**GLAUCOMA DETECTION IN HUMAN EYE**

**A PROJECT REPORT**

***Submitted by***

|  |  |
| --- | --- |
| **D. Sathishkumar** | **(211417104261)** |
| **P. Viswanath** | **(211417104284)** |
| **G.R. Venkatesulu** | **(211417104277)** |
|  |  |

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF ENGINEERING**

***in***

**COMPUTER SCIENCE AND ENGINEERING**

**PANIMALAR ENGINEERING COLLEGE, POONAMALLEE**

**ANNA UNIVERSITY: CHENNAI 600 025**

**APRIL 2021**

**ANNA UNIVERSITY: CHENNAI 600 025**

## BONAFIDE CERTIFICATE

Certified that this project report **“GLUCOMA DETECTION IN HUMEN EYE”** is the Bonafide work of “**SATHISHKUMAR D (211417104261), VISWANATH P (211417104284) VENKATESULU G R (211417104277),**” who carried out the project work under my supervision.

|  |  |
| --- | --- |
| **SIGNATURE** | **SIGNATURE** |
| **Dr. S. MURUGAVALLI, M.E., Ph.D.,**  **HEAD OF DEPARTMENT**      DEPARTMENT OF CSE,  PANIMALAR ENGINEERING COLLEGE, NAZARATHPETTAI,  POONAMALLEE,  CHENNAI-600 123. | **Mr. G. SENTHILKUMAR., M.C.A.,M.Phil.,M.B.A., M.E., (Ph.D),**  **SUPERVISOR**  ASSOCIATE PROFESSOR  DEPARTMENT OF CSE,  PANIMALAR ENGINEERING COLLEGE, NAZARATHPETTAI,  POONAMALLEE,  CHENNAI-600 123 |

Certified that the above candidate were examined in the university project viva-voice held on 26.03.2021

|  |  |
| --- | --- |
| **INTERNAL EXAMINER** | **EXTERNAL EXAMINER** |

**ACKNOWLEDGEMENT**

A project of this magnitude and nature requires kind co-operation and support from many for its successful completion. We wish to express our sincere thanks to all those who were involved in the completion of this project.

We would like to express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.**, for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We express our sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Thiru.C.SAKTHIKUMAR,M.E.,** and **Tmt. SARANYASREE SAKTHIKUMAR B.E.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.,**  who helped us in completing the project.

We thank the Head of the CSE Department, **Dr. S.MURUGAVALLI , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank our internal guide**, Dr. S.MURUGAVALLI , M.E.,Ph.D.,** and all the faculty members of the Department of CSE for their valuable guidance and continuous support for the development of our project.

### D SATHISHKUMAR (211417104)

### P VISWANATH (211417104306)

### G R VENKATESULU (211417104296)

### ABSTRACT

GLAUCOMA is an unending and irreversible eye infection in which the optic nerve is consistently hurt, inciting disintegrating in vision and individual fulfillment. In this errand, we develop a figuring in which we recognize GLAUCOMA in starting periods by using Recurrent Neural Network (RNN) in light of visual picture getting ready. This visual picture getting ready in perspective of the Deep Learning. A significant learning structure is proposed remembering the ultimate objective to get a different leveled depiction of FUNDUS pictures to isolate among GLAUCOMA and NON-GLAUCOMA outline.

The condition of the vascular arrangement of human the eye is an imperative characteristic factor in ophthalmology. Its division in fundus imaging is a troublesome endeavor due to distinctive anatomical structures like vein, optic glass, optic plate, macula, and fovea. In this proposed work, a robotized division of anatomical structures in fundus pictures, for instance, vein and optic circle is done using Recurrent Neural Networks (RNN). A Recurrent Neural Network is a composite of various simple taking care of units, each including a couple weighted wellsprings of data and one yield, performing convolution of input signals with weights and changing the outcome with some sort of nonlinearity. The advantage of RNN is that it can be set up on and on so more features can be found. A typical precision of 95.64% is settled in the gathering of a vein or not. The gained estimation of CDR is differentiated and the given estimations of the case pictures and in this way the execution of the proposed structure in which Recurrent Neural Networks for the division is used is magnificent in the motorized revelation of sound and Glaucoma pictures.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO** | **TITLE** | **PAGE NO** |
|  | **ABSTRACT** | iii |
|  | **LIST OF FIGURES** | 04 |
|  | **LIST OF SYMBOLS, ABBREVIATIONS** | 05 |
|  | **INTRODUCTION** | 06 |
| **1.** | 1.1 OVERVIEW | 07 |
|  | 1.2 PROBLEM DEFINITION | 08 |
|  | **LITERATURE SURVEY** | 09 |
| **2.** | **SYSTEM ANALYSIS** | 12 |
| **3.** | 3.1 EXISTING SYSTEM | 13 |
|  | 3.2 PROPOSED SYSTEM |  |
|  | 3.3 REQUIREMENT ANALYSIS AND SPECIFICATION | 14 |
|  | 3.3.1 Hardware Requirements |  |
|  | 3.3.2 Software Requirements |  |
|  | **SYSTEM DESIGN** | 16 |
| **4.** | 4.1 UML DIAGRAMS | 17 |
|  | 4.1.1 Use Case Diagram |  |
|  | 4.1.2 Collaboration Diagram | 18 |
|  | 4.1.3 Flow Diagram | 19 |
|  | 4.1.4 Sequence Diagram |  |
|  | **SYSTEM ARCHITECTURE** | 21 |
| **5.** | 5.1 ARCHITECTURE OVERVIEW | 22 |
|  | 5.2 SOFTWARE SPECIFICATION |  |
|  | 5.2.1 Python |  |
|  | 5.3 SOFTWARE DISCRIPTION  5.3.1 Python | 23 |
|  | 5.3.2 Libraries  5.3.3 Development |  |
|  | **SYSTEM IMPLEMENTATION** | 28 |
| **6.** | 6.1 CODING |  |
|  | **SYSTEM TESTING** | 41 |
| **7.** | 7.1 INTRODUCTION |  |
|  | 7.2 TYPES OF TESTING |  |
|  | 7.3 TEST CASES & REPORTS / PERFORMANCE ANALYSIS | 46 |
|  | **CONCLUSION** | 51 |
| **8.** | 8.1 CONCLUSION AND FUTURE ENHANCEMENTS |  |
|  | **REFERENCES** | 54 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **NAME OF THE FIGURE** | **PAGE NO** |
| 4.1.1 | Use case Diagram | 17 |
| 4.1.2 | Collaboration Diagram | 18 |
| 4.1.3 | Flow Diagram | 19 |
| 4.1.4 | Sequence Diagram | 19 |
| 5.1.1 | Architecture Diagram | 21 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**List Of Abbreviations**

|  |  |
| --- | --- |
| CDR-cup-to-disc  OD-Oculus Dexter  ROI-Region Of Retinal  OCT-Optical Coherence Tomogaphy  FCM-Fuzzy C Means Clustering  PEP-Python Enhancement Proposal  EFP-Extended Feature Projection  RNN-Recurrent Neura Network |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

**CHAPTER 1**

**INTRODUCTION**

**CHAPTER 1**

**INTRODUCTION**

* 1. **OVERVIEW**

Glaucoma is a common eye illness that is irreversible and the second driving clarification behind visual weakening. Due to nonappearance of a skilled early screening structure, it twists up recognizably watchful just in the last times of glaucoma. A measure of individuals with glaucoma was 64.3 million, additionally, is depended upon to ascend to 76.0 million of each 2020. Early examination and treatment are crucial to avoid loss of vision in glaucoma patients. Glaucoma can be seen suitable on time by checking retinal fundus pictures condition. The fundus photo of the eye intertwines the retina, optic circle, fovea, macula and back post. Retinal fundus pictures have remained the best standard for assessing the changes in retina. Out of a couple of frameworks utilized for clinical affirmation of glaucoma, fundus picture examination is the one most fitting for seeing. An altered structure for glaucoma region is proposed here, which impacts utilization of fundus to picture utilizing CDR. The optic plate (OD) or optic nerve head in the retina where cell axons leave the eye to shape the optic nerve. Optic plate division utilizing RNN is the fundamental stages in the proposed approach. The optic plate has a focal wretchedness, denied of material tissue. Glaucoma, depicted by the loss of nerve tissue causes extending of this circle zone, happening in the unavoidable and substandard localities initially times. An optic glass is segmented by Fuzzy C suggests utilizing ROI additionally, morphological undertaking. The parameter Cup-to-Disk Extent is figured to check for glaucoma. Clear Statement of the problem. Glaucoma is an ailment which exceedingly influences those persons who are under the age of 40 to 60 years of age. Its rate over all the world is 5% which is expanding step by step. Location of a sickness (GLAUCOMA) in light of profound learning by utilizing picture preparing on (FUNDUS, OCT).

* 1. **PROBLEM DEFINITION**

In our project, we develop a figuring in which we recognize GLAUCOMA in starting periods by using Recurrent Neural Network (RNN) in light of visual picture getting ready. This visual picture getting ready in perspective of the Deep Learning. A significant learning structure is proposed remembering the ultimate objective to get a different leveled depiction of FUNDUS pictures to isolate among GLAUCOMA and NON-GLAUCOMA outline. Wide examinations are performed on the FUNDUS and OCT pictures datasets. The condition of the vascular arrangement of human the eye is an imperative characteristic factor in ophthalmology. Its division in fundus imaging is a troublesome endeavor due to distinctive anatomical structures like vein, optic glass, optic plate, macula, and fovea. Vein division can help the distinguishing proof of masochist changes which are possible markers for arteriosclerosis, retinopathy? micro aneurysms and macular degeneration. The division of optic plate and optic holder from retinal pictures are used to figure a basic pointer, glass – to plate extent (CDR) accurately to help the specialists around Glaucoma in fundus pictures. In this proposed work, a robotized division of anatomical structures in fundus pictures, for instance, vein and optic circle is done using Recurrent Neural Networks (RNN).

