GLAUCOMA DETECTION BASED ON DEEP NEURAL NETWORKS

Team Members:-
16071A0519 : Yashna Gutha
16071A0528 : Loka Ruthvika Reddy
16071A0532 : Madhav Kode
16071A0554 : Sri Laasya Garapati

Guide:-
Mr. M. Ravikanth
Assistant Professor
CSE

CONTENTS

AGENDA

- We have seen the existing systems, their drawbacks and how our proposed system can overcome those making use of both functional and non-functional requirements. The system architecture that has been implemented is seen along with the UML diagrams.

- The agenda set for today is to understand the implementation of the training code to get required results and validate the test cases. We would also see how this could be developed more in future to improvise the method of detection.

INTRODUCTION

- Glaucoma is an ocular disease. Glaucoma can lead to blindness if it is not detected and treated in proper time.
- The proper diagnosis of glaucoma is key due to the fact that it occurs without symptoms and can lead to irreversible loss of sight.
- The risk of blindness depends on the intraocular pressure, the severity of the disease, the age of glaucoma onset, as well as family history of this disease
- It is the second leading cause of blindness worldwide, affecting approximately 6.7 million people
- The main objective of this project is to analyze and diagnose the glaucoma using digital fundus images.

ABSTRACT

- Glaucoma is a disease caused due to neuro degeneration of the optic nerve which leads to blindness.
- It has been nicknamed as the “Silent Thief of Sight”.
- It can be evaluated by monitoring intra ocular pressure (IOP), visual field and the optic disc appearance (cup-to-disc ratio). Glaucoma increases the cup to disc ratio (CDR), affecting the peripheral vision loss.
- In this project, we develop a deep learning (DL) architecture with convolutional neural network for automated glaucoma diagnosis.
- The algorithms are tested on publicly available fundus images and the results are compared.

LITERATURE SURVEY

S.NO	TITLE	AUTHOR NAMES AND YEAR	METHODOLOGY	PERFORMANCE
1	Automated Glaucoma Detection using Center Slice of Higher Order Statistics	Rahul Sharmaa,Pradip Sircar, R.B.Pachori, Sulatha V. Bhandary,U. Rajendra Acharya(2019)	Locality sensitive discriminant analysis (LSDA) data reduction technique method is implemented. Features are fed to support vector machine (SVM) classifier.	Used SVM,which is effective in high dimensional spaces and memory efficientDrawback is that the Radon transform does not preserve the phase information. RT further increases the mathematical complexity and introduce error.Achieved classification accuracy of 98.8% and 95% using entire private and public databases respectively.
2	Automatic Glaucoma Detection Method Applying a Statistical Approach to Fundus Images	Anindita Septiarini,Dyna M. Khairina,MKom, Awang H. Kridalaksana, MKom,Hamdani Hamdani(2018)	Used statistical features and the k-nearest neighbor algorithm as the classifier	Feature selection tries to eliminate candidate features that are irrelevant, thereby decreasing the complexity of the model.Feature selection does not guarantee improved performance.Achieved 95.24% accuracy.
3	Using hemorrhage detection through ROI segmentation	Mrs. Pavithra G,Anushree G,Dr. T.C.Manjunath,Dr. Dharmanna Lamani(2017)	Extract blood vessels through green channel using gabor filter	Accuracy of 93% achieved on dataset of 140 images
4	Automatic Glaucoma Detection by Using Funduscopic Images	Atheesan S., Yashothara S(2016)	Digital image processing is used.Glaucoma is identified through cup to disc ratio (CDR) cal and by the orientation of the blood vessels	Poor image quality may degrade the performance. Maintained F-score value about 96%

LITERATURE SURVEY

S.NO	TITLE	AUTHOR NAMES AND YEAR	METHODOLOGY	PERFORMANCE
5	Glaucoma Diagnosis with Machine Learning Based on Optical Coherence Tomography and Color Fundus Images	Guangzhou An,Kazuko Omodaka,Kazuki Hashimoto(2019)	Transfer learning of convolutional neural network (CNN) was used . Random Forest classifier was used.	Proposed CNNs do not use the disc center detection, their classification performance would be unaffected by a failure to detect the centers of disc and fovea . This study include a relatively small study population, which may have affected the statistical power of analyses . Achieved 10-fold CV AUC of 0.963
6	optic cup feature analysis using fundus images	Andres diaz Sandra morales (2016)	Used ROI localization followed by CMYK color space analysis. Cup segmentation done using stochastic watershed segmentation	0.81 specificity using 53 images dataset
7	detection of glaucoma through color fundus images	Zailing chen Rhongchang zho Shuo li (2017)	By analyzing heterogenous portion of peri-papillary-atrophy area	88% accuracy achieved on 100 images
8	Glaucoma analysis using fundus images through RNFL texture analysis (2017)	Somasis roy Xiao loui (2019)	the loss of RNFL thickness is considered as the main parameter, achieved using image segmentation	650 images dataset with an accuracy of 84%

PROPOSED SYSTEM

- In our system, we will develop a Deep Learning architecture with CNN working at its core for automating the detection of glaucoma.
- Deep Learning systems, such as Convolutional Neural Networks can deduce a hierarchical representation of images to discriminate between Glaucoma and non-Glaucoma patterns for diagnostic decisions.
- Different deep learning architectures like ResNet50 will be used for image classification tasks.
- Our model which is trained using convolutional neural network will be more efficient for the diagnosis of glaucoma than previously published models.
- CNNs eliminate the need for manual feature extraction, so you do not need to identify features used to classify images. This automated feature extraction makes deep learning models highly accurate for computer vision tasks such as object classification.

