

Improved Convolutional neural network Based on Guided filter for Glaucoma Detection

A PROJECT REPORT

Submitted by

Rishali Rishu : 19BEC1108

Sushant Kumar : 19BEC1078

in partial fulfillment for the award of the degree of

BACHELORS OF ENGINEERING

in

ELECTRONICS AND COMMUNICATION ENGINEERING



Chandigarh University

November 2022

Improved Convolutional neural network Based on Guided filter for Glaucoma Detection

**A report submitted in partial fulfilment of the
requirement for the award of degree of**

**BACHELORS OF ENGINEERING
in
ELECTRONICS AND COMMUNICATION ENGINEERING**

**Submitted By
Rishali rishu 19BEC1108
Sushant Kumar : 19BEC1078**

**under the guidance of
Dr. Sonali Dash**

Electronics and Communication Engineering





BONAFIDE CERTIFICATE

Certified that this project report **Improved Convolutional neural network based on Guided filter for Glaucoma Detection** is the bonafide work of **Rishali Rishu (19BEC1108) and Sushant Kumar (19BEC1078)** who carried out the project work under my/our supervision.

SIGNATURE

Dr. Ashutosh Tripathi
HEAD OF THE DEPARTMENT

Electronics and Communication

SIGNATURE

Dr. Sonali Dash
SUPERVISOR

Electronics and Communication

Submitted for the project viva-voce examination held on

INTERNAL EXAMINER

EXTERNAL EXAMINER

TABLE OF CONTENTS

List of Figures.....	i
List of Tables	ii
Abstract.....	iii
Graphical Abstract	iv
Abbreviations.....	v
Symbols	vi
Chapter 1.....	1
1.1. Identification of Relevant Contemporary Issues.....	1
1.2. Identification of Problem.....	2
1.3. Identification of Tasks.....	5
1.4. Timeline.....	8
1.5. Organization of the Report.....	9
Chapter 2.....	10
2.1. Timeline of the Reported Problem.....	10
2.2. Proposed Solution.....	12
2.3. Problem Definition.....	13
2.4. Goal/Objective.....	15
Chapter 3.....	14
3.1. Evaluation and Selection of Specification/Features.....	14
3.2. Design Constraint.....	15
3.3. Design Flow.....	16
3.4. Design Selection.....	17
3.5. Implementation Plan/ Methodology.....	20
Chapter 4.....	34
4.1. Implementation of Solution.....	34
4.2. Coding simulation.....	34
4.3. Results and Discussions.....	45

Chapter 5.....	50
5.1. Conclusion.....	50
5.2. Future Work.....	50
References.....	51
ECE ARCHIVES PROJECT SUBMISSION FORM.....	55
User Manual.....	56

List of Figures

Fig 1.1.1	Projection of Glaucoma by 2050	1
Fig 1.2.1	Anatomy Of Human an Eye	3
Fig 1.2.2	Types of Glaucoma	4
Fig 1.3.1	Difference between a Healthy eye and a Glaucomatous Eye 20	5
Fig 1.3.2	Fundus image 21	6
Fig 1.3.3	Major optical disc structures The neuroretinal rim is the area between the red and green lines, which represent the boundaries of the optic disc and the cup, respectively.	6
Fig 2.2.1	Images of the fundus that highlight the Optic Disc, Optic Cup, blood vessels, and Neuro-retinal Rim	13
Fig 3.4.1	Design Flow 1	17
Fig 3.4.2	Design Flow 2	18
Fig 3.5.1	Final Design Flow	19
Fig 3.6.1	Maps with soft Classification samples and marks. 4 black expert marks for the OD (left) and cup in the top row (right) OD (red) and cup (green) borders were averaged. Soft classification maps in the bottom row	21
Fig 3.6.2	Examples of Images from Training set for Healthy Eyes	22
Fig 3.6.3	Examples of Images from Training set for Glaucomatous Eyes	23
Fig 3.6.4	Examples of Images from Testing set for Healthy Eyes	23
Fig 3.6.5	Examples of Images from Testing set for Glaucomatous Eyes	24
Fig 3.6.6	Examples of Mask	24
Fig 3.6.7	Fundus images after applying filter masks	25
Fig 3.6.8	sigmoid function Curve	26
Fig 3.6.9	Tanh vs Sigmoid function Curve at a glance	27
Fig 3.6.10	ReLU vs Sigmoid function Curve at a glance	27
Fig 3.6.11	U-Net architecture	29
Fig 3.6.12	U-Net for Classification Of fundus Images	30

Fig 3.6.13	Vgg-16	31
Fig 3.6.14	VGG16 Model Architecture	31
Fig 3.6.16	VGG-16	41
Fig 4.1	Fundus image with OD and OC mask applied	45
Fig 4.2	Fundus Images with all Masks applied and neatly displayed	46
Fig 4.3	An example of fundus image after passing through the guided filter	46
Fig 4.4	Graphical Representation of the accuracy	48

List of Tables

Table 4.1	Results obtained from the suggested method	47
------------------	--	----

Acknowledgment

I really appreciate that I got an opportunity to express my hearty gratitude and respect to everyone who helped me in every stage and helped me complete my BE project, without them I wouldn't be able to complete this project.

It is difficult to overstate my greatest gratitude to **Dr. Sonali Dash**, and ECE Department. I want to start by expressing my gratitude to her for her patient guidance and inspiration throughout my study period. Second, I sincerely appreciate their encouragement and assistance with my project work, which gave me the bravery and confidence to face challenges. Finally, I want to express my gratitude for their helpful advice and suggestions as well as for investing so much time in the project. Without their assistance, I would have been lost.


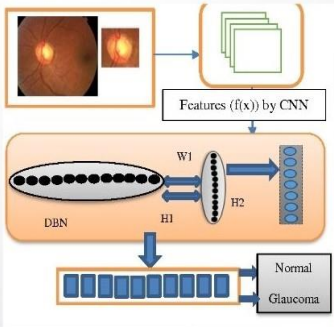
I owed (Dr.) Ashutosh Tripathi, the head of the ECE department, for his tremendous support during every stage of my academic career and yearned to see this accomplishment realized.

ABSTRACT

A series of illnesses known as glaucoma harm the optic nerve in the eye. Glaucoma can cause lifelong blindness or vision loss if it is not addressed. Because glaucoma frequently has no symptoms in its early stages, it is particularly difficult. If symptoms start to show up, it might be too late to stop blindness. There are numerous ways to identify glaucoma, including tonometry, which measures the pressure inside the eye, ophthalmoscopy, which looks at the optic nerve's form and color, and perimeter, which measures the entire field of vision. However, because to the fact that each person's glaucoma is unique, these procedures do not allow for the detection of all forms of glaucoma. A visual assessment technique that can identify glaucoma is cup-to-disc ratio. By image processing methods like binarization the cup-to-disc ratio is calculated in this project and utilized to assess the glaucomatous status of an eye using super pixel classification.

GRAPHICAL ABSTRACT

Improved Convolutional neural network Based on Guided filter for Glaucoma Detection

PROBLEM	SOLUTION	CONCLUSION
<div></div> <p>An estimated 57.5 million people worldwide are affected</p> <p>Glaucoma is sometimes called the “silent thief of sight” because it slowly damages the eyes and can cause irreparable harm before there is any vision loss</p>	<p>The damage caused by glaucoma can't be reversed. But treatment and regular checkups can help slow or prevent vision loss That's why early detection of glaucoma is very Necessary</p> <p>Several methods exists for Classification Of Glaucoma with any small iterations which can make a lot of difference in the results.</p> <div></div> <p>AI Models can be implemented for automated Detection of Glaucoma</p>	<p>For Imprvement in the Network a thorough comparison between various Functions is to be made between various functions used and the difference in accuracy if any filter is applied to it or not.</p>

ABBREVIATIONS/SYMBOLS

S.No	Abbreviations	Full Form
1	OC	Optic Cup
2	OD	Optic Disk
3	OCR	Optic Cup ratio
4	ONH	Optic nerve hypoplasia
5	CNN	Convolutional Neural Network
6	ROI	Region of interest
7	DWT	Discrete wavelet transform
8	RDR	Rim to Disk ratio

CHAPTER-1

INTRODUCTION

1.1 Identification of relevant Contemporary issue

The terminology "glaucoma" refers to a series of eye illnesses that cause progressive and permanent damage to the optic nerve, the nerve in the eye that controls vision, and which, if left untreated, can cause permanent blindness. Increased ocular pressure is the key reasons, however even persons with normal eye pressure might develop glaucoma. The two most prevalent kinds of glaucoma, according to the WHO, are angle closure glaucoma (ACG), which is less frequent and typically manifests itself more severely, and primary open angle glaucoma (POAG), which starts off slowly and sneakily.

Glaucoma is the second most common cause of blindness in the world. The WHO estimates that glaucoma causes the blindness of 4.55 million people Minimum 12 million individuals in India are affected with glaucoma, which leaves about 1.2 million of them legally blind. In India, glaucoma is the main factor causing irreversible blindness. Over 90% of glaucoma cases in the general community are untreated. Glaucoma prevalence is influenced by age. Additionally, a new study estimates that the number of glaucoma patients in the nation is higher than what ophthalmologists believe [1, 2].

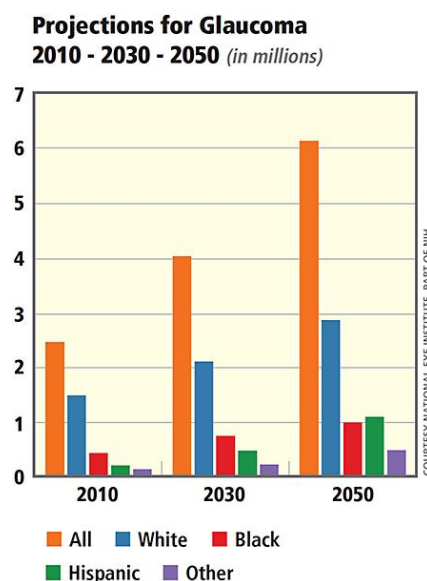


Fig. 1.1.1 Projection of Glaucoma by 2050

As observed in the above image it is clear that Glaucoma will spike in numbers in the upcoming years and it's a global Eye condition which can affect anyone at any stage in their life. Loss of eyesight due to glaucoma cannot be recovered. Further eyesight loss may be stopped with treatment and/or surgery. Open-angle glaucoma requires ongoing monitoring because it is a chronic condition. The first step in keeping our vision is diagnosis. Everyone, from young children to elderly people, is susceptible to glaucoma. There is a higher chance of glaucoma in older adults, although it can also be present before birth. Glaucoma can affect young adults as well. Particularly African Americans are more prone when they are younger. The most prevalent type of glaucoma, open-angle glaucoma, has almost no symptoms. Typically, increased ocular pressure causes little pain. Loss of vision starts with lateral or peripheral vision unconsciously compensating for this by moving our head to the side may prevent us from noticing anything until considerable vision loss has occurred [3]. Getting examined is the greatest method to safeguard our vision from glaucoma. Treatment for glaucoma can start right away.

1.2 Identification of Problem

The human eye is a special sense organ that can take in visual information and send it to the brain. The result of attaching a significant amount of a huge, less strongly curved sphere to a small, strongly curved portion of a larger sphere, the eyeball is not a simple spherical. The cornea, which makes up about one-sixth of the whole structure, is the tiny, transparent section with a radius of 8 mm (0.3 inch). The scleral segment, which makes up the remaining portion, is opaque and has a radius of 12 mm (0.5 inch). The limbus is the ring that connects the two regions. Since the cornea is transparent, while looking directly into the eye from the front, one perceives a ring of tissue within the eye instead of the cornea due to the white sclera surrounding it.

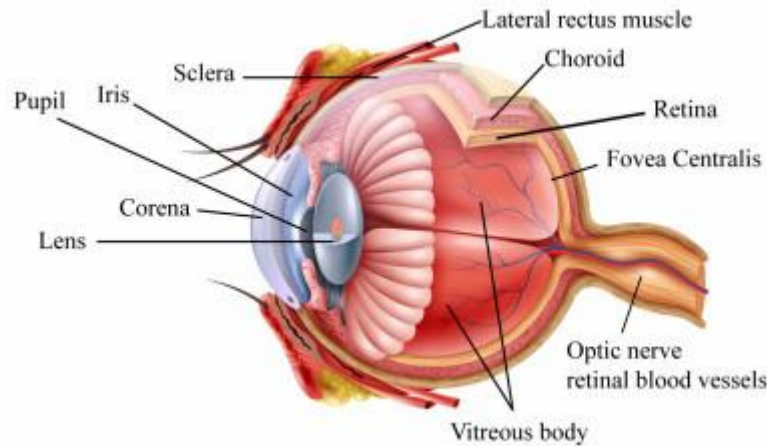


Fig 1.2.1 Anatomy of Human an Eye

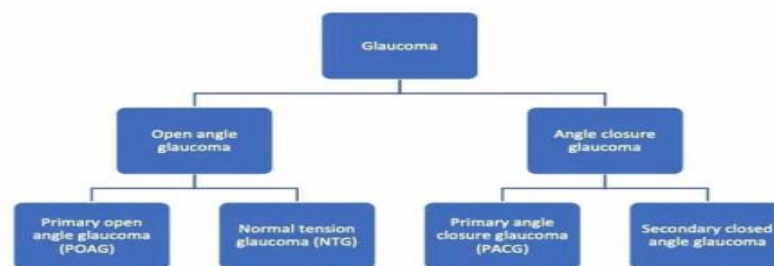
The eyeball is not a simple sphere, but rather the consequence of joining a sizable section of a large, less strongly curved sphere to a small, strongly curved portion of a larger sphere. The cornea is the small, transparent portion with a radius of 8 mm (0.3 inch), which makes up about one-sixth of the entire structure. The scleral segment, which makes up the remaining portion, is opaque and has a radius of 12 mm (0.5 inch). The limbus is the ring that connects the two regions. Since the cornea is transparent, while looking directly into the eye from the front, one perceives a ring of tissue within the eye instead of the cornea due to the white sclera surrounding it. Iris of the eye. The component that determines the color of the eye is the iris. The pupil is the term for this ring's centre. It appears gloomy because little or no light entering the eye is reflected back. The appearance of the inside lining of the globe can be discerned using an ophthalmoscope, a device that enables the view our to illuminate the interior of the eyeball while looking through the pupil. The major blood veins that carry blood to the retina are known as the fundus oculi, and they are particularly noticeable because they cross across the pallid optic disc, also known as the papilla, which is where the optic nerve fibres exit the eyeball [4].

Approximately 24 mm (about one inch) is the sagittal (vertical) diameter of the eye, and it is frequently smaller than the transverse diameter. Measurements of the eye are reasonably stable, varying by roughly a mm or two between healthy individuals. When a child is born, the sagittal diameter is roughly 16 to 17 mm (0.65 inch), but it quickly grows to 22.5 to 23 mm (0.89 inch) by the time they are three, and it achieves its maximum size between the ages of three and thirteen. It has a volume of approximately 6.5 cm³ and a weight of approximately 7.5 grammes (0.25 ounce) (0.4 cubic inch) [4].

The aqueous fluid is encircled by the three coats of the eye, lens, and vitreous body, which

are all optically transparent. The intermediate coat, which contains the majority of the eye's blood, is made up of the choroid, ciliary body, and iris, while the cornea and sclera make up the outermost coat. The choroid is where the retina, the body's thinnest layer, is found. The retina gets most of its sustenance from the choroid's blood vessels. Retinal vessels on the retina's surface provide the remaining sustenance; these vessels can be seen with an ophthalmoscope [5, 6, 7, 8].

Attachment A: Types of Glaucoma



Attachment B: Characteristics of POAG and PACG

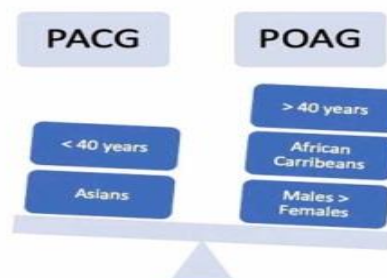


Fig 1.2.2 Types of Glaucoma

To enable proper monitoring and treatment, as well as to reduce the risk of irreversible visual field loss, early identification of glaucoma is crucial. Considering that numerous studies have been performed and numerous more are being conducted in numerous locations across the globe, e decided to do something original in the sector as well.

First, we decided to make a hybrid model to compare a few metrics before moving on to the detection phase. Based on that decision, we then decided to move forward with the detection phase. In our preparatory phase, we studied various models and methods available and discovered that many of the ones being used today are not very defined.