**CHAPTER 2**

**LITERATURE SURVEY**

**CHAPTER 2**

**LITERATURE SURVEY**

**Paper 1:” Local Entropy Thresholding Based Fast Retinal Vessels Segmentation by Modifying Matched Filter”**

The retinal blood vessels are highly responsible for the detection of retinal pathology such as glaucoma, hypertension, arteriosclerosis and diabetes. So the segmentation of retinal blood vessels from their background is a prominent task. The objective of this paper is to present an automatic local entropy thresholding based fast, efficient and accurate retinal blood vessels segmentation method by modifying the standard Gaussian shaped matched filter reported in other papers in literature. Another objective is to identify the thin blood vessels together with large blood vessel segments, which is not considering in some existing blood vessels segmentation methods in literature. The proposed method has been implemented on forty retinal images taken from DRIVE database and segmented results are compared with hand-labeled ground truth images also available in the DRIVE database. The efficacy of the proposed method was examined and presented in terms of overall sensitivity, specificity and accuracy. Further, the performance of the proposed algorithm is compared with some other existing standard methods for the same task available in literature and the performance of the proposed method is found to be performing significantly better.

**Paper 2: “Optic Disk Segmentation in Retinal Images Using Active Contour Model based on Extended Feature Projection.”**

Accurate localization and segmentation of an optic disk (OD) is an important problem in the analysis of abnormality conditions such as optic disk shrinking/swelling, pale optic disk and glucoma. Hence, this paper proposes an automated fast and accurate OD localization and segmentation technique. In this work, OD localization is performed using the extended feature projection method (EFP) based on retinal vessel orientation and average intensity variance. Multiple OD candidate locations, obtained from OD localization technique (EFP), are used as the initialization points of the active contour model to detect the OD boundaries. Next, we use a decision tree based on the OD features such as the area of vessels, the brightness and the entropy to select the final OD candidate. The proposed technique has been tested on STARE dataset to evaluate comparative studies on the localization and segmentation of OD in retinal images. The accuracy of the OD localization is 90.12% with an average computing time of 13 seconds per image. The performance of the OD segmentation in terms of sensitivity is 74.62 % and positive predictive value is 60.22% with an average computing time of 20 seconds per image. The proposed approach improves the accuracy of conventional feature projection method by 12.34% and runs as quickly as the conventional one.

**CHAPTER 3**

**SYSTEM ANALYSIS**

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3.1** **EXISTING SYSTEM**

Most existing techniques for OD confinement are run based, either abusing the OD appearance legitimate ties or the spatial connection between the OD and the fundamental vascular arcade. The location of OD variations from the norm has been performed through the discovery of injuries, for example, hemorrhages or through estimating glass to circle proportion. Therefore, these strategies result in mind-boggling and resolute picture investigation calculations restricting their relevance to vast picture sets got either in epidemiological examinations or in screening for retinal or optic nerve infections.

**EXISTING SYSTEM DISADVANTAGES**

As seen from results for traditional methods, there is need of proper detection of the OD edges to detect its abnormality. The proposed method is to solve all the problems with respect to traditional method.

**3.2 PROPOSED SYSTEM**

The problems which are occurred in the existing applications are overcome in proposed application. In this application we are implemented all of the functionality by using any device with internet access for organization to handling the meetings and built better communication environment between managers/employees or workers of organization. The proposed application supports secured audio/ video streaming as compared to existing applications. Minimum latency is supported by encoder. This application is cost effective solution for users. Organization must have to pay for what they use. They have to pay minimum cost for per meeting. We are using following RNN architecture. RNN (Recurrent neural network) is a network architecture in which we define a modal consist of multiple Recurrent and dropout layers. We give the images data to input layer in the form of arrays and apply different filters for extract the features from given image. GLAUCOMA disclosure Model has the uncommon impact on our lives. The fact of the matter is to give straightforward access to patient of GLAUCOMA from the unmistakable countries, data on a cloud by just using quick web affiliation and to make a contraption self-ruling application. This application will grow the general advancement of an affiliation and no prerequisite for sorting out partitioned spaces for social occasions. This application has the straightforward interface. All exchanges between each one of the customers are done through a quick web affiliation. Most of the streams are in the control of director that is stream grouped and likelihood for record streams. This is furthermore the shrewd plan and no gear or programming need.

**Advantage:**

The proposed system uses recurrent neural network (RNN) to automatically detect glaucoma. Using digital fundus images of affected patients quickly an accurate.

**3.3** **REQUIREMENT ANALYSIS AND SPECIFICATION**

**3.3.1 HARDWARE REQUIREMENTS**

* System : Intel Pentium.
* Hard Disk : 120 GB.
* Monitor : 15’’ LED
* Input Devices : Keyboard, Mouse
* Ram : 2 GB

**3.3.2 SOFTWARE REQUIREMENTS**

* Operating system : Windows 7.
* Coding Language : Python
* Tools : Jupiter notebook

**CHAPTER 4**

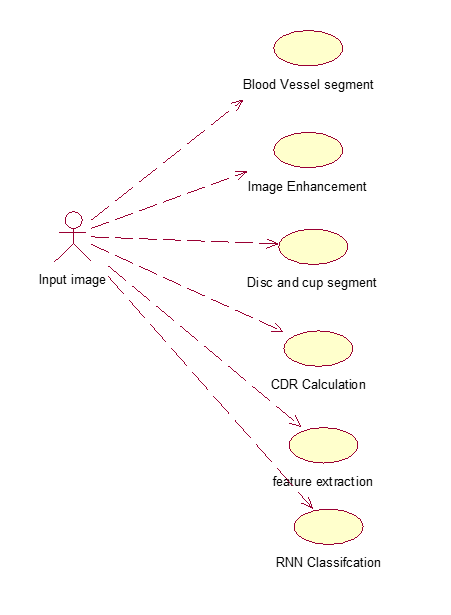
**SYSTEM DESIGN**

**CHAPTER 4**

**SYSTEM DESIGN**

**4.1 UML DIAGRAMS**

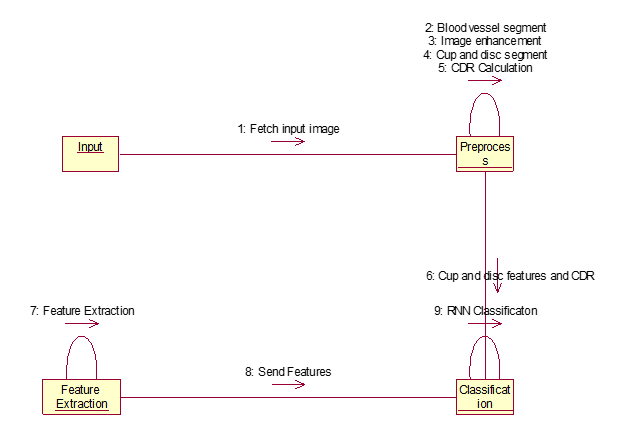
**4.1.1 USE CASE DIAGRAM**

A **use case diagram** at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

**Fig 4.1.1. Use case diagram**

**4.1.2 COLLABORATION DIAGRAM 0**

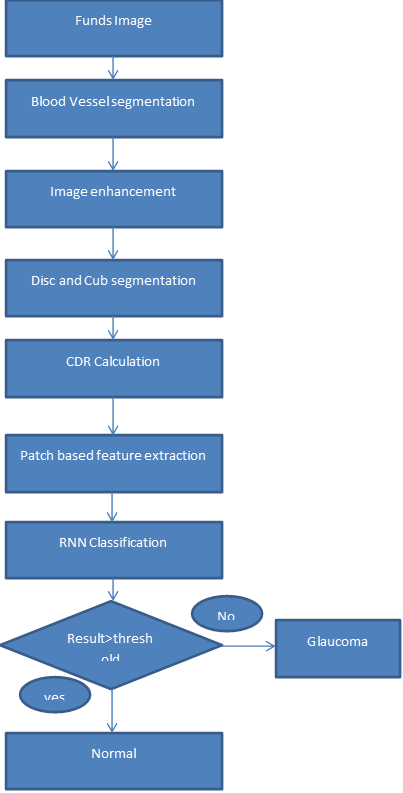
A collaboration diagram, a kind of structural diagram, shows how objects interact to perform the behavior of a particular use case, or a part of a use case. Along with sequence diagrams, collaboration is used by designers to define and clarify the roles of the objects that perform a particular flow of events of a use case. They are the primary source of information used to determining class responsibilities and interfaces.



**Fig 4.1.2. Package diagram**

**4.1.3 FLOW DIAGRAM**

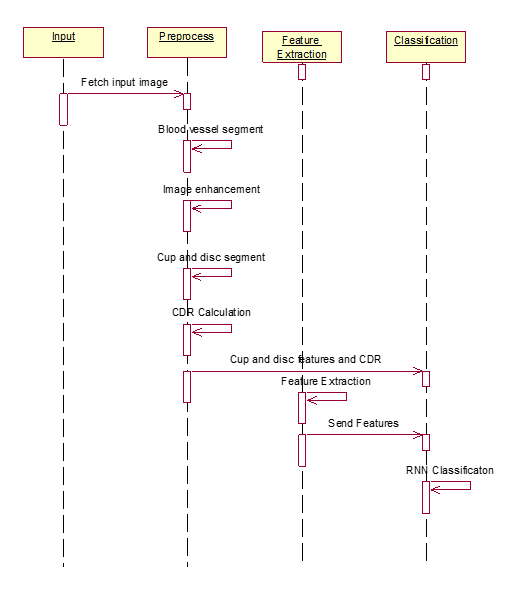
A diagram of the sequence of movements or actions of people or things involved in a complex system or activity.



**Fig 4.1.3. Flow diagram**

**4.1.4 SEQUENCE DIAGRAM**

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the sequence diagram of the system under development.