SYSTEM REQUIREMENTS

Functional requirements:

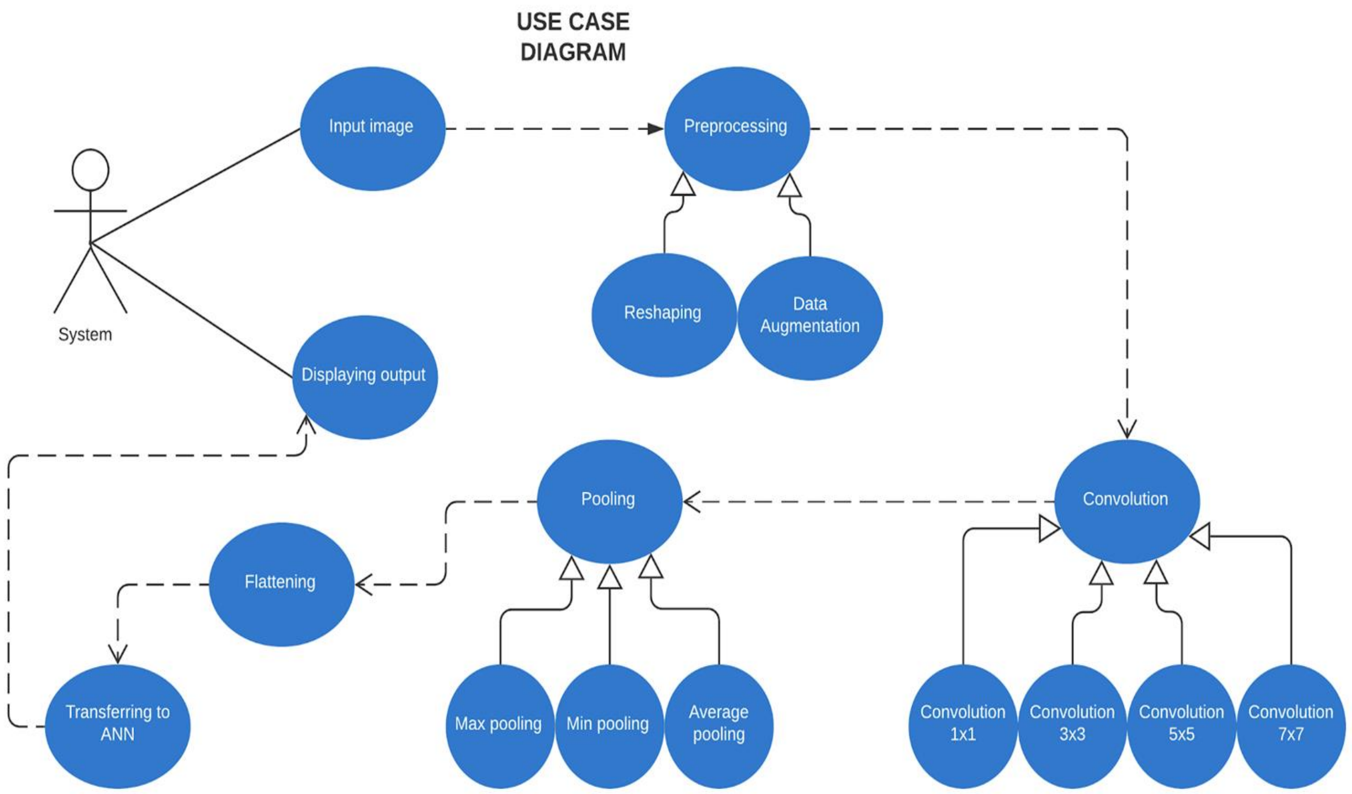
- User should be able to check the health status of his/her eye.
- To diagnose correctly on the image whether it is Glaucoma or not and if yes, measuring the severity of the effect using a deep learning model.
- System should summarize the results to users with accuracy.