1.3 Identification of Tasks

In real world the glaucoma test is quick and painless. Our vision will be evaluated by an eye doctor. Our pupils will be widened (dilated) with drops when they inspect our eyes. Our optic nerve will be examined for glaucoma symptoms. They might take pictures so they can document changes on our subsequent appointment. To check our eye pressure, a procedure known as tonometry will be performed. They may also undertake a visual field test to discover if we've lost peripheral vision. Our doctor may request specialized imaging examinations of our optic nerve if they have glaucoma suspicions.

But to make this task much easier we make use of technology in this field, for this we make use of AI/ML to design and develop good enough models to detect and diagnose Glaucoma and hence making it easier for the doctor to advice proper medication [9].

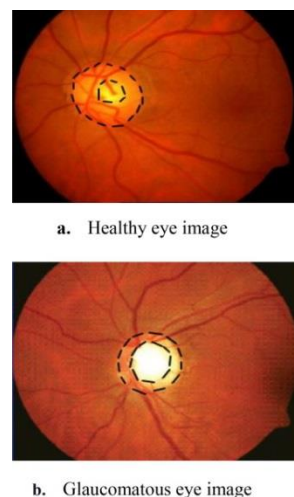


Fig 1.3.1 Difference between a Healthy eye and a Glaucomatous Eye

The accuracy of manual ophthalmic image analysis depends on the specialists' experience and takes a good amount of time. An essential tool is automatic analysis of retinal pictures. Automation helps with disease-related risk detection, diagnosis, and prevention. The analysis has made use of fundus pictures captured by a fundus camera. The image has undergone the necessary pre-processing processes, and depending on the approach, several classifiers have been employed to identify glaucoma. The accuracy of manual ophthalmic image analysis depends on the specialists' experience and takes a good amount of time. An essential tool is automatic analysis of retinal pictures. Automation helps with disease-related risk detection, diagnosis, and prevention. The analysis has made use of fundus pictures captured by a fundus camera [10].



Fig 1.3.2 Fundus image

The time- and money-consuming task of calculating the CDR is now exclusively done by professionals. Therefore, automatic Glaucoma assessment will be very beneficial. There are two ways to recognize the head of the optic nerve automatically. The first approach is based on the incredibly challenging issue of extracting visual information for the binary categorization of normal and abnormal situations. Nevertheless, the second method is more often used and is based on clinical signs like the cup-to-disc ratio and the inferior, superior, nasal, and temporal (ISNT) zones rule in the optic disc area.

The optic disc is formed by 1.2 million Ganglion cell axons as they exit the eye through the scleral canal to transmit visual information to the brain. Examining the optic disc helps to clarify the relationship between optic nerve cupping, vision loss, visual field and glaucoma. The neuroretinal rim, the cup (central area), and occasionally parapapillary atrophy are the three different parts that make up the optic disc. The cup-to-disc ratio is determined by comparing the vertical diameter of the cup to the vertical diameter of the disc (CDR) [11].

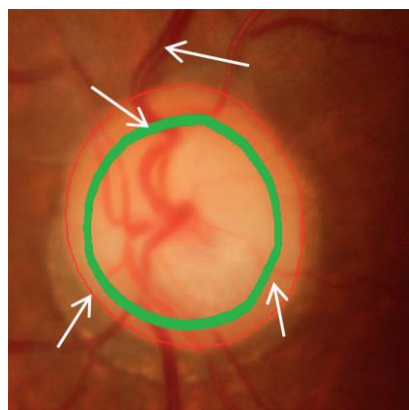
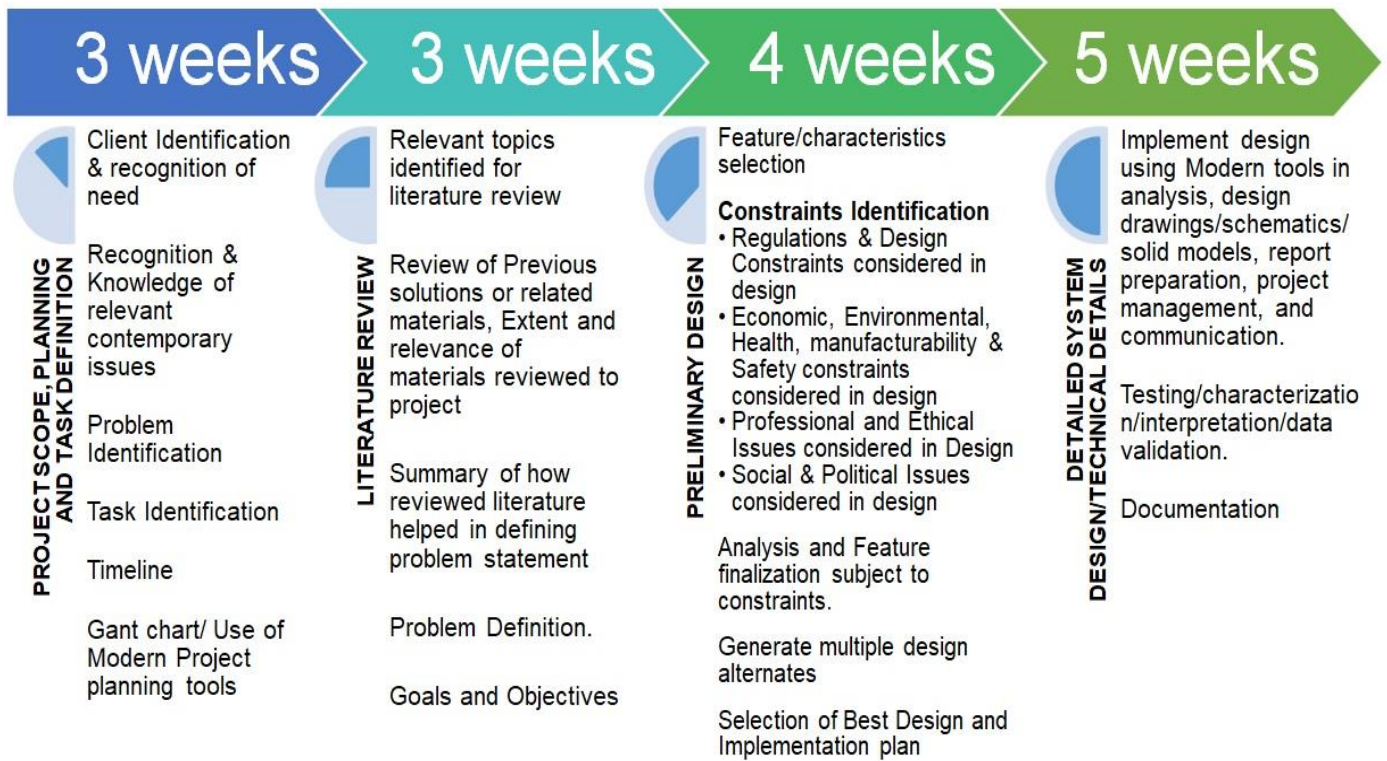


Fig 1.3.3 major optical disc structures The neuroretinal rim is the area between the red and green lines, which represent the boundaries of the optic disc and the cup, respectively.

Keeping everything written above in mind entire project can be dotted down in sequential tasks

- Information gathering/ learning: - The first step in designing a Detection Model
- Is to gather information. Many things need to be taken into consideration while selecting the dataset to be used for the model training. So we decided first to gather information about structure, design, overview, features, functionalities, and complexities of the project.
- Design: - After developing wireframes and planning a roadmap we will move to designing. We will create one or more models to test the accuracy for the given dataset and move ahead with the best model
- Execution :-
- Initial Execution: Run/train the model using the test dataset's initial parameters while describing our methodology and findings. Include the results in the report in the most comprehensible format possible.
- Beginning Evaluation: Assess these initial conditions in light of the scenario's validation dataset and our anticipated outcomes. How well do our findings match the validation dataset?
- Parameter Adjustments: Alter parameters to see how they affect the model's accuracy. What impact did altering the parameters have on our findings? Have any trends or patterns caught our attention?
- Model Deployment - Unlike software or application deployment, model deployment is a different. A simple ML model lifecycle would have stages like Scoping, Data Collection, Data Engineering, Model Training, Model Validation, Deployment, and Monitoring.
- Debugging/ Testing – We'll debug system to detect and remove existing and potential bugs in software that can lead to failure. Further testing our model manually and using automated software to ensure its proper functioning.

1.4 Timeline



1.5 Organization of the Report

Chapter-1: -

Here, we discuss the problems with the existing health system and analyzing issues using different criteria. Demonstrating the necessity for a remedy based on a survey or results from a survey. Also this work has identified the additional activities that must be done to create effective solutions.

Chapter-2

By analyzing data from across the world and compiling evidence of the incidences, we anticipate to be able to trace the problem's history back to the moment it is first reported to the world. Along with outlining the aims and objectives to work toward, we will now move on to suggesting remedies for the identified problem.

Chapter-3

The design flow and execution approaches are included in this section. Later working on incorporating other features found in the literature review. Models are tested in relation to

Chapter-4

The complete project implementation schedule will be covered. Gantt charts are used for documentation. Demonstrating the development work even further. Including outcome evidence and coded passages. Validating the data and testing the models as needed.

Chapter-5

This will also include the conclusion, which will discuss our expectations of the system, the outcomes or results following the performance, and how they differ from what is anticipated. In chapter 5, the attempt has taken to determine the cause of the discrepancy in findings. We also intend to discuss the system's potential future, which may involve proposals for expanding the solution or calling for necessary changes to the current method.

CHAPTER 2

LITERATURE REVIEW

2.1 Timeline of the reported problem

It is proposed that angle closure might have played a role in the early history of glaucoma because it is eventually recognized in antiquity as a condition of the crystalline lens. Today, it is known that the lens induces pupillary block, which results in angle closure. Angle closure is one of the more frequent glaucoma-related causes of vision loss worldwide. Incidence rates are greater in Asia. Angle closure had to be a thing back then, and medical writers may have been drawn to its dramatic depiction. We examined early accounts of glaucoma to look for evidence of angle closure [12].

- 1) Loss of vision
- 2) Anteriorly positioned bulging lens
- 3) Enlarged, fixated, or irascible pupils
- 4) The inability to treat the condition, or at least its difficulties (which is now considered to be an optic neuropathy brought on by ocular hypertension)
- 5) The presence of a glaucous (light blue, grey, or green) pupil (glaukos, the Arabic word zarqaa).

The existence of a glaucous pupillary tint. The evaluation of angle closure in modern ophthalmic training does not include the examination of pupillary color. Ophthalmoscopy, slit lamp biomicroscopy, gonioscopy, and tonometry are some of the most advanced tools we have. However, many early glaucoma descriptions mentioned a greenish tint to the pupil. The origin of this green colour is a topic of debate among historians. It has been proposed that candlelight examination is the cause. A different theory entails the deposition of "blood pigments" in the iris it is at once point suggested glaucoma with an angle closure could account for many instances of the green pupil, as observed in images of this condition due to the cataractous lens being visible due to the mid-dilated pupil, the green color is visible [13].

The first researcher to describe and publish a photograph of the glaucomatous disc's ophthalmoscopic appearance is Eduard Jaeger, the son and grandson of eminent Austrian ophthalmologists. He misinterpreted the monocular indirect ophthalmoscopy relative depth

clues and described (and drew) the glaucomatous disc as an enlargement of the papillary tissues relative to the surrounding retina [14]. A few months later, Albrecht von Graefe also went on record as having seen the papilla protrude in glaucoma. In his initial description of the glaucomatous disc, he provided a thorough explanation of the phenomena of retinal artery pulsing in the glaucomatous eye, which later proved to be a trustworthy and clinically practical observation. The term "halo glaucomatosus" refers to the ring-shaped region surrounding the "swollen" disc. By looking at the findings made in von Graefe's clinic, the issue of the allegedly bulging disc is resolved. Different observers couldn't agree upon ophthalmoscopic exams of a rabbit with a congenital fundus abnormality, i.e., a coloboma of uvea and of optic nerve, whether particular regions of the eye ground are elevated or depressed. Tissue flaws, or depressions, are discovered during the anatomical examination. Adolf Weber, one of von Graefe's assistants, is inspired to analyze the optics of monocular indirect ophthalmoscopy as a result. Adolf Weber eventually made substantial contributions to the knowledge of the mechanism of glaucoma. His examination of monocular indirect ophthalmoscopy showed numerous elements, some of which are optic and some of which are perception, which can lead to errors in estimating the fundus's relative depth.

Pathologic studies quickly supported the corrected ophthalmoscopic observation of a disc depression, which is then understood to be "pressure excavation," or the result of high pressure. This had a significant impact on von Graefe's thinking and led him to consider all known glaucoma symptoms in light of a potential connection to high pressure. Conducting this examination, he came to believe that the increased intraocular pressure is the "essence" of glaucoma rather than just a symptom [15].

In the decades that followed the development of the ophthalmoscope in the middle of the 19th century, glaucoma research advanced quickly. We are unaware of any prior accounts of neurosurgeon Harvey Cushing, MD's attempts to treat the disease while working at Johns Hopkins Hospital. The surgical records spanning the years 1896 to 1912 at the Johns Hopkins Hospital are examined. For the review, a case involving Cushing's unsuccessful surgical attempt to treat a patient with glaucoma is chosen. In 1905, Cushing removed the superior cervical ganglion from a patient who had previously undergone bilateral iridectomies and is thought to have chronic glaucoma with an acute episode. After the treatment, the patient indicated that his vision had stabilized and his pain had lessened. At the turn of the 20th century, renowned neurosurgeon Cushing began surgically treating glaucoma.

Sinthanayothin et al. presented the first technique for the automatic identification of the optic disc, fovea, and blood vessels from color fundus pictures in 1999. Automatic Classification

has since been successfully reproduced by other organisations, and it is currently believed to be necessary for algorithm-based glaucoma diagnosis from fundus pictures. Classification and structured learning appear to be the most dependable ways for analyzing fundus photos to identify glaucoma, with stated accuracy of over 95% in producing a positive diagnosis [16].

Additionally, glaucomatous and healthy eyes can be distinguished using artificial neural networks with characteristics like the cup-to-disc (C/D) ratio with an area under the receiver operating curve (AROC) of up to 0.90. AROC scores between 0.84 and 0.99 for glaucoma detection using DL algorithms have been reported by several groups. More recently, a startling sensitivity and specificity of 98% for the diagnosis of glaucoma was achieved by training a neural network on 1426 images of the fundus. 46 A detailed DL method has also been developed for calculating the degree of glaucomatous optic nerve damage from fundus images. By leveraging features from spectral-domain OCT pictures to train a DL algorithm, it demonstrates significant promise to predict neuroretinal damage from images of the optic disc.

2.2 Proposed solutions

A successful method for early glaucoma screening and diagnosis uses the crucial (CDR) feature as a structural signal to evaluate the harm to the optic nerve head (ONH). By combining a brightness criterion and a template matching technique, the method offers an efficient means to detect the optic disc (OD), even in the presence of light damages associated with clinical instances. Furthermore, U-Net architecture is used to do the classification of optic discs and optic cups. Finally, utilising Cup-to-Disc Ratio (CDR) computation, It is possible to distinguish between healthy and glaucomatous patients during glaucoma screening. Results: The proposed strategy for identifying and screening glaucoma is tested using the DRISHTI-GS1 dataset. Fifty retinal scans are provided by a qualified specialist, with labelling them as healthy or glaucomatous [17].

Initially our objective is to detect the Glaucoma image using U-Net but we then shifted to enhancing the models and studying the various model already available

We decided to check the accuracy of the advised models under different conditions

1. Performance difference with different activation functions
2. Variations in results with Fundus Images after adding a guided filter to increase the

gradient on it [18].

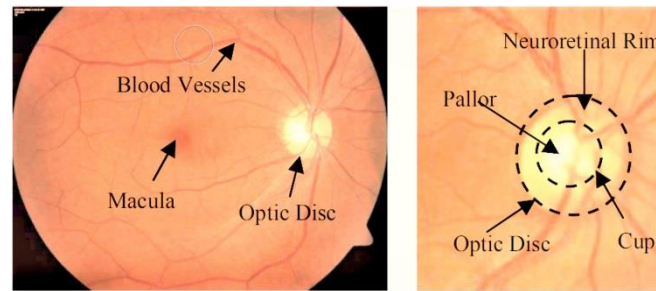


Fig 2.2.1 Images of the fundus that highlight the Optic Disc, Optic Cup, blood vessels, and Neuro-retinal Rim

2.4 Problem Definition

As we seen so far that Glaucoma is almost incurable and the only method for this is to detect it an at early stage as eyesight lost due to glaucoma remains permanent though out but if diagnosed at an early stage remaining eyesight could be saved . Hence a Solution which is able to classify Glaucoma effectively, therefore we proposed an improved Convolutional neural network Based on Guided filter for Glaucoma Detection.