**Fig 4.1.4. Sequence Diagram**

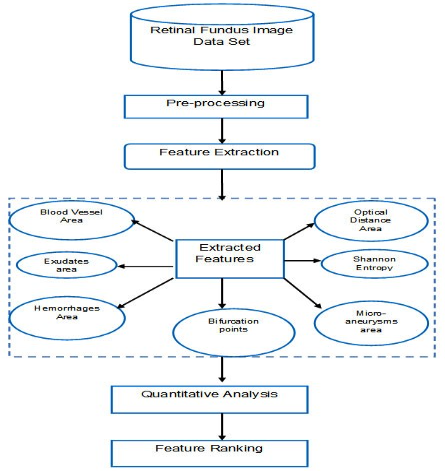
**CHAPTER 5**

**SYSTEM ARCHITECTURE**

**CHAPTER 5**

**SYSTEM ARCHITECTURE**

**5.1 ARCHITECTURE OVERVIEW**



**Fig 5.1.1 Architecture Diagram**

**5.2 SOFTWARE SPECIFICATION**

**5.2.1 PYTHON**

Python is an interpreted, high-level, general-purpose programming language. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. Python is often described as a “batteries included” language due to its comprehensive standard library. Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, an open-source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

**5.3 SOFTWARE DISCRIPTION**

**5.3.1 PYTHON**

Python is a multi-paradigm programming language. Object-oriented programming and structured programming are fully supported, and many of its features support functional programming and aspect-oriented programming. Many other paradigms are supported via extensions, including design by contract and logic programming.

Python uses dynamic typing, and a combination of reference counting and a cycle-detecting garbage collector for memory management. It also features dynamic name resolution (late binding), which binds method and variable names during program execution.

Python's design offers some support for functional programming in the Lisp tradition. It has filter map and reduce functions; list comprehensions, dictionaries, sets and generator expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from Haskell and Standard ML.

The language's core philosophy is summarized in the document The Zen of Python (PEP 20), which includes aphorisms such as:

* Beautiful is better than ugly.
* Explicit is better than implicit.
* Simple is better than complex.
* Complex is better than complicated.
* Readability counts.

Rather than having all of its functionality built into its core, Python was designed to be highly extensible. This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with ABC, which espoused the opposite approach.

Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to Perl's "there is more than one way to do it" motto, Python embraces a "there should be one—and preferably only one—obvious way to do it" design philosophy. Alex Martelli, a Fellow at the Python Software Foundation and Python book author, writes that "To describe something as 'clever' is not considered a compliment in the Python culture."

Python's developers strive to avoid premature optimization, and reject patches to non-critical parts of the CPython reference implementation that would offer marginal increases in speed at the cost of clarity. When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use PyPy, a just-in-time compiler. Cython is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter.

An important goal of Python's developers is keeping it fun to use. This is reflected in the language's name a tribute to the British comedy group Monty Python and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (from a famous Monty Python sketch) instead of the standard foo and bar.

0 A common neologism in the Python community is pythonic, which can have a wide range of meanings related to program style. To say that code is pythonic is to say that it uses Python idioms well, that it is natural or shows fluency in the language, that it conforms with Python's minimalist philosophy and emphasis on readability. In contrast, code that is difficult to understand or reads like a rough transcription from another programming language is called Unpythonic.

Users and admirers of Python, especially those considered knowledgeable or experienced, are often referred to as Pythonistas.

**5.3.2 LIBRARIES**

Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary precision decimals, manipulating regular expressions, and unit testing.Some parts of the standard library are covered by specifications, but most modules are not. They are specified by their code, internal documentation, and test suites (if supplied). However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

As of March 2018, the Python Package Index (PyPI), the official repository for third-party Python software, contains over 130,000 packages with a wide range of functionality, including:

* Graphical user interfaces
* Web frameworks
* Multimedia
* Databases
* Networking
* Test frameworks
* Automation
* Web scraping
* Documentation
* System administration
* Scientific computing
* Text processing
* Image processing

**5.3.3 DEVELOPMENT**

Python's development is conducted largely through the Python Enhancement Proposal (PEP) process, the primary mechanism for proposing major new features, collecting community input on issues and documenting Python design decisions. Python coding style is covered in PEP 8. Outstanding PEPs are reviewed and commented on by the Python community and the steering council.

Enhancement of the language corresponds with development of the CPython reference implementation. The mailing list python-dev is the primary forum for the language's development. Specific issues are discussed in the Roundup bug tracker maintained at python.org. Development originally took place on a self-hosted source-code repository running Mercurial, until Python moved to GitHub in January 2017.

CPython's public releases come in three types, distinguished by which part of the version number is incremented:

* Backward-incompatible versions, where code is expected to break and need to be manually [ported](https://en.wikipedia.org/wiki/Ported). The first part of the version number is incremented. These releases happen infrequently for example, version 3.0 was released 8 years after 2.0.
* Major or "feature" releases, about every 18 months, are largely compatible but introduce new features. The second part of the version number is incremented. Each major version is supported by bugfixes for several years after its release.
* Bugfix releases, which introduce no new features, occur about every 3 months and are made when a sufficient number of bugs have been fixed upstream since the last release. Security vulnerabilities are also patched in these releases. The third and final part of the version number is incremented.

Many alpha, beta, and release-candidates are also released as previews and for testing before final releases. Although there is a rough schedule for each release, they are often delayed if the code is not ready. Python's development team monitors the state of the code by running the large unit test suite during development, and using the BuildBot continuous integration system.

The community of Python developers has also contributed over 86,000 software modules (as of 20 August 2016) to the Python Package Index (PyPI), the official repository of third-party Python libraries.

The major academic conference on Python is PyCon. There are also special Python mentoring programmes, such as Pyladies.

**CHAPTER 6**

**SYSTEM IMPLEMENTATION**

**CHAPTER 6**

**SYSTEM IMPLEMENTATION**

**6.1 SAMPLE CODING**

**import cv2**

**import numpy as np**

**import matplotlib.pyplot as plt**

**import pandas as pd**

**clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(10,10))**

**val=['Glaucomatous',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Normal',**

**'Normal',**

**'Normal',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Normal',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Normal',**

**'Normal',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Normal',**

**'Glaucomatous',**

**'Normal',**

**'Normal',**

**'Normal',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Normal',**

**'Normal',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Normal',**

**'Normal',**

**'Glaucomatous',**

**'Glaucomatous',**

**'Glaucomatous'**

**]**

**#main**

**def load\_image(path):**

**return cv2.imread(path)**

**import sys**

**import os**

**import glob**

**#function to load image and their name**

**def load\_set(folder, shuffle=False):**

**img\_list = sorted(glob.glob(os.path.join(folder, '\*.png')) + \**

**glob.glob(os.path.join(folder, '\*.jpg')) + \**

**glob.glob(os.path.join(folder, '\*.jpeg')))**

**if shuffle:**

**np.random.shuffle(img\_list)**

**data = []**

**filenames = []**

**for img\_fn in img\_list:**

**img = load\_image(img\_fn)**

**data.append(img)**

**filenames.append(img\_fn)**

**return data, filenames**

**def extract\_DRISHTI\_GS\_train(db\_folder,cdr,train\_data):**

**file\_codes\_all,exp1,exp2,exp3,exp4 = [], [], [], [], []**

**if train\_data:**

**set\_path = os.path.join(db\_folder, 'DRISHTI\_GS')**

**else:**

**set\_path = os.path.join(db\_folder, 'DRISHTI\_GS')**

**images\_path = os.path.join(set\_path, 'images')**

**X\_all, file\_names = load\_set(images\_path)**

**rel\_file\_names = [os.path.split(fn)[-1] for fn in file\_names]**

**rel\_file\_names\_wo\_ext = [fn[:fn.rfind('.')] for fn in rel\_file\_names]**

**if train\_data:**

**file\_codes = [fn[fn.find('\_'):] for fn in rel\_file\_names\_wo\_ext]**

**else:**

**file\_codes = [fn[fn.find('\_'):] for fn in rel\_file\_names\_wo\_ext]**

**file\_codes\_all.extend(file\_codes)**

**for fn in rel\_file\_names\_wo\_ext:**

**if cdr:**

**if train\_data:**

**CDR = open(os.path.join(set\_path, 'GT', fn,fn + '\_cdrValues.txt'),'r')**

**else:**

**CDR = open(os.path.join(set\_path, 'Test\_GT', fn,fn + '\_cdrValues.txt'),'r')**

**CDR = list(CDR)**

**CDR = CDR[0].split()**

**exp1.append(float(CDR[0]))**

**exp2.append(float(CDR[1]))**

**exp3.append(float(CDR[2]))**

**exp4.append(float(CDR[3]))**

**return X\_all, file\_codes\_all,exp1,exp2,exp3,exp4,file\_names**

**#vessel objective:**

**def vesselsegment(filename):**

**fundus = cv2.imread(filename)**

**segmentedimage=[]**

**plt.imshow(fundus, cmap='gray')**

**plt.show()**

**b,green\_fundus,r = cv2.split(fundus)**

**clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))**

**contrast\_enhanced\_green\_fundus = clahe.apply(green\_fundus)**

**# applying alternate sequential filtering (3 times closing opening)**

**r1 = cv2.morphologyEx(contrast\_enhanced\_green\_fundus, cv2.MORPH\_OPEN, cv2.getStructuringElement(cv2.MORPH\_ELLIPSE,(5,5)), iterations = 1)**