SYSTEM REQUIREMENTS

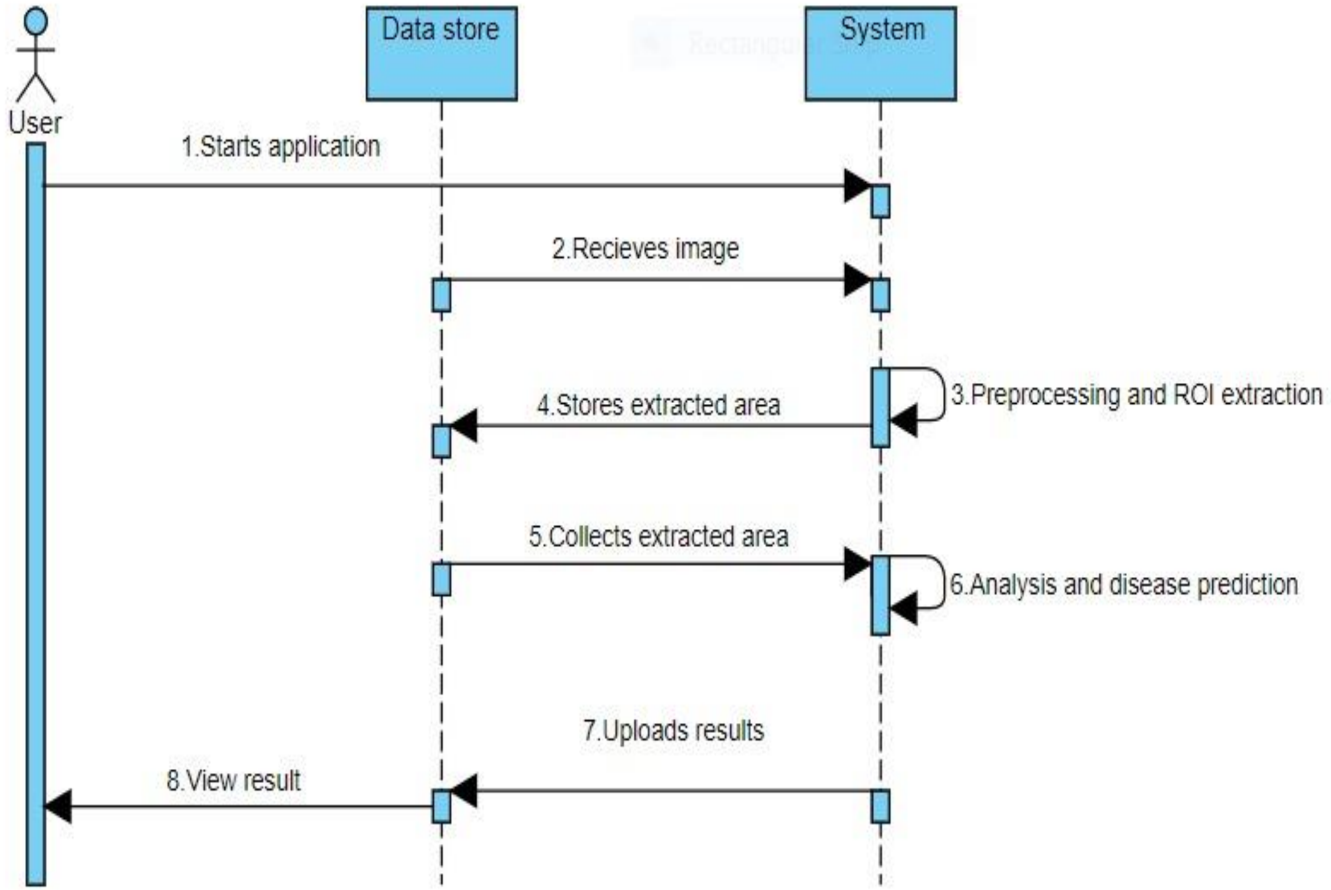
<u>Non-Functional requirements</u>**:**

- Accuracy: The system need to be accurate to predict the correct result
- Precision: The result should not be ambiguous.
- Usability: The system should not be complicated for the user to use. It must give best experience to user while using and managing the application.
- Stability: The system must function consistently over a period of time without getting failed or breakdown
- Scalability: The system should meet the increasing need of users and surrounding environment
- Efficiency: The system should handle capacity, throughput and response time adroitly.
- Performance: The system should function fast and accurately.