2.5 Goals/Objectives

By automatically categorizing photographs of normal and glaucoma eyes based on the histograms, textures, and fractals of fundus images, this work aims to increase classification accuracy. It is necessary to research and select important criteria for glaucomatous image categorization that are more precise and sensitive than the current standard in order to boost accuracy. Eye-quality assessments can be made more accurate and reproducible. The ophthalmologist must precisely describe the optical impression of the optic nerves in order to diagnose and establish a baseline from which alterations may be detected through examination. This study on glaucoma, a potentially blinding ailment, aims to detect changes in the disease's progression and prevent future vision loss.

CHAPTER 3

DESIGN FLOW

3.1 Evaluation & Selection of Specifications

As the proposed solution is entirely software based we had to keep certain things in mind while trying to implement the solution

1. Dataset

Data Acquisition is one of the most crucial part of any AI/ML project as we shouldn't ever allow this bias to flow into the data selection sample because data is frequently biased, mostly due to the nature of the organization, the regions in which it is operated, seasonal variations, and numerous other reasons and It's not the model where we fit the problem and generate a logical consequence that determines, it's the step before selection of a right sample! Machine learning and predictive algorithms are based on the right data sample.

2. Programming Language

Why did we choose Python above other programming languages when we made our decision? Python offers readable and transparent code. Developers may build reliable systems with Python's simplicity, but machine learning and artificial intelligence are driven by complex algorithms and flexible workflows. Developers get to focus exclusively on solving an ML problem, rather than on the language's technical specifics.

Many developers find Python appealing because it is simple to learn. Python code is accessible by humans, making it simple to comprehend and use our work for future reference. This makes it easier to develop machine learning models.

2. Platform for Development

We wanted a rather flexible platform where everything is easily available and is able to provide us with very interactive tool

We first decided to use the conventional method using IDE but soon we realized that this method is very restricting as in Libraries used in the model, some are supported some are not and some require certain version of installments which might not be supported by other then we decided to shift to Online platforms such as Colab but the lack of computational power became a big issue

At last we decided to move it to kaggle.

Easy to use Notebook, ease of data acquisition, easy to debug, ease of installing dependencies and all the work can be made public with a single click.

3.2 Design Constraints

Following constraints are kept in check while developing this application

- **Health :**

Health care inputs market restrictions are typically a sign of market dysfunction, which may be brought on by the fact that these markets are highly regulated in order to address issues with information asymmetry.

- **Social:**

The Societal-Cognitive Processing (SCP) paradigm contends that social constraints (such as a lack of safe venues to express thoughts and feelings) prevent people from processing difficult life events emotionally and cognitively, which can lead to maladjustment. This study explores the connections between social restrictions and post-loss adjustment, the personal and loss-related correlates of social constraints throughout mourning, and whether social constraints have an impact on the ties between intrusive thoughts related to the loss and adjustment.

Some project constraints to consider –

- **Scope-**

Project scope, also known as the quality, level of detail, and deliverables of a project, define its size. The scale of the project determines how much time and money will be required to execute it. At every stage of the project, we must be aware of scope creep and take all required precautions to prevent it. By conducting thorough project planning and obtaining project stakeholders' approval prior to production, we may be able to prevent scope creep.

- **Time-**

1. Time management is crucial to the success of our project because there will be different time constraints for each step. There will be negative effects if we try to

extend the project timetable, such as more deadlines, changes to the team calendar, or less time for preparation.

2. Time management is crucial to the success of our project because there will be different time constraints for each step. There will be negative effects if we try to extend the project timetable, such as more deadlines, changes to the team calendar, or less time for preparation. other typical project constraints to take into account

- **Resources -**

These project criteria are closely tied to our project's financial constraints because they are pricey. Without sufficient resource allocation, projects could have higher costs, longer timeframes, and lower quality. Among the sources to take into account are:

1. People
2. Equipment or materials
3. Facilities
4. Software

Management of project Constrains –

- Recognize our restrictions:

We cannot control the restrictions of our project if we are not aware of them. Once we are aware of the project restrictions, we can plan how to go past them. Consider the risks we can face when designing a project, as well as the resources we'll require and their related expenses.

- Monitoring project quality :

Regularly reviewing our project plan and procedures will help us maintain project quality. Use work management software to monitor everyone's progress while our team handles various tasks throughout project execution. If changes are made, establish a change control method to prevent scope creep.

- Using risk analysis, identify potential project hazards, assess them, and make plans for them:

Having a strong risk management strategy in place allows us to avoid the most harmful project risks and be prepared for any unforeseen ones.

- Effective communication :

Effective team communication is essential for the successful handling of project constraints. Without clear communication, we can think we're controlling our boundaries while another team member unknowingly thwarts our meticulous work.

3.4 Design Flow

This picture below depicts the design flows for the said project

First selected design model is a simple Classification program which classifies the fundus images and gives the accuracy results accordingly , it is the most common form of classification method for Glaucoma.

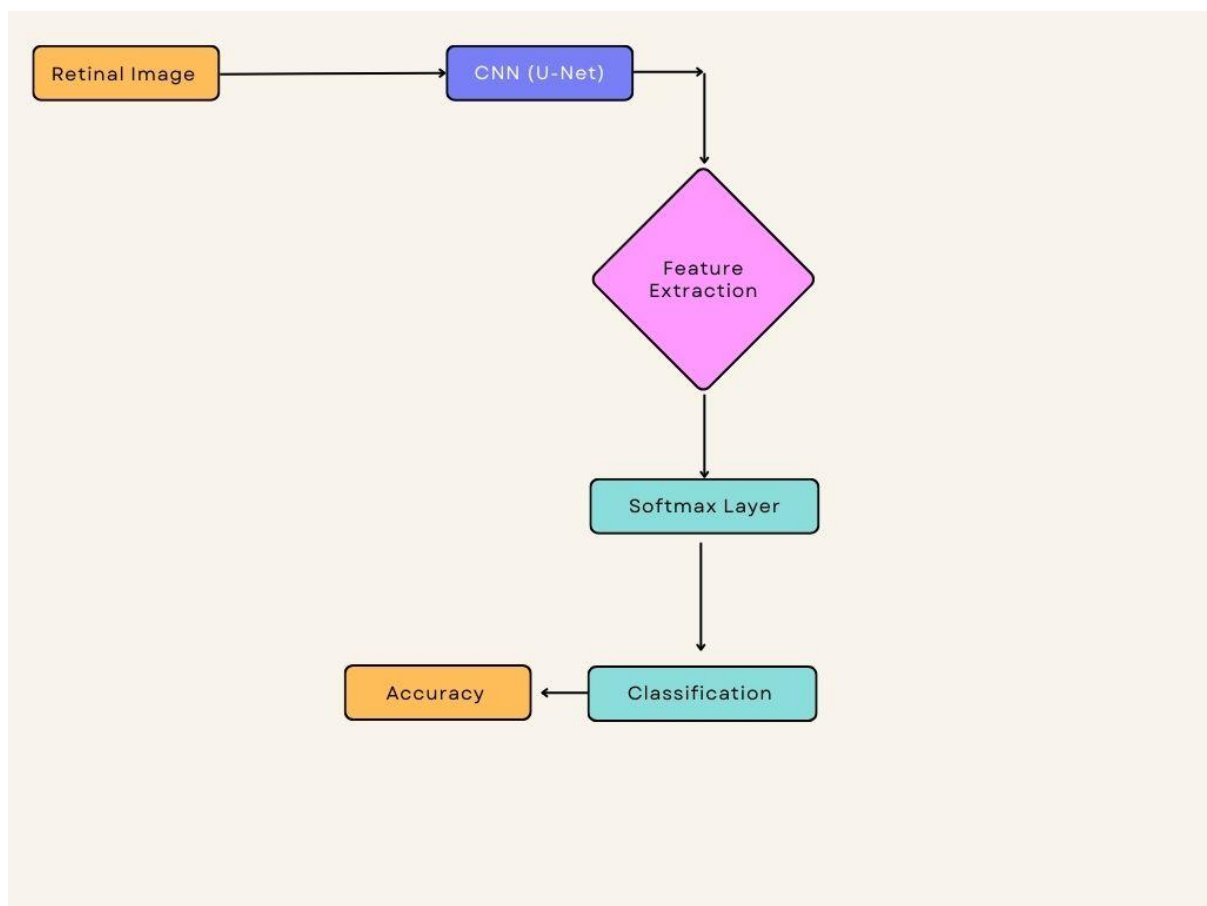


Fig 3.4.1 Design Flow 1

The Suggested method here not just classifies but also Detects glaucoma for the patient
It employs VGG-16 for feature extraction

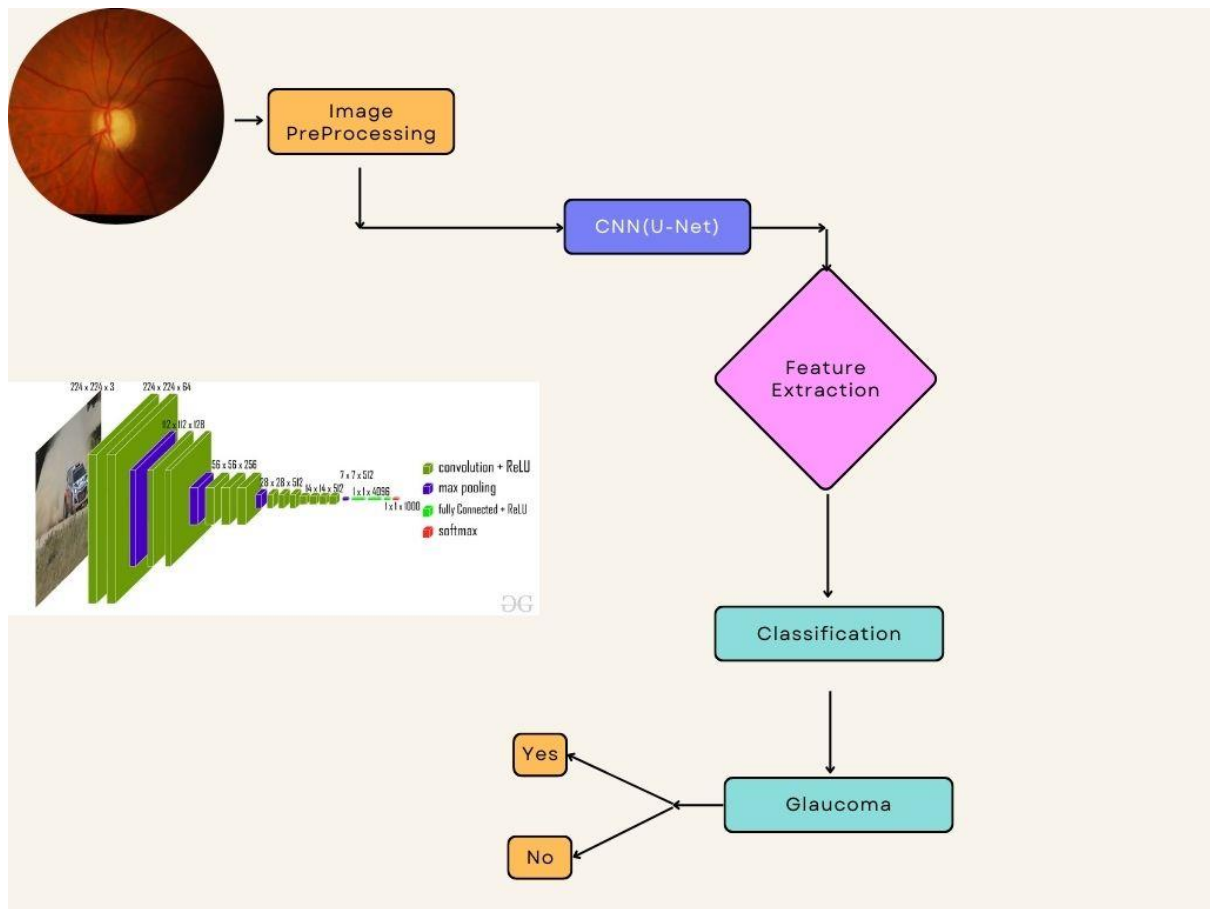


Fig 3.4.1 Design Flow 2

3.5 Design selection

Considering the various methods available and the task at hand it was important to improve the available Networks and for that we had to make comparison

As we can see in the Flowchart below in total 8 different networks are deployed out of which 4 are using a guided filter for the input images

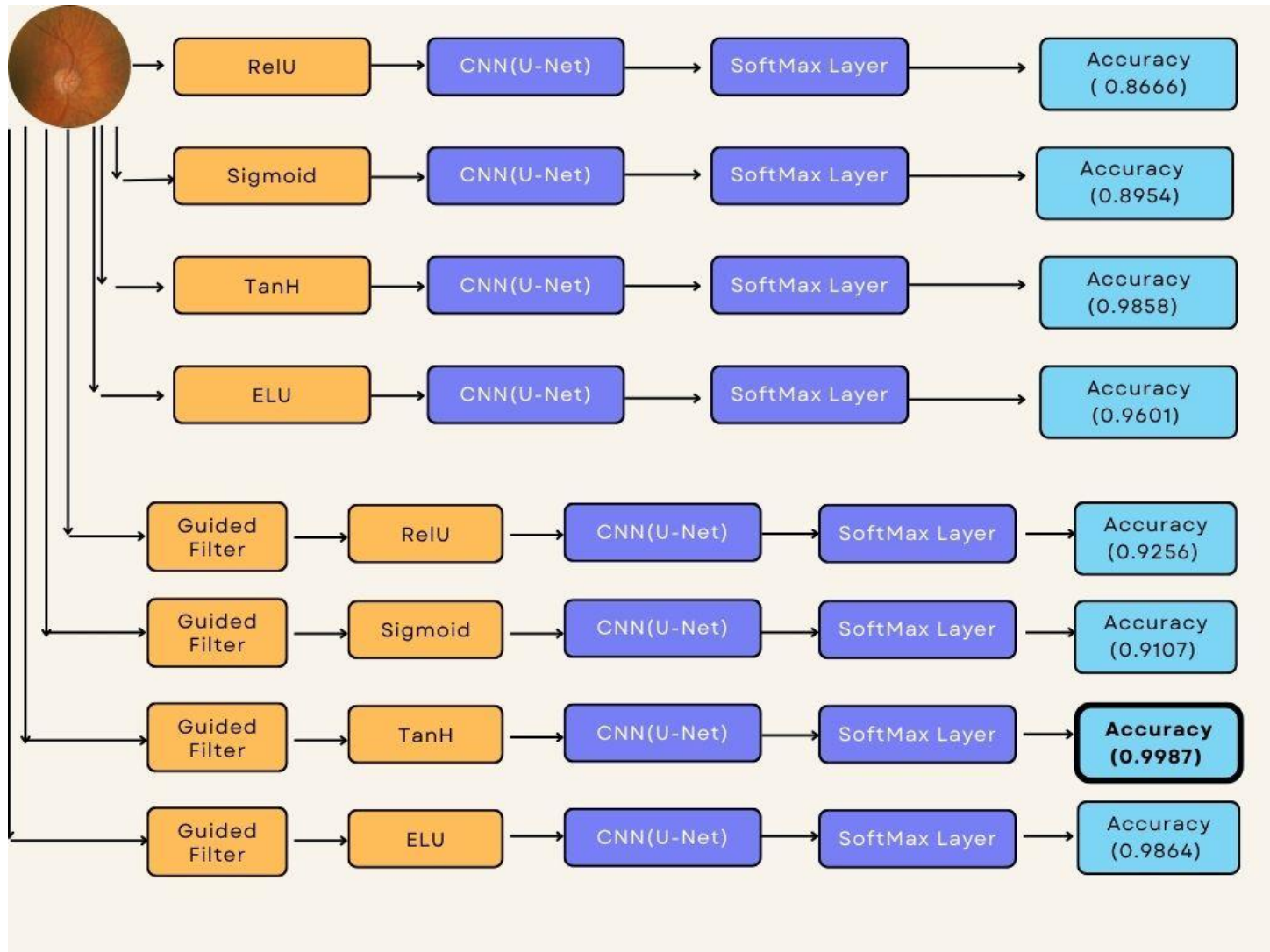


Fig 3.5.1 Final Design Flow

The above design flow is what we used for the project. It fulfils all are requirement for the said project and is having a lot of potential for the expansion of the project for the future tasks which is our main goal.