**R1 = cv2.morphologyEx(r1, cv2.MORPH\_CLOSE, cv2.getStructuringElement(cv2.MORPH\_ELLIPSE,(5,5)), iterations = 1)**

**r2 = cv2.morphologyEx(R1, cv2.MORPH\_OPEN, cv2.getStructuringElement(cv2.MORPH\_ELLIPSE,(11,11)), iterations = 1)**

**R2 = cv2.morphologyEx(r2, cv2.MORPH\_CLOSE, cv2.getStructuringElement(cv2.MORPH\_ELLIPSE,(11,11)), iterations = 1)**

**r3 = cv2.morphologyEx(R2, cv2.MORPH\_OPEN, cv2.getStructuringElement(cv2.MORPH\_ELLIPSE,(23,23)), iterations = 1)**

**R3 = cv2.morphologyEx(r3, cv2.MORPH\_CLOSE, cv2.getStructuringElement(cv2.MORPH\_ELLIPSE,(23,23)), iterations = 1)**

**f4 = cv2.subtract(R3,contrast\_enhanced\_green\_fundus)**

**cv2.imwrite('sub.png',f4)**

**plt.imshow(f4, cmap='gray')**

**plt.show()**

**f5 = clahe.apply(f4)**

**# removing very small contours through area parameter noise removal**

**ret,f6 = cv2.threshold(f5,15,255,cv2.THRESH\_BINARY)**

**mask = np.ones(f5.shape[:2], dtype="uint8") \* 255**

**contours, hierarchy = cv2.findContours(f6.copy(),cv2.RETR\_LIST,cv2.CHAIN\_APPROX\_SIMPLE)**

**for cnt in contours:**

**if cv2.contourArea(cnt) <= 200:**

**cv2.drawContours(mask, [cnt], -1, 0, -1)**

**im = cv2.bitwise\_and(f5, f5, mask=mask)**

**ret,fin = cv2.threshold(im,15,255,cv2.THRESH\_BINARY\_INV)**

**newfin = cv2.erode(fin, cv2.getStructuringElement(cv2.MORPH\_ELLIPSE,(3,3)), iterations=1)**

**# removing blobs of unwanted bigger chunks taking in consideration they are not straight lines like blood**

**#vessels and also in an interval of area**

**fundus\_eroded = cv2.bitwise\_not(newfin)**

**xmask = np.ones(fundus.shape[:2], dtype="uint8") \* 255**

**xcontours, xhierarchy = cv2.findContours(fundus\_eroded.copy(),cv2.RETR\_LIST,cv2.CHAIN\_APPROX\_SIMPLE)**

**for cnt in xcontours:**

**shape = "unidentified"**

**peri = cv2.arcLength(cnt, True)**

**approx = cv2.approxPolyDP(cnt, 0.04 \* peri, False)**

**if len(approx) > 4 and cv2.contourArea(cnt) <= 3000 and cv2.contourArea(cnt) >= 100:**

**shape = "circle"**

**else:**

**shape = "veins"**

**if(shape=="circle"):**

**cv2.drawContours(xmask, [cnt], -1, 0, -1)**

**finimage = cv2.bitwise\_and(fundus\_eroded,fundus\_eroded,mask=xmask)**

**blood\_vessels = cv2.bitwise\_not(finimage)**

**cv2.imwrite("test.png",blood\_vessels)**

**fundus\_out = cv2.imread('test.png')**

**plt.imshow(fundus\_out,cmap='gray')**

**plt.show()**

**segmentedimage.append(blood\_vessels)**

**return(segmentedimage)**

**#morphing**

**def cdr(cup,disc,plot):**

**#morphological closing and opening operations**

**R1 = cv2.morphologyEx(cup, cv2.MORPH\_CLOSE, cv2.getStructuringElement(cv2.MORPH\_ELLIPSE,(2,2)), iterations = 1)**

**r1 = cv2.morphologyEx(R1, cv2.MORPH\_OPEN, cv2.getStructuringElement(cv2.MORPH\_ELLIPSE,(7,7)), iterations = 1)**

**R2 = cv2.morphologyEx(r1, cv2.MORPH\_CLOSE, cv2.getStructuringElement(cv2.MORPH\_ELLIPSE,(1,21)), iterations = 1)**

**r2 = cv2.morphologyEx(R2, cv2.MORPH\_OPEN, cv2.getStructuringElement(cv2.MORPH\_ELLIPSE,(21,1)), iterations = 1)**

**R3 = cv2.morphologyEx(r2, cv2.MORPH\_CLOSE, cv2.getStructuringElement(cv2.MORPH\_ELLIPSE,(33,33)), iterations = 2)**

**img = R3**

**ret,thresh = cv2.threshold(img,127,255,0)**

**contours,hierarchy = cv2.findContours(thresh, cv2.RETR\_EXTERNAL,cv2.CHAIN\_APPROX\_SIMPLE) #Getting all possible contours in the segmented image**

**cup\_diameter = 0**

**largest\_area = 0**

**el\_cup = contours[0]**

**if len(contours) != 0:**

**for i in range(len(contours)):**

**if len(contours[i]) >= 5:**

**area = cv2.contourArea(contours[i])**

**if (area>largest\_area):**

**largest\_area=area**

**index = i**

**el\_cup = cv2.fitEllipse(contours[i])**

**cv2.ellipse(img,el\_cup,(140,60,150),3)**

**x,y,w,h = cv2.boundingRect(contours[index])**

**cup\_diameter = max(w,h)**

**cubx=w**

**cuby=h**

**#morphological closing and opening operations**

**R1 = cv2.morphologyEx(disc, cv2.MORPH\_CLOSE, cv2.getStructuringElement(cv2.MORPH\_ELLIPSE,(2,2)), iterations = 1)**

**r1 = cv2.morphologyEx(R1, cv2.MORPH\_OPEN, cv2.getStructuringElement(cv2.MORPH\_ELLIPSE,(7,7)), iterations = 1)**

**R2 = cv2.morphologyEx(r1, cv2.MORPH\_CLOSE, cv2.getStructuringElement(cv2.MORPH\_ELLIPSE,(1,21)), iterations = 1)**

**r2 = cv2.morphologyEx(R2, cv2.MORPH\_OPEN, cv2.getStructuringElement(cv2.MORPH\_ELLIPSE,(21,1)), iterations = 1)**

**R3 = cv2.morphologyEx(r2, cv2.MORPH\_CLOSE, cv2.getStructuringElement(cv2.MORPH\_ELLIPSE,(33,33)), iterations = 1)**

**r3 = cv2.morphologyEx(R3, cv2.MORPH\_OPEN, cv2.getStructuringElement(cv2.MORPH\_ELLIPSE,(43,43)), iterations = 1)**

**img2 = r3**

**ret,thresh = cv2.threshold(img2,127,255,0)**

**contours,hierarchy = cv2.findContours(thresh, cv2.RETR\_EXTERNAL,cv2.CHAIN\_APPROX\_SIMPLE)**

**disk\_diameter = 0**

**largest\_area = 0**

**el\_disc = el\_cup**

**if len(contours) != 0:**

**for i in range(len(contours)):**

**if len(contours[i]) >= 5:**

**area = cv2.contourArea(contours[i])**

**if (area>largest\_area):**

**largest\_area=area**

**index = i**

**el\_disc = cv2.fitEllipse(contours[i])**

**cv2.ellipse(img2,el\_disc,(140,60,150),10)**

**x,y,w,h = cv2.boundingRect(contours[index])**

**disk\_diameter = max(w,h)**

**diskx=w**

**disky=h**

**if plot:**

**plt.imshow(img2, 'gray',interpolation = 'bicubic')**

**plt.axis("off")**

**plt.title("Optic Disk")**

**plt.show()**

**plt.imshow(img, 'gray')**

**plt.axis("off")**

**plt.title("Optic Cup")**

**plt.show()**

**if(disk\_diameter == 0): return 1**

**cdr = cup\_diameter/disk\_diameter**

**return cdr,cubx,cuby,diskx,disky**

**CHAPTER 7**

**SYSTEM TESTING**

**CHAPTER 8**

**System Testing**

### 7.1 INTRODUCTION

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**7.2 TYPES OF TESTING**

**Unit Testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive.

Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Unit Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.

**Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one

step up – software applications at the company level – interact without error.

**Test Results:**

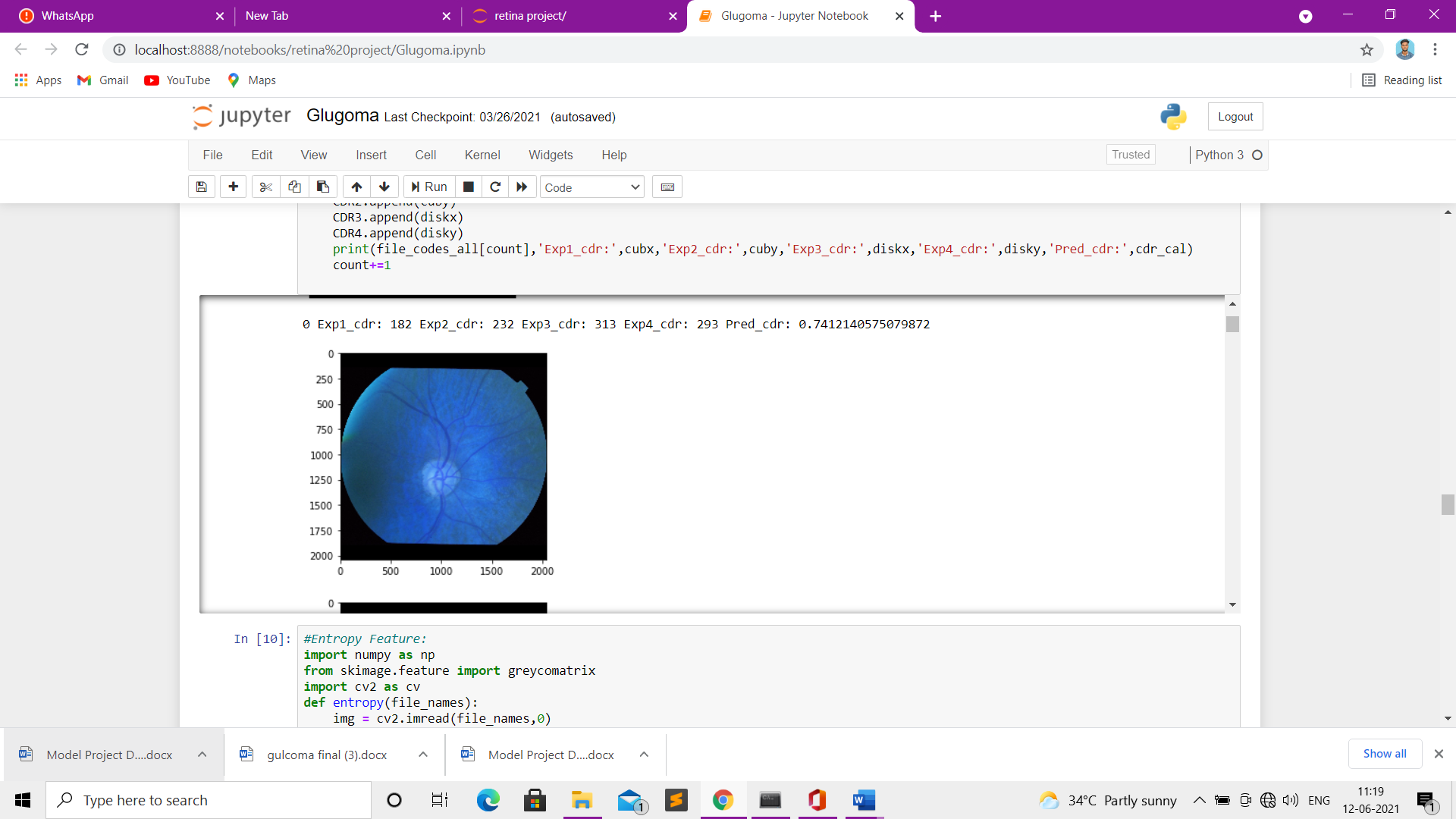
All the test cases mentioned above passed successfully. No defects encountered.