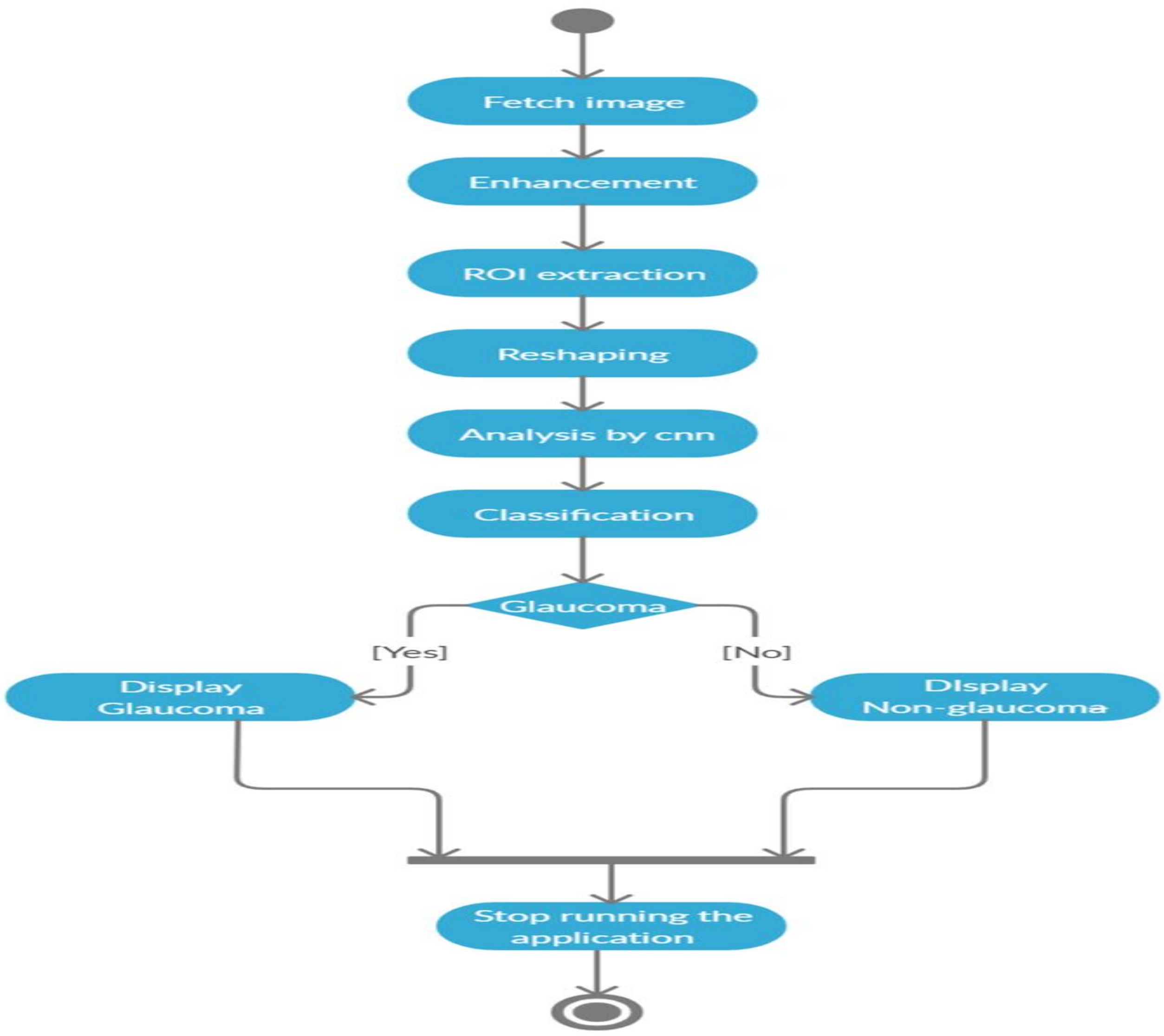
USE CASE DIAGRAM



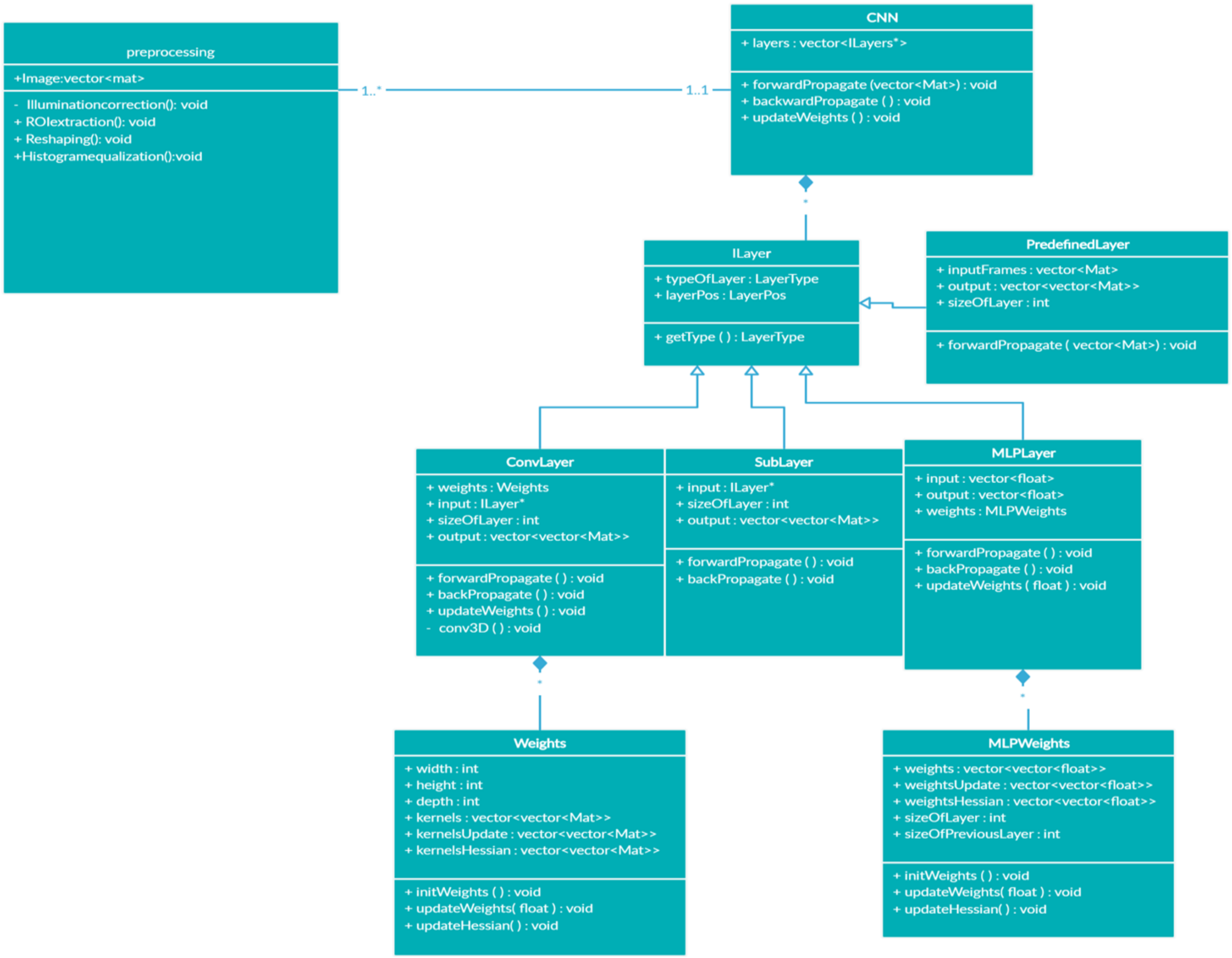
SEQUENCE DIAGRAM



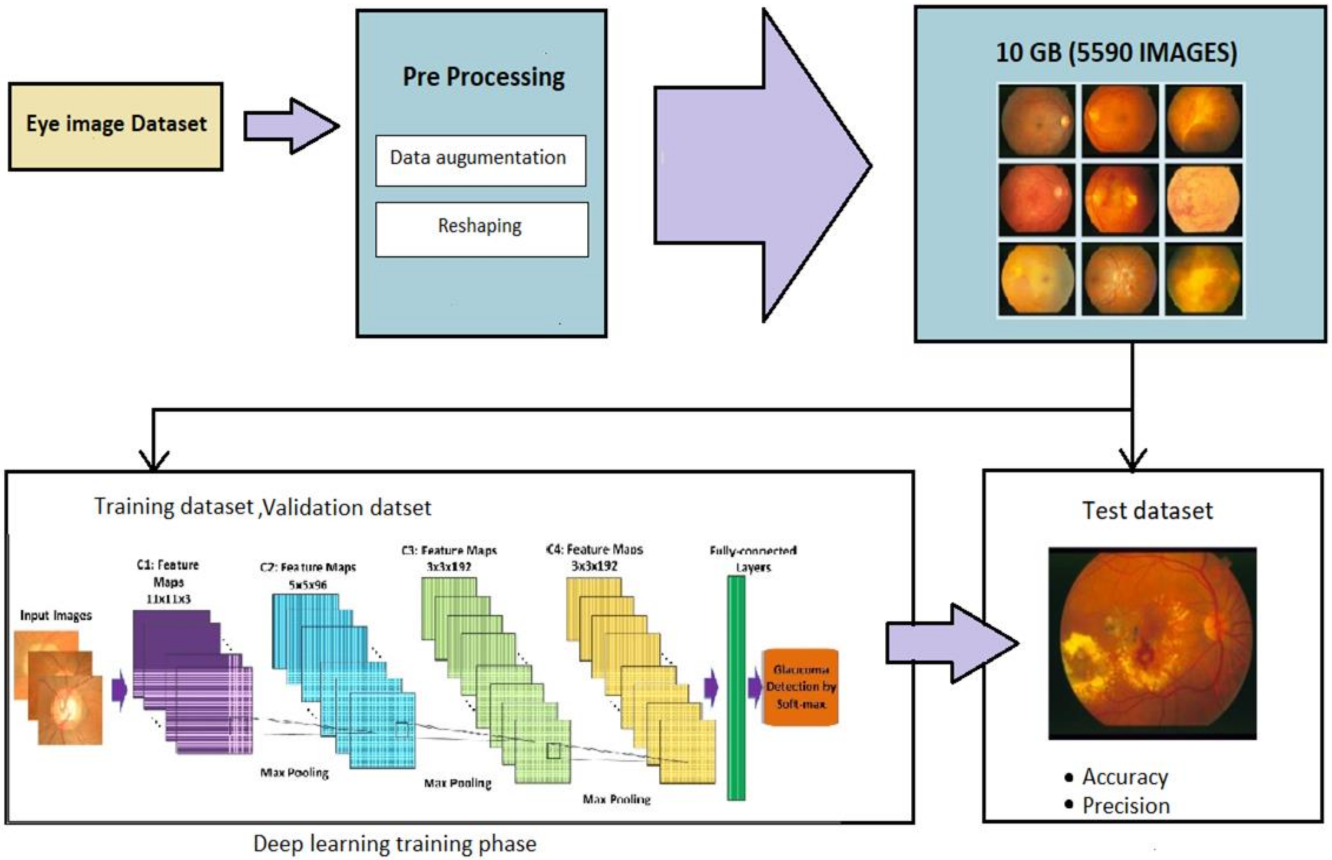
ACTIVITY DIAGRAM



CLASS DIAGRAM

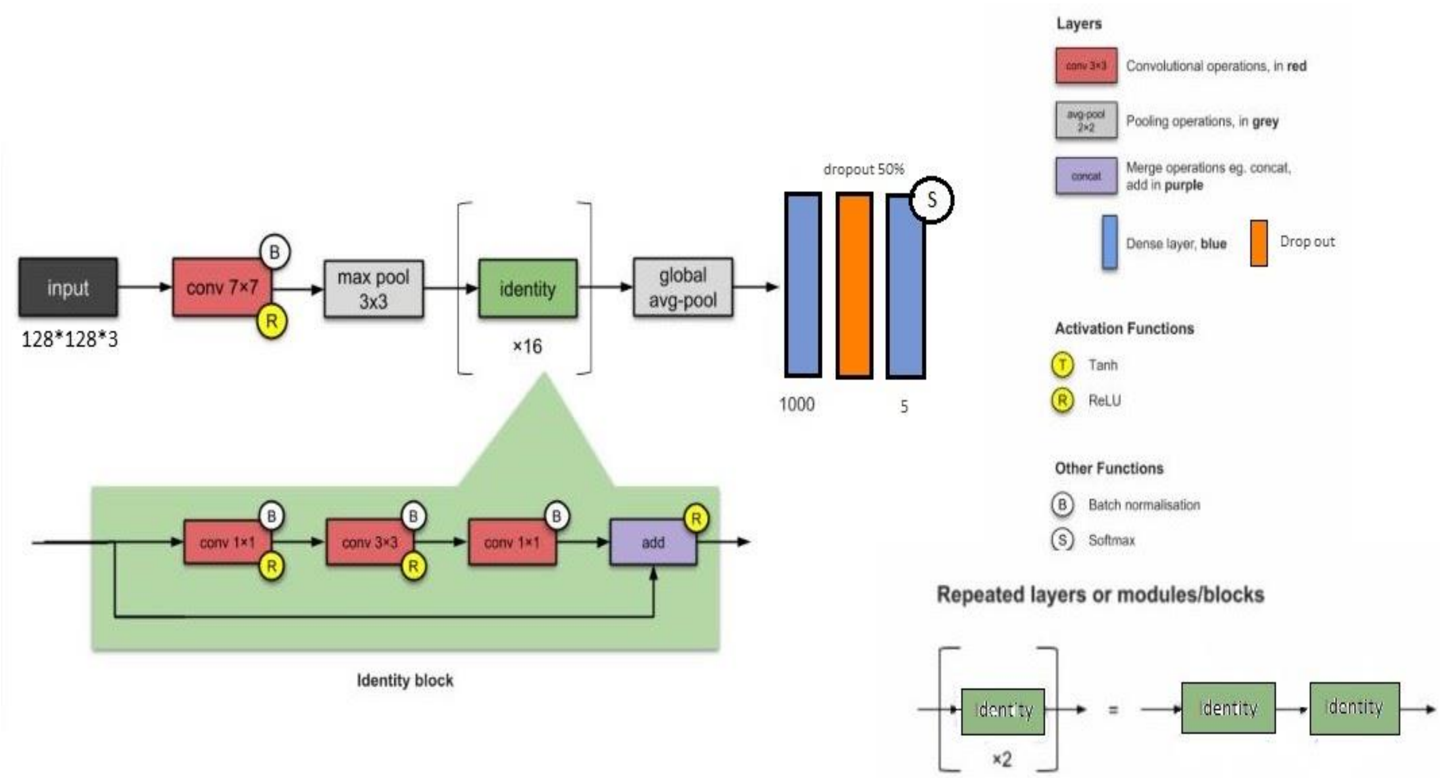


SYSTEM ARCHITECTURE



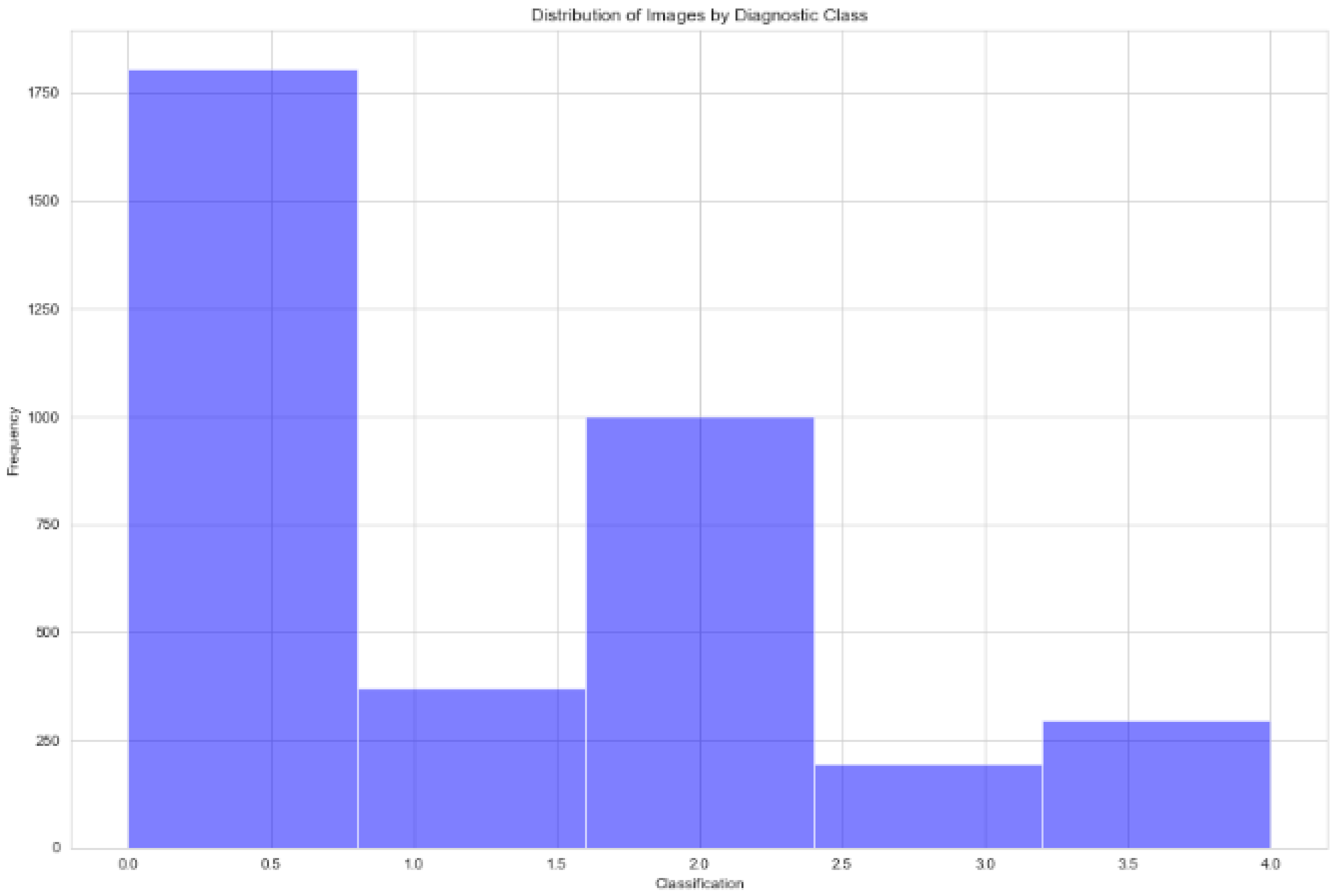
SYSTEM ARCHITECTURE

CNN Architecture



IMPLEMENTATION SCREENSHOTS

Data Distribution by diagnostic class (0-4)



IMPLEMENTATION SCREENSHOTS

ResNet Model

+ Code + Text

```
# instantiate and compile a model
model = create_model(input_shape=(HEIGHT, WIDTH, COLORS), n_out=N_CLASSES)

for layer in model.layers:
    layer.trainable = False

for i in range(-5, 0):
    model.layers[i].trainable = True

metric_list = ["accuracy"]
optimizer = optimizers.Adam(lr=WARMUP_LEARNING_RATE)
model.compile(optimizer=optimizer, loss="categorical_crossentropy",  metrics=metric_list)
model.summary()
```

```
________________________________________________________________________________
res5b_branch2a (Conv2D)         (None, 4, 4, 512)    1049088     activation_92[0][0]
________________________________________________________________________________
bn5b_branch2a (BatchNormalizati (None, 4, 4, 512)    2048        res5b_branch2a[0][0]
________________________________________________________________________________
activation_93 (Activation)      (None, 4, 4, 512)    0           bn5b_branch2a[0][0]
________________________________________________________________________________
res5b_branch2b (Conv2D)         (None, 4, 4, 512)    2359808     activation_93[0][0]
