Iris Image Recognition Through Goggles Using CNN and Mathematical Models

Submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Electrical and Electronics Engineering

by

SUBHADEEP MAL (20BEE0179)

RIDHIMA CHOPRA (20BEE0328)

KEVIN GERARD (20BEI0079)

Under the guidance of

Dr. Rajini G. K.

School of Electrical Engineering

VIT,Vellore.

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

May,2024

DECLARATION

We here by declare that the thesis entitled "**Iris Image Recognition Through Goggles Using CNN and Mathematical Models**" submitted by us, for the award of the degree of *Bachelor of Technology in Electrical and Electronics Engineering* to Vellore Institute of Technology, University is a record of bonafide work carried out by me under the supervision of **Dr. Rajini G.K.**, HOD, EIE, SELECT, VIT University, Vellore.

We further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Ridhima

Subhadeep Mal

A. Kevin gerard

Place: Vellore

Date: 09/05/2024

Signature of the Candidate

CERTIFICATE

This is to certify that the thesis entitled "**Iris Image Recognition Through Goggles Using CNN and Mathematical Models**" submitted by **Subhadeep Mal (20BEE0179)**, SELECT School, Vellore Institute of Technology, Vellore, for the award of the degree of *Bachelor of Technology in Electrical and Electronics Engineering* , is a record of bonafide work carried out by him under my supervision during the period, 01.01.2024 to 09.05.2024, as per the Vellore Institute of Technology code of academic and research ethics.

The contents of this report have not been submitted and will not be submitted either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university. The thesis fulfills the requirements and regulations of the University and in my opinion, meets the necessary standards for submission.

Place: Vellore

Date: 09/05/2024

Signature of the Guide

(Dr. Rajini G.K.)

The thesis is satisfactory / unsatisfactory

Approved by

Head of the Department
EEE

Dean,SELECT

Acknowledgements

With immense pleasure and a deep sense of gratitude, we wish to express my sincere thanks to our supervisor **Dr. Rajini G.K.**, HOD EIE, SELECT School, Vellore Institute of Technology, University, without her motivation and continuous encouragement, this project work would not have been successfully completed.

We are grateful to the Chancellor of Vellore Institute of Technology, University, **Dr. G.Viswanathan**, the Vice Presidents, and the Vice Chancellor for motivating us to carry out research in Vellore Institute of Technology, University and also for providing me with infrastructural facilities and many other resources needed for our project.

We express our sincere thanks to **Dr. Mathew M. Noel**, Dean, SELECT, Vellore Institute of Technology, University for his kind words of support and encouragement. We like to acknowledge the support rendered by **our colleagues** in several ways throughout our project work.

We wish to extend our profound sense of gratitude to **our parents** for all the sacrifices they made during our research and also for providing us with moral support and encouragement whenever required.

Place: Vellore

Date: 09/05/2024 **Subhadeep Mal**

Executive Summary

Biometric recognition, utilizing physiological or behavioural characteristics, serves as a cornerstone in authenticating the identity of individuals. The iris, with its unique and scarcely affected pattern by aging, stands out as a reliable biometric identifier. It also eliminates contact between individuals authenticating their identity unlike in case of fingerprint recognition, where fingerprint is verified by means of touch.

In this study, we present a comprehensive approach for iris detection employing VGG16, a Neural Network model, alongside MATLAB's image processing toolbox. Our methodology encompasses training and fine-tuning VGG16 parameters as well as mathematical approach for iris localization. It is followed by employing various mathematical techniques for iris segmentation, feature extraction, and database matching to ascertain individual identity. Furthermore, we address the challenge of specular reflections commonly encountered on glass surfaces, particularly when individuals wear goggles, which can impede accurate iris recognition.

Through systematic experimentation and analysis, we demonstrate the effectiveness and robustness of our proposed iris recognition system, offering significant advancements in biometric authentication technology. We have also put forward a comparison of various approaches taken for the different steps in iris recognition.

TABLE OF CONTENTS

Acknowledgement i
Executive Summary ii
List of Figures vi
List of Tables viii
List of Terms and Abbreviations ix
List of Symbols x

1 INTRODUCTION 1
1.1 Objective 1
1.2 Motivation 1
1.3 Background 2

2 PROJECT DESCRIPTION AND GOALS 3
2.1 Review of Literature 3
2.2 Project Description 5
  2.2.1 About Iris 5
  2.2.2 The Proposed Iris Recognition System 6
2.3 Goals 7

3 TECHNICAL SPECIFICATION 8
3.1 Software Specification 8
  3.1.1 MATLAB R2021a 8
  3.1.2 Python 8
  3.1.3 Visual Studio Code (VS Code) 10
3.2 Hardware Specification 10

4 DESIGN APPROACH AND DETAILS 11
4.1 Methodology 11

4.2 About CASIA Dataset 12
4.3 Pre-Processing 13
  4.3.1 Histogram Equalization (HE) 13
  4.3.2 Contrast Limited Adaptive Histogram Equalization (CLAHE) . . . 15
  4.3.3 Unsharp Masking 16
4.4 Iris Localization Using Mathematical Model 16
  4.4.1 Canny Edge Detection 17
  4.4.2 Hough Transform 18
4.5 Iris Localization Using CNN 20
  4.5.1 About Neural Networks: 20
  4.5.2 CNN 20
  4.5.3 Overview of CNN model training for Iris localization: 23
4.6 Iris Segmentation 29
4.7 Iris Normalization 29
4.8 Feature Extraction 30
  4.8.1 Discrete Wavelet Transform + Principle Component Analysis . . . 30
  4.8.2 Local Binary Pattern 31
  4.8.3 Gabor Filter 32
4.9 Feature Matching 33
4.10 Tackling Specular Reflections 33

5 SCHEDULE, TASKS AND MILESTONES 34
5.1 Schedule 34
5.2 Tasks and Milestones 35
  5.2.1 First Review 36
  5.2.2 Second Review 36
  5.2.3 Third Review 36

6 PROJECT DEMONSTRATION 37
6.1 Pre-Processing 37
6.2 Iris Localization using Mathematical Model 39
6.3 Iris Localization using VGG16 Model 39
6.4 Iris Segmentation 41

6.5 Iris Normalization 42
6.6 Feature Extraction 42
6.7 Feature Matching 43
6.8 Tackling Specular Reflections 44

7 COST ANALYSIS / RESULT AND DISCUSSION 45
7.1 Cost Analysis 45
7.2 Results and Discussion 45

8 SUMMARY 48
8.1 Summary 48
REFERENCES 48
Curriculum Vitae 51

Capstone Project Summary 52

LIST OF FIGURES

2.1 Human Eye Anatomy 5

2.2 Iris Features 6

2.3 Block Diagram of Iris Recognition System 7

4.1 Our Proposed Methodology for Iris Recognition 11

4.2 CASIA V3 and V4 Dataset 12

4.3 Example of Histogram Equalization 13

4.4 Example of CLAHE 15

4.5 Canny Edge Detection Performed on Lena Image 17

4.6 Illustration of Hough Transform to detect Circles 19

4.7 CNN Basic Architecture 20

4.8 Graphs of Activation Functions 21

4.9 CNN Methodology 22

4.10 CASIA datasets comparison 23

4.11 Labelme 24

4.12 Annotated output file 24

4.13 Iris located example image used to train the model 25

4.14 VGG16 Architecture 26

4.15 VGG16 parameter summary 27

4.16 Daughman's Rubber Sheet Model 29

4.17 Lena Image Decomposition using DWT 30

4.18 Example of a Gabor Filter Bank 32

5.1 Schedule of the Project 34

6.1 Original Image 37

6.2 Histogram Equalized Image 37

6.3 Contrast Limited Adaptive Histogram Equalized Image 38

6.4 Histogram Equalized Image with Unsharp Masking 38

6.5 Contrast Limited Adaptive Histogram Equalized Image with Unsharp Masking . . . 38
6.6 Localization Using Built-In imfindcircles function . . . 39
6.7 Localization Using User-Defined imfindcircles function . . . 39
6.8 Confusion Matrix . . . 40
6.9 Losses for 10 epochs . . . 40
6.10 Losses for 15 epochs . . . 40
6.11 On Test data . . . 41
6.12 On External image . . . 41
6.13 Segmented Iris . . . 41
6.14 Normalized Iris . . . 42
6.15 Stored Feature Vectors . . . 42
6.16 Match Found in the Database . . . 43
6.17 Match Not Found in the Database . . . 43
6.18 Tackling Specular Reflection . . . 44

LIST OF TABLES

5.1 Tasks and Milestones for the Project 36
7.1 MSE Values of different Pre-processed Iris Image 45
7.2 PSNR Values (in dB) of different Pre-processed Iris Image 46
7.3 Performance Evaluation of various Feature Extraction Methods 47

LIST OF TERMS AND ABBREVIATIONS

CLAHE	Contrast Limited Adaptive Histogram Equalization	1
CNN	Convolutional Neural Network	1, 6
DWT	Discrete Wavelet Transform	1, 30
HE	Histogram Equalization	1
LBP	Local Binary Pattern	1, 30, 47
MSE	Mean Squared Error	1
NIR	Near Infrared	53
PCA	Principle Component Analysis	1, 30
PSNR	Peak Signal to Noise Ratio	1
VGG	Visual Graphics Group	6

LIST OF SYMBOLS

1. σ - standard deviation
2. G - Gaussian Blur
3. M - Gradient magnitude
4. θ - Gradient direction
5. W_ϕ - Discrete Wavelet coefficients
6. $\phi_{j_0,m,n}$ - Wavelet function
7. z - Standardized or z-score value
8. μ - Mean of the sample
9. λ - Wavelength of the sine component
10. ϕ - Phase offset
11. γ - Spatial aspect ratio

CHAPTER 1

INTRODUCTION

1.1 Objective

Innovative approaches to identity verification and authentication, known as biometric methods, rely on an individual's distinct physical or behavioral traits. Biometric methods are a more convenient and secure way to verify someone's identity than traditional methods like passwords or access cards, which can be misplaced, stolen, or forgotten. Iris recognition technology has become a leading biometric method because of its distinct benefits. Its unmatched precision, non-intrusive operation, and resilience to fraud make it a favored option for various purposes, from access control to healthcare.

The objective of the project lies in recognizing the iris of an individual even through the goggles. The specular reflection on the goggles hinders the process of iris recognition. Thus, this thesis aims at tackling the specular reflections that occur on the glass surfaces. Moreover, by iris recognition, we aim to provide a contactless solution of biometric authentication as it can be highly useful during pandemics like COVID-19 where being contactless is highly important. This thesis also tabulates the performance of different pre-processing methods like Histogram Equalization (HE), Contrast Limited Adaptive Histogram Equalization (CLAHE) along with a combination of Unsharp Masking methods using metrics like Mean Squared Error (MSE), Peak Signal to Noise Ratio (PSNR) etc. Iris Localization is performed by Convolutional Neural Network (CNN) as well as through Canny Edge Detection and Hough Transform. Features from normalized iris are extracted using a combination of Discrete Wavelet Transform (DWT) and Principle Component Analysis (PCA), Local Binary Pattern (LBP) and Gabor Filter. Their performances have also been tabulated.

1.2 Motivation

Previously, we undertook a comprehensive course on "Digital Image Processing" as part of our academic curriculum. This course delved into fundamental concepts such as image pre-processing, enhancement, restoration, and object recognition, providing us with a strong foundation in handling visual data. As a culmination of this learning, we were tasked with a project that applied these principles to a real-life problem.

Our motivation for choosing a project in iris recognition through goggles arose from the widespread use of eyewear across various environments, including workplaces, public spaces, sports, and healthcare. Goggles serve multiple purposes, from protection and vision correction to fashion statements, making them a ubiquitous accessory. However, the existing iris recognition systems often encounter difficulties in accurately capturing and recognizing iris patterns when individuals wear goggles. This limitation results in decreased performance and reliability, undermining the effectiveness of biometric authentication systems.

By addressing this challenge, our project aims to enhance the performance of iris recognition technology in scenarios where goggles are commonly worn. This includes developing algorithms and techniques specifically tailored to handle iris images captured through goggles, ensuring robust and accurate identification even in diverse environmental conditions. Such advancements are essential for improving security measures, enhancing user experience, and promoting the seamless integration of biometric technology into everyday applications.

1.3 Background

There have been numerous studies conducted on iris recognition due to its applications in biometrics and security. However, several issues and challenges exist in iris recognition technology. Specular reflection occurs when light is reflected directly off a surface, such as the cornea of the eye. This reflection can distort the appearance of iris patterns captured in images, leading to inaccuracies in feature extraction and matching algorithms.

They can reduce the contrast between iris patterns and the surrounding areas in an image. This reduction, in contrast, makes it challenging for iris recognition algorithms to distinguish fine details and extract accurate features for matching. Moreover, Iris recognition systems use non-contact imaging techniques, typically based on NIR light, to capture high-resolution images of the iris. This non-contact approach eliminates the need for physical contact between the user and the biometric sensor, making it a touchless authentication method. This project aims at a comparative analysis of various approaches for different steps of iris recognition.

CHAPTER 2

PROJECT DESCRIPTION AND GOALS

2.1 Review of Literature

A literature review is defined as the critical analysis and evaluation on previously existing work on a particular field. Some of the journals that talk about out the related work in the iris recognition field are as follows:

Mattar (2013) gave his overview on Principal Components Analysis(PCA) Based Iris Recognition and Identification System. His research lies in adapting the PCA and ANN techniques, commonly used for general image recognition, specifically for the iris recognition problem by encoding iris patterns as eigen irises. The promising results on non-ideal data highlight the potential of this PCA-based iris recognition approach.

H. and Malisuwan (2014) gave their views on Image Enhancement for Iris Recognition. Specifically, they have explored and compared the performance of techniques like Histogram Equalization, Brightness Preserving Bi-Histogram Equalization, Dynamic Histogram Equalization, Adaptive Histogram Equalization, Contrast Limited Adaptive Histogram Equalization, Dualistic Sub-Image Histogram Equalization, and Multi-Scale Adaptive Histogram Equalization within an iris recognition system. Their study finds that CLAHE and MAHE give the highest recognition performance among the explored enhancement methods.

Omran and AlShemmary (2020) suggested the idea of using a Deep convolutional Neural Network for iris Recognition. The novel aspects are using a custom DCNN called IRISNet for automatic iris feature learning and classification, without relying on hand-crafted iris encoding methods. Preprocessing is still done using conventional techniques.

Jusman et al. (2020) proposed their research on ”Performances of proposed normalization algorithm for iris recognition”. They proposed an enhanced iris recognition pipeline focused on a new normalization algorithm that improves recognition accuracy compared to existing techniques when combined with GLCM feature extraction and discriminant analysis classification on iris images.

Tahir et al. (2021) proposed their work on iris recognition. The key contributions of their work include the new method for pupil boundary detection, the modified Wildes' method for limbic boundary detection, and the adoption of the R-C-E-S method for eyelid detection. The authors claim that their system is adaptive to different datasets and achieves reasonable accuracy and EER compared to previous methods.

Jayalaxmi et al. (2022) introduced a novel iris recognition algorithm integrating patch-wise processing, empirical mode decomposition (EMD) for feature extraction, and Support Vector Machine (SVM) classification, enhancing performance through localized information and data-driven features.

Farouk et al. (2022) introduced a hybrid technique combining edge detection, segmentation, CNN, and HD for biometric recognition using iris data from datasets like CASIA-Iris-Interval V4, IITD, and MMU. The proposed model achieves high recognition accuracies of 94.88%, 96.56%, and 98.01% on these datasets, demonstrating its superiority over other classifiers in the literature.

Jan et al. (2021) proposed a robust iris localization scheme that combines preprocessing techniques, adaptive thresholding, morphological operations, centroid-based pupil localization, Circular Hough transform (CHT) for iris outer boundary marking, and Fourier series for boundary refinement. This scheme achieves improved results compared to contemporary iris localization methods on public iris databases like IITD V1.0, CASIA-Iris-Interval, and MMU V1.0.

