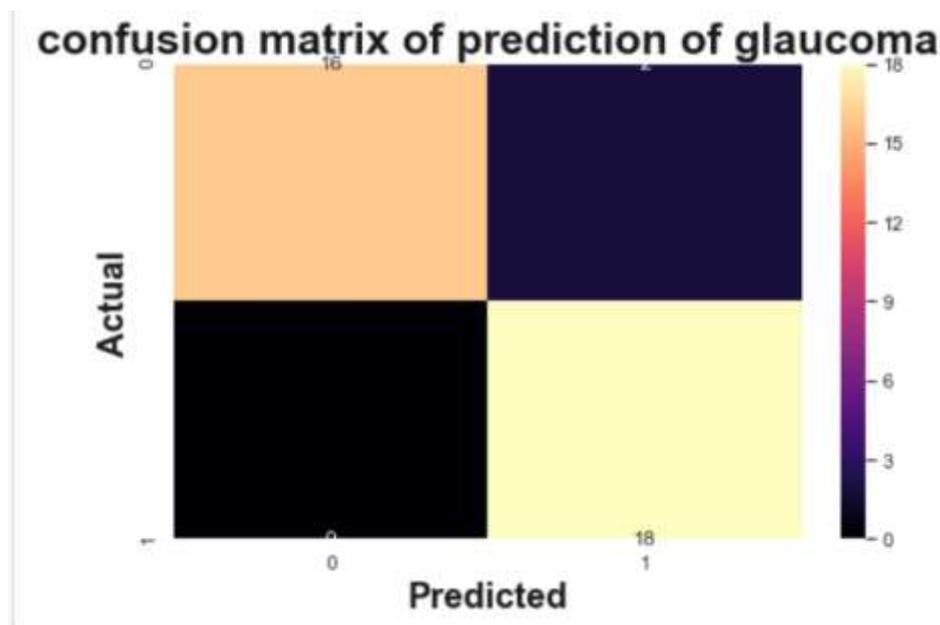


Fig 7: confusion matrix of prediction of glaucoma

In Fig 7, it has been observed the performance of the model and the plot describes that only 2 predictions are predicted wrong out of all predictions.

7. Comparison of test and predicted values of tested images

In this plot we compared the test values and predicted values of test images.

Code: -

```

b1 = go.Bar(
    x = x,
    y = ytest,
    name = "tested data",
    marker = dict(color = 'purple',
        line=dict(color='rgb(0,0,0)',width=1.5)))

b2 = go.Bar(
    x = x,
    y = ypred,
    name = 'predicted data',
    marker = dict(color = 'crimson',
        line=dict(color='rgb(0,0,0)',width=1.5)))

data = [b1, b2]
layout = go.Layout(barmode = "group",title = 'Tested and predicted
data',width=1000, height=500,
    xaxis = dict(title= 'Images',ticklen=5,tickangle=-50),
    yaxis = dict(title= 'tested and predicted values',ticklen= 5),
    font=dict(
        family="Aries",

```

```

size=15,
color="black"))
fig = go.Figure(data = data, layout = layout)
fig.show()
    
```

Comparing tested and predicted data

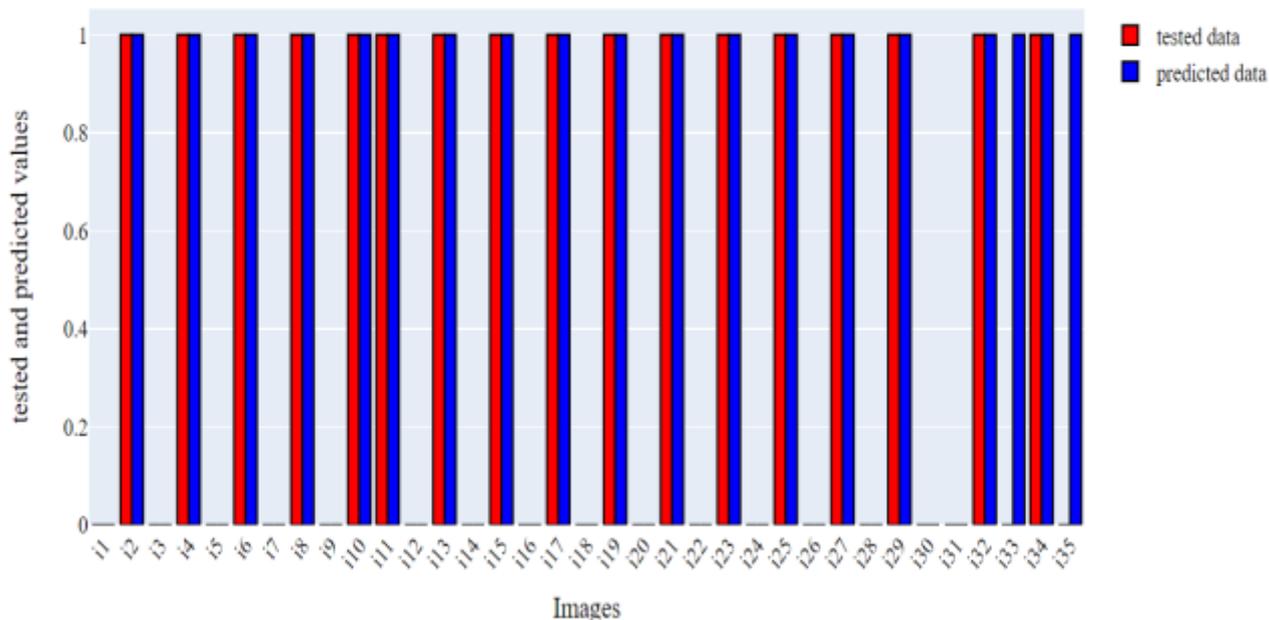


Fig 8: Comparing test and predicted values of tested images

In Fig 8, it has been observed that almost all images categorical values are correctly predicted by our model. Only images 33 and 35 are wrongly predicted. Due to this we got 94.4% accuracy.