________________________________________________________________________________
bn5b_branch2b (BatchNormalizati (None, 4, 4, 512)    2048        res5b_branch2b[0][0]
________________________________________________________________________________
activation_94 (Activation)      (None, 4, 4, 512)    0           bn5b_branch2b[0][0]
________________________________________________________________________________
res5b_branch2c (Conv2D)         (None, 4, 4, 2048)   1050624     activation_94[0][0]
________________________________________________________________________________
bn5b_branch2c (BatchNormalizati (None, 4, 4, 2048)   8192        res5b_branch2c[0][0]
________________________________________________________________________________
add_31 (Add)                    (None, 4, 4, 2048)   0           bn5b_branch2c[0][0]
                                                                 activation_92[0][0]
________________________________________________________________________________
activation_95 (Activation)      (None, 4, 4, 2048)   0           add_31[0][0]
________________________________________________________________________________
res5c_branch2a (Conv2D)         (None, 4, 4, 512)    1049088     activation_95[0][0]
________________________________________________________________________________
bn5c_branch2a (BatchNormalizati (None, 4, 4, 512)    2048        res5c_branch2a[0][0]
________________________________________________________________________________
activation_96 (Activation)      (None, 4, 4, 512)    0           bn5c_branch2a[0][0]
```

RESULTS

Model's training accuracy

```
history_finetunning = model.fit_generator(generator=train_generator,
                              steps_per_epoch=STEP_SIZE_TRAIN,
                              validation_data=val_generator,
                              validation_steps=STEP_SIZE_VALID,
                              epochs=3, # revert to: epochs=EPOCHS
                              callbacks=callback_list,
                              verbose=1).history

model.save('/content/drive/My Drive/ResNetCompleteModel.h5')
```

```
Epoch 1/3
91/91 [==============================] - 1240s 14s/step - loss: 0.2300 - acc: 0.9043 - val_loss: 0.2818 - val_acc: 0.9136
Epoch 2/3
91/91 [==============================] - 1236s 14s/step - loss: 0.1932 - acc: 0.9192 - val_loss: 0.2486 - val_acc: 0.9171
Epoch 3/3
91/91 [==============================] - 1243s 14s/step - loss: 0.1762 - acc: 0.9280 - val_loss: 0.2257 - val_acc: 0.9237
```

```
[ ] history = {'loss': history_warmup['loss'] + history_finetunning['loss'],
               'val_loss': history_warmup['val_loss'] + history_finetunning['val_loss'],
               'acc': history_warmup['acc'] + history_finetunning['acc'],
               'val_acc': history_warmup['val_acc'] + history_finetunning['val_acc']}

    sns.set_style("whitegrid")
    fig, (ax1, ax2) = plt.subplots(2, 1, sharex='col', figsize=(20, 14))

    ax1.plot(history['loss'], label='Train loss')
    ax1.plot(history['val_loss'], label='Validation loss')
    ax1.legend(loc='best')
```