Othman et al. (2016) discussed the evolution of the open-source iris recognition system OSIRIS, focusing on versions OSIRISV2, OSIRISV4, and OSIRISV4.1. Developed within the BioSecure Association, OSIRIS provides a reference for the scientific community with modules such as segmentation, normalization, feature extraction, and template matching. OSIRISV4.1 introduces a novel iris normalization approach and shows performance improvements on databases like ICE2005 and CASIA-IrisV4-Thousand, making it a recommended baseline system for iris recognition research and comparison with other algorithms.

2.2 Project Description

In an era where security and identity verification are of paramount importance, biometric authentication methods have gained significant attention for their reliability and effectiveness. Among these methods, iris recognition stands out as a highly accurate and secure technology for identifying individuals based on the unique patterns present in their irises. The human eye anatomy is shown in **Fig. 2.1** for a better understanding.

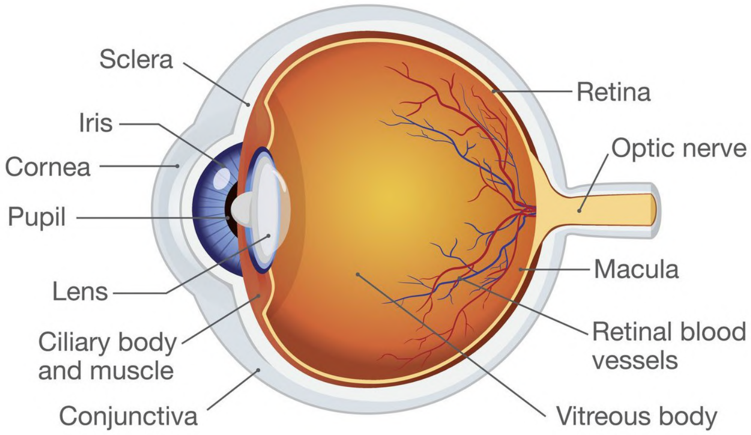


Fig. 2.1 Human Eye Anatomy

2.2.1 About Iris

The Iris shown in **Fig.2.2**, situated behind the cornea, is a muscular structure in the human eye responsible for regulating pupil size by contracting or expanding. When light enters the eye, certain nerve cells at the rear transmit signals to the brain, prompting adjustments in iris size based on light intensity.

When observing An iris, presents a combination of colors and patterns, with the three primary patterns being:

- Pigmented rings: Broadly colored bands encircling the pupil.
- Crypts: Varying in size, these diamond-shaped gaps are dispersed across the iris.
- Furrows: Curved, pale lines traversing the iris.

With its intricate and stable structure, the iris offers a rich source of biometric information that remains relatively unchanged throughout a person's life.

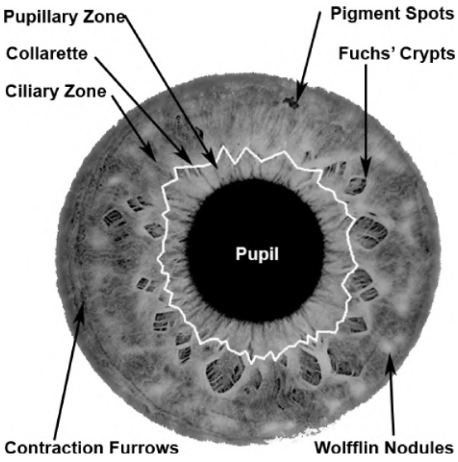


Fig. 2.2 Iris Features

2.2.2 The Proposed Iris Recognition System

The Iris Recognition system is described in **Fig. 2.3**. We have used CASIA-IrisV3 Interval images which contain a total of 22,035 iris images from more than 700 subjects. All iris images are 8-bit gray-level JPEG files, collected under near-infrared illumination. Some of these images are included in the dataset and are then subjected to preprocessing through various methods such as Histogram Equalization, Contrast Limited Histogram Equalization, and then a combination with Unsharp Masking Method.

This paper compares the various preprocessing methods using performance measures like Mean Squared Error and Peak Signal to Noise Ratio. Preprocessing enhances image quality for better performance. Next, we identify the Iris using the VGG16 Neural Network model.VGG16 is a popular CNN architecture developed by the Visual Graphics Group (VGG) at the University of Oxford. It consists of 16 layers, including 13 convolutional layers and 3 fully connected layers, and is known for its simplicity and high performance in image classification tasks.

Mathematically, Iris is detected using Canny Edge detection and Hough Transform. The Iris is segmented and then it is normalized using Daughman's Rubber Sheet Model. The iris region is transformed from Cartesian coordinates to polar coordinates. This transformation helps in normalizing the iris image and making it invariant to changes in pupil dilation, eyelid position, and eye rotation.

The normalized iris obtained is subjected to various feature extraction methods like Discrete Wavelet Transform and Principle Component Analysis, Local Binary Pattern, and Gabor Filter. The performance of these feature extraction methods is evaluated based on the time taken to extract features. Hamming Distance is used to match the feature vectors of the input image with the feature vectors stored in the database. Feature

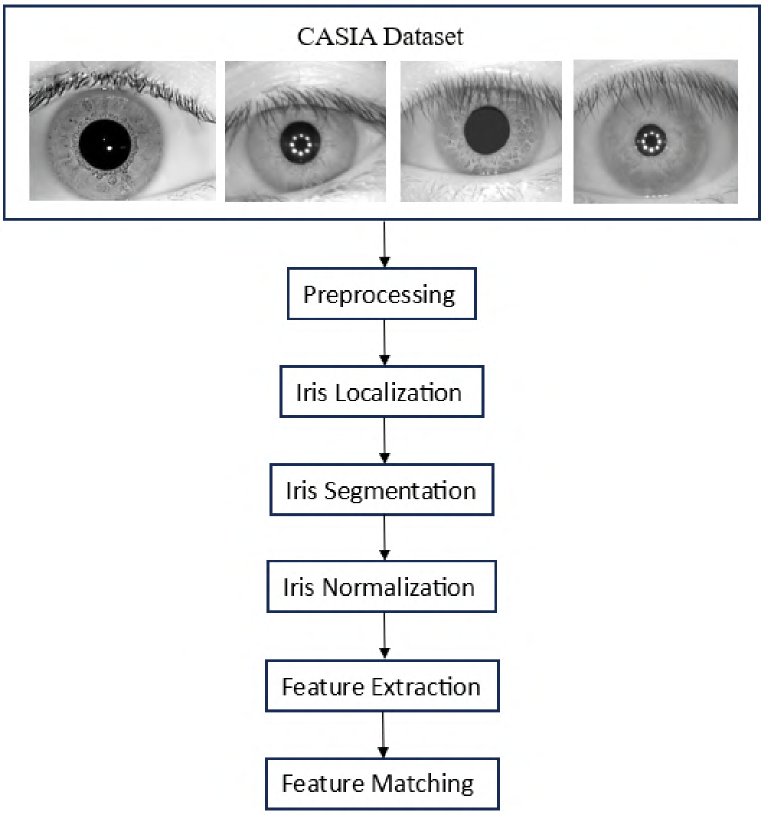


Fig. 2.3 Block Diagram of Iris Recognition System

vectors are matched perfectly when the Hamming Distance is zero.

2.3 Goals

The primary objective of our project is to gain a comprehensive understanding of how iris recognition systems operate and to conduct a comparative analysis of various approaches used for iris recognition. We aim to explore the nuances of different methodologies and algorithms, evaluating their efficacy and identifying areas for improvement.

One of the key challenges we seek to address is the impact of specular reflections on iris recognition accuracy. These reflections can significantly hinder the precision and reliability of biometric systems. To mitigate this issue, we intend to implement advanced software-level techniques specifically designed to counteract specular reflections.

By developing and testing these innovative solutions, we aim to enhance the overall performance and robustness of iris recognition systems, making them more effective and reliable in real-world scenarios where specular reflections are prevalent.

CHAPTER 3

TECHNICAL SPECIFICATION

3.1 Software Specification

The softwares that are used for setting up the iris recognition system are:

3.1.1 MATLAB R2021a

MATLAB, designed primarily for scientists and engineers, empowers users to study, innovate, and develop transformative products and systems. Central to MATLAB is its matrix-based language, facilitating intuitive computational mathematics expressions. This language, known for its simplicity and versatility, forms the backbone of MATLAB's capabilities, enabling seamless integration of mathematical algorithms, data analysis, and visualization. With MATLAB, users can efficiently tackle complex scientific and engineering challenges, ranging from signal processing and image analysis to machine learning and control systems, contributing to advancements that shape and impact the world.

3.1.1.1 Image Processing Toolbox

For image processing, analysis, visualization, and algorithm development, Image Processing Toolbox offers an extensive collection of industry-standard algorithms and workflow applications. Both deep learning and conventional image processing methods can be used to accomplish image segmentation, image enhancement, noise reduction, geometric transformations, and image registration. Processing of 2D, 3D, and infinitely large images is supported by the toolbox.

3.1.2 Python

Python is a programming language known for its interpretative nature, object-oriented approach, and high-level design. It offers many built-in libraries and built-in data structures. Additionally, it serves well as a scripting or integration language, connecting various components seamlessly. Python's straightforward syntax prioritizes readability, reducing maintenance costs. It promotes modularity and code reuse through support for modules and packages. The Python interpreter and comprehensive standard library are open-source across major platforms, allowing for unrestricted distribution.

3.1.2.1 Libraries Used:

- OS: This module offers a versatile approach to accessing operating system-specific features. When our task involves simple file operations like reading or writing, we utilize the open() function. For handling paths, the os.path module comes into play, while for reading lines from files specified on the command line, the file input module serves the purpose.
- Time: This module provides various time-related functions. For related functionality, there is also the datetime and calendar modules.
- OpenCV: OpenCV (Open Source Computer Vision Library) is a freely available software library dedicated to computer vision and machine learning. It was developed with the aim of offering a unified platform for various computer vision applications and promoting the integration of machine perception into commercial products. With its Apache 2 license, OpenCV facilitates businesses in utilizing and adapting the code according to their needs.
- Tensorflow: TensorFlow is a freely available software library designed for efficient numerical computation. Its adaptable structure enables seamless distribution of computation across diverse platforms, including CPUs, GPUs, and TPUs, catering to desktops, server clusters, as well as mobile and edge devices.
- Json: Python has a built-in package called JSON, which is used to work with JSON data.
- Numpy: NumPy serves as the cornerstone for scientific computing in Python. It offers a Python library featuring a multi-dimensional array object, along with related objects like masked arrays and matrices. Additionally, NumPy provides a wide range of functions for swift array operations, covering mathematical, logical, shape manipulation, sorting, selection, input/output tasks, discrete Fourier transforms, basic linear algebra, statistical operations, random simulations, and beyond.
- Matplotlib: Matplotlib is a powerful tool in Python for making different kinds of visuals, like pictures that don’t move, ones that do move, and even ones where you can click and interact with them. It’s like a toolbox for creating all sorts of cool images and charts in Python.

- Labelme: Labelme is a graphical image annotation tool. It is written in Python and uses Qt for its graphical interface.
- Albumentations: Argumentation is a versatile and speedy library for augmenting images. It's highly popular across various domains including industry, deep learning research, machine learning competitions, and open source initiatives.

3.1.3 Visual Studio Code (VS Code)

Visual Studio Code, commonly known as VS Code, is a versatile source-code editor developed by Microsoft, compatible with Windows, Linux, and macOS platforms. Its feature-rich environment includes robust debugging tools, syntax highlighting for various programming languages, intelligent code suggestions, a vast library of code snippets, and utilities for code organization and restructuring. Additionally, VS Code seamlessly integrates Git version control, enabling efficient collaboration and version management for software development projects. With its user-friendly interface and extensive plugin ecosystem, VS Code has become a preferred choice for developers across different domains, enhancing productivity and streamlining the coding experience.

3.2 Hardware Specification

The above software is run on the laptop with the following specifications:

- Processor: The Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz with a maximum turbo frequency of 3.6 GHz is a capable processor commonly found in midrange laptops. Its quad-core design provides efficient multitasking and handling of moderate computational workloads, making it suitable for tasks like programming, data analysis, and light gaming.
- RAM: With 8.00 GB of RAM (7.79 GB usable), the laptop offers sufficient memory for running multiple applications simultaneously and handling moderate to heavy workloads. This amount of RAM is well-suited for tasks involving software development, image and video editing, as well as running virtual machines for testing and development purposes.
- System Type: Being a 64-bit operating system with an x64-based processor, the laptop supports advanced computing capabilities, including access to larger memory spaces and improved performance for 64-bit applications. This architecture is essential for running modern software, particularly data-intensive applications and computational tools like MATLAB that benefit from 64-bit processing power.

CHAPTER 4

DESIGN APPROACH AND DETAILS

4.1 Methodology

This iris recognition system deals with two approaches for the localization of the iris. The methodology is as shown in **Fig. 4.1**.

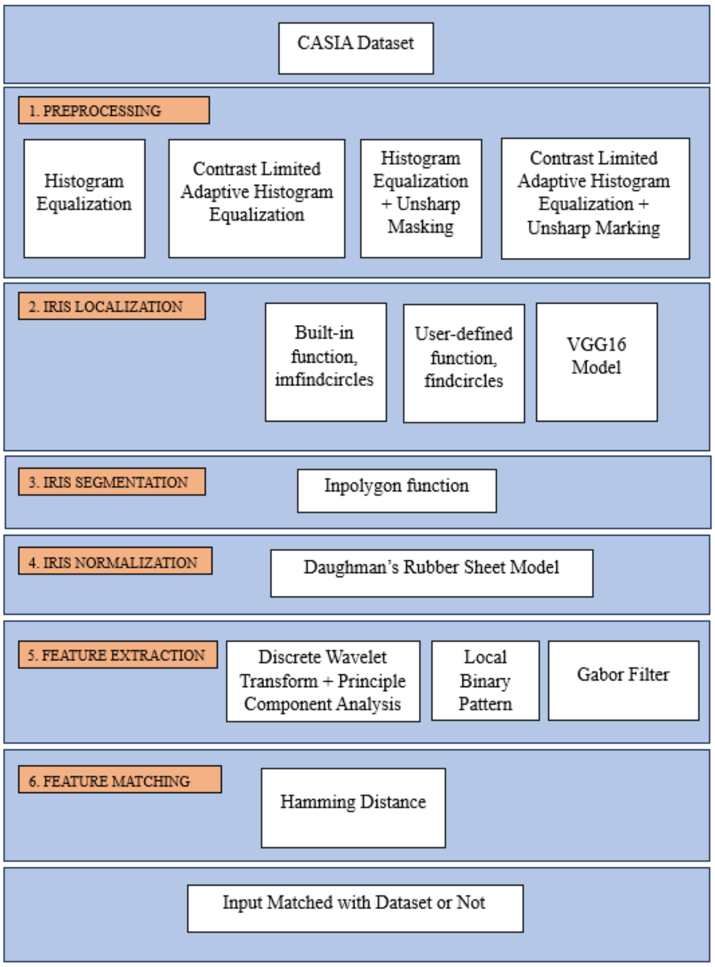

CASIA Dataset
1. PREPROCESSING
Histogram Equalization
Contrast Limited Adaptive Histogram Equalization
Histogram Equalization + Unsharp Masking
Contrast Limited Adaptive Histogram Equalization + Unsharp Marking
2. IRIS LOCALIZATION
Built-in function, imfindcircles
User-defined function, findcircles
VGG16 Model
3. IRIS SEGMENTATION
Inpolygon function
4. IRIS NORMALIZATION
Daughman's Rubber Sheet Model
5. FEATURE EXTRACTION
Discrete Wavelet Transform + Principle Component Analysis
Local Binary Pattern
Gabor Filter
6. FEATURE MATCHING
Hamming Distance
Input Matched with Dataset or Not

Fig. 4.1 Our Proposed Methodology for Iris Recognition

Preprocessing in iris biometric systems involves cleaning and enhancing raw iris images to improve accuracy. Then, iris localization identifies and isolates the iris region from the rest of the eye image. Iris segmentation further refines this by extracting the iris texture and removing unwanted areas. This is followed by iris normalization, which standardizes iris images to a consistent size and orientation for reliable comparisons. Feature extraction techniques like Gabor filters or DWT isolates unique iris patterns for recognition. Storage and retrieval systems manage the database of encoded iris templates securely. Feature matching algorithms such as Hamming Distance compare extracted features to stored templates to identify individuals accurately. Post-matching verification stages ensure reliable recognition results. Overall, these steps form a robust iris recognition system for biometric authentication.