3.6 Implementation plan/methodology

Medical practitioners frequently refer to glaucoma as "silent robbery of sight" since it damages the optic nerve head and reduces peripheral vision as a result. Early glaucoma detection and classification will allow patients to receive the right care and help from their eye surgeons, improving their quality of life. The huge number of potential patients and the small pool of available ophthalmologists is a significant challenge in the management of health care. In this suggested work, a glaucoma early detection approach is put forth to help an ophthalmologist make an early diagnosis of the condition. A deep convolutional neural network analyses an image and is often used to find low-, mid-, and high-level properties.

- **Classification of Images**

Image Classification is a method that splits a digital image into a number of smaller groupings known as Image segments, which makes the original image less complicated and thus easier to manage or analyse in the future. Simply put, categorization is the process of assigning labels to pixels. Each pixel or element of a picture that falls under the same category has its own unique name. Consider the following scenario as an illustration: The input for object detection must be a photo. Instead of looking at the entire image, a portion of the image chosen by the classification system can be supplied into the detector. Through this the detector will only process a portion of the image, saving time during inference [19].

1. Retinal Image Data Set and Pre-processing

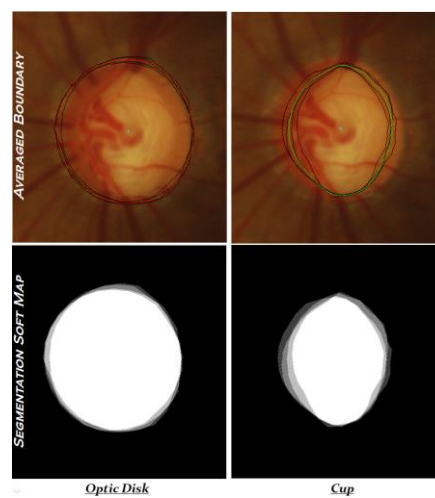
Contrary to the enormous number of vascular Classification datasets and datasets for diabetic retinopathy, there are very few public datasets for glaucoma assessment. Public datasets that only support OD Classification include Rim-one and Drions-db. The majority of the 169 photos in Rim-one are normal, and each one has undergone five hand Classifications. Drions-db. has 110 photos, each of which includes two manual annotations. Together, OD and cup Classification, two significant components of ONH Classification, serve as the foundation for glaucoma assessment. Cup Classification algorithms will be evaluated using a reference dataset, according to. The public cannot access it for free.

There are 101 images in the DRISHTI-GS dataset. It is split into 51 testing images and 50 training images.

- **Image Acquisition**

All photos are taken with the guests' permission at the Arvind Eye Hospital in Madurai. Clinical investigators chose patients with glaucoma based on their examination's clinical results. Ages of the chosen patients ranged from 40 to 80, with a fairly equal number of men and women. The normal class is selected from patients who are undergoing standard refraction testing and did not have glaucoma.

All photographs are taken while the subject's eyes are dilated and are all captured using the PNG uncompressed image format, centered on the OD, with a 30-degree field of view and a size of 2896 1944 pixels. Other than this, there are no other imaging restrictions placed on the acquisition procedure. Four sources of ground truth are gathered for each photograph.



**Fig 3.6.1 Maps with soft Classification samples and marks. 4 black expert marks for the OD (left) and cup in the top row (right)
OD (red) and cup (green) borders were averaged. Soft classification maps in the bottom row**

Region Boundary: Average boundaries are determined from the manually drawn borders for both the OD and cup regions. The average boundary is calculated by averaging the manual markings in each of the 80 equal angular sectors that the disc centre divides the image region into. A total of pixel locations $[x, y]$ conforming to the averaged boundaries of the OD and cup are contained in two separate text files with the name Image Name RegionTypeAvgBoundary.txt.

- **Classification of Map softly:** Four expert manual Classifications are combined to

create a soft map $B(p)$ for the OD and cup regions. B can have values ranging from 0 to 1, with 0 and 1 denoting complete disagreement and agreement that p is a border point, respectively. The same is true for $B_{0:75}$, which denotes agreement amongst at least three experts. Classification soft maps in PNG file format are offered for the OD and cup regions of each image. Figure 3's bottom row displays example Classification soft maps for the OD and cup region [20].

In the fig 3.6.2 we can see few examples of the healthy eyes from the training set

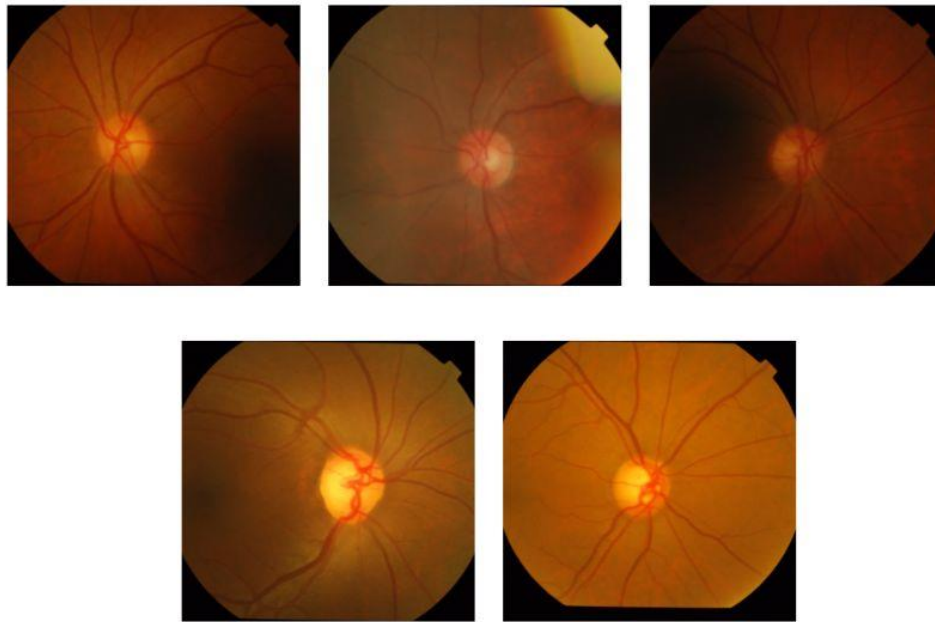


Fig 3.6.2 Examples of Images from Training set for Healthy Eyes

In the fig 3.6.3 we can see few examples of the Glaucomatous eyes from the training set

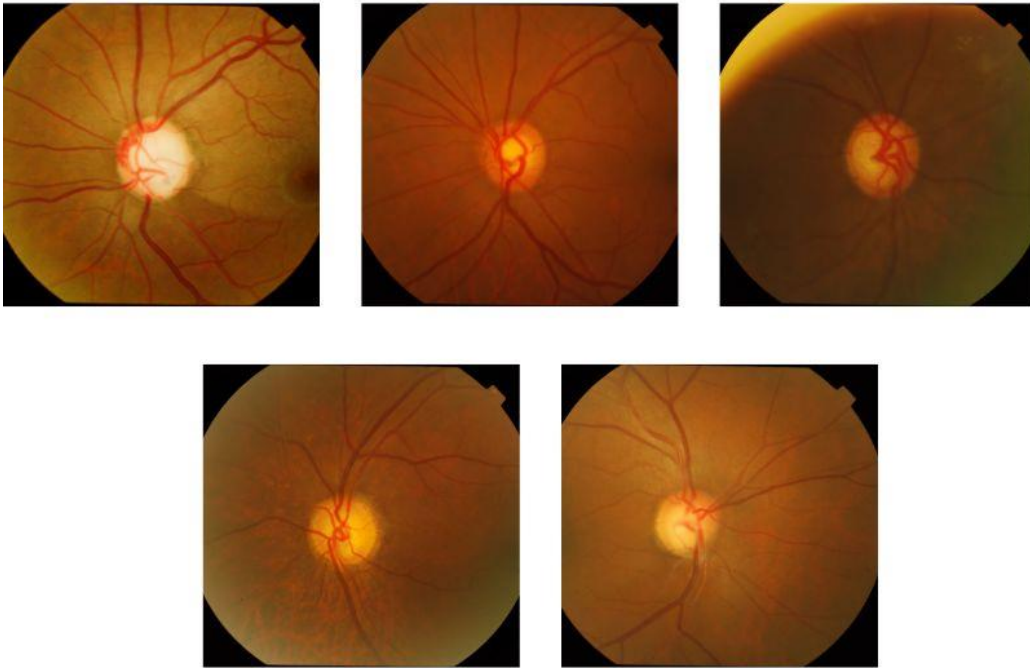


Fig 3.6.3 Examples of Images from Training set for Glaucomatous Eyes

In the fig 3.6.4 we can see few examples of the Healthy eyes from the testing set

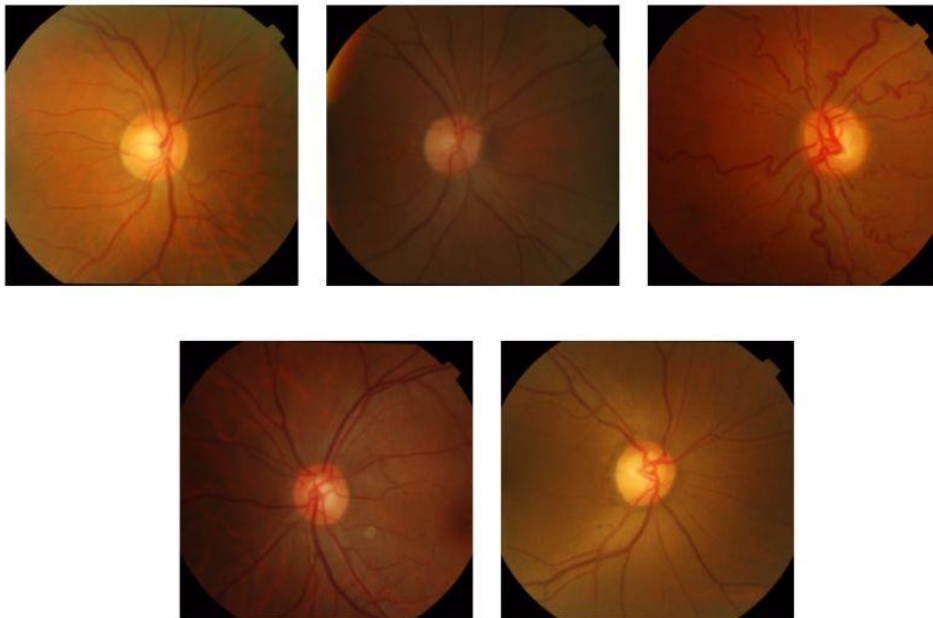


Fig 3.6.4 Examples of Images from Testing set for Healthy Eyes

In the fig 3.6.5 we can see few examples of the Glaucomatous eyes from the testing set

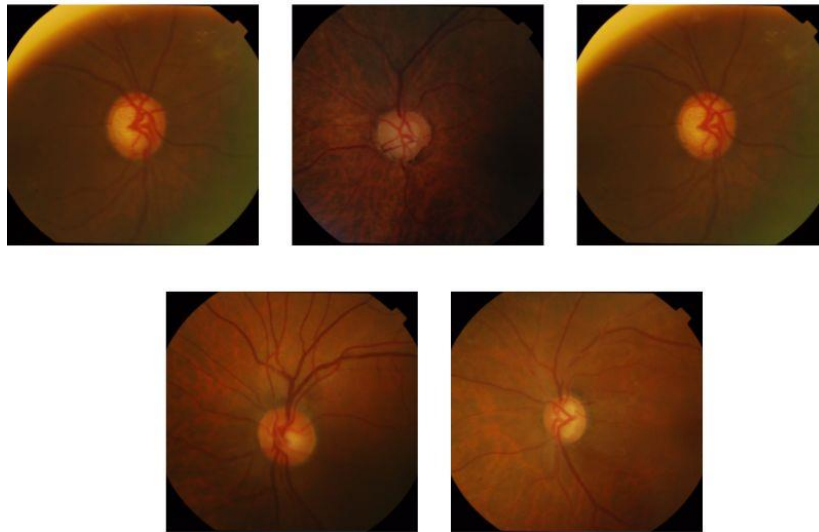


Fig 3.6.5 Examples of Images from Testing set for Glaucomatous Eyes

In the fig 3.6.6 we can see few examples of the Mask in the dataset



Fig 3.6.6 Examples of Mask

In the fig 3.6.7 we can see few examples of the fundus image after applying the Guided Filter

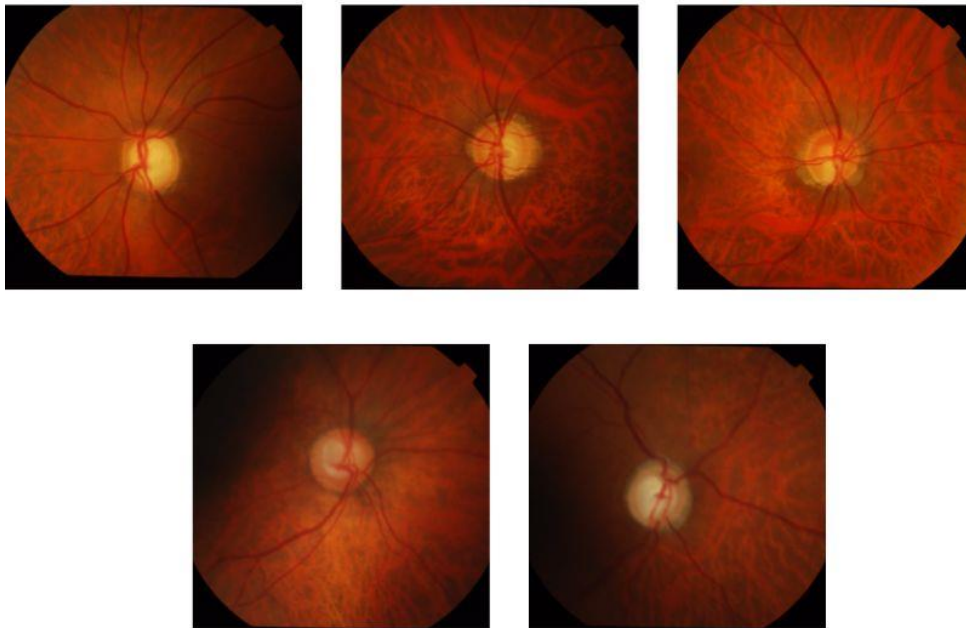


Fig 3.6.7 Fundus images after applying Guided filter

2. Activation functions

The weighted total is determined by the activation function, which then adds bias to it to determine whether or not to activate a neuron. The goal of the activation function is to increase the nonlinearity of a neuron's output. We are aware that the weight, bias, and individual activation functions of each neuron in the neural network determine how well they function. A neural network's weights and biases for each neuron would be adjusted based on the output error. Back-propagation is the name of this procedure. Activation functions enable back-propagation by providing the gradients and error required since weights must be updated accordingly [21].

- **Activation Functions utilized in the proposed models-**

1. Sigmoid or Logistic Activation Function

The Sigmoid Function curve appears to be in the shape of a S. The sigmoid function is used in part because it occurs between (0 to 1). Since the output of such models is a probability

prediction, they are particularly used in this context. Due to the fact that anything only has a probability of occurring between 0 and 1, the sigmoid is the best choice.

The function might take on a variety of forms. We can thus determine the sigmoid curve's slope between any two locations. The derivative is not monotonic, despite the function being monotonic. A neural network may become stuck during training if it uses the logistic sigmoid function. Softmax functionality, Multiclass classification uses a more broadly based logistic activation function. [22].

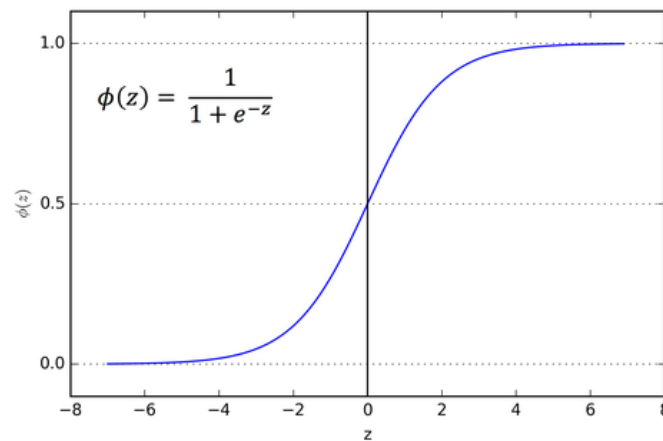


Fig 3.6.8 sigmoid function Curve

2. Tanh or hyperbolic tangent Activation Function

Tanh is comparable to a better logistic sigmoid. The tanh function has a range of between (-1 to 1). Tanh is sigmoidal as well (s - shaped). The tanh graph will map the zero inputs close to zero and the negative inputs strongly negative, which is a benefit of this.