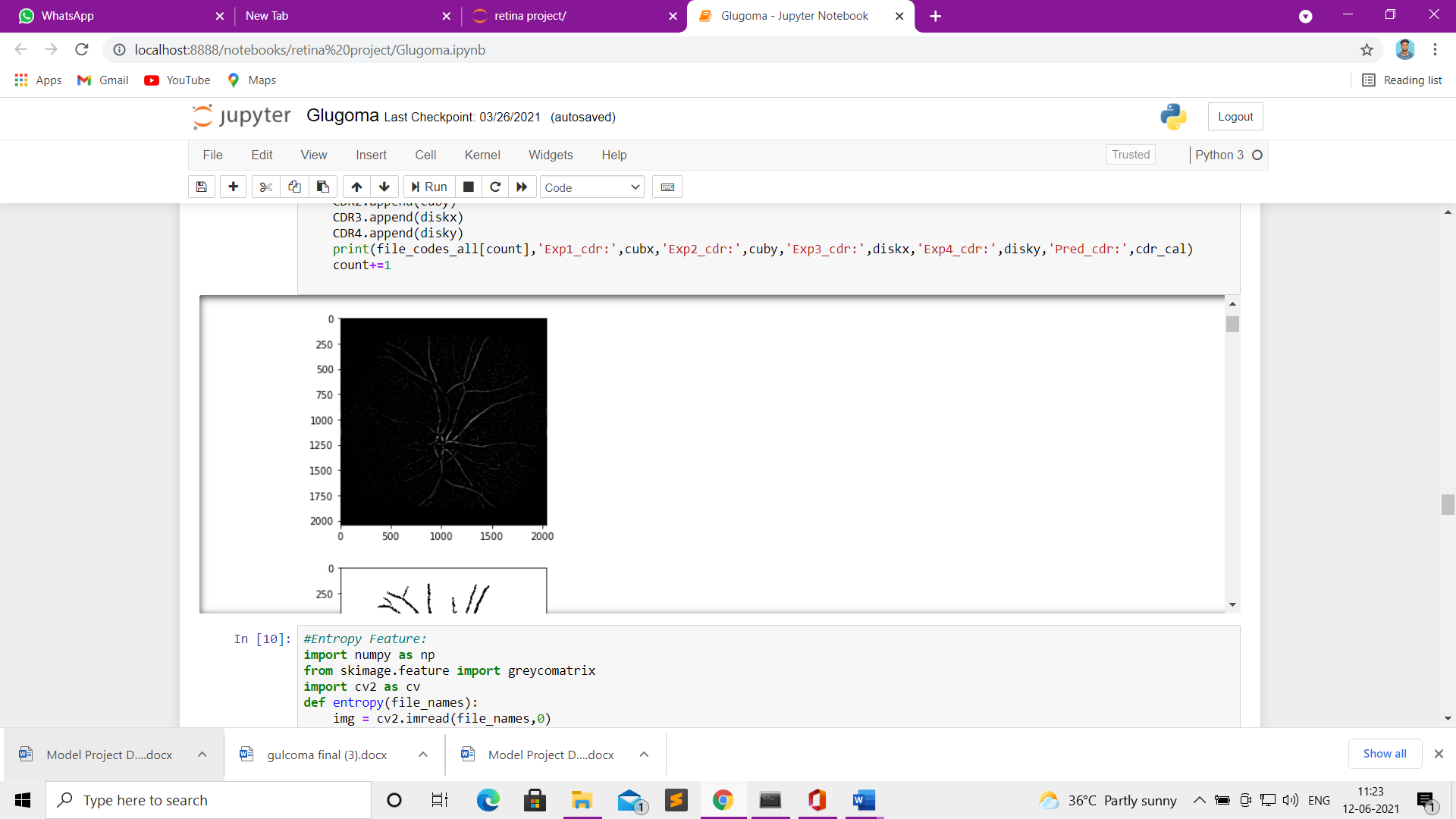
**Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

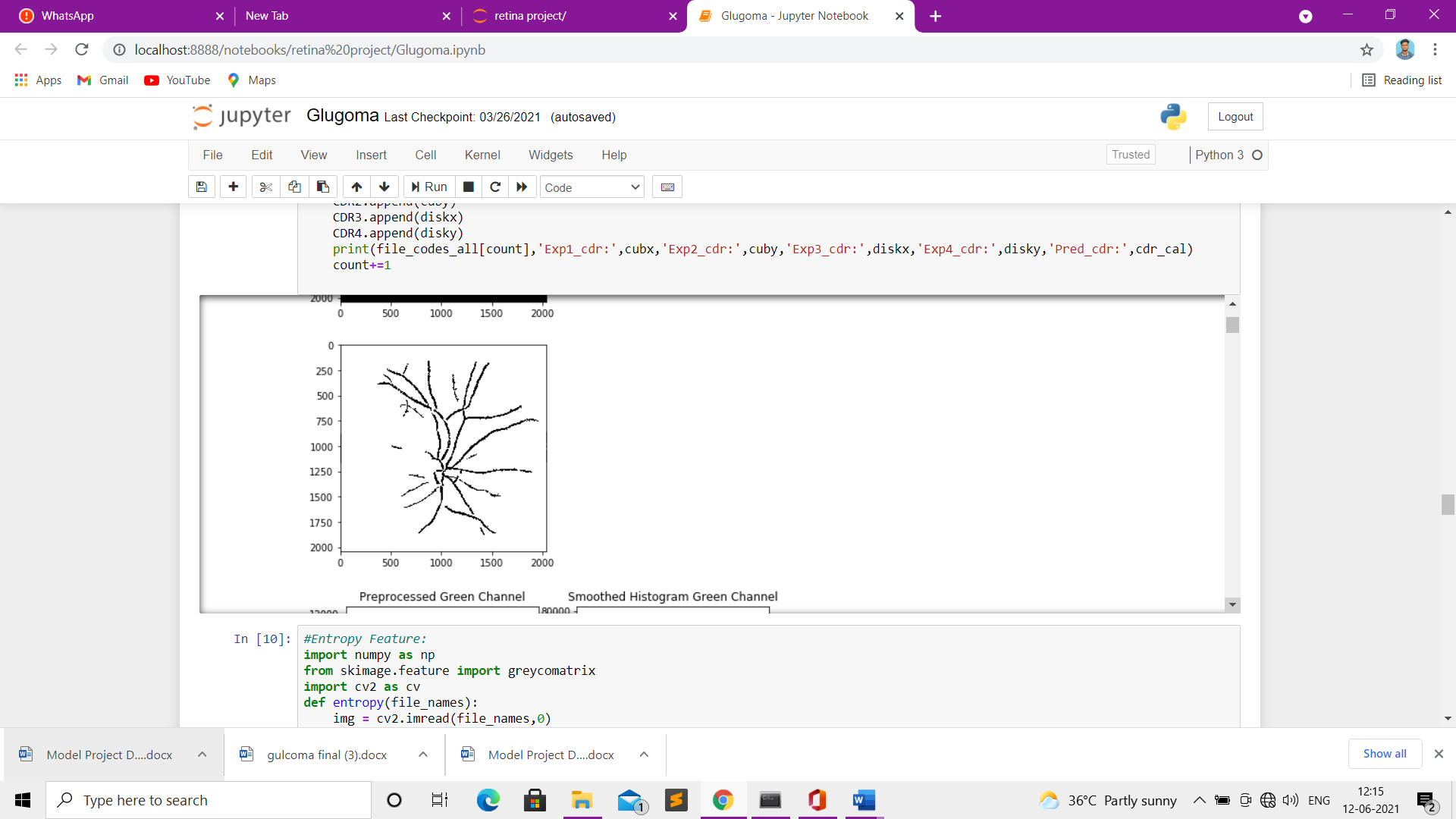
**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