4.2 About CASIA Dataset

A total of 22,035 iris images from over 700 subjects are included in CASIA-IrisV3. Three subsets, dubbed CASIA-Iris-Interval, CASIA-Iris-Lamp, and CASIA-Iris-Twins, are part of CASIA-IrisV3. Iris images are all 8 bit gray-level JPEG files that were taken in near infrared light. CASIA-IrisV4 is an extension of CASIA-IrisV3. It contains subsets from CASIA as well as new subsets like CASIA-Iris-Distance, CASIA-Iris-Thousand and CASIA-Iris-Syn as shown in **Fig. 4.2**.We have used CASIA-Iris-Interval and CASIA-Iris-Thousand for our project (**CBSR** (**2010**)).

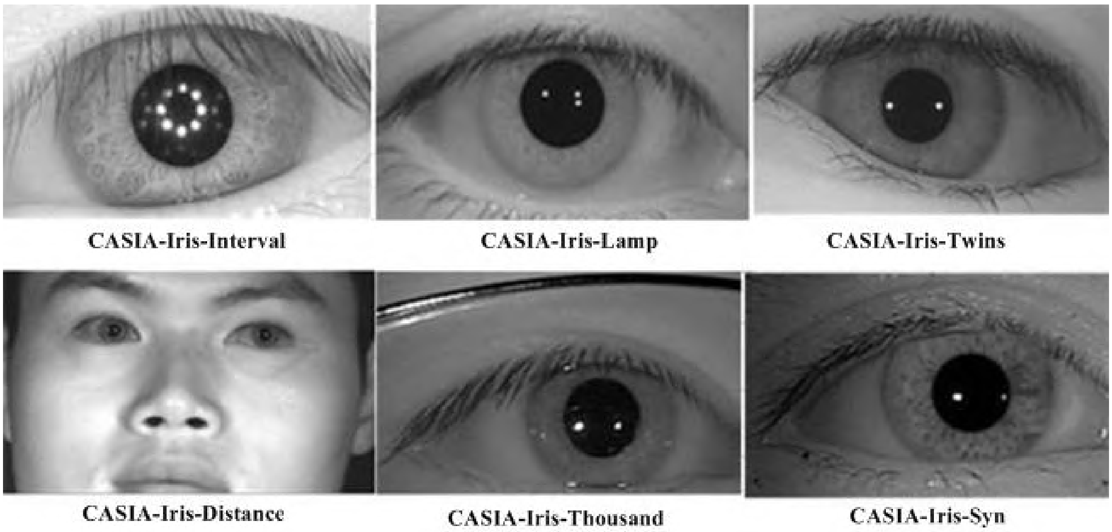


Fig. 4.2 CASIA V3 and V4 Dataset

A close-up iris camera was used to take iris images of the CASIA-Iris-Interval. The most intriguing aspect of our iris camera is the circular NIR LED array created, which has the right amount of light flux for iris imaging. This innovative design allows our iris camera to capture extremely clear iris images. The CASIA-Iris-Interval is a useful tool for examining the fine-grained texture characteristics of iris pictures.
CASIA-Iris-Thousand contains 20,000 iris images from 1000 subjects. It also contains

eye images with glasses and specular reflections. These images were collected using IKEMB-100 camera.

4.3 Pre-Processing

Images must be pre-processed to normalize pixel values, reduce noise, extract features, augment data, reduce dimensionality, and enhance image quality for better performance.

4.3.1 Histogram Equalization (HE)

Histogram equalization redistributes pixel intensities in such a way that the histogram of the image becomes more evenly spread out across the intensity range as depicted in **Fig. 4.3 (H. and Malisuwan (2014))**.. It brings out the details in areas which are too dark or too bright by spreading out the intensity values more uniformly across the histogram. Hence, it increases the global contrast of images. Histogram equalization cannot be applied separately to the red, green, and blue components of the image as it leads to dramatic changes in the image's colour balance. Since, CASIA dataset consists of grayscale images only, HE seemed to be a good choice.

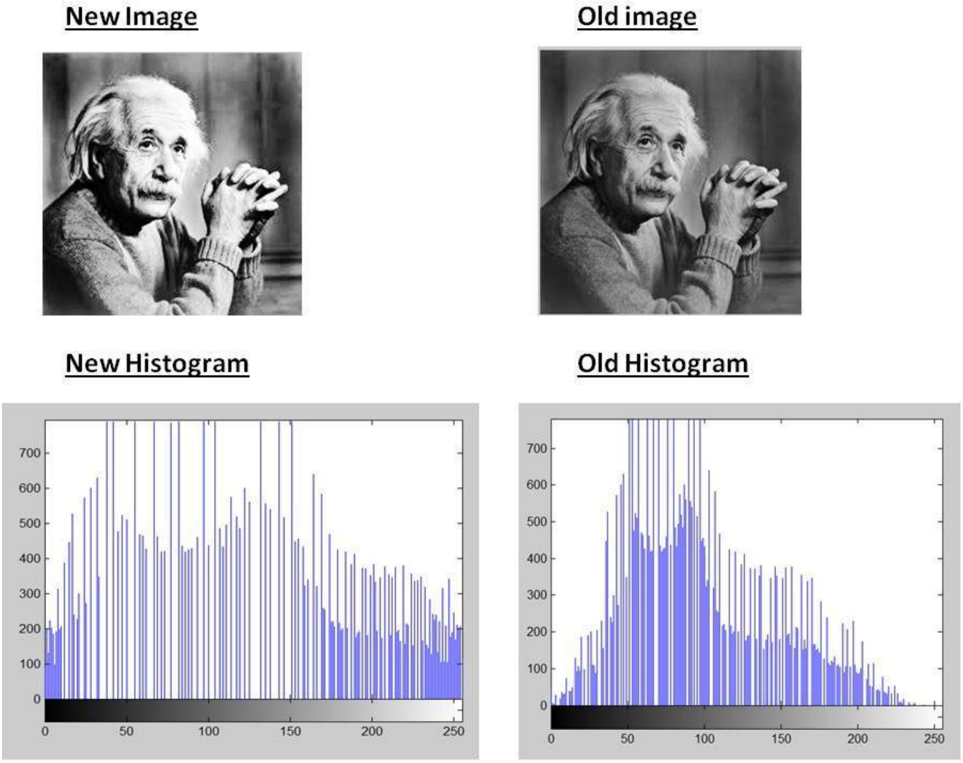


Fig. 4.3 Example of Histogram Equalization

Histogram Equalization is given by **Eqn.4.1**

$$S_k = T(r_k) = \sum_{k=0}^{k} \frac{n_k}{n} = \sum_{k=0}^{k} p_k(r_k) \tag{4.1}$$

where r_k is the kth gray level, n_k is the number of pixels in the image with that gray level, n number of pixels in that image, and k=0,1....L-1.

Features:

- Contrast Enhancement
- Brightness Correction
- Global Operation
- Non-linear Transformation

Algorithm:

1. *Compute Histogram:* Calculate the histogram of the input image. The histogram represents the frequency of each intensity level (0 to 255 for an 8-bit CASIA image) in the image.
2. *Compute Cumulative Distribution Function (CDF):* Compute the cumulative distribution function from the histogram. The CDF at intensity level *i* is given by **Eqn.4.2**

   $$CDF(i) = \sum_{j=0}^{i} Histogram(j) \tag{4.2}$$

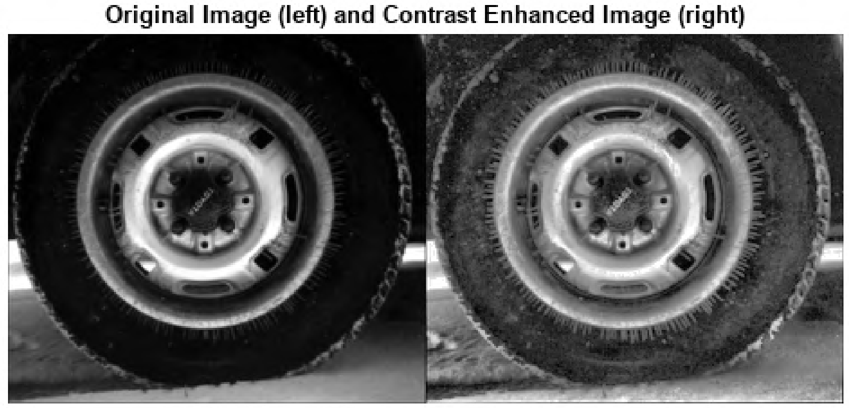
   where Histogram(j) is the value of the histogram at intensity level j.
3. *Normalize CDF:* Normalize the CDF values to the range [0, 255] by scaling them as in **Eqn.4.3** :

   $$CDF_{normalized}(i) = \frac{CDF(i) - min(CDF)}{max(CDF) - min(CDF)} * 255 \tag{4.3}$$

   where min(CDF) and max(CDF) are the minimum and maximum values of the CDF, respectively.
4. *Map Intensities:* For each pixel in the input image with intensity level i, replace the intensity with the corresponding normalized CDF value, $CDF_{normalized}(i)$.
5. *Output Equalized Image:* The resulting image after mapping intensities is the histogram-equalized image.

4.3.2 Contrast Limited Adaptive Histogram Equalization (CLAHE)

Adaptive Histogram Equalization differs from ordinary histogram equalization in the respect that the adaptive method computes several histograms, each corresponding to a distinct section of the image, and uses them to redistribute the lightness values of the image. **(H. and Malisuwan (2014))**.It is therefore suitable for improving the local contrast and enhancing the definitions of edges in each region of an image as shown in **Fig. 4.4**. CLAHE was developed to prevent the over-amplification of noise that adaptive histogram equalization can give rise to. Thus, CLAHE was chosen over AHE for our project.



Original Image (left) and Contrast Enhanced Image (right)

Fig. 4.4 Example of CLAHE

Features:

- Local Contrast Enhancement
- Adaptive Limiting
- Non-linear Transformation
- Reduced Noise Amplification
- Enhanced Image Details
- Computational Efficiency

Algorithm:

1. *Divide Image into Tiles:* Divide the input image into non-overlapping tiles of a specified size (e.g., 8x8 or 16x16 pixels).

2. *Apply Histogram Equalization to Each Tile:* Apply histogram equalization independently to each tile to enhance local contrast within that tile. This step creates several locally equalized versions of the image.
3. *Clip Histograms:* Clip the histograms of each tile to limit the contrast enhancement. This prevents over-amplification of noise and avoids contrast saturation.
4. *Combine Tiles:* Combine the locally equalized tiles to obtain the final CLAHE-enhanced image.

4.3.3 Unsharp Masking

Unsharp masking is used to improve the sharpness and clarity of an image by enhancing the edges and fine details.It also enhances contrast in areas of the image where there are transitions between light and dark regions, further improving the overall visual quality.

Features:

- Edge Enhancement
- High-pass Filtering
- Controlled Sharpening
- Enhancement of Textures

Algorithm:

1. Unsharp masking involves creating a blurred version of the original image (the ”unsharp” mask) using a Gaussian or other smoothing filter.
2. The unsharp mask is then subtracted from the original image to produce a high-pass filtered image containing edge information.
3. This high-pass filtered image is then added back to the original image, effectively boosting the edges and details while maintaining overall image structure and smoothness.

4.4 Iris Localization Using Mathematical Model

Iris localization refers to the process of identifying and extracting the iris region from an image or video frame. The iris is the colored part of the eye that surrounds the pupil. Iris localization is often a crucial step in iris recognition systems, where the unique patterns in the iris are used for biometric identification purposes.(**Jan et al. (2021)**)

4.4.1 Canny Edge Detection

Canny edge detection and the Hough transform are commonly used in combination for detecting circular shapes in images due to their complementary strengths and properties (**Song et al. (2017)**). The Canny edge detector is known for its ability to accurately localize edges in an image while minimizing noise. This ensures that only relevant edge pixels, which could potentially belong to a circular shape, are detected. It relies on the gradient of intensity changes in the image, making it effective for detecting boundaries between regions with different intensities, which often correspond to object edges. Canny Edge detection performed on Lena image is given in **Fig. 4.5**.

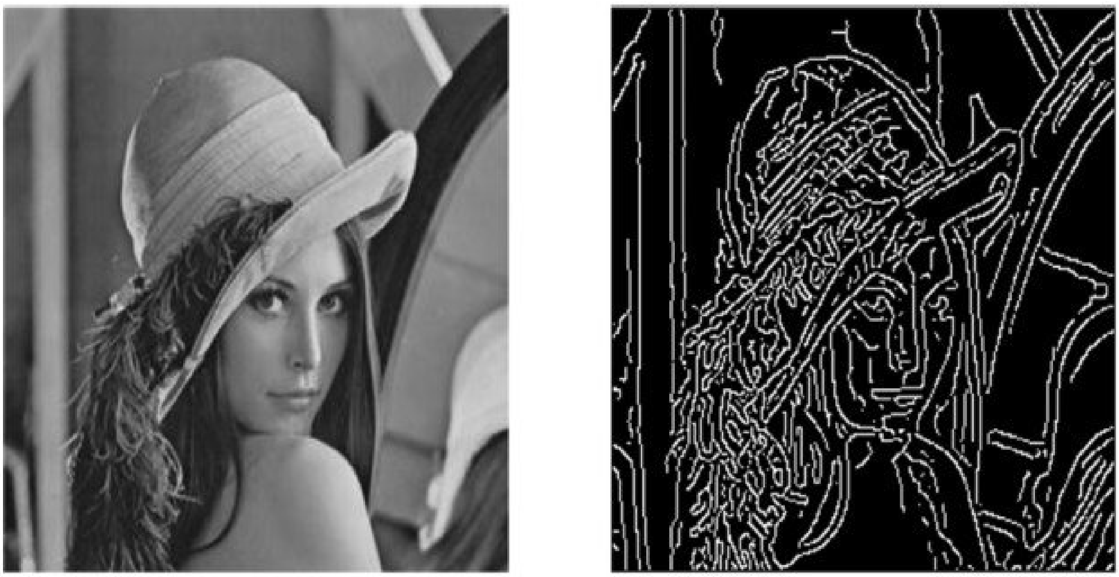


Fig. 4.5 Canny Edge Detection Performed on Lena Image

Features:

- High Edge Detection Accuracy
- Edge points are well localized
- Single edge point response
- Noise Reduction

Algorithm:

1. *Apply Gaussian Filter:* Smooth the input image with Gaussian filter given in **Eqn. 4.4**

$$GaussianBlurr(G_\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4.4}$$

2. *Find Gradient and Magnitude:* Compute the gradient magnitude(M) and angle images(θ) as given in **Eqn. 4.5** and **Eqn. 4.6**

$$EdgeGradient(M) = \sqrt{{G_x}^2 + {G_y}^2} \tag{4.5}$$

$$Angle(\theta) = tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{4.6}$$

3. *Non-maximum Suppression:* Apply nonmaximum suppression on the gradient magnitude (M) image. For each pixel in M:
    (a) Determine the gradient direction θ for the pixel
    (b) Compare the magnitude of the pixel's gradient with its neighbors in the direction perpendicular to θ.
    (c) Keep the pixel's magnitude if it is the local maximum; otherwise, set it to zero
4. Reduce false edge: double thresholding and connectivity analysis to detect and link edges
    (a) High-threshold $\Longrightarrow$ strong edge pixels $\Longrightarrow$ valid edge pixels
    (b) Low-threshold $\Longrightarrow$ weak edge pixels $\Longrightarrow$ valid only when connected to strong edge pixels

4.4.2 Hough Transform

The Hough transform accumulates evidence of edge points that could belong to a circular shape by mapping them to a parameter space as illustrated in **Fig. 4.6**, where circular shapes correspond to distinct peaks or curves.By combining Canny edge detection, which accurately identifies edge pixels, with the Hough transform, which robustly identifies circular shapes based on edge evidence, it becomes possible to detect circles in images effectively. This combination makes Canny edge detection and the Hough transform a powerful technique for circle detection in image processing applications **(Jan et al. (2021))**.