The function could take many different shapes. Although the derivative of the function is not monotone, the function itself is. Most often, the tanh function is employed to divide data into two groups. [23].

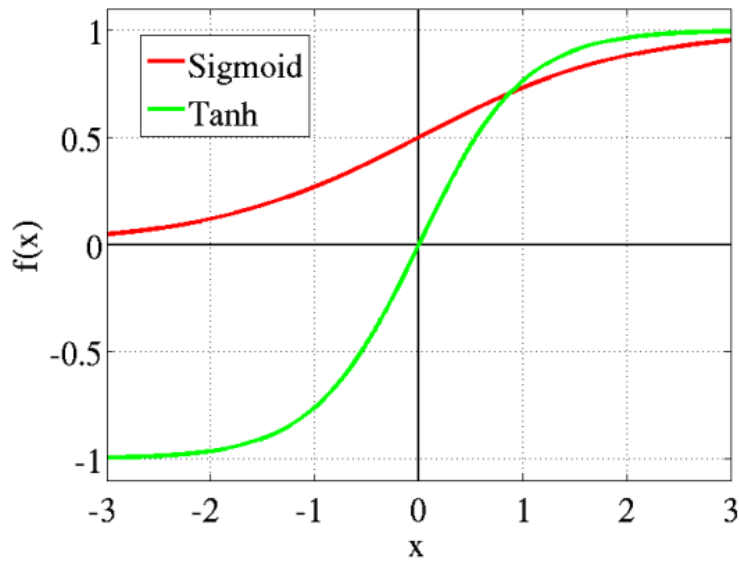


Fig 3.6.9 Tanh vs Sigmoid function Curve at a glance

3. ReLU (Rectified Linear Unit) Activation Function

The activation function that is currently used most commonly worldwide is the ReLU. Considering that nearly all deep learning or convolutional neural networks use it. The range is from 0 to infinity. Its derivative and the behaviour of the function are monotonic.

The issue is that because all negative values quickly become zero, it is more difficult for the model to fit the data appropriately or to be trained from it. Or, to put it another way, Every negative input to the ReLU activation function instantly becomes a graph value of zero[24].

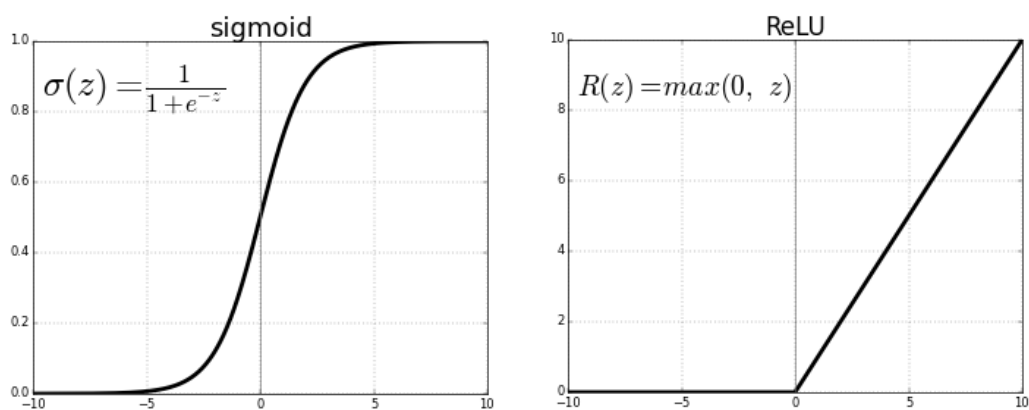


Fig 3.6.10 ReLU vs Sigmoid function Curve at a glance

4. ELU (Exponential Linear Unit)

The function known as the exponential linear unit, or ELU for short, tends to converge to zero faster and yield more precise results. ELU features an additional alpha constant that must be a positive quantity, unlike other activation functions.

ELU and RELU are extremely similar, with the exception of negative inputs. Both of these take the form of an identity function for non-negative inputs. ELU progressively smooths out, as opposed to RELU, which does so sharply, until its output equals [25].

- **U-Net: for Image Classification**

In the semantic process when classifying an image, also known as pixel-based classification, we categorise each pixel into one of several classes. Roads and buildings can be distinguished from other land cover types in satellite imagery using classification in GIS.

The initial application of U-net is classification. Its design can be summed up as an encoder network followed by a decoder network. Semantic analysis is different from classification, where the deep network's final output is the only thing that counts. In addition to pixel-level discrimination, a technique for projecting the discriminative features that are learned throughout the encoder's many phases onto the pixel space is needed for classification.

Encoder is located in the first half of the architecture diagram. It is often a pre-trained classification network like VGG/ResNet where convolution blocks are applied followed by a maxpool down sampling to encode the input image into feature representations at many distinct levels [26].

The decoder is the second element of the architecture. By semantically projecting the discriminative characteristics (lower resolution) learned by the encoder onto the pixel space, the goal is to provide a dense classification (higher resolution). Concatenation, up sampling.

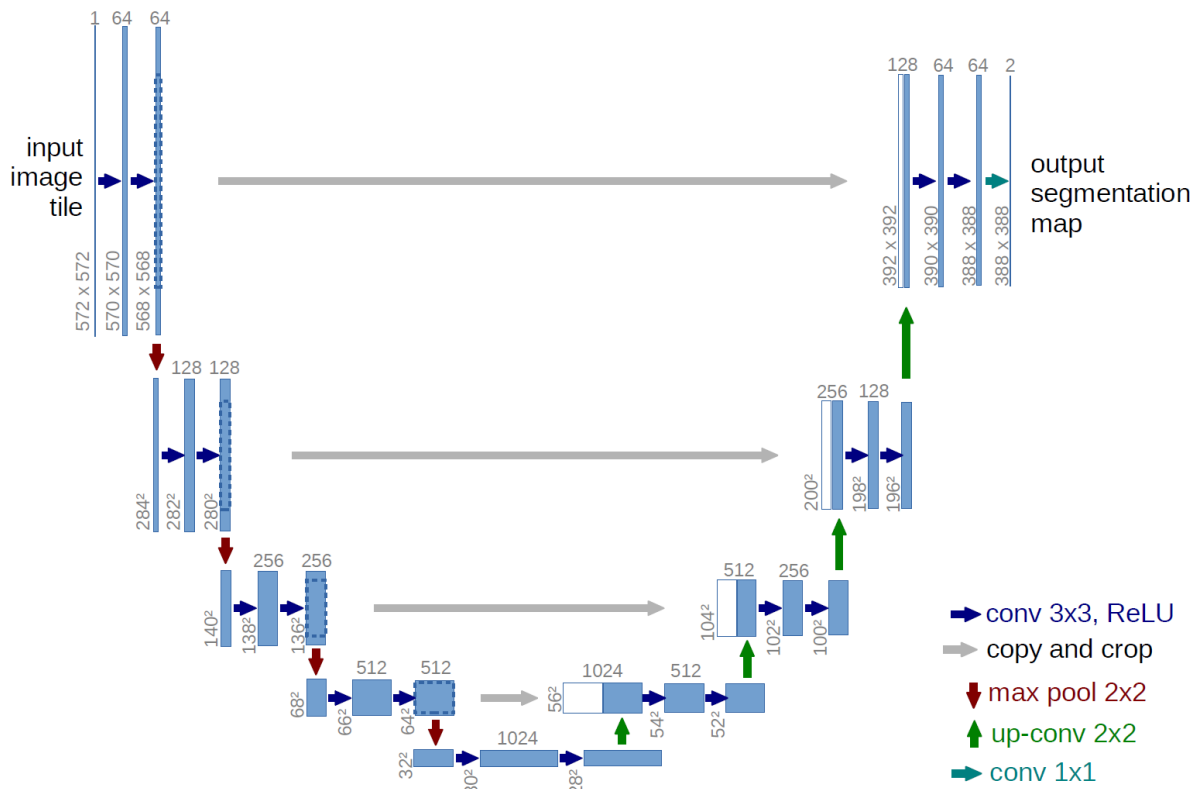


Fig 3.6.11 U-Net architecture

This architecture features multilayer skip connections that connect a contracting path on the left and an expansive way on the right (middle). The final layer, which follows the expanding path, produces predictions using retinal fundus images as input to the contracting path. Before using a rectified linear activation unit, which has the functional form $f(z) = \max(0, z)$, each filter bank in a convolution layer applies three 3 by 3 padded convolutions $(0, z)$. Three convolutional blocks are present in both the contracting and expanding paths. Two convolutional layers follow a block in the contracting route. a layer with a pool size of two (2) and maximum pooling. Two convolutional layers, a dropout layer, and a 2 2 up sampling layer are all present in a block, and a concatenation of the matching block and the contracting path (i.e., a merged layer). The linking path contains two convolutional layers. A 1 1 convolutional layer with a sigmoid activation and one filter to generate pixel-wise class scores makes up the final output layer. Each convolution layer in blocks 1, 2, and 3 has 112, 224, and 448 filters in the contracting path, but the expansive paths of blocks 5, 6, and 7 contain 224, 122, and 122 filters, respectively. The connection path's 448 filters are distributed throughout each convolutional layer [27].

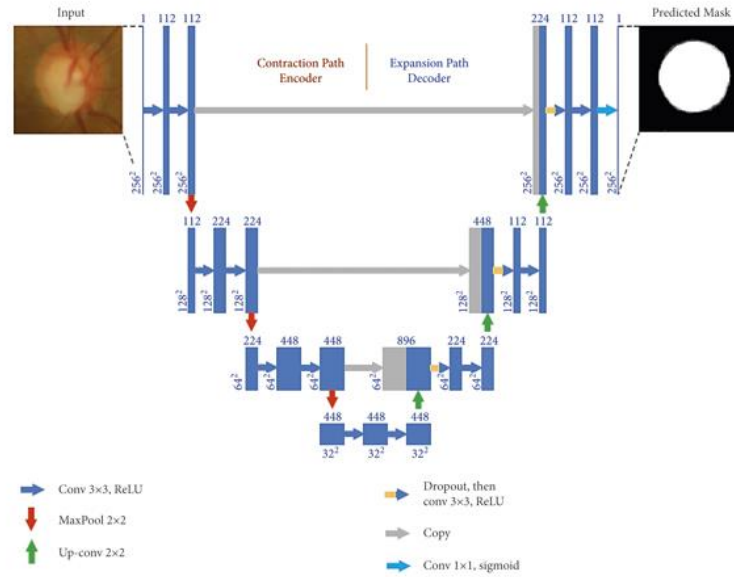


Fig 3.6.12 U-Net for Classification Of fundus Images

- **Feature Extraction from Pretrained CNN Architecture**

The method of feature extraction keeps the information from the original data collection while transforming raw data into manageable numerical characteristics. It yields superior results when compared to utilising machine learning on the raw data directly.

Manual or automated feature extraction is possible. Manual feature extraction requires both the development of an extraction technique as well as the identification and description of the qualities that are relevant to a specific situation. Having a firm understanding of the context or domain can frequently help in deciding which traits might be beneficial.

Deep networks or specialised algorithms can automatically extract information from signals or pictures. Automated feature extraction eliminates the need for human participation. This approach can be quite useful when we need to swiftly transition from gathering raw data to developing machine learning algorithms [28].

The initial layers of deep networks have primarily taken on the task of feature extraction with the emergence of deep learning, albeit mostly for picture data. The first challenge that necessitates a high level of expertise is feature extraction before creating powerful prediction models for signal and time-series applications [29].

1. VGG-16

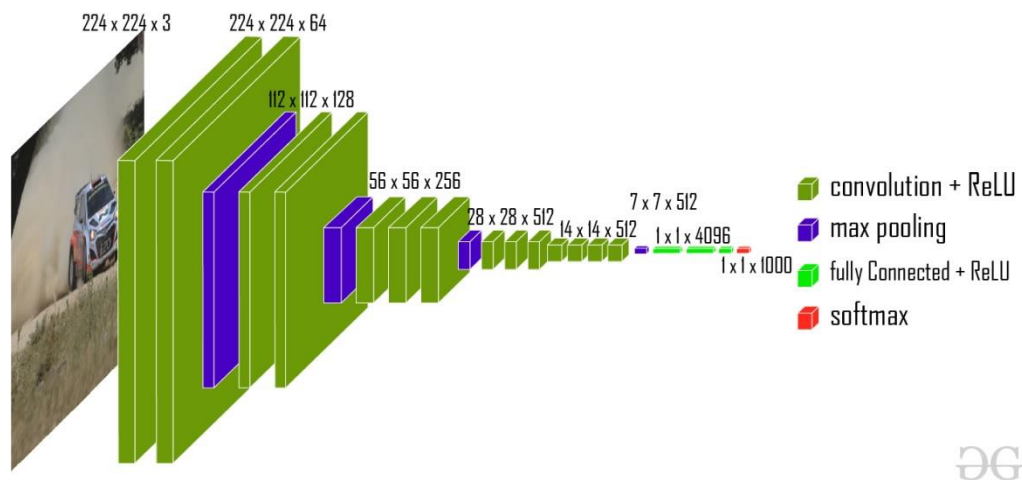


Fig 3.6.13 Vgg-16

Approximately 14 million photos modeling into 22,000 categories make up the ImageNet dataset, which was used to train the convolutional neural network VGG16. Very Deep Convolutional Networks for Large-Scale Image Recognition, a 2015 paper by K. Simonyan and A. Zisserman, makes this model recommendation.

VGG16, which includes photographs of vehicles, has been trained on millions of images. Its taught weights and convolutional layers can recognize common features like edges, colors, wheels, windshields, etc.

Our photos will be sent through the convolutional layers of VGG16, which will provide a Feature Stack containing the recognized visual features. From this point, it is simple to flatten the three-dimensional feature stack into a NumPy array, making it ready for any modeling we choose to do [30].

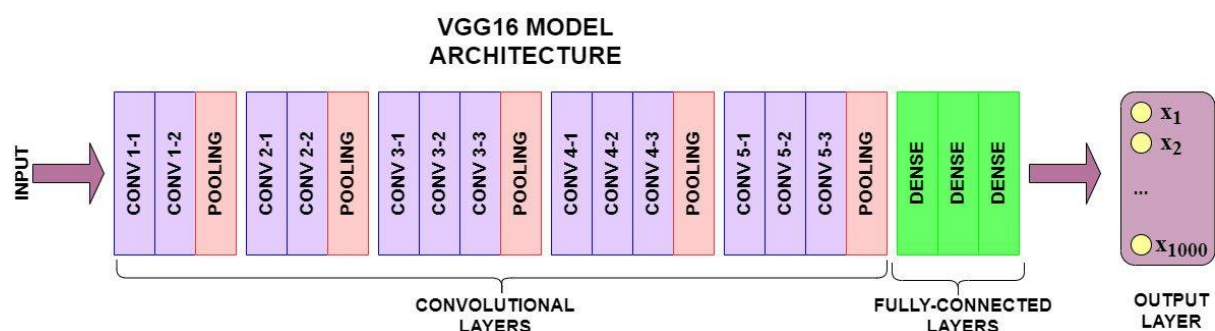


Fig 3.6.14 VGG16 Model Architecture

2. **VGG Architecture:** The network is fed with a dimensioned image (224, 224, 3). The first two layers include the same padding and 64 channels with a 3*3 filter size. Following a max pool layer of stride (2, 2), two layers have convolution layers of 128 filter size and filter size (3, 3). A max-pooling stride (2, 2) layer that is the same as the layer preceding it comes next. Then, 256 filters with filter widths of 3 and 3 are distributed over 2 convolution layers. After that, there are two sets of three convolution layers, and then a max pool layer comes next. Each filter contains 512 filters and the same padding (3, 3). We employ 3*3 filters in these convolution and max-pooling layers instead of 7*7 filters, ZF-11*11 Net, and AlexNet layers. Additionally, some of the layers use 1*1 pixels to change the number of input channels. After each convolution layer, 1-pixel padding is applied to prevent the image's spatial characteristic. After adding a convolution and max-pooling layer to the stack, we got a (7, 7, 512) feature map. To construct a feature vector with the value 1, 25088, this result is flattened. There are then three layers that are entirely linked to one another. The first layer outputs a vector with a length of (1, 4096) and the second layer also generates a vector. It takes the most recent feature vector as input.