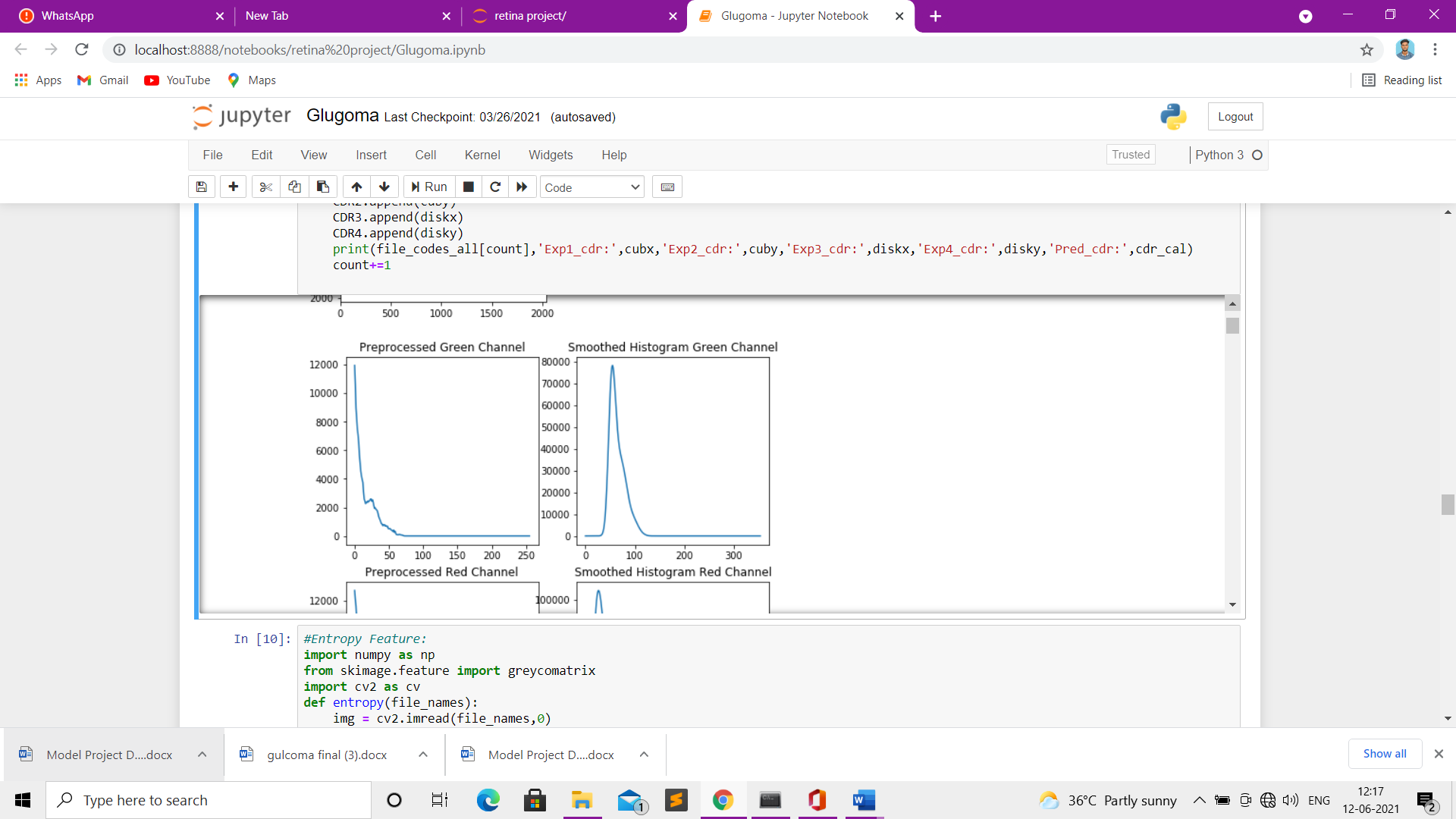
**7.3 TEST CASES & REPORTS / PERFORMANCE ANALYSIS**