Features:

- Accurate Circle Localization
- Robustness to Noise
- Edge-Based

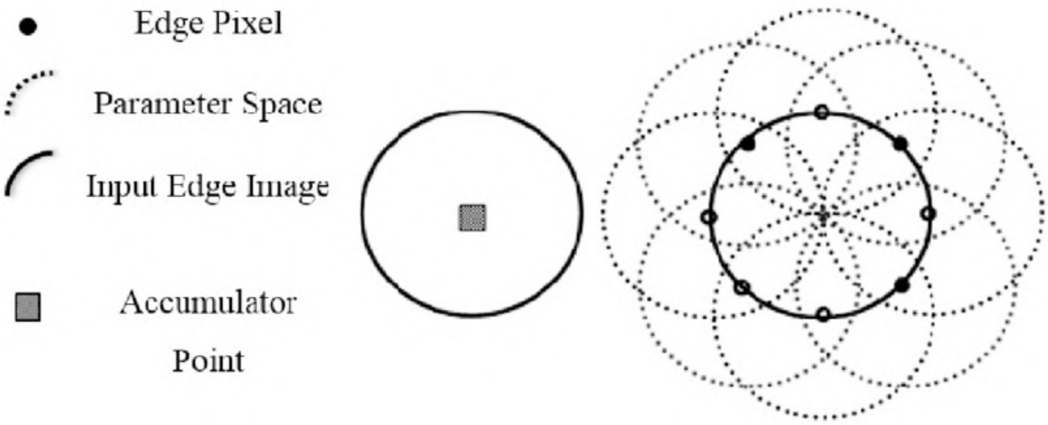


Fig. 4.6 Illustration of Hough Transform to detect Circles

Algorithm:

1. *Input:* Input image I, edge map E (obtained using an edge detection algorithm like Canny), and parameters such as the minimum line length, maximum gap between line segments, and threshold for line detection.
2. *Initialize Accumulator Array:* Create a three-dimensional accumulator array A to store votes for possible circle parameters (x_c, y_c, r) where (x_c, y_c) are the coordinates of the circle's center and r is the radius. The dimensions of the accumulator array depend on the range of possible circle parameters.
3. Voting: For each edge pixel (x, y) in the edge map E:
    (a) Iterate through possible circle parameters (x_c, y_c, r)
    (b) Calculate the corresponding circle equation **Eqn. 4.7**:

$$(x - x_c)^2 + (y - y_c)^2 = r \tag{4.7}$$

    (c) Vote in the accumulator array A for the calculated circle parameters
4. *Thresholding:* After all edge pixels are processed, apply a threshold to the accumulator array A to identify significant peaks (indicating potential circles).
5. *Post-processing:* For each significant peak in the accumulator array:
    (a) Extract the corresponding circle parameters (x_c, y_c, r)
    (b) Optionally, perform circle fitting or refinement to improve circle accuracy and continuity.
6. *Output:* Obtain the detected circles based on the extracted circle parameters and post-processed results

4.5 Iris Localization Using CNN

4.5.1 About Neural Networks:

Neural networks are computational systems inspired by the structure and function of the human brain's neural networks. They're made up of interconnected nodes, known as neurons, organized in layers. Each neuron processes input signals using an activation function and passes the result to the next layer's neurons.

These networks can learn and adjust based on input data through a process called training. During training, they fine-tune their parameters, like weights and biases, to improve their predictions by minimizing the error. This adjustment is typically done using optimization techniques such as gradient descent.

There are various types of neural networks, like feedforward, convolutional, and recurrent networks, as well as more specialized architectures such as GANs and transformers. Neural networks have wide-ranging applications across fields like image and speech recognition, natural language processing, autonomous vehicles, healthcare, finance, and many others.

4.5.2 CNN

A Convolutional Neural Network (CNN) is a Deep Learning algorithm that can analyze images, determine the significance of different features or objects within the image, and distinguish between them. Convolution, between two functions in mathematics, produces a third function, expressing how the shape of one function is modified by the other. An image is nothing but a matrix of pixel values. It contains a series of pixels arranged in a grid-like fashion that contains pixel values to denote how bright and what color each pixel should be.

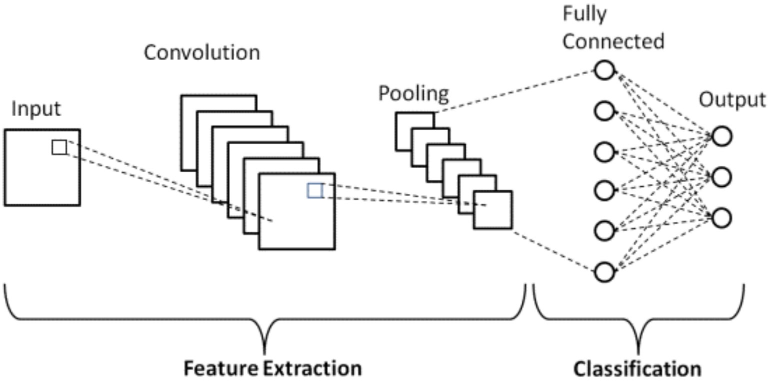


Fig. 4.7 CNN Basic Architecture

4.5.2.1 Convolutional Neural Network Architecture

A CNN has 4 layers as shown in **Fig. 4.7**:

- Convolution Layer: In this layer, a dot product operation is performed between two matrices: one matrix represents the learnable parameters, also called a kernel, while the other matrix represents a restricted area of the input, known as the receptive field. The kernel is smaller spatially compared to the input image but extends deeper. For instance, in an image with three (RGB) channels, the kernel's height and width are spatially small, but its depth covers all three channels.

- Pooling Layer: Pooling layers reduce the size of feature maps produced by convolutional layers through operations like max pooling or average pooling. These methods retain crucial information while decreasing the spatial dimensions of the feature maps. Pooling aids in creating a more compact representation that remains informative.

- Activation Functions: Activation functions introduce non-linearities into the network, allowing it to learn complex relationships in the data. Some common activation functions used in CNNs include:

  1. Sigmoid (**Fig. 4.8a**):

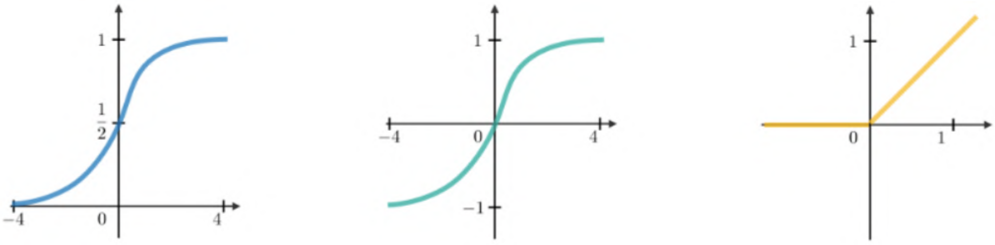
$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{4.8}$$

  2. tanh (**Fig. 4.8b**):

$$tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} = \frac{1-e^{-2x}}{1+e^{-2x}} \tag{4.9}$$

  3. ReLU (**Fig. 4.8c**):

$$Relu(z) = max(0, z) \tag{4.10}$$



(a) Sigmoid Function **(b)** tanh Function **(c)** ReLU Function

Fig. 4.8 Graphs of Activation Functions

- Fully Connected Layers: After several convolutional and pooling layers, the high-level reasoning in the neural network is done through fully connected layers. These layers take the features extracted by the convolutional layers and map them to the output classes through one or more dense layers.

Since no pre-trained model is available for detecting the iris in an image. Hence for better iris localization with more accuracy and to tackle the main issue behind this paper i.e. ”tackling reflection”, a CNN model is trained to localize the iris easily and work with the further steps of biometric authentication. The methodology is shown below in **Fig. 4.9**.

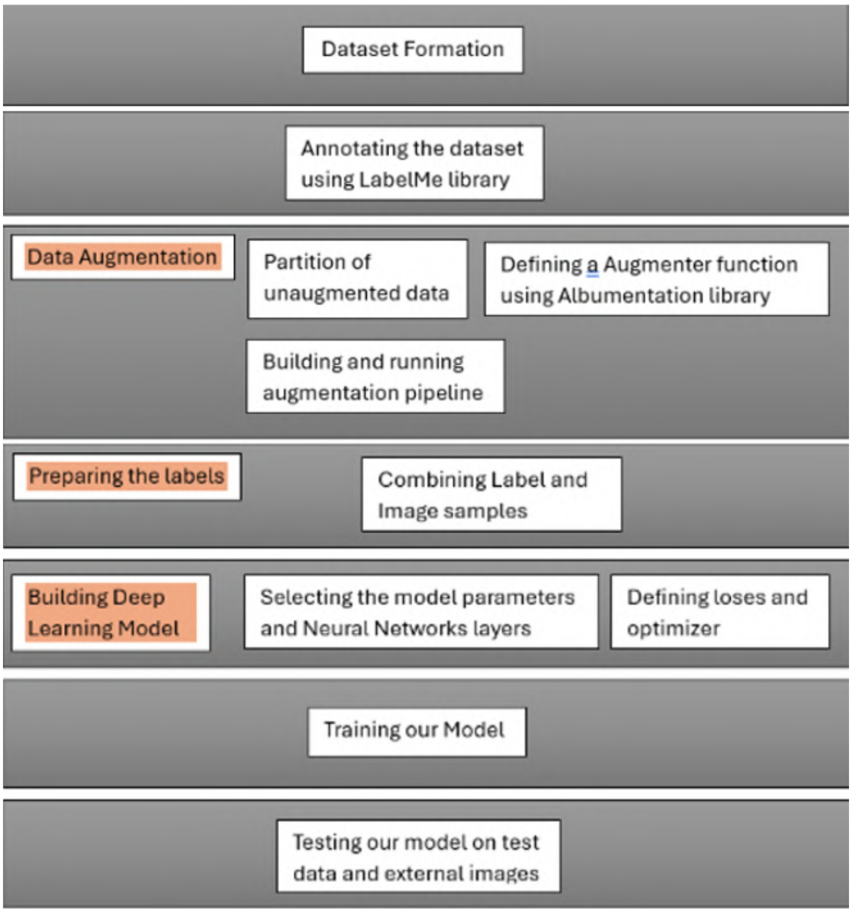

Dataset Formation
Annotating the dataset using LabelMe library
Data Augmentation
Partition of unaugmented data
Defining a Augmenter function using Albumentation library
Building and running augmentation pipeline
Preparing the labels
Combining Label and Image samples
Building Deep Learning Model
Selecting the model parameters and Neural Networks layers
Defining loses and optimizer
Training our Model
Testing our model on test data and external images

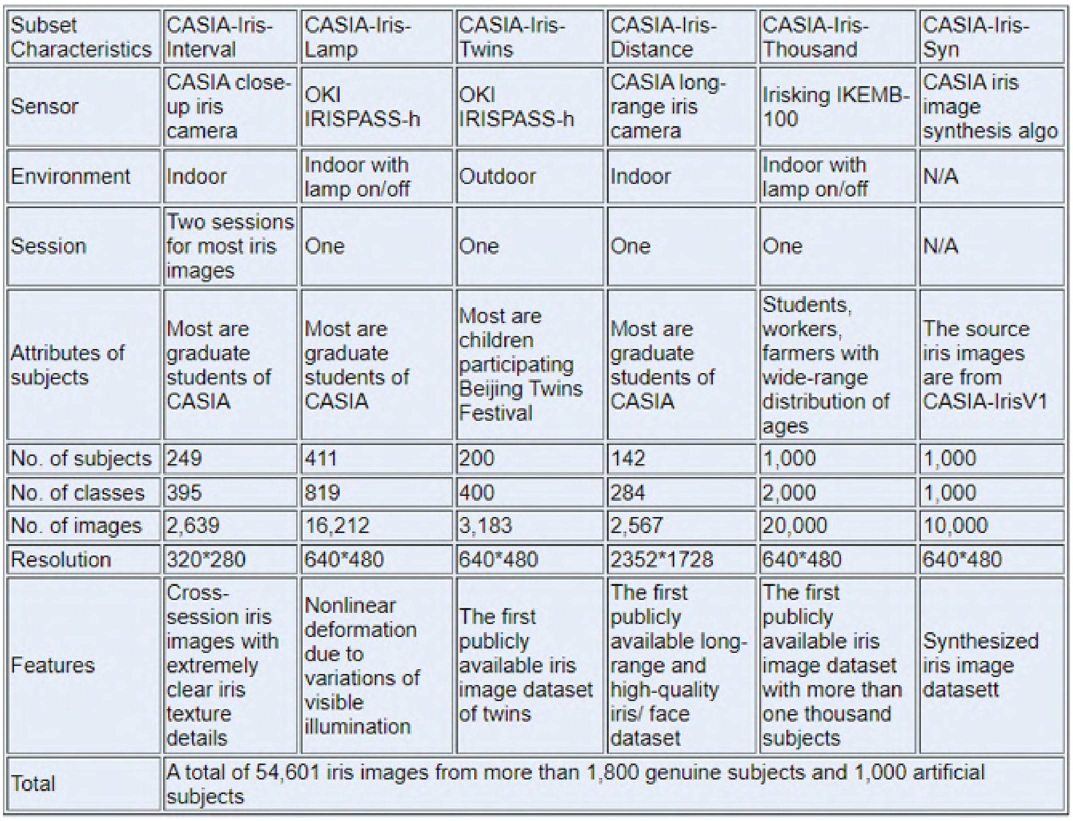
Fig. 4.9 CNN Methodology

For building a CNN Model involving iris detection, the process starts with dataset gathering. Since no pre-annotated dataset is available for our task, the Next step involved, annotating the dataset. Augmentation of data gives us much more variety and diversity among the data. Combining the labels with their images was the next step. After preprocessing the dataset and making it ready, the final step was Model training. Testing our trained model gave the overview and performance of our model.

4.5.3 Overview of CNN model training for Iris localization:

4.5.3.1 Dataset formation

For preparing the dataset, 100+ eye images were taken from different datasets of the CASIA Iris database. the important thing that was kept in mind while preparing the dataset was our novelty, i.e. to tackle specular reflections. For this images of persons, wearing spectacles with specular reflections were included in the dataset to train the model on those types of images also (**Othman et al. (2016)**).
The comparison of various datasets of the CASIA database is given in the image below in **Fig. 4.10** taken from their website.