RESULTS

Predicting output

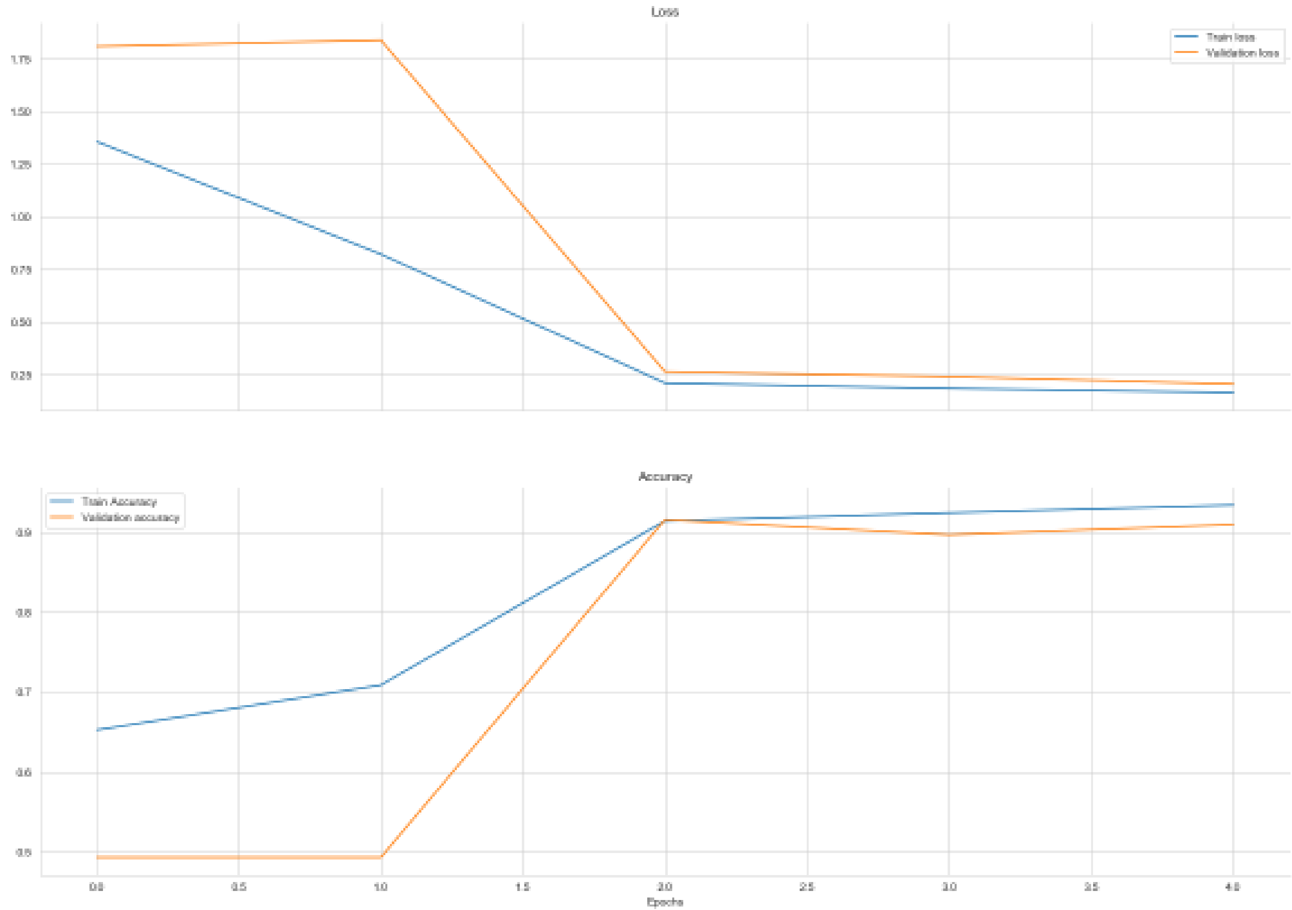
```
imageRealTimeTestDir='/content/drive/My Drive/Test'
Predictions=[]
Actual_labels=[]
for foldername in os.listdir(imageRealTimeTestDir):

  for filename in os.listdir(imageRealTimeTestDir+"//"+foldername):
    img=cv2.imread(imageRealTimeTestDir+"//"+foldername+"//"+filename)
    backup=img
    img=cv2.resize(img,(128,128))
    img=np.reshape(img,[1,128,128,3])
    output=model.predict([img])
    ind=np.argmax(output[0])
    print('For image {} predicted label is {}'.format(filename,ind))
    Predictions.append(ind)
    Actual_labels.append(int(foldername))
```

```
For image 360832d84ce0.png predicted label is 0
For image 050bb1eafa76.png predicted label is 0
For image 35362d43e753.png predicted label is 0
For image 441affbe99aa.png predicted label is 2
For image 01d9477b1171.png predicted label is 0
For image 49eb73968c44.png predicted label is 0
For image 441affbe99aa15.png predicted label is 2
For image 486e852a3b4d.png predicted label is 0
For image 2b10f138e67d.png predicted label is 0
For image 0da321efbce6.png predicted label is 0
For image 540e4973829e.png predicted label is 0
```