8. Comparing amount of glaucomatous images in tested and predicted data

In this graph we visualized the percentage of glaucomatous images present in test and predicted data.

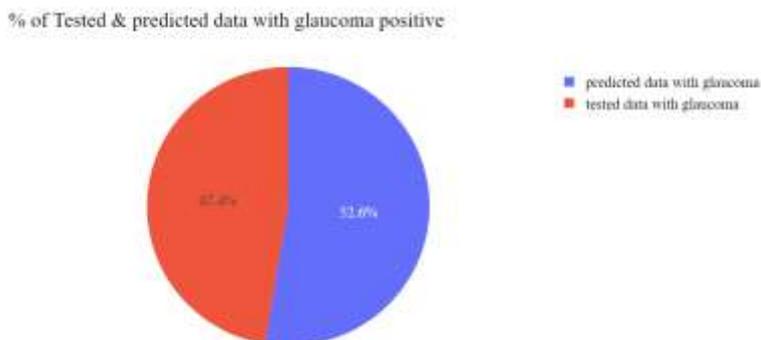


Fig 9: Amount of glaucomatous images in test and predicted data

In Fig 9 it has been observed that tested data contains 47.4% glaucoma positive images and predicted data contains 52.6% glaucoma positive images. By the above graph it has been concluded that 5% of glaucoma positive images data was wrongly predicted by our model. As 5% glaucomatous images are predicted wrongly, we should get same 5% wrong prediction in healthy images^[7].

9. Comparing amount of healthy images in tested and predicted data

In this graph we visualized the percentage of healthy images present in test and predicted data^{[8][9]}.

% of Tested & predicted data with glaucoma negative

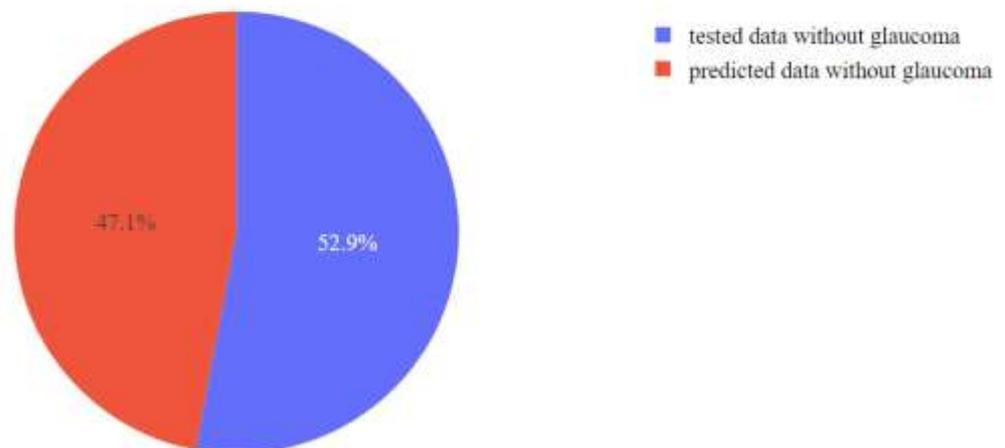


Fig 10: Amount of healthy images present in test and predicted data

In Fig 10 it has been observed that tested data contains 47.1% healthy images and predicted data contains 52.9% healthy images. It has been concluded that 5% of healthy images prediction is wrong^[10].

Conclusion

This paper presents the visual analysis of prediction of glaucoma using supervised machine learning algorithm called support vector machine on retinal images dataset where we gained 94.4% accuracy. The graphs which are visualized in this paper shows sample dataset which we used, data after pre-processing, percentage of trained and tested data, trained data after classification, feature extraction of retinal image, heat map which describes confusion matrix of our model, comparison of trained and tested data, to check amount of healthy and glaucomatous image in test and predicted data.

References

1. <https://www.webmd.com/eye-health/glaucoma-eyes#1>
2. <https://www.glaucoma.org/glaucoma/>
3. Development of machine learning models for diagnosis of glaucoma. Seong Jae Kiml, Kyong Jin Cho, Sejong
4. Review of Machine Learning Techniques for Glaucoma Detection and Prediction, Conference Paper · August 2014

5. Glaucoma screening pipeline based on hidden features-2019
6. <https://www.simplilearn.com/data-visualization-in-python-using-matplotlib-tutorial>
7. <https://robertmitchellv.com/blog-bar-chart-annotations-pandas-mpl.html>
8. <https://www.kaggle.com/kanncaa1/plotly-tutorial-for-beginners>
9. <https://www.kdnuggets.com/2019/08/advanced-data-visualisation-data-scientists.html>
10. Kourchi Adil, Lahcen Oughdir, F. Messaoudi “Successful machine learning project”, Article in International Journal of Scientific and Engineering Research · September 2019