Subset Characteristics	CASIA-Iris-Interval	CASIA-Iris-Lamp	CASIA-Iris-Twins	CASIA-Iris-Distance	CASIA-Iris-Thousand	CASIA-Iris-Syn
Sensor	CASIA close-up iris camera	OKI IRISPASS-h	OKI IRISPASS-h	CASIA long-range iris camera	Irisking IKEMB-100	CASIA iris image synthesis algo
Environment	Indoor	Indoor with lamp on/off	Outdoor	Indoor	Indoor with lamp on/off	N/A
Session	Two sessions for most iris images	One	One	One	One	N/A
Attributes of subjects	Most are graduate students of CASIA	Most are graduate students of CASIA	Most are children participating Beijing Twins Festival	Most are graduate students of CASIA	Students, workers, farmers with wide-range distribution of ages	The source iris images are from CASIA-IrisV1
No. of subjects	249	411	200	142	1,000	1,000
No. of classes	395	819	400	284	2,000	1,000
No. of images	2,639	16,212	3,183	2,567	20,000	10,000
Resolution	320*280	640*480	640*480	2352*1728	640*480	640*480
Features	Cross-session iris images with extremely clear iris texture details	Nonlinear deformation due to variations of visible illumination	The first publicly available iris image dataset of twins	The first publicly available long-range and high-quality iris/ face dataset	The first publicly available iris image dataset with more than one thousand subjects	Synthesized iris image datasett
Total	A total of 54,601 iris images from more than 1,800 genuine subjects and 1,000 artificial subjects					

Fig. 4.10 CASIA datasets comparison

4.5.3.2 Annotating the dataset

Since no pre-annotated dataset was available, so after forming our won dataset, the next step was to annotate the dataset. For annotation, the LabelMe library is used in Python (**Aljabri et al. (2022)**). LabelMe is an open-source graphical annotation tool for image and video data, that is publicly available on GitHub. It is written in Python, and it uses Qt for its graphical interface. LabelMe is extremely lightweight and easy to use. We can use it to create annotations for object detection, semantic segmentation, and panoptic

segmentation for both images and videos.
The interface and how it is used for iris annotation is shown below in **Fig. 4.11**

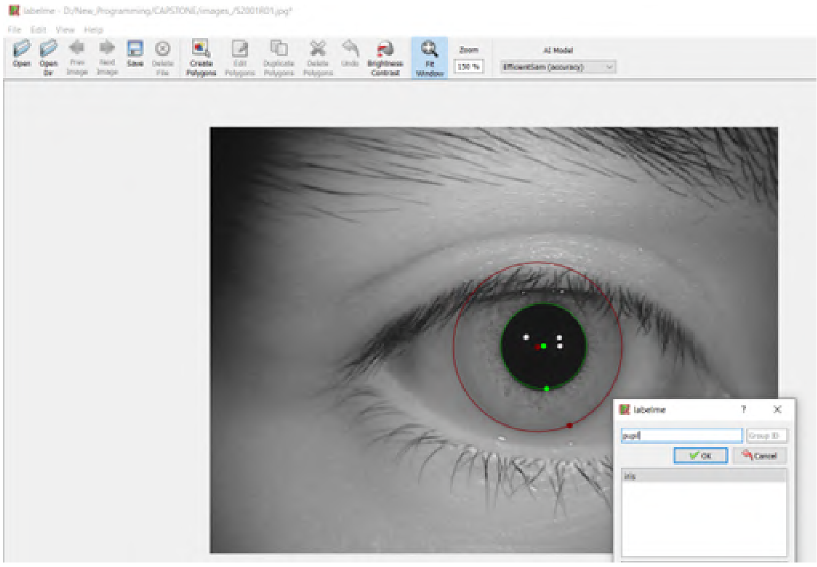
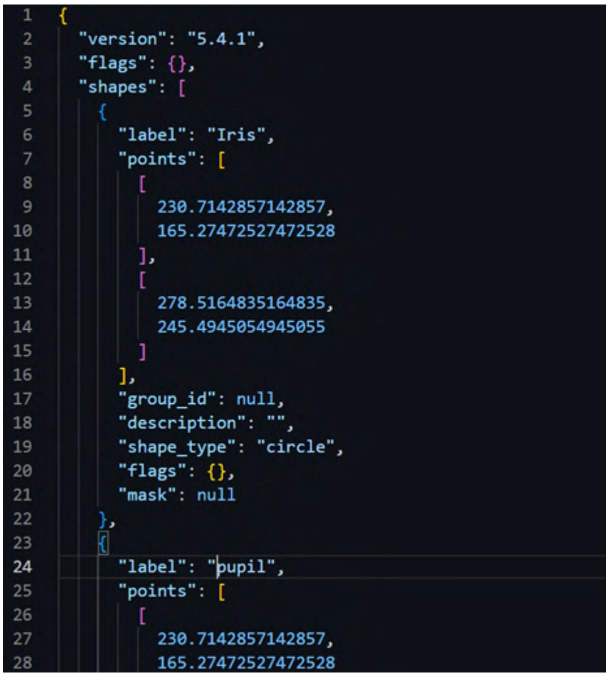


Fig. 4.11 Labelme

The output of annotation is saved as a JSON file as shown below in **Fig. 4.12**



```
{
  "version": "5.4.1",
  "flags": {},
  "shapes": [
    {
      "label": "Iris",
      "points": [
        [
          230.7142857142857,
          165.27472527472528
        ],
        [
          278.5164835164835,
          245.4945054945055
        ]
      ],
      "group_id": null,
      "description": "",
      "shape_type": "circle",
      "flags": {},
      "mask": null
    },
    {
      "label": "pupil",
      "points": [
        [
          230.7142857142857,
          165.27472527472528
```

Fig. 4.12 Annotated output file

As we can see, the iris and pupil are marked differently with two points, and their coordinates are shown separately under both labels. One coordinate is the center and the other is a point on the circumference of the circle marked.

4.5.3.3 Data Augmentation

Next, an augmentation pipeline is made for the dataset. Data augmentation is a collection of methods utilized in machine learning and data preprocessing to expand the dataset size by generating altered or artificial versions of the original data. These adjustments may involve transformations like rotation, flipping, resizing, cropping, introducing noise, or altering the color spectrum. The objective of data augmentation is to increase the diversity and variability within the training data, ultimately enhancing the resilience and generalization capabilities of machine learning models.
The augmentation methods used by us here are listed below:

1. alb.RandomCrop()
2. alb.HorizontalFlip()
3. alb.RandomBrightnessContrast()
4. alb.RandomGamma()
5. alb.RGBShift()
6. alb.VerticalFlip()

This augmentation pipeline is followed by the calculation of the radius for marking a circle around the iris. For radius calculation, the formula used is:

$$(x-h)^2 + (y-k)^2 = r^2 \tag{4.11}$$

The output of the above steps is shown in an example figure in **Fig. 4.13**

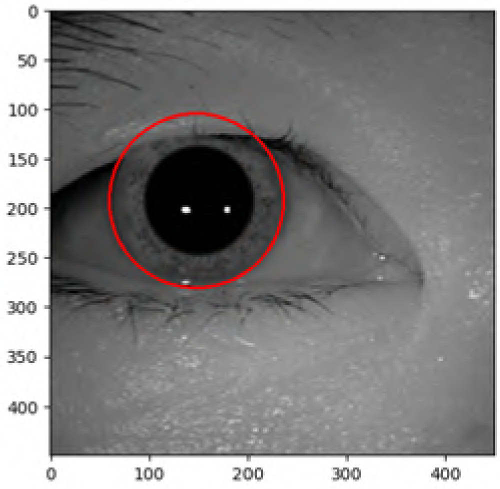


Fig. 4.13 Iris located example image used to train the model

4.5.3.4 Combining the labels with their respective images

After the image augmentation, the images are combined with their respective labels. After running our dataset through the Augmentation pipeline, our final dataset was of the size:
Training data images and labels: 3840
Test data images and labels: 360
Validation data images and labels: 240

4.5.3.5 Training our Model

VGG16: It is a convolutional neural network (CNN) architecture developed by the Visual Geometry Group (VGG), belongs to the broader VGG family, which encompasses variations like VGG19 and VGG13 **(Sitaula and Hossain (2021)**).

- **Architecture of VGG16:** Specifically, VGG16 comprises 16 layers: 13 convolutional layers followed by 3 fully connected layers and a softmax layer for classification. It employs 3x3 filters with a stride of 1 and padding for spatial resolution preservation in convolutional layers. Max-pooling layers with 2x2 filters and a stride of 2 are used for downsampling. The architecture diagram is given below in figure **Fig. 4.14**. The 16 in VGG16 refers to 16 layers that have weights.

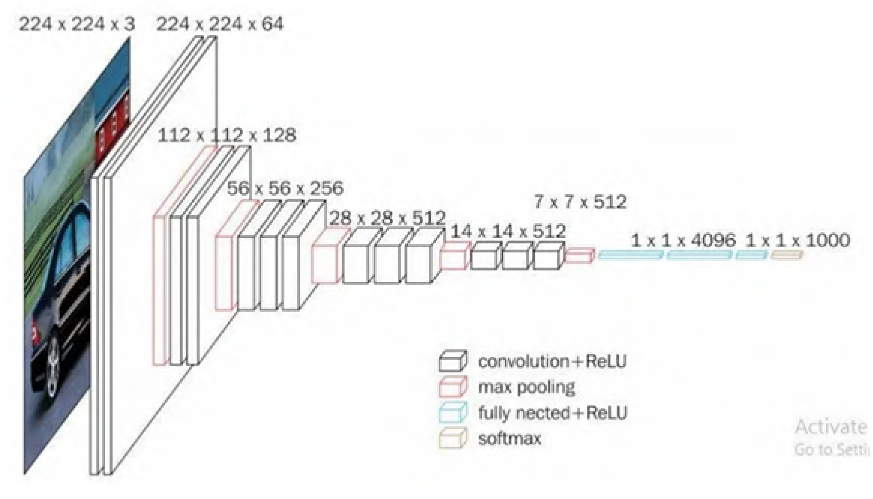
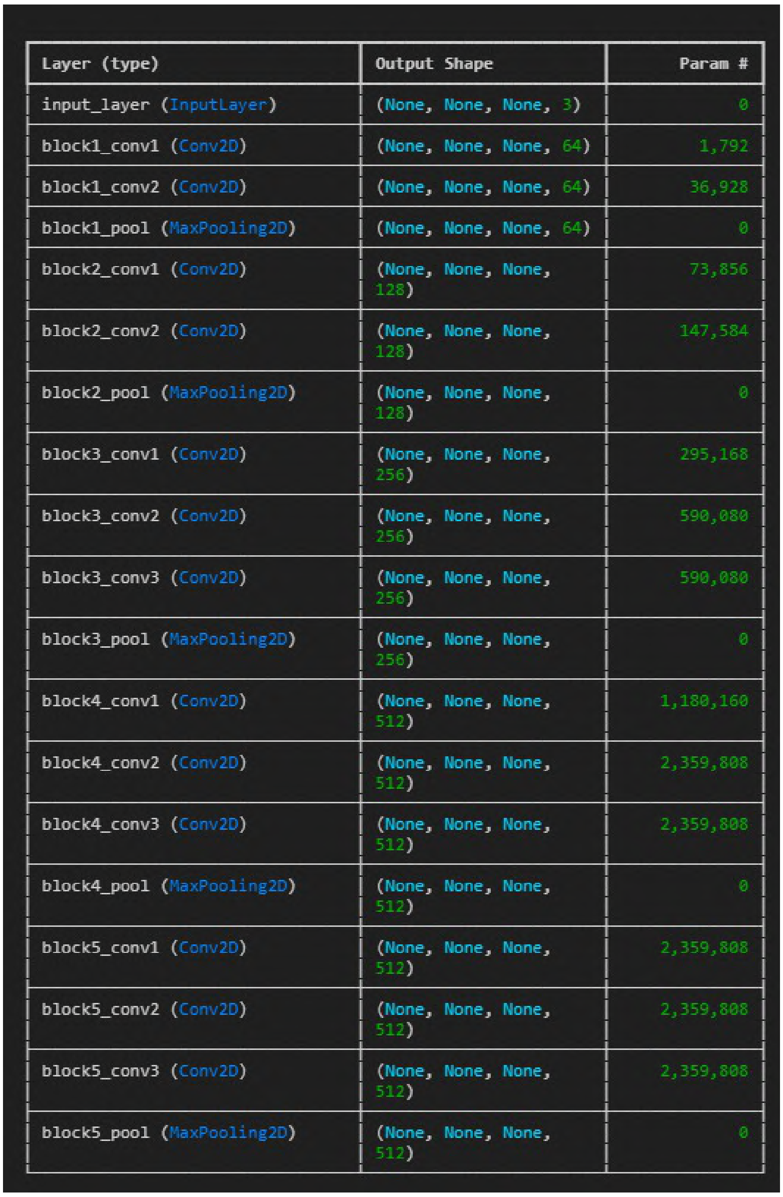


Fig. 4.14 VGG16 Architecture

In VGG16 there are thirteen convolutional layers, 5 Max Pooling layers, and 3 Dense layers which sum up to 21 layers but it has only 16 weight layers i.e., the learnable parameters layer.

- **Pre-Trained Models:** The Pre-trained versions of VGG16 were trained on vast image datasets like ImageNet. These pre-trained models have gleaned generic features from a wide array of images and can be fine-tuned for specific tasks using smaller datasets.
- **Performance:** VGG16 demonstrates competitive performance on benchmark datasets like ImageNet in tasks such as image classification.

The VGG16 summary of parameters is given below in the figure **Fig. 4.15**



```
Layer (type)                     Output Shape                    Param #
input_layer (InputLayer)         (None, None, None, 3)                 0
block1_conv1 (Conv2D)            (None, None, None, 64)            1,792
block1_conv2 (Conv2D)            (None, None, None, 64)           36,928
block1_pool (MaxPooling2D)       (None, None, None, 64)                0
block2_conv1 (Conv2D)            (None, None, None,               73,856
                                 128)
block2_conv2 (Conv2D)            (None, None, None,              147,584
                                 128)
block2_pool (MaxPooling2D)       (None, None, None,                    0
                                 128)
block3_conv1 (Conv2D)            (None, None, None,              295,168
                                 256)
block3_conv2 (Conv2D)            (None, None, None,              590,080
                                 256)
block3_conv3 (Conv2D)            (None, None, None,              590,080
                                 256)
block3_pool (MaxPooling2D)       (None, None, None,                    0
                                 256)
block4_conv1 (Conv2D)            (None, None, None,            1,180,160
                                 512)
block4_conv2 (Conv2D)            (None, None, None,            2,359,808
                                 512)
block4_conv3 (Conv2D)            (None, None, None,            2,359,808
                                 512)
block4_pool (MaxPooling2D)       (None, None, None,                    0
                                 512)
block5_conv1 (Conv2D)            (None, None, None,            2,359,808
                                 512)
block5_conv2 (Conv2D)            (None, None, None,            2,359,808
                                 512)
block5_conv3 (Conv2D)            (None, None, None,            2,359,808
                                 512)
block5_pool (MaxPooling2D)       (None, None, None,                    0
                                 512)
```

Fig. 4.15 VGG16 parameter summary

Optimizer:

Optimizers and their role in the Neural network model: Optimizers are algorithms or methods used to minimize the error or loss function during the training of a model. The optimizers adjust the parameters of the model to minimize the error between the predicted outputs and the actual targets.

Optimizer used: *Adam Optimizer*

The Adam optimizer(Adaptive Moment Estimation), is an advancement of stochastic gradient descent (SGD) applied in the training of deep learning models. It determines customized learning rates for individual parameters by taking into account both the first and second moments of the gradients.

Benefits of Adam Optimizer:

- Adaptive learning rates for each parameter.
- Momentum to accelerate convergence.
- Robustness to noisy gradients and sparse gradients.

Loss function:

In deep learning, a loss function, interchangeably referred to as a cost or objective function, gauges the disparity between the model's predicted output and the actual target output.

Loss Function used: *Binary Cross Entropy*

Binary cross-entropy serves as a prevalent loss function for binary classification tasks. It evaluates the dissimilarity between two probability distributions: those produced by the model's predicted probabilities and the actual binary labels of the dataset.

Binary Cross-Entropy(BCE) can be calculated as:

$$-(y\log(p) + (1-y)\log(1-p)) \tag{4.12}$$

If $M > 2$ (i.e. multiclass classification), we calculate a separate loss for each class label per observation and sum the result:

$$-\sum_{c=1}^{M} y_{o,c}\log(p_{o,c}) \tag{4.13}$$

Benefits of Binary Cross Entropy:

- Handles class imbalance.
- Encourages probabilistic outputs.
- Ease of optimization.

4.6 Iris Segmentation

Localizing the iris and the pupil gives the center coordinates and their radius respectively. Using them, the boundary coordinates of the iris and pupil are obtained. The inpolygon function in MATLAB is used to create a binary mask that masks the pixels inside the specified polygon i.e. the region between the pupil and the iris.

4.7 Iris Normalization

Daughman's Rubber Sheet Model shown in **Fig. 4.16** **Jusman et al. (2020)** is a mathematical model used in iris recognition systems to represent the normalized iris texture for feature extraction and comparison. It is a concept in iris recognition that treats the iris texture as if it were mapped onto a flexible surface, where unique patterns like crypts and furrows become deformations. This model extracts iris features for biometric identification by analyzing these deformations, contributing to accurate and reliable recognition systems.It was proposed by John Daugman, a prominent researcher in the field of biometrics and iris recognition. It compensates for the differences in the scale of the iris and variations in pupil size.