The Softmax function is used to categorise 1000 classes in the third fully linked layer. A vector of size (1, 4096) is also produced by the second layer, and the third layer produces 1000 channels. Every hidden layer uses ReLU as its activation function. ReLU is more computationally effective since it encourages speedier learning and reduces the possibility of vanishing gradient problems [31].

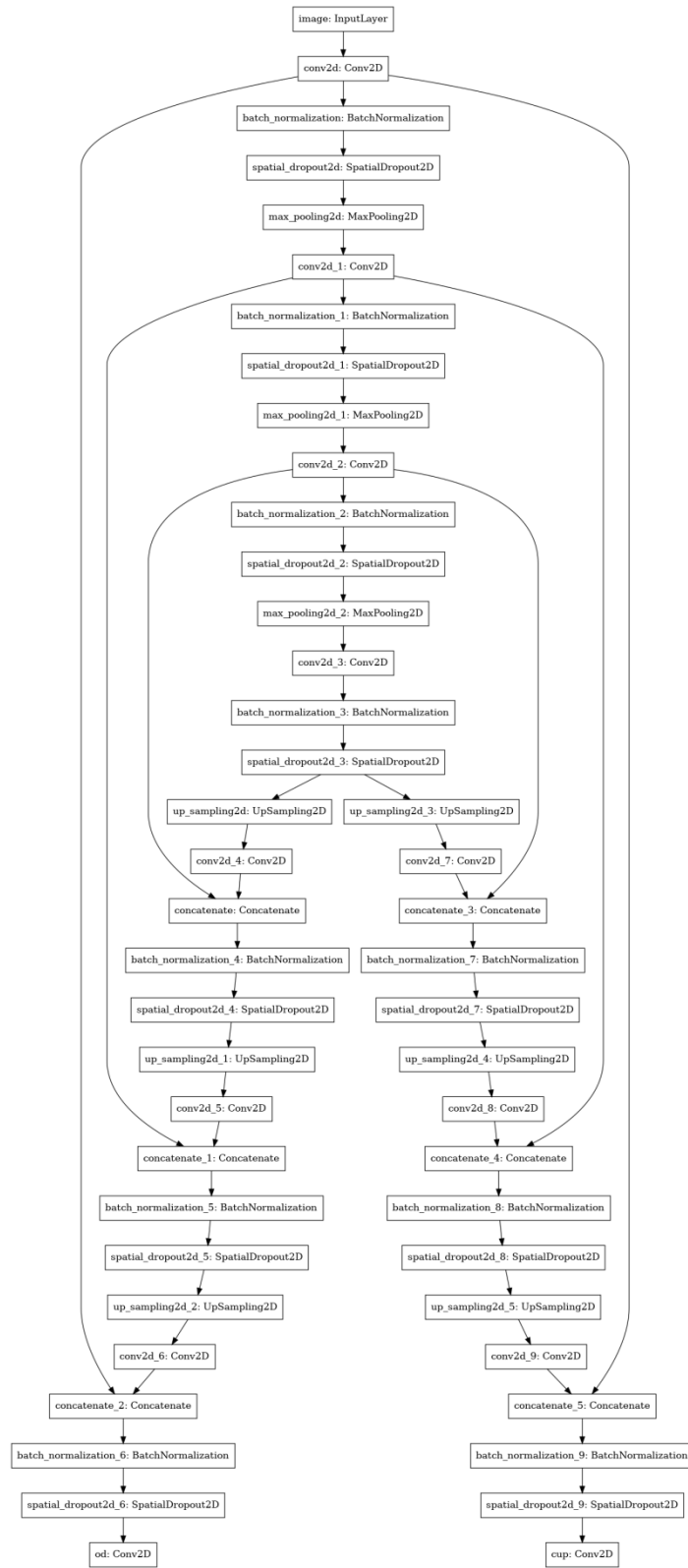


Fig 3.6.16 Vgg-16

- **Guided Filter**

An edge-preserving smoothing light filter is known as a guided filter. Similar to a bilateral filter, it can keep sharp edges while removing noise or texture. The guided image filter has two benefits over the bilateral filter: While guided image filters use simpler algorithms with linear computational complexity, bilateral filters have high computational complexity. Sometimes, bilateral filters introduce undesirable gradient reversal artefacts and distort the image. The linear combination-based guided image filter maintains the gradient direction of the guiding picture in the output image, preventing gradient reversal. Filtering with edge preservation when the filtering input p and the guiding image are identical. The guided filter maintains sharp edges while removing noise from the input image [32].

To be more precise, a "flat patch" or a guided filter's option `display style epsilon` can be used to specify a "high variance patch." Patches with variances significantly lower than the value of the parameter "`display style epsilon epsilon`" will be smoothed, while patches with variances significantly greater than this value will be retained. Similar to how `displaystyle epsilon epsilon` played a part in the guided filter; the range variance plays a similar role in the bilateral filter. Both of them specify the noise/flat patches that should be smoothed and the edge/high variance patches that should be kept.

The edges of an image may contain artefacts after applying the bilateral filter. This is due to the sharp shift in pixel value at the edge. Due to edges, these artefacts are both inevitable and difficult to avoid and appear in a variety of images. Gradient reversal is better avoided by the guided filter. Additionally, in some circumstances, it is possible to guarantee that gradient reversal won't happen.

It is possible to transfer the structure from the guidance `displaystyle II` to the output `displaystyle qq` thanks to the local linear model of `displaystyle q=aI+b`. Some specialised filtering-based applications, such as feathering, matting, and dehazing, are made possible by this characteristic [33, 34].

CHAPTER 4

RESULTS ANALYSIS AND VALIDATION

4.1 Implementation of the solution

For the final implementation of the solution we used Kaggle as it provides a very flexible environment and supports Python and R which are very versatile and general purpose language

4.2 Coding Simulation:

```
#importing libraries#
import numpy as np
import pandas as pd
import tensorflow as tf
from tensorflow.keras.layers import Conv2D, UpSampling2D ,
BatchNormalization, SpatialDropout2D, Input, MaxPool2D, concatenate
from tensorflow.keras.models import Model
from tensorflow.keras.applications import vgg16
from tensorflow.keras.callbacks import EarlyStopping
from tensorflow.data import Dataset, AUTOTUNE
import tensorflow.strings
from tensorflow.keras.utils import plot_model
from keras.models import load_model

import matplotlib.pyplot as plt

from glob import glob
from pathlib import Path
import os
from keras.callbacks import ModelCheckpoint

print("Setup completed!")

#UpSampling2D doubles the dimensions of the input#
#In Convolutional neural network, the kernel is nothing but a filter that is used to
extract the features from the images#
#Batch normalization (also known as batch norm) is a method used to make
training of artificial neural networks faster and more stable through
normalization of the layers' inputs by re-centering and re-scaling.# #it drops
entire 2D feature maps instead of individual elements#
#VGG-16 is a convolutional neural network that is 16 layers deep#
##
#setting path for the training variable#
PATH TO TRAINING = r'./input/drishtigs-retina-dataset-for-onh-
Classification/Training-20211018T055 246Z-001/Training/'
```

```

PATH_TO_TRAINING_TENSOR = tf.Variable(PATH_TO_TRAINING)

TRAIN_GLAUCOMA_IMAGES =
np.array(glob(os.path.join(PATH_TO_TRAINING, 'Images/GLAUCOMA/*')))
TRAIN_NORMAL_IMAGES =
np.array(glob(os.path.join(PATH_TO_TRAINING, 'Images/NORMAL/*' )))

TRAIN_GLAUCOMA_GT = np.array(glob(PATH_TO_TRAINING
+ '/GT/GLAUCOMA/*')) TRAIN_NORMAL_GT =
np.array(glob(PATH_TO_TRAINING + '/GT/NORMAL/*'))

#setting path for the testing variable#
PATH_TO_TEST = r'../input/drishtigs-retina-dataset-for-onh-Classification/Test-
20211018T060000Z-001 /Test'
TEST_GLAUCOMA_IMAGES =
np.array(glob(os.path.join(PATH_TO_TEST, 'Images/glaucoma/*')))
TEST_NORMAL_IMAGES = np.array(glob(os.path.join(PATH_TO_TEST,
'Images/normal/*')))

SEED = 2
EPSILON = 10 ** -4
STATELESS_RNG= tf.random.Generator.from_seed(SEED, alg='philox')
IMAGE_SIZE = (224, 224)
CHANNELS = 3
SHUFFLE_BUFFER = len(TRAIN_NORMAL_IMAGES) +
len(TRAIN_GLAUCOMA_IMAGES) BATCH_SIZE =
SHUFFLE_BUFFER // 2
early_stopping = EarlyStopping(monitor = 'val_loss', restore_best_weights = True, patience = 5)

"""This Dataset contains 50 train images and 51 test Images
In Each Directory there are two folders one is images and the second
one is GT. The GT folder contains Optic Disk and Cup masks associated
with the images in Images folder."""

Path(TRAIN_GLAUCOMA_IMAGES[0]).stem
TEST_GLAUCOMA_IMAGES
#location of the path#

#localising the csv file for saving output results later#
csvpath=r'../input/epochh/Results.xlsx'
print (csvpath)

def load_image_and_gt(path, test = False):

    print(path)

    filename = Path(path).stem
    if test:
        path_to_OD_softmap = os.path.join(PATH_TO_TEST, 'Test_GT', filename,
        'SoftMap', filename + '_ODsegSoftmap.png')
        path_to_cup_softmap = os.path.join(PATH_TO_TEST, 'Test_GT', filename,
        'SoftMap', filename + '_cupsegSoftmap.png')

```



```

else:
    path_to_OD_softmap = os.path.join(PATH_TO_TRAINING, 'GT', filename,
    'SoftMap', filename + ' ODsegSoftmap.png')
    path_to_cup_softmap = os.path.join(PATH_TO_TRAINING, 'GT', filename,
    'SoftMap', filename + ' cupsegSoftmap.png')
    image = tf.io.read_file(path)
    image = tf.io.decode_png(image)
    od = tf.io.read_file(path_to_OD_softmap)
    od = tf.io.decode_png(od)

```

```

cup = tf.io.read_file(path_to_cup_softmap)
cup = tf.io.decode_png(cup)
return image, od, cup

```

```

def show_image_with_masks(image, od, cup, ax, index, label):

```

```

    ax[5 * index].imshow(image)
    ax[5 * index].set_title(label)
    ax[5 * index + 1].imshow(od)
    ax[5 * index + 1].set_title('Optical Disk Mask')
    ax[5 * index + 2].imshow(od * image)
    ax[5 * index + 2].set_title('Image with OD Mask')
    ax[5 * index + 3].imshow(cup)
    ax[5 * index + 3].set_title('Cup Mask')
    ax[5 * index + 4].imshow(cup * image)
    ax[5 * index + 4].set_title('Image with CUP Mask')

```

```

def load_image_with_masks(path, dice = False, test = False):

```

```

    filename = tf.strings.split(path, sep = '.')[-2]
    filename = tf.strings.split(filename, sep = '/')[-1]
    directory_path = tf.strings.split(path, sep = 'Images')[0]
    if test:
        softmap_path = tf.strings.join([directory_path, 'Test_GT/',
        filename, '/SoftMap/'])
    else:
        softmap_path = tf.strings.join([directory_path, 'GT/',
        filename, '/SoftMap/'])
    od_path =
    tf.strings.join([softmap_path, filename,
    ' ODsegSoftmap.png'])
    cup_path =
    tf.strings.join([softmap_path, filename,
    ' cupsegSoftmap.png'])
    image = tf.io.read_file(path)
    image = tf.io.decode_png(image, channels = 3)
    image = tf.image.resize(image, IMAGE_SIZE)
    image = image / 255.0

    od = tf.io.read_file(od_path)
    od = tf.io.decode_png(od, channels = 1)
    od = tf.image.resize(od, IMAGE_SIZE)
    od = od / 255.0
    if dice:
        od = tf.where(od >= 0.5, 1.0, 0.0)

    cup = tf.io.read_file(cup_path)
    cup = tf.io.decode_png(cup, channels = 1)
    cup = tf.image.resize(cup, IMAGE_SIZE)

```

```

cup = cup / 255.0
if dice:
cup = tf.where(cup >= 0.5, 1.0, 0.0)

return {'image' : image}, {'od' : od, 'cup' : cup}

def augment_images(image, od, cup, single_target = False):

seeds = STATELESS_RNG.make_seeds(2)[0],
STATELESS_RNG.make_seeds(2)[0]
def augment(img, seeds):

img =
tf.image.stateless_random_flip_left_right(i
mg, seeds[0]) img =
tf.image.stateless_random_flip_up_down(i
mg, seeds[1])

img = tf.image.resize_with_crop_or_pad(img, IMAGE_SIZE[0] + 20,
IMAGE_SIZE[1] + 20) img = tf.image.stateless_random_crop(img, size =
(*IMAGE_SIZE, img.shape[2]), seed = seeds[0])
return img

image = augment(image, seeds)
od = augment(od, seeds)
cup = augment(cup, seeds)

if single_target:
return image, od
return {'image' : image}, {'od' : od, 'cup' : cup}

image, od, cup = load_image_and_gt(TRAIN_NORMAL_IMAGES[0])
plt.imshow(od)
image.shape

rows = 6
cols = 5
fig, axes = plt.subplots(rows, cols, figsize = (15, 15))
axes = axes.flatten()

normal_images = np.random.choice(TRAIN_NORMAL_IMAGES, rows // 2)
glaucoma_images = np.random.choice(TRAIN_GLAUCOMA_IMAGES, rows - (rows // 2))

for idx, file in enumerate(normal_images):
image, od, cup = load_image_and_gt(file)
show_image_with_masks(image, od, cup, axes, index = idx, label = 'Normal :' +
os.path.basename(file))
for idx, file in enumerate(glaucoma_images):
image, od, cup = load_image_and_gt(file)
show_image_with_masks(image, od, cup, axes, index = idx + rows // 2, label =
'Glaucoma :' + os.path.basename(file))

plt.tight_layout()

```

```

def get_avgboundary(file):
    image = tf.io.read_file(file)
    image = tf.io.decode_png(image)

    cup_mask = np.zeros(shape = (image.shape[0], image.shape[1]))
    od_mask = np.zeros(shape = (image.shape[0], image.shape[1]))
    center_mask = np.zeros(shape = (image.shape[0], image.shape[1]))

    filename = Path(file).stem
    boundary_path = os.path.join(PATH_TO_TRAINING, 'GT',
    filename, 'AvgBoundary')
    cup_file = os.path.join(boundary_path, filename +
    '_CupAvgBoundary.txt')
    od_file = os.path.join(boundary_path, filename + '_ODAvgBoundary.txt')
    center_file = os.path.join(boundary_path, filename + '_diskCenter.txt')

    count = 0
    with open(cup_file) as cf:
        for line in cf:
            a, b = line.split()

            a = int(a)
            b = int(b)
            cup_mask[a, b] = 255
            count += 1

    with open(od_file) as of:
        for line in of:
            a, b = line.split()
            a = int(a)
            b = int(b)
            od_mask[a, b] = 255

    with open(center_file) as ctrf:
        for line in ctrf:
            a, b = line.split()
            a = int(a)
            b = int(b)
            center_mask[a, b] = 255
    return od_mask.reshape(image.shape[0], image.shape[1], 1), \
    cup_mask.reshape(image.shape[0], image.shape[1], 1), \
    center_mask.reshape(image.shape[0], image.shape[1], 1)

glaucoma_avgboundary_test = TRAIN_GLAUCOMA_IMAGES[3]

od_mask, cup_mask, center_mask =
get_avgboundary(glaucoma_avgboundary_test)
image, od,
cup = load_image_and_gt(glaucoma_avgboundary_test)

fig, axes = plt.subplots(2, 3, figsize = (20, 20))
axes = axes.flatten()

axes[0].imshow(image)
axes[0].set_title('Test Image')
axes[1].imshow(od)

```

```

axes[1].set_title('Optical Disk Mask')
axes[2].imshow(od_mask)
axes[2].set_title('Optical Disk Avg Boundary')
axes[3].imshow(cup)
axes[3].set_title('Cup Mask')
axes[4].imshow(cup_mask)
axes[4].set_title('Cup Avg Boundary')
axes[5].imshow(center_mask)
axes[5].set_title('Center Avg Boundary')
plt.tight_layout()