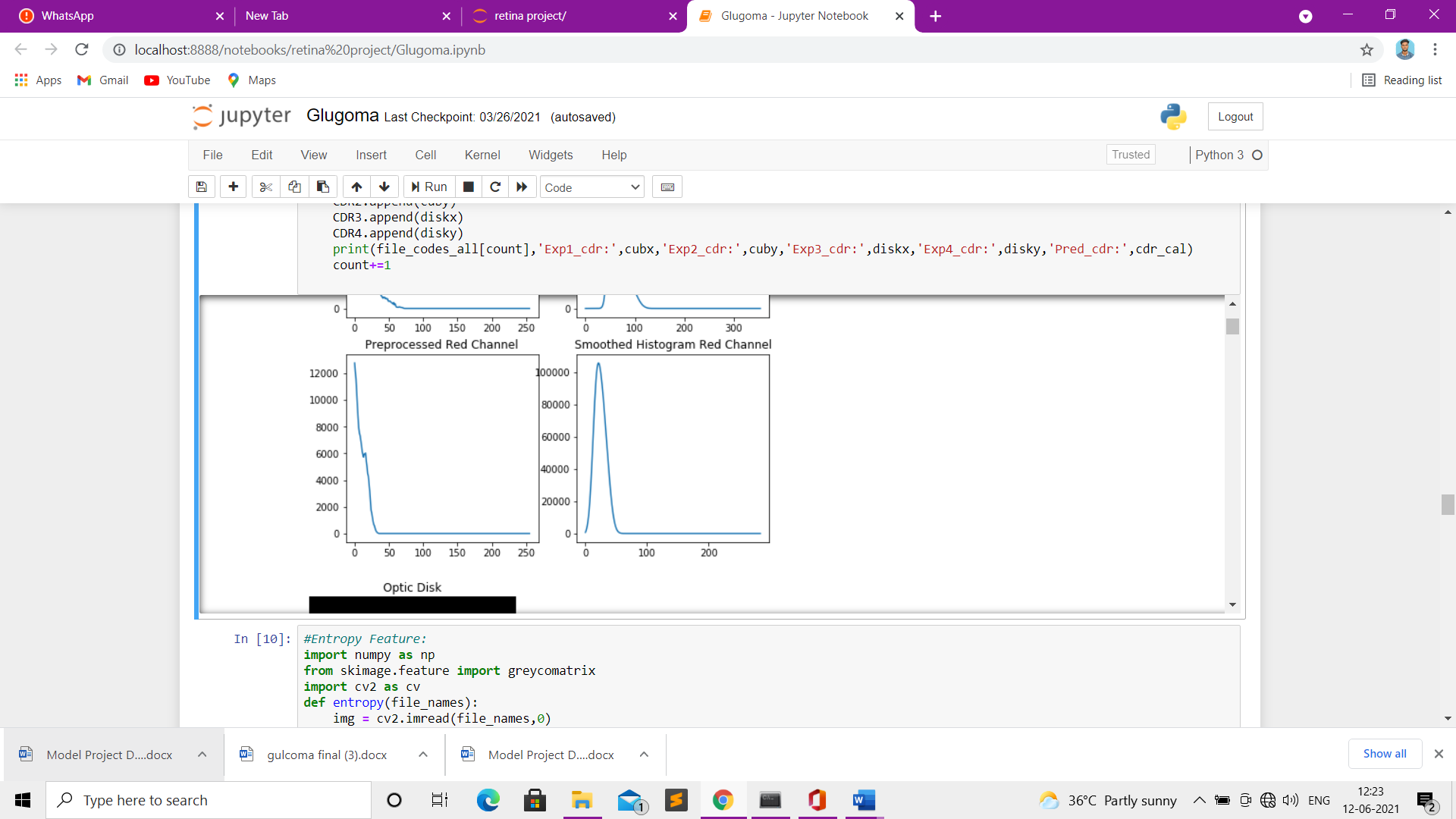
 **Fig 7.3.1 Primary Eye Screenshot**

**Fig 7.3.2 Black Scale Screenshot**

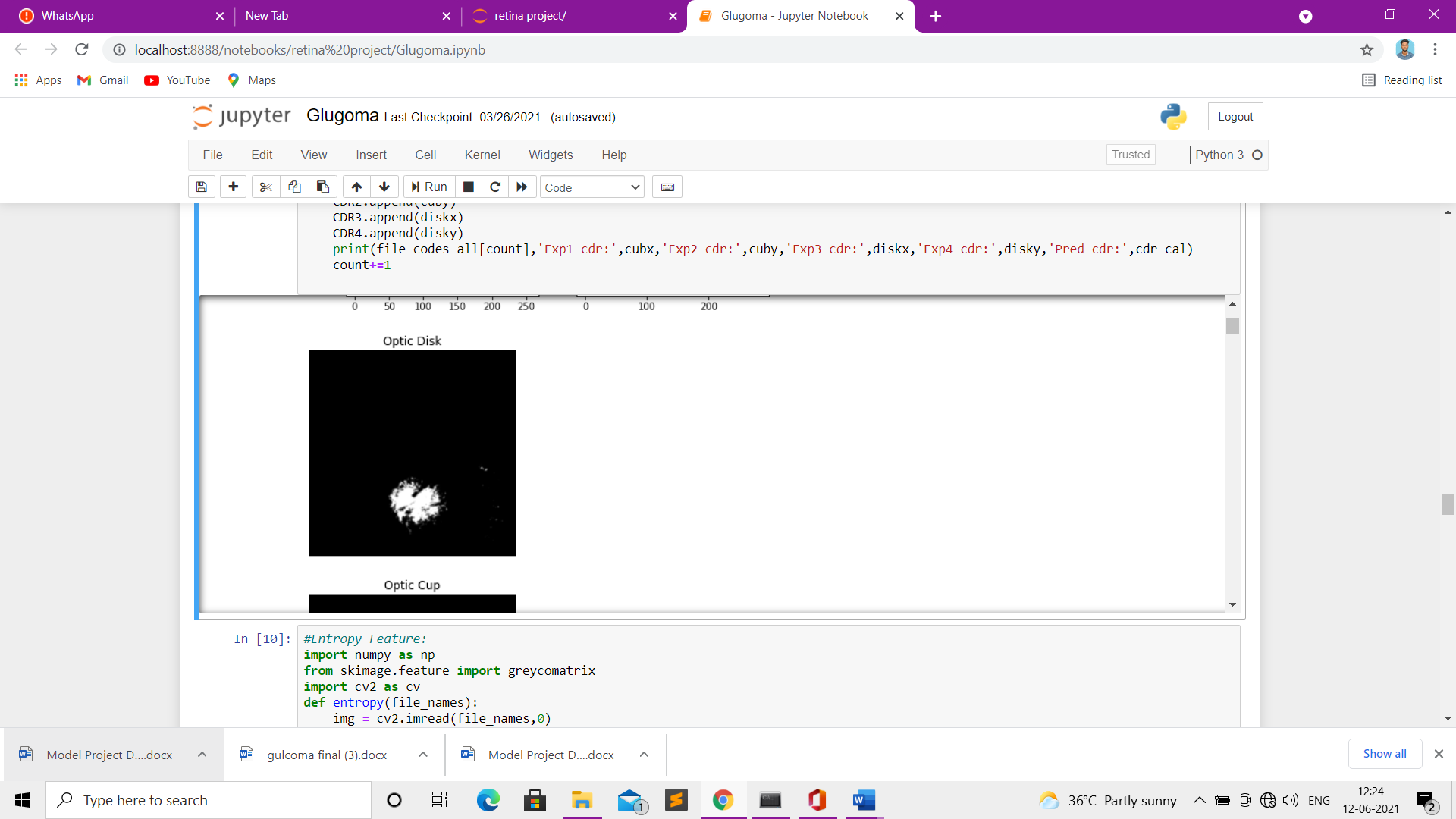
 **Fig 7.3.3 Gray scale Screenshot**



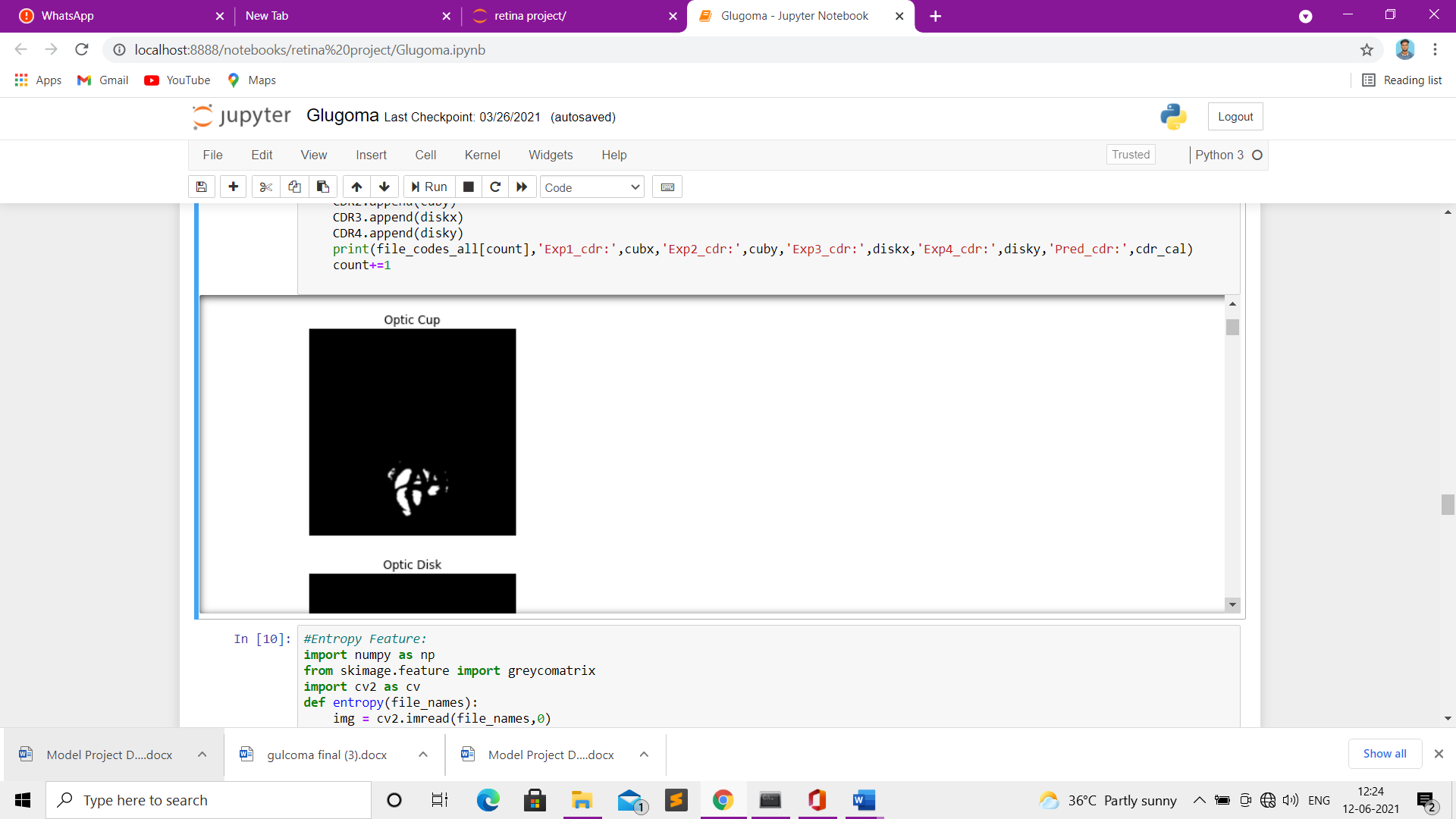
**Fig 7.3.4 Preprocessed green and Smoothed Histogram Green channel Screenshot**