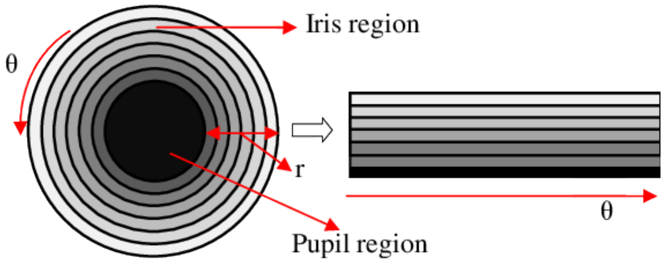


Fig. 4.16 Daughman's Rubber Sheet Model

Features:

- Polar Coordinate Representation
- Rubber Sheet Deformation
- Invariant Features

Algorithm:

1. The polar coordinates are converted to cartesian coordinates by the **Eqn. 4.14**:

$$x = r\cos(\theta) \quad y = r\sin(\theta) \tag{4.14}$$

4.8 Feature Extraction

An iris is recognized by its texture, so we have chosen methods that can extract discriminative features from the normalized iris based on its texture pattern like Discrete Wavelet Transform (DWT) Local Binary Pattern (LBP), and Gabor Filter. Principle Component Analysis (PCA) is basically used to reduce the dimension to make the process of iris recognition faster.

4.8.1 Discrete Wavelet Transform + Principle Component Analysis

DWT allows the decomposition of signals into different frequency bands at varying resolutions as shown in **Fig. 4.17**. This multi-resolution property enables capturing both high and low-frequency components of a signal simultaneously. It has inherent noise reduction properties due to the energy compaction feature. By concentrating signal energy into fewer coefficients, it can suppress noise and enhance the visibility of signal features, improving the robustness of feature extraction algorithms. This makes it suitable for Feature extraction.

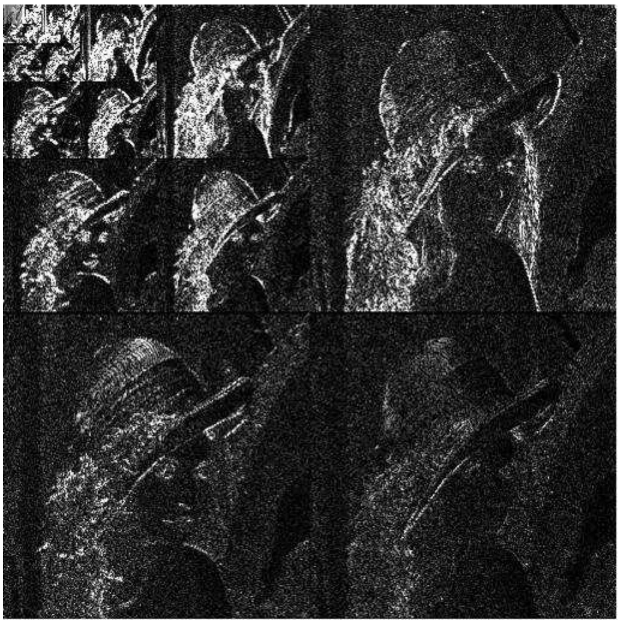


Fig. 4.17 Lena Image Decomposition using DWT

DWT is described by the **Eqn. 4.15** and **Eqn. 4.16**:

$$W_\phi(j_0, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)\phi_{j_0,m,n} \tag{4.15}$$

$$W_\psi^i(j_0, m, n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)\phi_{j,m,n}^i \tag{4.16}$$

where $i=\{H,V,D\}$
PCA transforms high-dimensional data into a lower-dimensional representation by projecting it onto a new subspace of orthogonal axes called principal components. This reduction in dimensionality can help in simplifying the data representation, visualization, and computation. By selecting a subset of the principal components that explain the majority of the variance in the data, PCA can achieve significant data compression without losing critical information **(Mattar (2013))**.. It is obtained by **Eqn. 4.17** followed by **Eqn. 4.18**:

$$z = \frac{x-\mu}{\sigma} \tag{4.17}$$

$$cov(X,Y) = \frac{\sum_{i=1}^{n} \left(X_i - X'\right)\left(Y_i - Y'\right)}{n-1} \tag{4.18}$$

Then eigen values and eigen vectors are obtained and the eigen vectors with the highest values are chosen.

Features:

- Multiresolution Analysis
- Localization in Time and Frequency
- Signal Denoising
- Compression

4.8.2 Local Binary Pattern

LBP operates on small neighborhoods of pixels within an image, making it suitable for capturing local texture information. It computes a binary pattern for each pixel by comparing its intensity value with that of its neighbors. It is inherently robust to changes in illumination. This makes it a suitable method of extracting features from normalized iris. LBP is given by the **Eqn. 4.19** and **Eqn. 4.20**:

$$LBP = \sum_{n=0}^{7} s\left(i_n - i_c\right) 2^n \tag{4.19}$$

where i_c is the center pixel value and i_n is the neighbouring pixel values.

$$\text{s}(z) = \begin{cases} 1, z >= 0 \\ 0, z < 0 \end{cases} \tag{4.20}$$

Features:

- Local Texture Descriptor
- Robustness to Illumination Changes:
- Binary Representation
- Rotation and Scale Invariance:
- Computational Efficiency

4.8.3 Gabor Filter

Gabor filters given in **Fig.4.18** are capable of analyzing images at multiple scales and orientations simultaneously. They provide a multiresolution representation of texture features, enabling the extraction of both fine and coarse texture details. They are sensitive to orientation, have spatial localization properties, and can detect both high-frequency details, such as fine textures, and low-frequency components, such as smooth regions or large-scale patterns. Features extracted from Gabor-filtered images can be further processed and analyzed using techniques such as histogramming, feature pooling, or dimensionality reduction.

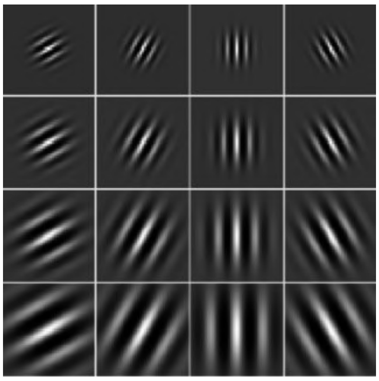


Fig. 4.18 Example of a Gabor Filter Bank

Gabor filter is represented by the **Eqn. 4.21**:

$$g(x,y;\lambda,\theta,\psi,\sigma,\gamma) = exp\left(-\frac{x'^2+\gamma^2y'^2}{2\sigma^2}\right)exp\left(i\left(2\pi\frac{x'}{\lambda}+\psi\right)\right) \tag{4.21}$$

where λ is the wavelength of the sine component, θ is the orientation of the filter, ψ is the phase offset, σ is the std. deviation of the gaussian envelope, γ is the spatial aspect ratio and $x' = x\cos(\theta) + y\sin(\theta)$; $y' = -x\sin(\theta) + y\cos(\theta)$

Features:

- Frequency and Orientation Selectivity
- Spatial Localization
- Multi-Resolution Analysis
- Filter Bank Design

4.9 Feature Matching

The Hamming distance is a metric used in iris recognition to quantify the similarity or dissimilarity between two iris templates. Specifically, it counts the number of differing bits between the templates, providing a measure of their similarity. This distance calculation is particularly valuable for feature vectors of fixed length, such as those commonly used in biometric recognition systems. By analyzing the Hamming distance, one can assess the degree of resemblance or variation between iris templates efficiently (**Farouk et al. (2022)**).
The equation for HD is given by **Eqn. 4.22**:

$$D_H = \sum_{i=1}^{k} |x_i - y_i| \tag{4.22}$$

$x = y \implies D = 0; x \neq y \implies D = 1$

4.10 Tackling Specular Reflections

An endeavor is undertaken to combat specular reflections on glass surfaces that hinder iris recognition accuracy. Utilizing software tools, a binary mask is created through thresholding a grayscale image containing these reflections. Pixels with intensities surpassing the threshold are designated as 1 (white), while others are set as 0 (black).

Morphological dilation, expanding foreground pixel areas in a binary image, is then applied using a disk-shaped structuring element to the mask.(**Jan et al. (2021)**) Subsequently, an inpainting algorithm fills the regions identified by the binary mask with appropriate pixel values derived from their surrounding context. This comprehensive process aims to mitigate the effects of specular reflections, thus enhancing the precision and reliability of iris recognition systems.

CHAPTER 5

SCHEDULE, TASKS AND MILESTONES

5.1 Schedule

The schedule of the project is given in **Fig. 5.1**

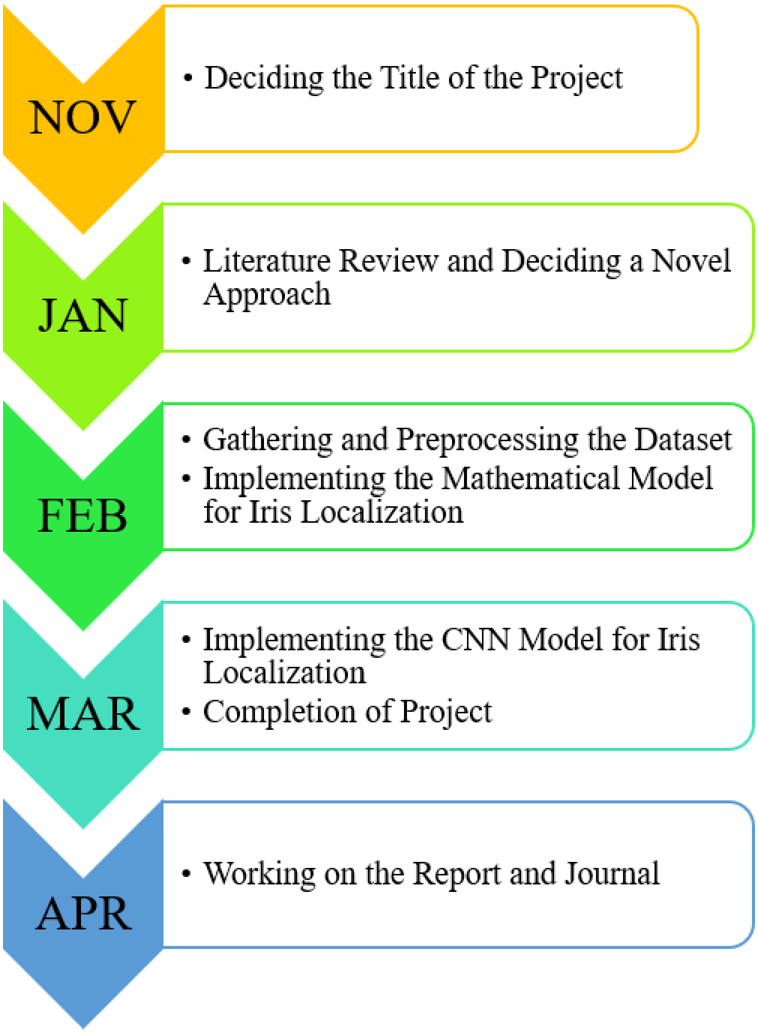

NOV
• Deciding the Title of the Project
JAN
• Literature Review and Deciding a Novel Approach
FEB
• Gathering and Preprocessing the Dataset
• Implementing the Mathematical Model for Iris Localization
MAR
• Implementing the CNN Model for Iris Localization
• Completion of Project
APR
• Working on the Report and Journal

Fig. 5.1 Schedule of the Project

The title of the project was finalized by November end after a confirmation from our Guide.

The Subsequent months of December and January were dedicated to an exhaustive Literature Survey, aiming to gain a profound understanding of iris recognition within the realm of image processing, a relatively new domain for our team. As we delved deeper, we carefully considered existing work in the field, striving to identify avenues for introducing novelty.

During this period, we also focused on acquiring the necessary dataset and formulating strategies for dataset preprocessing. By the end of February, significant progress was made as we successfully localized the iris using mathematical methodologies. Concurrently, we initiated exploration into the utilization of Convolutional Neural Networks (CNN) for iris localization, recognizing its potential for enhancing accuracy and efficiency.

March marked a pivotal phase of the project, witnessing the culmination of our efforts. We achieved the milestone of completing the project, which included the localization of iris using the CNN Model, along with subsequent processes such as segmentation, normalization, feature extraction, and matching with the database features.

5.2 Tasks and Milestones

Tasks and milestones play complementary roles in project management, ensuring progress alignment with plans, goal achievement, and overall project trajectory. Milestones serve as prominent indicators of progress, highlighting key achievements and phases within the project timeline. On the other hand, tasks represent actionable steps that collectively contribute to reaching these milestones. Efficient tracking of tasks and milestones is essential for evaluating project performance, identifying potential bottlenecks or delays, and making necessary adjustments to maintain project momentum.

By integrating task management with milestone tracking, teams can effectively monitor progress, manage resources, and ensure timely delivery of project objectives, ultimately leading to successful project completion.

The Tasks and Milestones are tabulated in **Table 5.1**

Table 5.1 Tasks and Milestones for the Project

Review	Tasks	Milestones Achieved
Review 1	Objectives, Methodologies, Literature Review and Schedule of the project were defined	Finalising the problem statement, completed Literature Review survey and started working on Iris Localization using Mathematical Model
Review 2	Methodology, Results and Discussions were defined	Completed the Mathematical and CNN Approach for Iris Localization, as well as finished the Project, Addressed the problem of Specular Reflection
Review 3	Preparing the report and journal based on the final results	Final documentation of the functioning model and report submission

5.2.1 First Review

The aim of the first review was to lay down the title of the project and the objective. The literature review was done on the project to get an idea about the iris recognition system. We were able to finalize the problem statement and complete the Literature Review survey. Since, we were able to do all this well before our decided date, we even started working on Iris Localization using Mathematical Model.

5.2.2 Second Review

Methodology of the project was explained in detail. It was followed by the results obtained and the discussions on those obtained results. The panel coordinator asked few queries which were answered by our team. By the second review, we were able to complete the Mathematical and CNN Approach for Iris Localization, as well as we were able to address the problem of Specular Reflection occuring on the glass surfaces.

5.2.3 Third Review

The project's progress will be showcased to the external faculty-in-charge for validation. Simultaneously, the completion and approval of the project report and journal will signify the successful conclusion of all tasks and deliverables.

CHAPTER 6

PROJECT DEMONSTRATION

This section demonstrates the various steps carried out on a sample image **Fig.6.1** from the dataset.

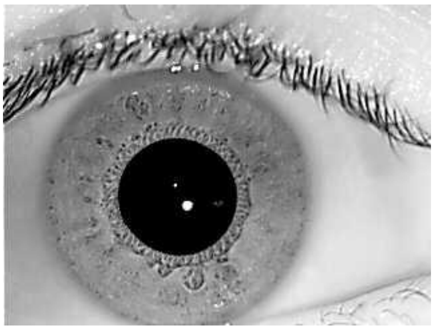
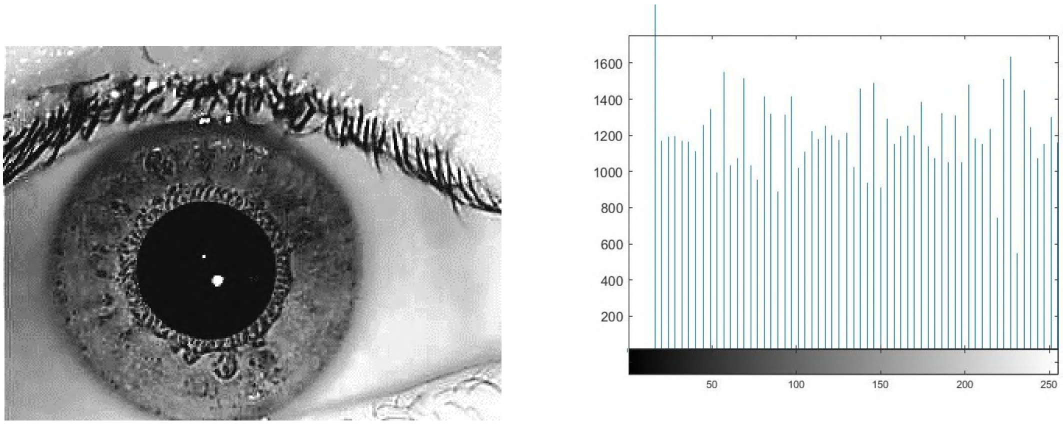


Fig. 6.1 Original Image

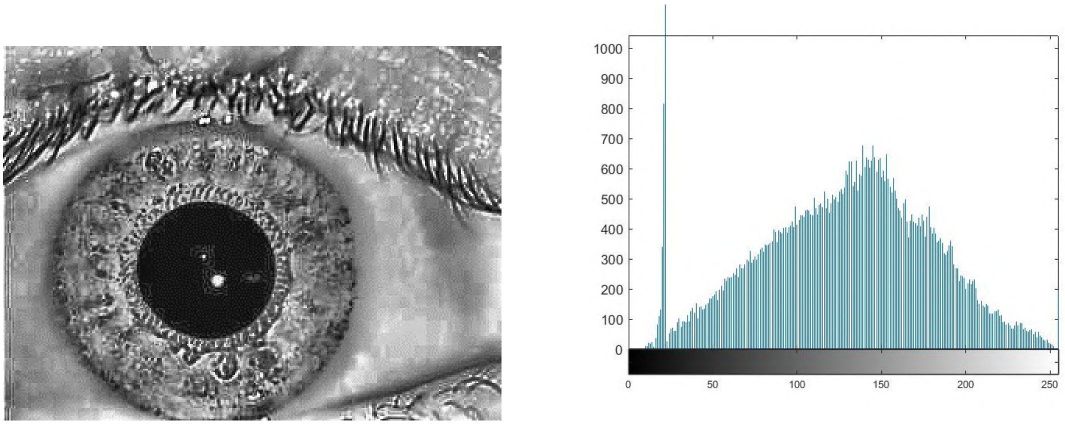
6.1 Pre-Processing

The Pre-processed images along with their histograms are shown in **Fig.6.2**, **Fig.6.3**, **Fig.6.4** and **Fig.6.5**.