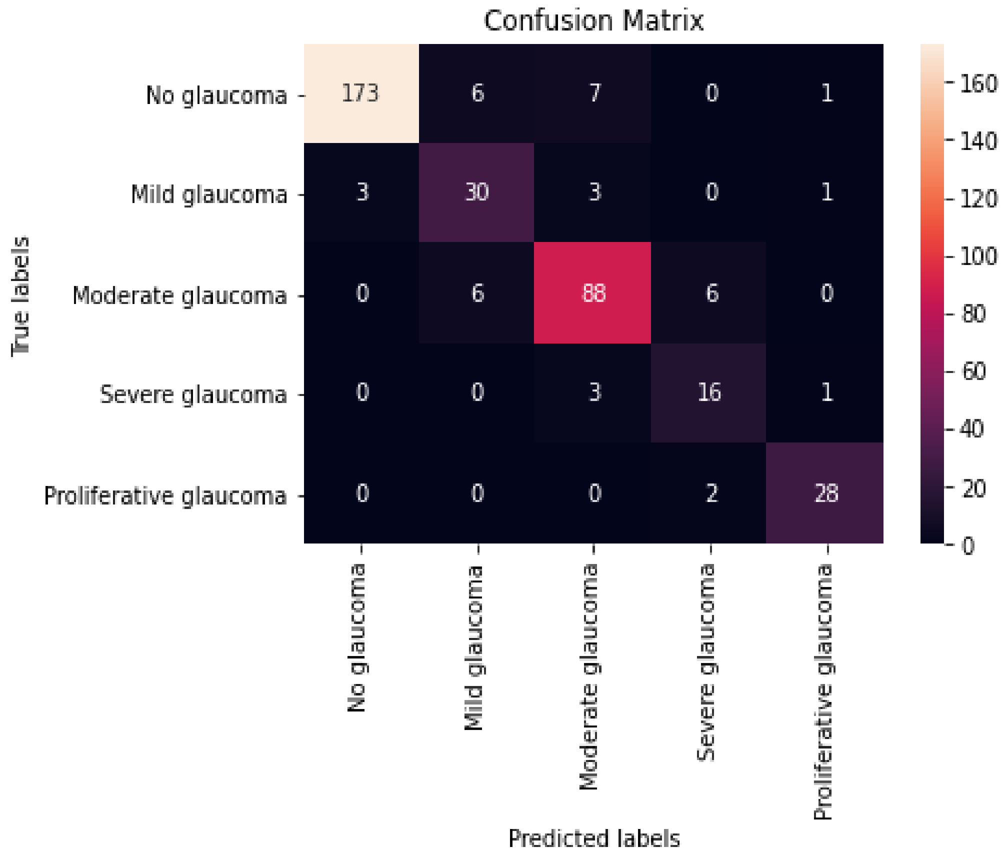
RESULTS

Model's loss, accuracy against train data and validation data



RESULTS

Confusion Matrix



TEST CASES

TEST CASE ID	TEST CASE NAME	DESRIPTION	EXPECTED RESULTS	ACTUAL RESULTS	TEST RESULTS
1	Verification of data set	Test case deals with images	Images should be clear and coloured	Images should be clear and coloured	Pass
2	Validation of custom architecture	To evaluate the model performance	High training and validation accuracy which are close	Achieved training accuracy of 51% and validation accuracy of 54%	Fail
3	Validation of RESNET50 architecture with 3 Epochs	To evaluate the model performance	High training and validation accuracy which are very close	Achieved training accuracy of 95% and validation accuracy of 47%	Fail

TEST CASES

TEST CASE ID	TEST CASE NAME	DESRIPTION	EXPECTED RESULTS	ACTUAL RESULTS	TEST RESULTS
4	Validation of RESNET50 architecture with 5 epochs	To evaluate the model performance	High training and validation accuracy which are very close	Achieved training accuracy of 92% and validation accuracy of 92%	Pass
5	Test working of libraries and packages	All the imported libraries and packages should work properly	Worked as expected	Worked as expected	Pass
6	Test working of dataset loading	Dataset must be imported without any uncertainty	Data picked from correct folders	Data picked from correct folders	Pass

CONCLUSION

- We presented a deep neural network framework to detect Glaucoma and also stages of its severity. In this project, a structure of deep learning for Glaucoma disease detection relies on significant CNN, which can get the discriminative features that better depict the hidden models related to Glaucoma.
- This project aims to implement a model for the hospitals, which will help predict Glaucoma disease at the earliest possible using fundus images.
- We have used 1928 images for Glaucoma detection in which we have achieved training accuracy of 93.41% and validation accuracy of 90.96%.

FUTURE SCOPE

- Future scope of our work is to improve the accuracy with less cost effectiveness and also to implement a procedure of checking every fundus image generated in the hospital for glaucoma so that a patient can take necessary steps at his earliest convenience.

REFERENCES

- *N. A. Diptu et al., "early Detection of Glaucoma Using Fuzzy Logic in Bangladesh Portugal, 2018, pp. 87-93.*
- *N. Sengar, M. K. Dutta, R. Burget and M. Ranjoha, "Automated detection of suspected glaucoma in digital fundus images," 2017 40th international Conference on Telecommunications and Signal Processing (TSP), Barcelona, 2017, pp. 749-752.*
- *Septiarini, Anindita et al. "Automated Detection of Retinal Nerve Fiber Layer by Texture-Based Analysis for Glaucoma evaluation." Healthcare informatics research vol. 24,4 (2018): 335-345. doi:10.4258/hir.2018.24.4.335*
- *Rahul Sharma, Pradip Sircas et al. "Automated glaucoma detection using center slice of higher order statistics", Journal of Mechanics in Medicine and BiologyVol. 19, No. 01, 1940011 (2019), https://doi.org/10.1142/S0219519419400116*
- *Context," 2018 international Conference on intelligent Systems (IS), Funchal - Madeira, R. Zhao, X. Chen, L. Xiyao, C. Zailiang, F. Guo and S. Li, "Direct Cup-to-Disc Ratio estimation for Glaucoma Screening via Semi-supervised Learning," in ieee Journal of Biomedical and Health informatics.*


THANK YOU