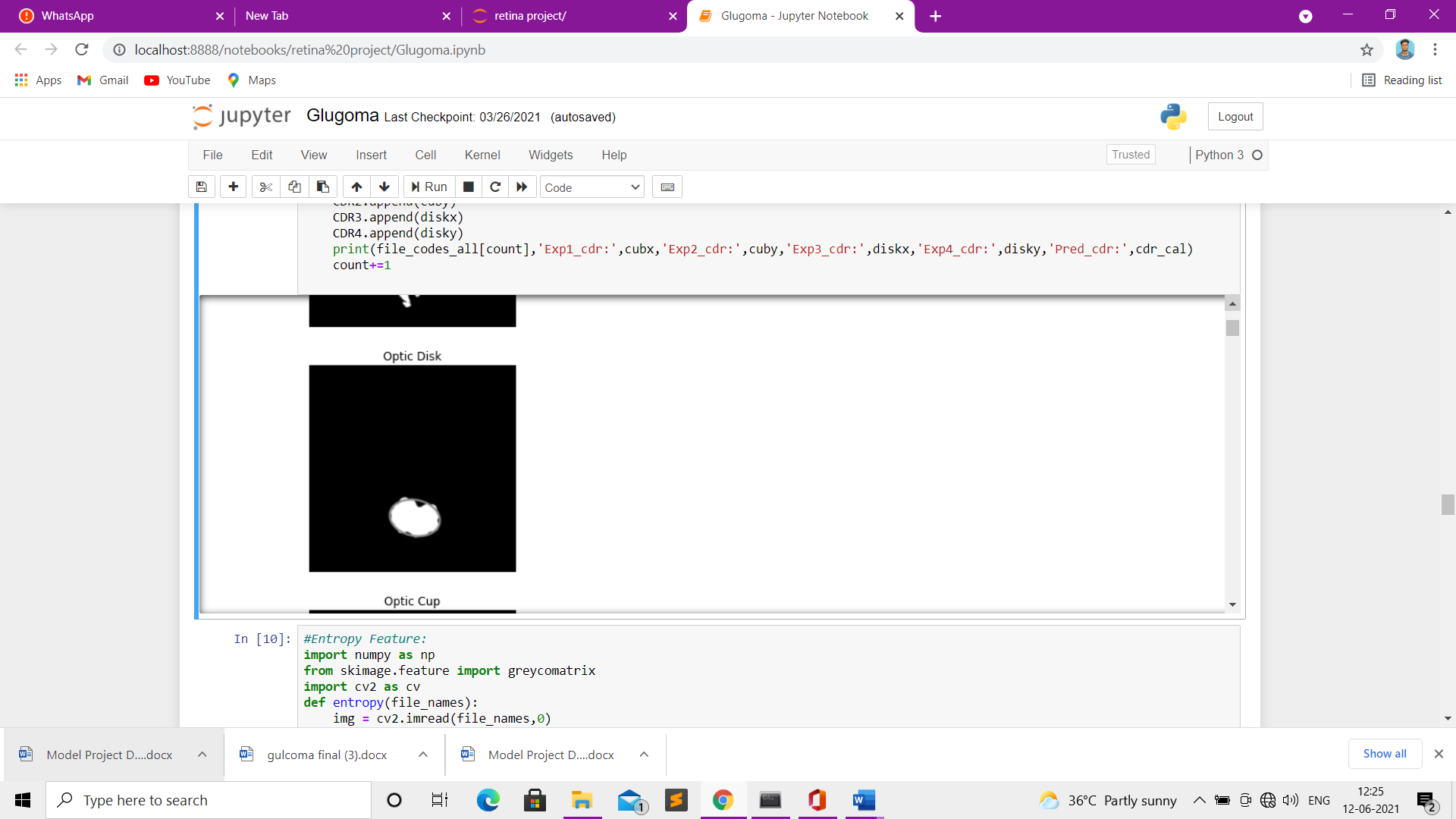


**Fig 7.3.5 Preprocessed Red and Smoothed Histogram Red channel Screenshot**

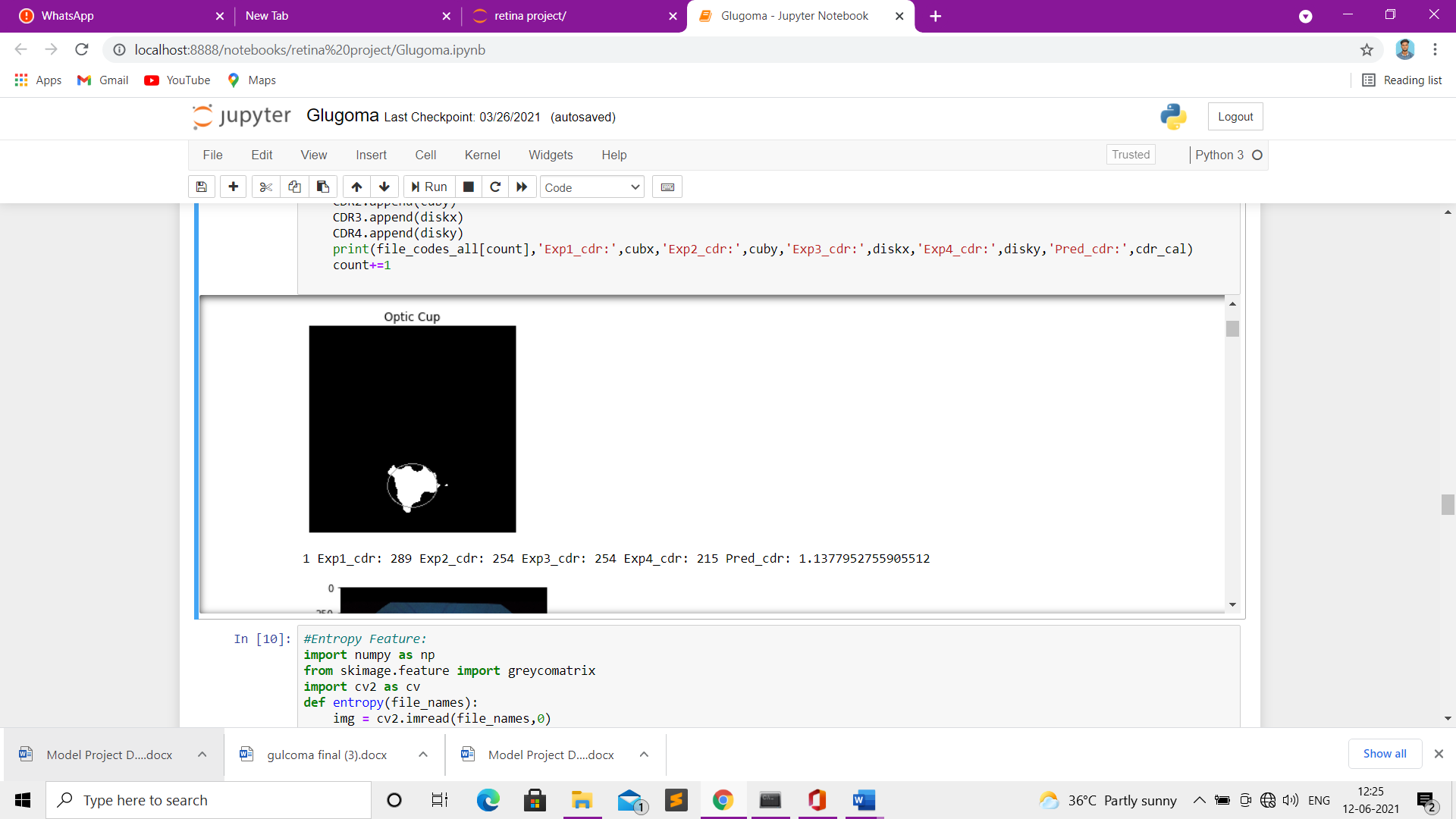


**Fig 7.3.6 Normal Optic Disk Nerve**

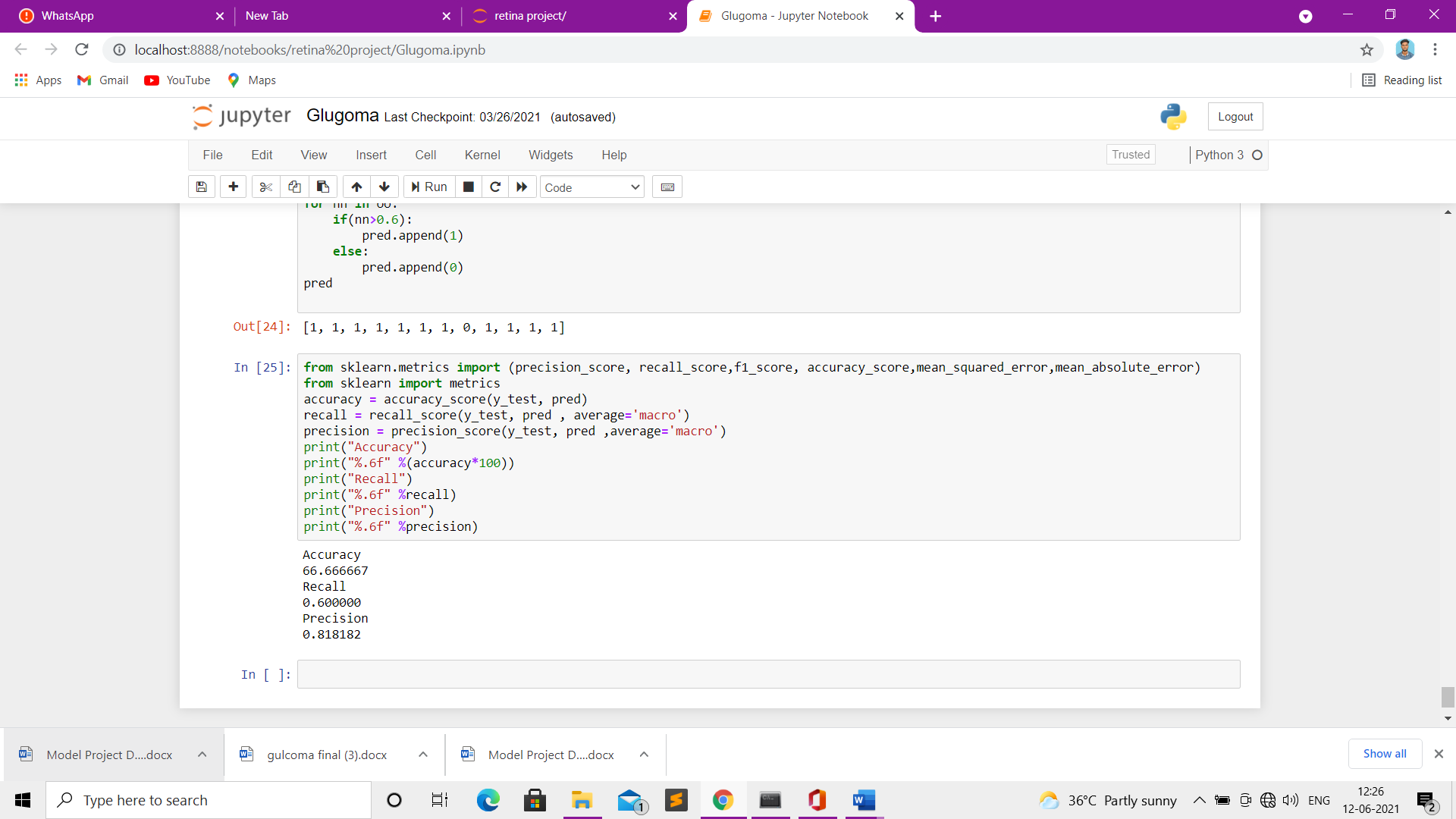
**Fig 7.3.7 Normal Optic Cup Nerve**



**Fig 7.3.8 Normal Optic Disk Nerve**



**Fig 7.3.9 Advanced Optic Cup Nerve**



**Fig 7.3.10 Output ScreenShot**

**CHAPTER 8**

**CONCLUSION**

**CHAPTER 8**

**CONCLUSION**

**8.1 CONCLUSION AND FUTURE ENHANCEMENTS**

**8.1.1 CONCLUSION**

This study brings evidence that deep neural networks are available methodology for medical imaging, even though they solve the task in question in a different way than virtually all well documented past work. We find this encouraging, in particular. This study brings evidence that deep neural networks are a viable methodology for medical imaging, even though they solve the task in question in a different way than virtually all well-documented past work. We find this encouraging, in particular given the entirely supervised character of the neural approach, which learns from raw pixel data and does not rely on any prior domain knowledge on vessel structure. While learning, a network autonomously extracts low-level features that are invariant to small geometric variations, then gradually transforms and combines them into higher order features. In this way, the raw raster image is transformed into a more abstract and a priori unknown representation that fosters effective vessel segmentation. The features learned at multiple levels of abstraction are then automatically composed into a complex function that maps an input patch to its label. An average accuracy of 95.64% is determined in the classification of blood vessel or not. Optic cup is also segmented from the optic disc by Fuzzy C Means Clustering (FCM). This proposed algorithm is tested on a sample of hospital images and CDR value is determined. The obtained values of CDR is compared with the given values of the sample images and hence the performance of proposed system in which Recurrent Neural Networks for segmentation is employed, is excellent in automated detection of healthy and Glaucoma images. given the entirely supervised character of the neural approach, which learns from raw pixel data and does not rely on any prior domain knowledge on vessel structure. While learning, a network autonomously extracts low-level features that are invariant to small geometric variations, then gradually transforms and combines them into higher order features. In this way, the raw raster image is transformed into a more abstract and a priori unknown representation that fosters effective vessel segmentation. The features learned at multiple levels of abstraction are then automatically composed into a complex function that maps an input patch to its label. An average accuracy of 95.64% is determined in the classification of blood vessel or not. Optic cup is also segmented from the optic disc by Fuzzy C Means Clustering (FCM). This proposed algorithm is tested on a sample of hospital images and CDR value is determined. The obtained values of CDR is compared with the given values of the sample images and hence the performance of proposed system in which Recurrent Neural Networks for segmentation is employed, is excellent in automated.

* + 1. **FUTURE ENHANCEMENTS**
* We will convert that model into real time application for detection of GLAUCOMA. The application will be easy to access to each perso

**REFERENCES**

**REFERENCES**

1. Donghyeon Kim\*, Heejin Park, Jeasoo Kim, Joo Young Hahn, Joung-Soo Park ” MIMO Communication Based on Adaptive Passive Time Reversal in Deep Water” IEEE access 2019
2. M. Leeson and M. Higgins, “Optical wireless and millimeter waves for 5g access networks,” in The Fifth Generation (5G) of Wireless Communication. IntechOpen, 2018.
3. Jeneel Pravin Kachhadiya, Abhinav Kashyap “Air-Water Optical Communication using an Airborne Radio controlled Aircraft” IEEE access 2018
4. T. D. P. Perera, D. N. K. Jayakody, S. K. Sharma, S. Chatzinotas, and J. Li, “Simultaneous wireless information and power transfer (swipt):Recent advances and future challenges,” IEEE Communications Surveys & Tutorials, vol. 20, no. 1, pp. 264–302, 2017.
5. Yuezhong Li, Xiaoqiang Yan, Lingyuan Zeng, Hualing Wu”Research on Water Meter Reading System Based on LoRa Communication” IEEE access 2017
6. H. Kaushal and G. Kaddoum, “Underwater optical wireless communication,” IEEE access, vol. 4, pp. 1518–1547, 2016.
7. C. Gussen, P. Diniz, M. Campos, W. A. Martins, F. M. Costa, and J. N. Gois, “A survey of underwater wireless communication technologies,” J. Commun. Inform. Sys, vol. 31, no. 1, 2016.