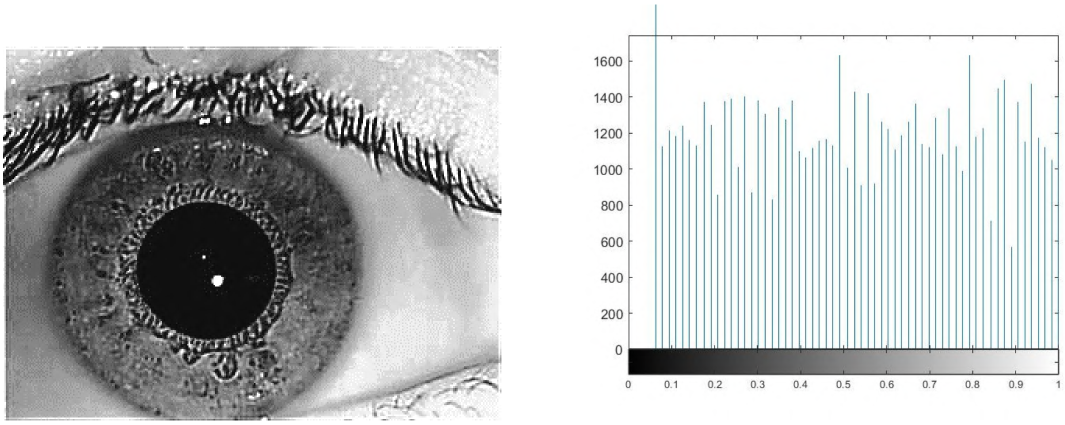
(a) Preprocessed Image **(b)** Histogram

Fig. 6.2 Histogram Equalized Image



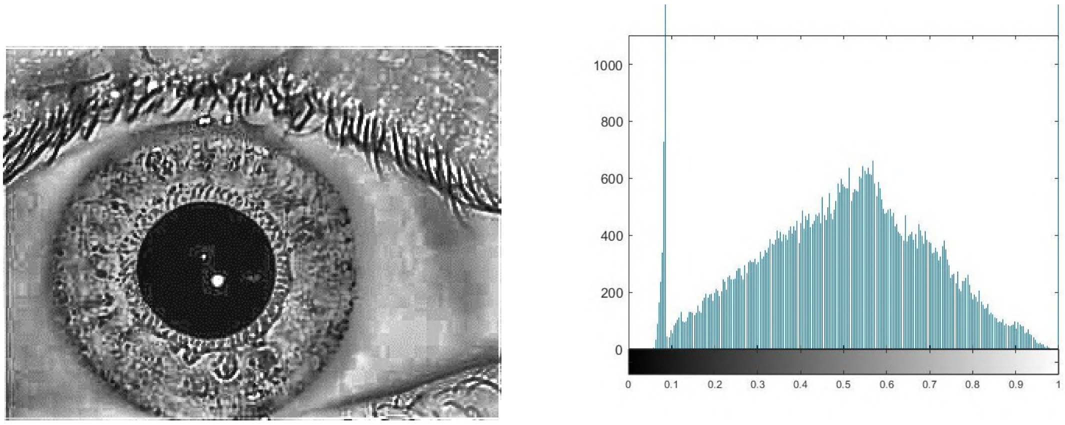
(a) Preprocessed Image **(b)** Histogram

Fig. 6.3 Contrast Limited Adaptive Histogram Equalized Image



(a) Preprocessed Image **(b)** Histogram

Fig. 6.4 Histogram Equalized Image with Unsharp Masking



(a) Preprocessed Image **(b)** Histogram

Fig. 6.5 Contrast Limited Adaptive Histogram Equalized Image with Unsharp Masking

6.2 Iris Localization using Mathematical Model

After preprocessing the Iris is localized using the built-in imfindcircles (**Fig.6.6**) and user-defined functions findcircles (**Fig.6.7**).

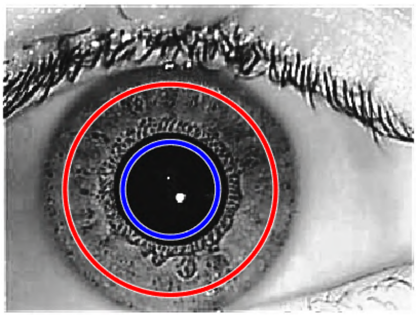
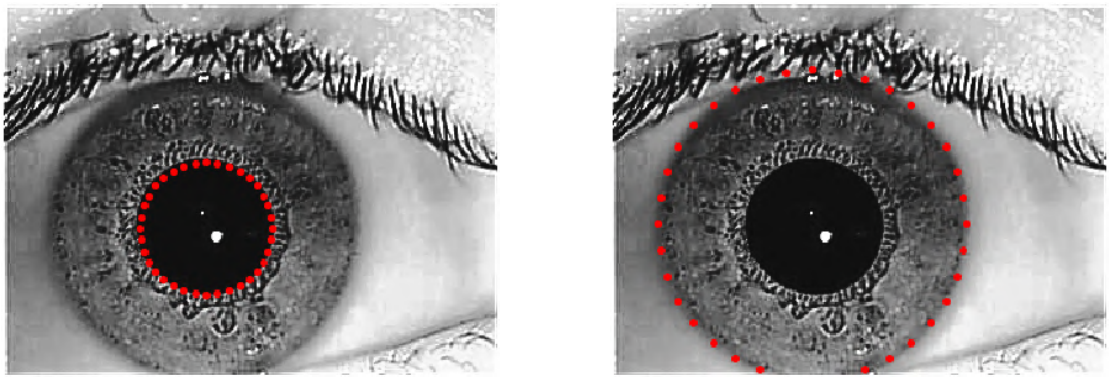


Fig. 6.6 Localization Using Built-In imfindcircles function



(a) Localized Pupil

(b) Localized Iris

Fig. 6.7 Localization Using User-Defined imfindcircles function

6.3 Iris Localization using VGG16 Model

The confusion matrix of our trained model is shown in the below figure (**Fig.6.8**). Each cell in the confusion matrix represents the number of correct identifications (true positives) and misidentifications (false positives) for each individual, assisting in evaluating the accuracy and performance of the iris recognition system for human authentication. Our model was trained for two different sets of Epochs: 10 and 15. A comparison of losses is shown in the below figure graphically for 10 epochs (**Fig.6.9**) and 15 epochs (**Fig.6.10**).

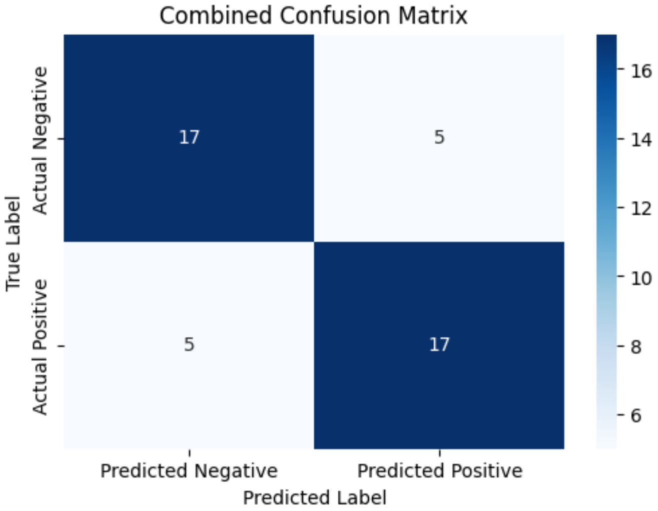


Fig. 6.8 Confusion Matrix

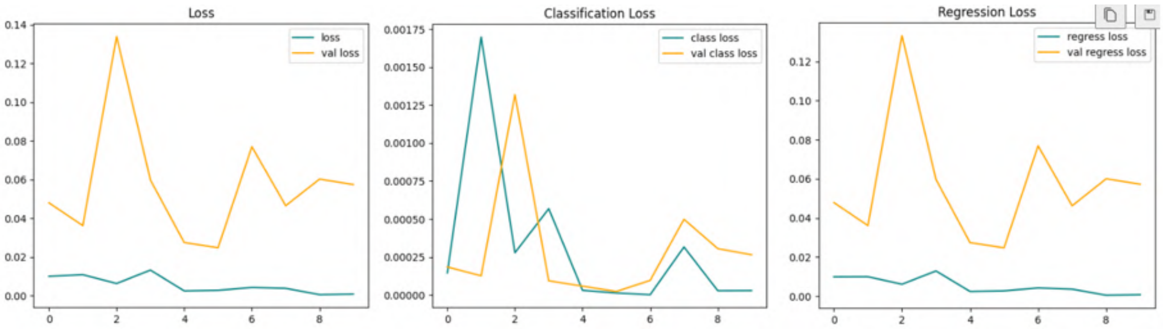


Fig. 6.9 Losses for 10 epochs

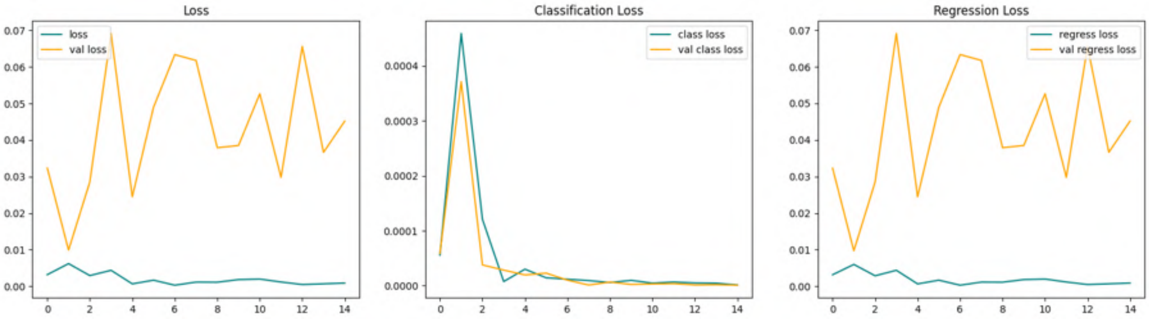


Fig. 6.10 Losses for 15 epochs

The final result of the localized iris by our self-trained model is shown in (**Fig.6.11**) and (**Fig.6.12**):

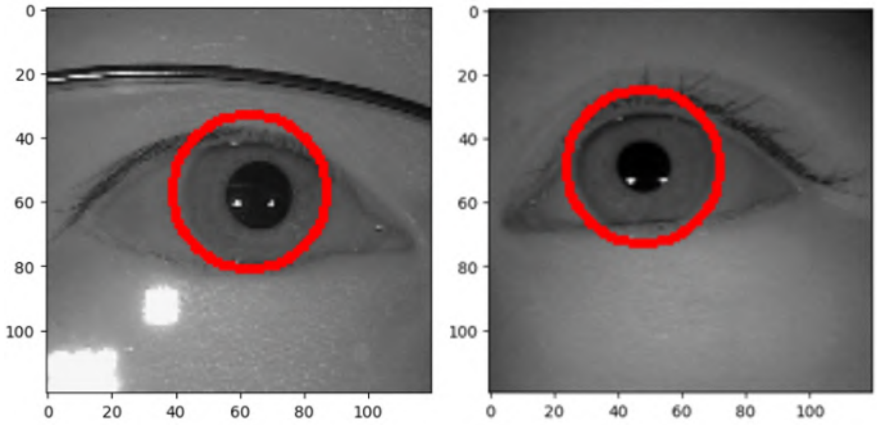


Fig. 6.11 On Test data

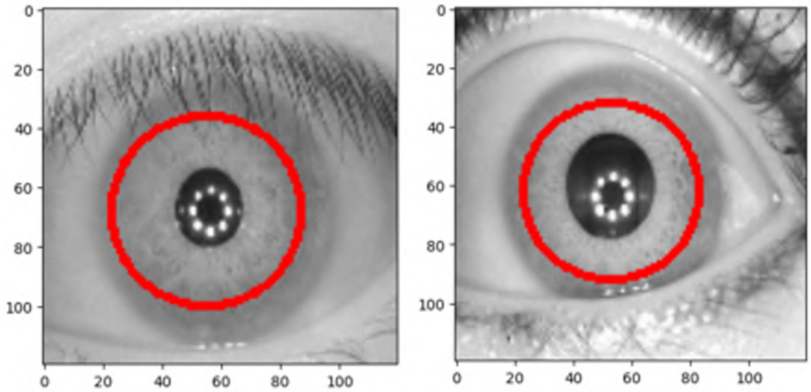


Fig. 6.12 On External image

6.4 Iris Segmentation

Fig.6.13 shows the segmented iris.

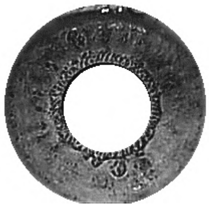


Fig. 6.13 Segmented Iris

6.5 Iris Normalization

This is followed by the normalization of the iris as shown in **Fig.6.14**.

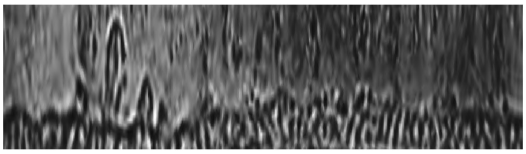


Fig. 6.14 Normalized Iris

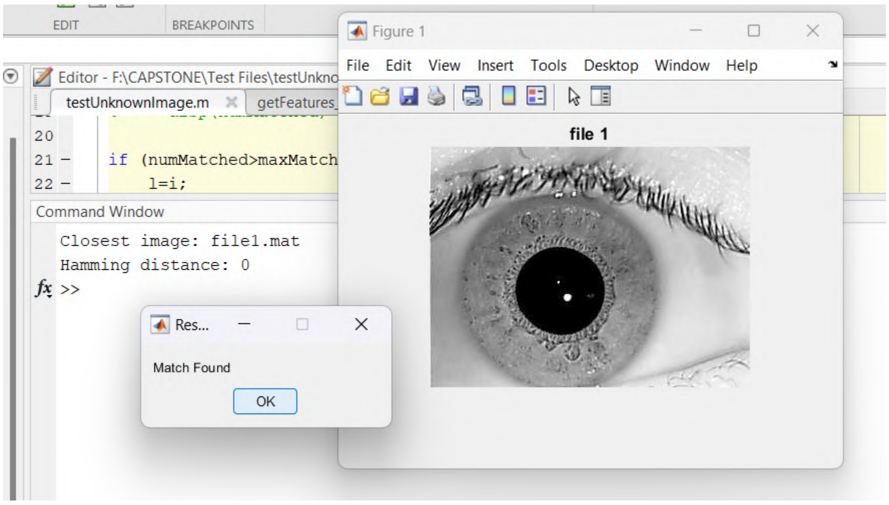
6.6 Feature Extraction

The extracted features are stored as .mat files as shown in **Fig.6.15**. They are loaded to compare the extracted features from the input unknown image.

Fig. 6.15 Stored Feature Vectors

6.7 Feature Matching

The features are matched from the existing database are it is stated if the match is found (**Fig.6.16**) or not (**Fig.6.17**).