```

```

train_images = np.concatenate([TRAIN_NORMAL_IMAGES,
                                TRAIN_GLAUCOMA_IMAGES]) test_images =
np.concatenate([TEST_NORMAL_IMAGES,
TEST_GLAUCOMA_IMAGES])

```

```

train_ds_dice = Dataset.from_tensor_slices(train_images)\
    .shuffle(buffer_size = SHUFFLE_BUFFER)\
    .map(lambda x: load_image_with_masks(x, dice = True),
num_parallel_calls = AUTOTUNE)\
    .map(lambda image, targets:
augment_images(image['image'], targets['od'], targets['cup']))\
    .batch(batch_size = BATCH_SIZE)\
    .prefetch(1)\
    .cache()

```

```

train_ds_no_dice = Dataset.from_tensor_slices(train_images)\
    .shuffle(buffer_size = SHUFFLE_BUFFER)\
    .map(lambda x: load_image_with_masks(x, dice = False),
num_parallel_calls = AUTOTUNE)\
    .map(lambda image, targets:
augment_images(image['image'], targets['od'], targets['cup']))\
    .batch(batch_size = BATCH_SIZE)\
    .prefetch(1)\
    .cache()

```

```

test_ds_dice = Dataset.from_tensor_slices(test_images)\
    .map(lambda x: load_image_with_masks(x, dice = True, test = True),
num_parallel_calls = AUTOTUNE)\
    .batch(batch_size = BATCH_SIZE)

```

```

test_ds_no_dice = Dataset.from_tensor_slices(test_images)\
    .map(lambda x: load_image_with_masks(x, dice = False, test = True),
num_parallel_calls = AUTOTUNE)\
    .batch(batch_size = BATCH_SIZE)
train_images = np.concatenate([TRAIN_NORMAL_IMAGES,
                                TRAIN_GLAUCOMA_IMAGES]) test_images =
np.concatenate([TEST_NORMAL_IMAGES,
TEST_GLAUCOMA_IMAGES])

```

```

train_ds_dice = Dataset.from_tensor_slices(train_images)\
    .shuffle(buffer_size = SHUFFLE_BUFFER)\
    .map(lambda x: load_image_with_masks(x, dice = True),
num_parallel_calls = AUTOTUNE)\
    .map(lambda image, targets:
augment_images(image['image'], targets['od'], targets['cup']))\
    .batch(batch_size = BATCH_SIZE)\

```

```

.prefetch(1)\
.cache()

train_ds_no_dice = Dataset.from_tensor_slices(train_images)\
.shuffle(buffer_size = SHUFFLE_BUFFER)\
.map(lambda x: load_image_with_masks(x, dice = False),
num_parallel_calls = AUTOTUNE)\
.map(lambda image, targets:
augment_images(image['image'], targets['od'], targets['cup']))\
.batch(batch_size = BATCH_SIZE)\
.prefetch(1)\
.cache()

test_ds_dice = Dataset.from_tensor_slices(test_images)\
.map(lambda x: load_image_with_masks(x, dice = True, test = True),
num_parallel_calls = AUTOTUNE)\
.batch(batch_size = BATCH_SIZE)

test_ds_no_dice = Dataset.from_tensor_slices(test_images)\
.map(lambda x: load_image_with_masks(x, dice = False, test = True),
num_parallel_calls = AUTOTUNE)\
.batch(batch_size = BATCH_SIZE)

train_ds_no_dice_st = Dataset.from_tensor_slices(train_images)\
.shuffle(buffer_size = SHUFFLE_BUFFER)\
.map(lambda x: load_image_with_masks(x, dice = False), num_parallel_calls =
AUTOTUNE)\
.map(lambda image, targets: augment_images(image['image'],
targets['od'], targets['cup'], single_target = True))\
.batch(batch_size = BATCH_SIZE)\
.prefetch(1)\
.cache()

test_ds_no_dice_st = Dataset.from_tensor_slices(test_images)\
.map(lambda x: load_image_with_masks(x, dice = False, test = True),
num_parallel_calls = AUTOTUNE)\
.map(lambda image, targets: (image['image'], targets['od']))\
.batch(batch_size = BATCH_SIZE)

def upsampling_block(inp, skips, filters, kernels, rates):

    activation = 'sigmoid'
    x = inp
    for f, kernel, skip, rate in zip(filters, kernels, skips, rates):
        x = UpSampling2D()(x)
        x = Conv2D(filters = f, kernel_size = kernel, strides = 1, activation = activation,
padding = 'same')(x)
        x = concatenate([x, skip])
        x = BatchNormalization()(x)
        x = SpatialDropout2D(rate = rate)(x)
    return x

def conv_block(inp, filters, kernels, strides, rates):

    activation = 'ReLU'
    x = inp
    skips = []

```

```

for f, kernel, stride, rate in zip(filters, kernels, strides, rates):
    if stride == 2:
        skips.append(skip)
        x = MaxPool2D()(x)
        x = Conv2D(filters = f, kernel_size = kernel, strides = 1, activation = activation,
padding = 'same')( x)
        skip = x
    else:
        x = Conv2D(filters = f, kernel_size = kernel, strides = 1, activation = activation,
padding = 'same')( x)
        skip = x

x = BatchNormalization()(x)
x = SpatialDropout2D(rate = rate)(x)

return skips, x

```

```

def res_unet(conv_filters, conv_kernels, conv_strides, conv_rates, up_filters, up_kernels,
up_rates):

```

```

    inp = Input(shape = (*IMAGE_SIZE, 3), name = 'image')
    x = inp

    skips, x = conv_block(x, conv_filters, conv_kernels,
conv_strides, conv_rates)
    skips = skips[::-1]

    od = upsampling_block(x, skips, up_filters, up_kernels, up_rates)
    od = Conv2D(filters = 1, kernel_size = 1, strides = 1, padding = 'same',
activation = 'sigmoid', name = 'o d')(od)

    cup = upsampling_block(x, skips, up_filters, up_kernels, up_rates)
    cup = Conv2D(filters = 1, kernel_size = 1, strides = 1, padding = 'same',
activation = 'sigmoid', name = ' cup')(cup)

```

```

    return Model(inputs = inp, outputs = [od, cup])

```

```

def res_unet_st(conv_filters, conv_kernels, conv_strides, conv_rates, up_filters, up_kernels,
up_rates):

```

```

    inp = Input(shape = (*IMAGE_SIZE, 3), name = 'image')
    x = inp

    skips, x = conv_block(x, conv_filters, conv_kernels,
conv_strides, conv_rates)
    skips = skips[::-1]

    od = upsampling_block(x, skips, up_filters, up_kernels, up_rates)
    od = Conv2D(filters = 1, kernel_size = 1, strides = 1, padding = 'same',
activation = 'sigmoid', name = 'o d')(od)

    return Model(inputs = inp, outputs = od)

```

```

def dice(y, y_pred):
    epsilon = EPSILON
    numerator = 2 * tf.reduce_sum(y * y_pred, axis = [1, 2])
    denominator = tf.reduce_sum(y + y_pred, axis = [1, 2])
    dice = tf.reduce_mean((numerator + epsilon)/(denominator + epsilon))
    return 1 - dice

conv_filters = [8, 16, 32, 64]
conv_kernels = [3, 3, 3, 3]
conv_strides = [1, 2, 2, 2]
conv_rates = [0.8, 0.8, 0.8, 0.8]

upsampling_filters = [32, 16, 8, 8]
upsampling_kernels = [3, 3, 3, 3]
upsampling_rates = [0.5, 0.5, 0.5, 0.5]

mod1 = res_unet(conv_filters, conv_kernels, conv_strides, conv_rates,
upsampling_filters, upsampling_kernels, upsampling_rates)
mod1.compile(optimizer = 'Adam', loss = {'od' : dice, 'cup' : dice}, metrics = ['accuracy'])

mod2 = res_unet(conv_filters, conv_kernels, conv_strides, conv_rates,
upsampling_filters, upsampling_kernels, upsampling_rates)
mod2.compile(optimizer = 'Adam', loss = {'od' : 'binary_crossentropy', 'cup' :
'binary_crossentropy'}, metrics = ['accuracy'])
hist2 = mod2.fit(train_ds_no_dice, validation_data = test_ds_no_dice, callbacks
= [early_stopping], epochs = 200)

def plot_loss(epochs, hist, loss):
    plt.plot(epochs + .5, hist.history['loss'], 'r', label = 'Loss')
    plt.plot(epochs + .5, hist.history['od_loss'], 'r-', label = 'OD Loss')
    plt.plot(epochs + .5, hist.history['cup_loss'], 'r+', label = 'Cup Loss')
    plt.plot(epochs, hist.history['val_loss'], 'b', label = 'Val Loss')

    plt.plot(epochs, hist.history['val_od_loss'], 'b-', label = 'Val OD Loss')
    plt.plot(epochs, hist.history['val_cup_loss'], 'b+', label = 'Val Cup Loss')
    plt.legend()
    plt.xlabel('Epochs')
    plt.ylabel('Loss')
    plt.title(loss)

epochs1 = np.arange(0, len(hist1.history['loss']))
plot_loss(epochs1, hist1, 'Dice')

epochs2 = np.arange(0, len(hist2.history['loss']))
plot_loss(epochs2, hist2, 'Binary CrossEntropy')

to_predict = Dataset.from_tensor_slices(test_images)\

```

```

.map(lambda x: load_image_with_masks(x, dice = True, test = True),
num_parallel_calls = AUTOTUNE)\
.map(lambda image, targets: image['image'])\
.batch(batch_size = BATCH_SIZE)

predict1 = mod1.predict(to_predict)
predict2 = mod2.predict(to_predict)

test = next(iter(predict1))
test.shape

def plt_preds(ds, preds, super_title, binarizer_threshold = 0.5):
    ds = next(iter(ds))
    image, targets = ds
    image = image['image']
    od = targets['od']
    cup = targets['cup']
    preds = next(iter(preds))
    rows = 5
    cols = 7
    fig, axes = plt.subplots(rows, cols, figsize = (20, 20))
    axes = axes.flatten()
    for i in range(rows):
        axes[cols * i].imshow(image[i])
        axes[cols * i].set_title('Original Image')
        axes[cols * i + 1].imshow(od[i])
        axes[cols * i + 1].set_title('Original OD Mask')
        axes[cols * i + 2].imshow(cup[i])
        axes[cols * i + 2].set_title('Original Cup Mask')
        axes[cols * i + 3].imshow(preds[i])
        axes[cols * i + 3].set_title('Predicted OD Mask')
        axes[cols * i + 4].imshow(np.where(preds[i] > binarizer_threshold, 1, 0))
        axes[cols * i + 4].set_title('Binarized OD Mask')
        axes[cols * i + 5].imshow(preds[26 + i - 1])
        axes[cols * i + 5].set_title('Predicted Cup Mask')
        axes[cols * i + 6].imshow(np.where(preds[26 + i] >=
binarizer_threshold, 1, 0)) axes[cols * i +
6].set_title('Binarized Cup Mask')

plt.suptitle(super_title)

plt_preds(test_ds_dice, predict1, super_title = 'Dice Loss Trained on Binarized
GT Masks', binarizer_threshold = 0.5)

plt_preds(test_ds_no_dice, predict2, binarizer_threshold = 0.5, super_title =
'Binary Cross Entropy Trained on Default Masks')

#save model
%cd /kaggle/working
from IPython.display import FileLink

```



```

FileLink(r'Model 1.h5')

import torch
import torch.nn as nn
import torch.nn.functional as F
import torchvision.transforms.functional as TF
from torchvision.io import read_image
import matplotlib.pyplot as plt
import cv2 as cv
from filters import guided_filter

def disp(x): return (x.clamp(0,1)*255).byte().permute(1,2,0).cpu()

#Local contrast enhancement#
p = read_image("../input/drishtigs-retina-dataset-for-onh-Classification/Test-20211018T060000Z-001/Test /Images/glaucoma/drishtiGS_001.png") / 255

r = 13
amount = 5

fig, (ax,bx,cx) = plt.subplots(1,3, figsize=(25,25))
ax.set_title("Original")
ax.imshow(disp(p))

q = TF.gaussian_blur(p, r)
bx.set_title("Gaussian blur")
bx.imshow(disp(q))

c = q + (p - q) * amount
cx.set_title("Gradients boosted")
cx.imshow(disp(c))

```

4.3 Result and Discussions:

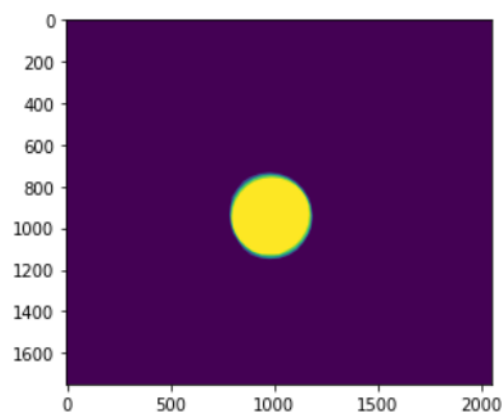


Fig 4.1 Fundus image with OD and OC mask applied

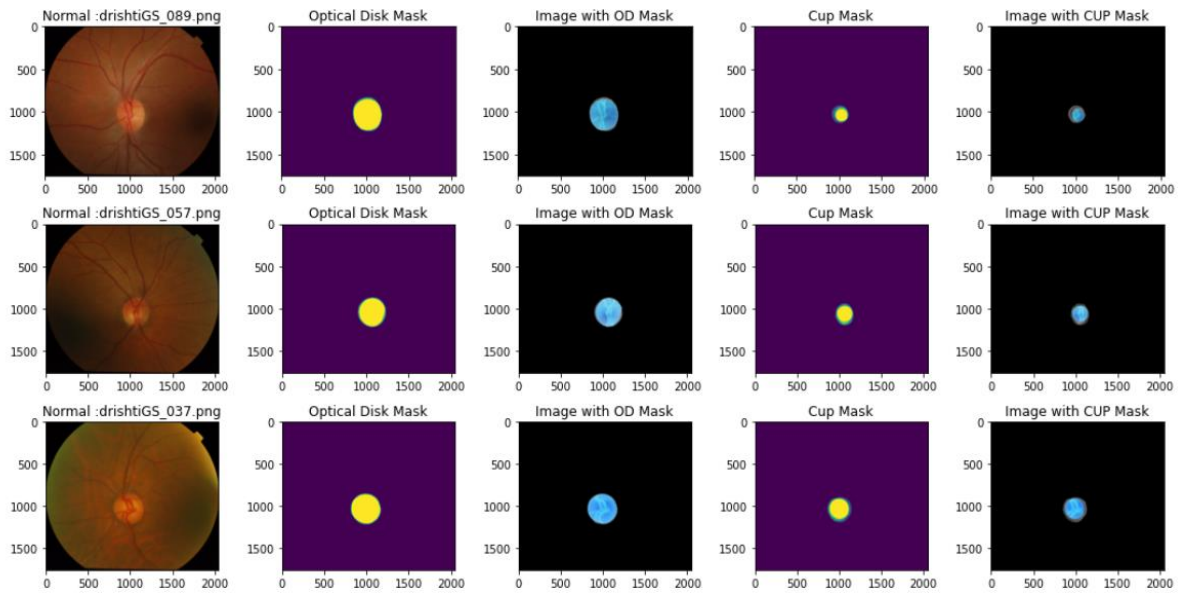


Fig 4.2 Fundus Images with all Masks applied and neatly displayed

[4]: <matplotlib.image.AxesImage at 0x7f152dd49750>

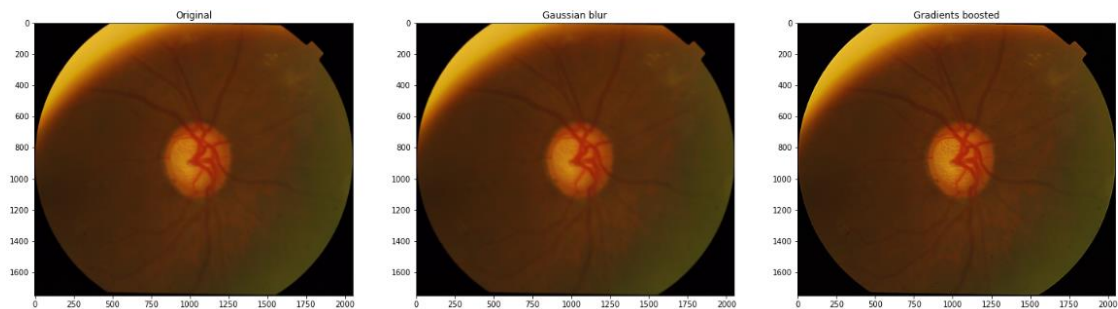


Fig 4.3 an example of fundus image after passing through the guided filter

Table 4.1 shows the difference in accuracy for the different Activation Functions and With Added mask