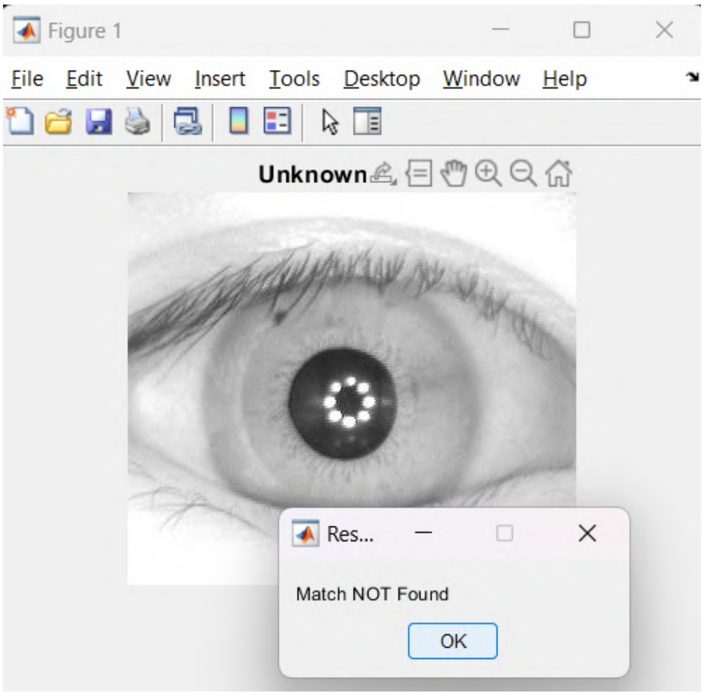


```
if (numMatched>maxMatch
  l=i;
```

```
Closest image: file1.mat
Hamming distance: 0
>>
```

Match Found

Fig. 6.16 Match Found in the Database

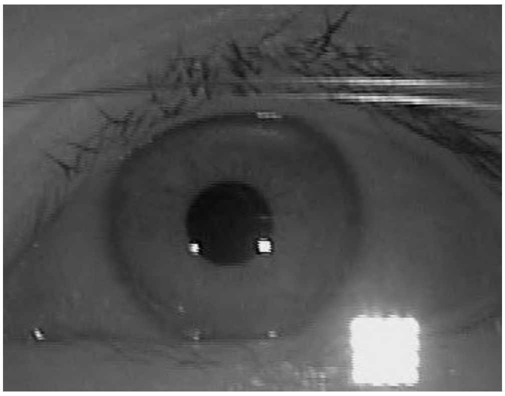


Match NOT Found

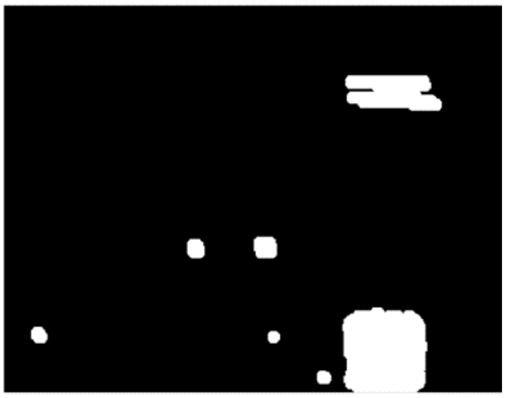
Fig. 6.17 Match Not Found in the Database

6.8 Tackling Specular Reflections

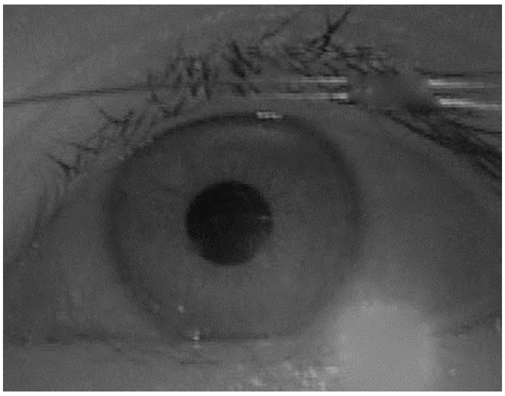
We have taken up an approach to address the specular reflections at the software level i.e using the MATLAB software as depicted in **Fig. 6.18**.



(a) Image with Reflection



(b) Morphological Dilated Mask



(c) Specular Reflection lessened

Fig. 6.18 Tackling Specular Reflection

CHAPTER 7

COST ANALYSIS / RESULT AND DISCUSSION

7.1 Cost Analysis

The software component of the project, executed on MATLAB and Python with VGG16, incurs no cost. MATLAB is covered under a university license, while Python and VGG16 are open-source, aligning with budget constraints and enabling efficient utilization of resources without financial burdens.

7.2 Results and Discussion

The various results obtained are discussed below.

1. The various pre-processing methods were compared against as shown in **Table 7.1** and **Table 7.2**.

Table 7.1 MSE Values of different Pre-processed Iris Image

Image	**Normal HE**	**CLAHE**	**Unsharp Masking + HE**	**Unsharp Masking + CLAHE**
1	1154.9291	1915.316	0.017907	0.032475
2	1918.418	2461.404	0.029245	0.040921
3	2363.3779	3030.7652	0.036133	0.048635
4	2363.3659	2702.3911	0.043798	0.043012
5	2782.9298	2589.4434	0.042111	0.040241
6	2648.9215	1464.0942	0.038771	0.025644
7	1011.8692	968.2536	0.014126	0.016339
Avg	**2112.263**	**1915.316**	**0.03172**7	**0.035324**

As can be seen from the above table, using an Unsharp Mask **reduces the MSE values** to a great extent. The Average value shows that the **best** pre-processing methods out of the following is using **Histogram Equalization in combination with the Unsharp Mask.**

Table 7.2 PSNR Values (in dB) of different Pre-processed Iris Image

Image	**Normal HE**	**CLAHE**	**Unsharp Masking + HE**	**Unsharp Masking + CLAHE**
1	17.5392	15.3424	65.6346	63.0493
2	15.3354	14.253	63.5042	62.0454
3	14.4295	13.3493	62.5858	61.2953
4	13.5327	13.8473	61.7502	61.8289
5	13.7198	14.0327	61.9208	62.1181
6	13.9341	16.5091	62.2797	64.075
7	18.1136	18.3049	66.6645	66.0326
Avg	**15.22919**	**15.09124**	**63.47711**	**62.92066**

A similar result is obtained for the PSNR Values. The Unsharp Mask **improves the PSNR values** to a great extent and the PSNR values also **favors the Histogram Equalization with Unsharp Mask.**

2. Out of the two functions ’imfindcircles’ and ’findcircles’ we were able to tune in the ’findcircles’ better. the ’imfindcircles’ was only able to localize the pupil and based on that it assumed the iris having a radius, say ’r’ px greater than the pupil radius. Since every iris has different radius, this was not a robust method. Even from **Fig. 6.7**, we can say that ’findcircles’ function is **more accurate** in localizing iris.

3. Our Deep Learning Model Metrics values are as follows:

   - **Accuracy:**
     This metric measures the overall correctness of the model’s predictions. In this case, an accuracy of **94.737%** means that the model correctly classified 94.737% of the iris samples.
   - **Precision:**
     Precision quantifies the accuracy of positive predictions made by the model. Specifically, it measures the proportion of true positive predictions (correctly identified positive cases) out of all positive predictions made. A precision of **94.444%** indicates that when the model predicts an iris as positive (belonging to a certain class), it is correct 94.444% of the time.
   - **Recall:**
     Recall, also known as sensitivity or true positive rate, evaluates the model’s ability to correctly identify all positive instances. It measures the proportion of true positive predictions out of all actual positive cases. A recall of

94.444% means that the model identified 94.444% of all positive iris samples correctly.

4. **TABLE 7.3** depicts the time elapsed for different feature extraction techniques in seconds for comparison purposes.

Table 7.3 Performance Evaluation of various Feature Extraction Methods

Image	DWT+PCA	LBP	Gabor Filter
1	0.128186	0.056524	0.429330
2	0.119805	0.056939	0.463508
3	0.122480	0.059595	0.471235
4	0.139157	0.060341	0.418886
5	0.122028	0.067032	0.444015
Avg	**0.126331**	**0.060086**	**0.445394**

We can figure out that the **best performance** is given by **Local Binary Pattern LBP method.**

5. Despite applying masking and dilation techniques, the persistent penumbra surrounding a substantial reflection in the corner continues to affect the inpainting process, resulting in a light gray appearance. Further dilation risks compromising iris details. Addressing this challenge at the hardware level through cross polaroids offers a more viable solution. Cross polaroids can effectively minimize or eliminate specular reflections, ensuring clearer iris images for improved inpainting outcomes without sacrificing crucial iris details during preprocessing stages.

6. **Future Scope:** The future of iris recognition involves incorporating different Neural Network models, such as the attention awareness model which offer potential improvements in localization precision, adaptability to variability, and lower rates of false acceptance and rejection. These models allocate attention dynamically to key areas of the iris, enhancing the flexibility of feature extraction and facilitating the integration of multiple biometric modalities. Emphasizing privacy protection and ongoing learning, attention-aware iris recognition systems aim for scalability, efficiency, and improved practical effectiveness

CHAPTER 8

SUMMARY

8.1 Summary

In this study, the focus is on biometric recognition, particularly the reliable identification of individuals through iris patterns. The uniqueness and resilience of iris patterns against aging make them a valuable biometric identifier. The research presents a methodical approach utilizing VGG16, a Neural Network model, in conjunction with MATLAB's image processing toolbox. The methodology covers training and fine-tuning VGG16 parameters, employing mathematical techniques for iris segmentation, feature extraction, and database matching to confirm individual identity. Experiments and analysis were conducted to demonstrate the effectiveness and robustness of various steps to iris recognition.

This research evaluates the effectiveness of different preprocessing techniques such as Histogram Equalization (HE), Contrast Limited Adaptive Histogram Equalization (CLAHE), and their combination with Unsharp Masking. Additionally, iris localization was performed using Canny Edge Detection, Hough Transform, and a VGG16 Model, achieving an accuracy of 94.737% and a precision and recall of both 94.444%. The study also compares various feature extraction methods including Discrete Wavelet Transform combined with Principal Component Analysis (DWT+PCA), Local Binary Patterns (LBP), and Gabor Filters. Furthermore, the research addresses the challenge of dealing with specular reflections on glasses that can impede iris recognition at the software level.

REFERENCES

Aljabri, M., AlAmir, M., AlGhamdi, M., Abdel-Mottaleb, M. and Collado-Mesa, F. (2022), 'Towards a better understanding of annotation tools for medical imaging: a survey', *Multimedia tools and applications* **81**(18), 25877–25911.

CBSR (2010), 'Casia iris image database'.
URL: *http://biometrics.idealtest.org/findTotalDbByMode.do?mode=Iris/datasetDetail/3*

Farouk, R. H., Mohsen, H. and El-Latif, Y. M. A. (2022), 'A proposed biometric technique for improving iris recognition', *International Journal of Computational Intelligence Systems* **15**(1), 79.

H., S. and Malisuwan, S. (2014), 'A study of image enhancement for iris recognition', *Journal of Industrial and Intelligent Information* **3**.

Jan, F., Min-Allah, N., Agha, S., Usman, I. and Khan, I. (2021), 'A robust iris localization scheme for the iris recognition', *Multimedia Tools and Applications* **80**, 4579–4605.

Jayalaxmi, H., Anitha, T., Sunil, S. H. and Prashanthi, H. (2022), 'Robust iris recognition algorithm using emd and support vector machine', *Journal of Positive School Psychology* pp. 4279–4288.

Jusman, Y., Ng, S. and Hasikin, K. (2020), 'Performances of proposed normalization algorithm for iris recognition', *International Journal of Advances in Intelligent Informatics* **6**, 161.

Mattar, E. (2013), 'Principal components analysis based iris recognition and identification system', *International Journal of Soft Computing and Engineering (IJSCE)* **3**, 430–436.

Omran, M. and AlShemmary, E. N. (2020), 'An iris recognition system using deep convolutional neural network', *Journal of Physics: Conference Series* **1530**(1), 012159.
URL: *https://dx.doi.org/10.1088/1742-6596/1530/1/012159*

Othman, N., Dorizzi, B. and Garcia-Salicetti, S. (2016), 'Osiris: An open source iris recognition software', *Pattern recognition letters* **82**, 124–131.

Sitaula, C. and Hossain, M. B. (2021), ‘Attention-based vgg-16 model for covid-19 chest x-ray image classification’, *Applied Intelligence* **51**(5), 2850–2863.

Song, R., Zhang, Z. and Liu, H. (2017), ‘Edge connection based canny edge detection algorithm’, *Pattern Recognition and Image Analysis* **27**, 740–747.

Tahir, A., Dawood, S. and Anghelus, S. (2021), ‘An iris recognition system using a new method of iris localization’.

Curriculum Vitae

Name: Subhadeep Mal
Father’s Name: Joydeb Mal
Date of Birth: 02/04/2002
Nationality: Indian
Sex: Male
Company Placed: TATA Consultancy Services
Permanent Address: Paschim Midnapur, West Bengal - 721102
Phone number: 7501415425
Mobile: 7501415425
E-mail ID: subhadeep.mal2020@vitstudent.ac.in

CGPA: 8.47

Placement Details: Placed through CDC.

Capstone Project

Project title:	Iris Image Recognition through Goggles Using CNN and Mathematical Models
Team Members:	Ridhima Chopra, Subhadeep Mal, Kevin Gerard
Faculty Guide:	Rajini G.K.
Semester/Year:	WIN SEM 2023-2024 / 4th Year
Project Abstract:	Biometric recognition is crucial for verifying individuals' identities using unique physiological or behavioral traits. The iris, known for its distinctiveness and resistance to aging effects, is a dependable biometric identifier. Our study introduces a comprehensive approach to iris detection by combining VGG16, a Neural Network model, with MATLAB's image processing tools. We meticulously train and fine-tune VGG16 parameters and utilize various mathematical techniques for iris segmentation, feature extraction, and database matching to confirm identity. We also tackle the challenge of specular reflections on glass surfaces, especially when individuals wear goggles, which can hinder accurate iris recognition. Through systematic experimentation and analysis, we demonstrate the effectiveness and robustness of our iris recognition system, marking significant progress in biometric authentication technology.

Project Title:	Iris Image Recognition Through Goggles using CNN and Mathematical Models
List **codes** and **standards** that significantly affect your project (Must)	**ANSI/INCITS 379-2004 Iris Interchange Format:** This standard uses a rectangular format for iris images, which can be raw or compressed and can vary in size. It's flexible but can result in larger file sizes. **ISO/IEC 19794-6:** 2005 Biometric Data Interchange Format – Part 6: This standard uses a polar image specification for iris images, with specific steps for preprocessing and segmentation. The resulting image contains only iris information and is much smaller in size compared to the ANSI format.
List at least two significant realistic design constraints that are applied to your project. (Must)	Our model for iris recognition depends on the pixel intensity. If there are variations in the illumination of the iris images subjected to recognition, it becomes difficult to recognise iris relying on the mathematical model. Moreover, specular reflections on the goggles is difficult to be tackled at the software level.
Briefly explain two significant trade-offs considered in your design, including options considered and the solution chosen (Must)	Since our proposed iris recognition system is pixel-intensity dependant, the same Near Infrared (NIR) camera must be used that is used to capture the dataset images. It has also been researched that the specular reflections cannot be just eliminated completely at the software level.Attempting to fill in the reflected region more will just lead to the degradation of the iris details. The best that can be done is by using cross polaroids and eliminating these unwanted reflections at the hardware level itself.
Describe the computing aspects, if any, of your project. Specifically identifying hardware-software trade-offs, interfaces, and/or interactions	**Deep Learning Framework Interaction:** The project involves using VGG16, a pre-trained deep learning model. This interaction includes loading the model, fine-tuning it if necessary, and using it for iris detection tasks. **MATLAB Interface:** MATLAB provides a graphical user interface (GUI) for developing and testing image processing algorithms. It also offers command-line interfaces for scripting and automation. **Performance Optimization:** Balancing computational load between CPU and GPU to maximize performance is crucial. This involves optimizing the code and leveraging parallel processing capabilities where applicable.