Table 4.1 Results obtained from the suggested method

Method	Epoch 1	Epoch 2	Epoch 3	Epoch 4	Epoch 5
UNet (Sigmoid)	0.8954	0.8951	0.8949	0.8956	0.8957
UNet (Relu)	0.8666	0.8657	0.8650	0.8663	0.8660
UNet (TanH)	0.9858	0.9851	0.9852	0.9854	0.9859
UNet (Elu)	0.9601	0.9594	0.9607	0.9600	0.9603
Guided filter+UNet (Sigmoid)	0.9101	0.9089	0.9103	0.9094	0.9107
Guided filter+UNet (Relu)	0.9256	0.9251	0.9250	0.9257	0.9256
Guided filter+UNet (Tanh)	0.9981	0.9985	0.9899	0.9987	0.9987
Guided filter+UNet (Elu)	0.9876	0.9868	0.9858	0.9848	0.9838

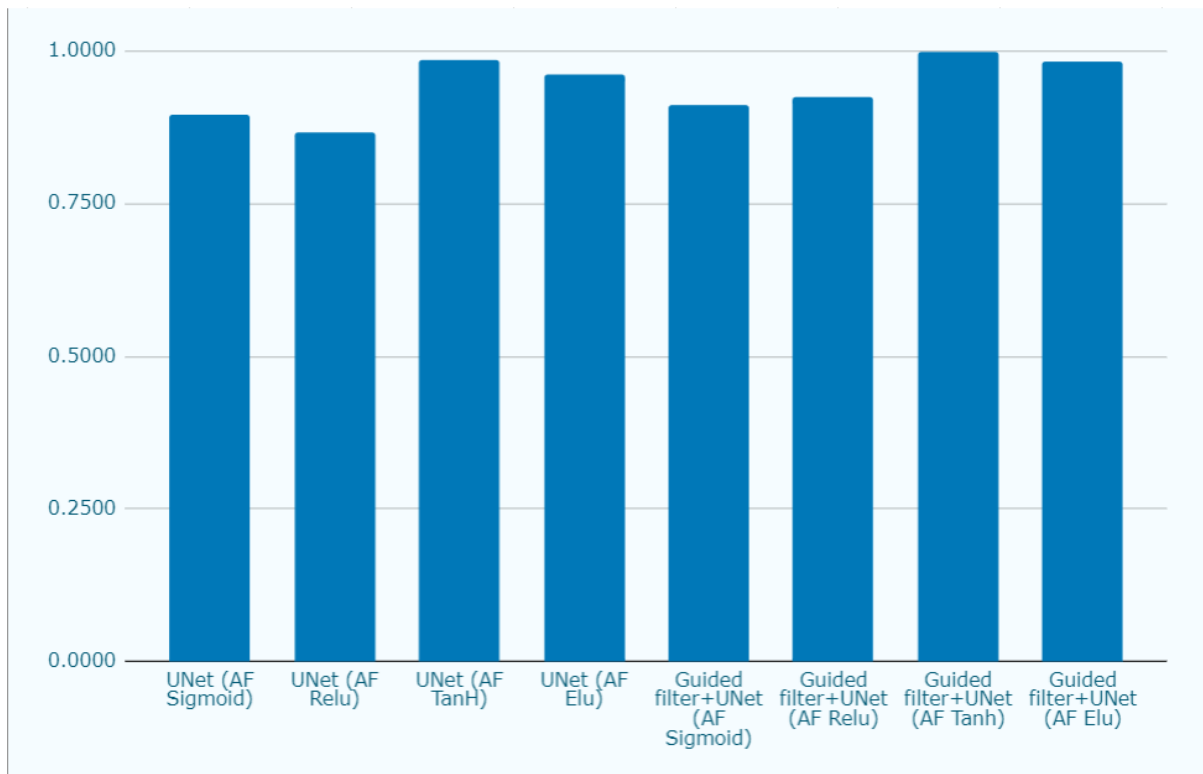


Fig 4.4 Graphical Representation of the accuracy

As Observed from the above table we can see the difference in performance of the model when different parameters are applied to it

It is observed the accuracy of the model is at its lowest with ReLU Activation function and it is highest with the added mask and Tanh activation function

- **Analysis**

We must regularly evaluate a project as it is being carried out. Failure to do so would result in unforeseen difficulties missed important details, or procedural defects that become apparent as the project progresses. Project analysis is necessary because of this. It involves examining each cost or issue associated with a project before beginning work on it and the results thereafter.

- **Design drawings/schematics/ solid models**

As this is a group project, so our prime aim for the completion of the project is to follow a strategic development.

- **Design:**

After developing wireframes and planning a roadmap we will move to designing. We will create one or more models for the proposed solution with each being different from the previous while keeping the errors and changes in mind and doing changes based of our requirements.

- **Development/ implementing: -**

Development of the solution started with citing research papers in the similar field and analyzing the different solutions given by them

Then the design is implemented roughly as the pseudo code and a rough image of the solution is drawn out of it. Solution is the coded using Python on Kaggle Notebook

- **Testing: -**

After the completion of all the stages testing will be our last stage where we are going to test the whole system and see if it meets the desired requirements. If some bugs occur or if the output is not according to the expectations then further changes will be made into the project.

- **Project management, and communication**

For timely work project management is done perfectly by our team leader it's really important for good communication between team members to relay all the information timely for good completion of the project.

- **Testing/characterization/interpretation/data validation**

Unit testing, integration testing, system testing, and acceptance testing are the four basic testing phases that must be finished before a programme is approved for usage.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

The Suggested Method for classification utilizes U-Net for feature extraction but unlike the original network we utilized different Activation function and added a Guided filter for the images within the dataset

As we can observe in the Table 4.1 Conventional Method with activation function ReLu, gives an average of **0.8659** accuracy for the epochs observed in the table the activation function Sigmoid following performing slightly better with an average accuracy of **0.8953**.

After changing the activation function we start to see improvement in the accuracy of the model where with Activation function Tanh it performed remarkably well with an average accuracy of **0.9855** which seems small but in reality is a significant improvement over the original method 8

For even further improvement A Guided filter is added to further enhance the images which once again can be observed in the output table, with Guided Filter accuracy for the original method increased but highest accuracy result is observed in the Activation Function TanH with guided filter applied, with average accuracy of **0.9968**

So the suggested method outperforms the original UNet network.

5.2 Future work

The work can be extended by using different CNN networks for future work. Additionally other enhancement and denoising techniques can be considered for feature enhancement. For example; homomorphic filter , anisotropic filter DWT (Discrete wavelet transform) , Small scale retinex, multi scale retinex etc. Also in the future researchers should consider the Classification part using the same model for calculation of cup to disc ratio (CDR) and rim to disc ratio (RDR) to detect the Glaucomatic conditions of the fundus image.

REFERENCES

- [1] Quigley HA. Number of people with glaucoma worldwide. *Br J Ophthalmol* , vol. 80, pp 389–93, 1996.
- [2] Traverso CE, Walt JG, Kelly SP, Hommer AH, Bron AM, Denis P, et al. Direct costs of glaucoma and severity of the disease: a multinational long term study of resource utilisation in Europe. *Br J Ophthalmol*, vol. 89, pp. 1245–1249, 2005.
- [3] Iester M, Zingirian M. Quality of life in patients with early, moderate and advanced glaucoma. *Eye*, vol. 16, pp. 44–49, 2002.
- [4] Herbert Gross, Fritz Blechinger, Bertram Aichtner. Handbook of Optical Systems: Survey of Optical Instruments. *Human Eye*, vol. 4, pp. 1-87, 2008.
- [5] Oyster, Clyde W. "The human eye." *Sunderland*, vol. 4, 1999.
- [6] Janz NK, Wren PA, Guire KE, Musch DC, Gillespie BW, Lichter PR. Collaborative Initial Glaucoma Treatment Study. Fear of blindness in the Collaborative Initial Glaucoma Treatment Study: patterns and correlates over time. *Ophthalmology*, vol. 114, pp. 2213–20, 1998.
- [7] Quigley, Harry A. "Open-angle glaucoma." *New England Journal of Medicine* 328, vol. 15, pp. 1097-1106, 1993.
- [8] Fraser, S. G., and R. P. L. Wormald. "Hospital Episode Statistics and changing trends in glaucoma surgery." *Eye* 22, vol. 1, pp. 3-7, 2008.
- [9] Lee, David A., and Eve J. Higginbotham. "Glaucoma and its treatment: a review." *American journal of health-system pharmacy* 62, vol. 7, pp. 691-699, 2007.
- [10] Weinreb, Robert N., Tin Aung, and Felipe A. Medeiros. "The pathophysiology and treatment of glaucoma: a review." *Jama* 311, vol. 18, pp. 1901-1911, 2014.
- [11] Virk, Jiwanpreet Kaur, Mooninder Singh, and Mandeep Singh. "Cup-to-disk ratio (CDR) determination for glaucoma screening." In *2015 1st International Conference on Next Generation Computing Technologies (NGCT)*, pp. 504-507. IEEE, 2015.

- [12]MacMonnies, C. W. Glaucoma history and risk factors. *Journal of optometry*, vol. 10(2), pp. 71-78, 2014.
- [13]Razeghinejad, M. Reza, and George L. Spaeth. "A history of the surgical management of glaucoma." *Optometry and Vision Science* 88, vol. 1, E39-E47, 2011.
- [14]Runge, Paul E. "Eduard Jaeger's Test-Types (Schrift-Scalen) and the historical development of vision tests." *Transactions of the American Ophthalmological Society*, vol. 98, pp. 375, 2000.
- [15]Calvin, Helen, Pamela Rupnow, and Theodore Grosvenor. "How good is the estimated cover test at predicting the von Graefe phoria measurement?." *Optometry and Vision Science: Official Publication of the American Academy of Optometry* 73, vol. 11, pp. 701-706, 1996.
- [16]Sinthanayothin, Chanjira, James F. Boyce, Helen L. Cook, and Thomas H. Williamson. "Automated localisation of the optic disc, fovea, and retinal blood vessels from digital colour fundus images." *British journal of ophthalmology* 83, vol. 8, pp. 902-910, 2004.
- [17]Sivaswamy, Jayanthi, S. R. Krishnadas, Gopal Datt Joshi, Madhulika Jain, and A. Ujjwaft Syed Tabish. "Drishti-gs: Retinal image dataset for optic nerve head (onh) segmentation." In 2014 *IEEE 11th international symposium on biomedical imaging (ISBI)*, pp. 53-56, IEEE 2014.
- [18]Foster, Paul J., Ralf Buhrmann, Harry A. Quigley, and Gordon J. Johnson. "The definition and classification of glaucoma in prevalence surveys." *British journal of ophthalmology* 86, vol. 2, pp. 238-242, 2002.
- [19]Venkatesh, Svetha, and Robyn Owens. "On the classification of image features." *Pattern Recognition Letters* 11, vol. 5, pp. 339-349, 2002.
- [20]Sivaswamy, J., Krishnadas, S. R., Joshi, G. D., Jain, M., & Tabish, A. U. S. (2014, April). Drishti-gs: Retinal image dataset for optic nerve head (onh) segmentation. In 2014 *IEEE 11th international symposium on biomedical imaging (ISBI)*, vol. 4, pp. 53-56, IEEE, 2014.
- [21]Ramachandran, Prajit, Barret Zoph, and Quoc V. Le. "Searching for activation functions." *preprint*, vol. 1, pp. 17-14, 2017.

- [22]Iliev, A., Nikolay Kyurkchiev, and S. Markov. "On the Approximation of the step function by some sigmoid functions." *Mathematics and Computers in Simulation* vol. 133, pp. 223-234, 2017.
- [23]Fan, Engui. "Extended tanh-function method and its applications to nonlinear equations." *Physics Letters A* 277, vol. 4(5), pp. 212-218, 2004.
- [24]Schmidt-Hieber, Johannes. "Nonparametric regression using deep neural networks with ReLU activation function." *The Annals of Statistics* 48, vol. 4, pp. 1875-1897, 2004.
- [25]Dubey, Shiv Ram, Satish Kumar Singh, and Bidyut Baran Chaudhuri. "Activation functions in deep learning: a comprehensive survey and benchmark." *Neurocomputing*, vol. 503, pp. 92-108, 2022.
- [26]Du, Getao, Xu Cao, Jimin Liang, Xueli Chen, and Yonghua Zhan. "Medical image segmentation based on u-net: A review." *Journal of Imaging Science and Technology*, vol. 64, pp. 1-12, 2020.
- [27]Siddique, Nahian, Sidike Paheding, Colin P. Elkin, and Vijay Devabhaktuni. "U-net and its variants for medical image segmentation: A review of theory and applications." *Ieee Access*, vol. 9, pp. 82031-82057, 2021.
- [28]Selesnick, Ivan W. "The double-density dual-tree DWT." *IEEE Transactions on signal processing* 52, vol. 5, pp. 1304-1314, 2004.
- [29]Jogin, Manjunath, M. S. Madhulika, G. D. Divya, R. K. Meghana, and S. Apoorva. "Feature extraction using convolution neural networks (CNN) and deep learning." In *2018 3rd IEEE international conference on recent trends in electronics, information & communication technology (RTEICT)*, pp. 2319-2323. IEEE, 2018.
- [30]Qassim, Hussam, Abhishek Verma, and David Feinzimer. "Compressed residual-VGG16 CNN model for big data places image recognition." In *2018 IEEE 8th annual computing and communication workshop and conference (CCWC)*, pp. 169-175. IEEE, 2018.
- [31]Mateen, Muhammad, Junhao Wen, Sun Song, and Zhouping Huang. "Fundus image classification using VGG-19 architecture with PCA and SVD." *Symmetry* 11, vol. 1, pp. 1, 2018.

[32]S. Dash, S. Verma, S. Bevinakoppa, M. Wozniak, J. hafi, and M. Fazal Ijaz. "Guidance image-based enhanced matched filter with modified thresholding for blood vessel extraction." *Symmetry* 14, vol. 2, pp. 194, 2022.

[33]Seow, Ming-Jung, and Vijayan K. Asari. "Ratio rule and homomorphic filter for enhancement of digital colour image." *Neurocomputing* 69, vol. 7(9), pp. 954-958, 2006.

[34]Rahman, Zia-ur, Daniel J. Jobson, and Glenn A. Woodell. "Multi-scale retinex for color image enhancement." *In Proceedings of 3rd IEEE international conference on image processing*, vol. 3, pp. 1003-1006. IEEE, 1996.



ECE ARCHIVES PROJECT SUBMISSION FORM



Project Code: **CU/ECE/20**____/Sem____/UID_____ (To be filled by Office)

Project Name: _____

Name and UID of student: _____

Team Members:

S. No.	Name	UID	Semester	Contact No.
1.				
2.				
3.				
4.				
5.				

Section to be filled by Project Mentor

Status (Please tick, whichever applicable)

Working		Not Working	
Marks Awarded		60	

Project Mentor Details:

Name _____

Employee ID _____

Sign _____

Date _____

Section to be filled by Project Examiner(s)

Status (Please tick, whichever applicable)

Working		Not Working	
---------	--	-------------	--

Project Examiner Signatures:

Internal _____

Employee ID _____

External _____

Employee ID _____

Date _____

USER MANUAL

User manual is useful because:

It is amazing to be able to offer people the items and services they require. We are not only generating a profit, but we are also assisting others in solving their difficulties. The user Handbook, however, is a crucial component that should never be disregarded when offering A solution which is to be utilized by many professional people hence the User manual should be well made and should act as a step by step guide for the future users. Without it, even though our items ought to be straightforward to understand, some of the people might not genuinely comprehend how they perform.

To run the project, follow these steps:

1. First create a fresh environment which supports Python
2. Pip installs all the dependencies given in the file.
3. Once installed, open the file
4. Check for the files and add path accordingly, keep on checking for proper path by running the desired block of code
5. After successfully setting the path copy rest of the code
6. Run the model and check for proper output
7. Make alterations and keep on tagging the outcome
8. Add the details of given parameters for prediction
9. Note the final result and then compile the rest

