A Project report on

IRIS RECOGNITION USING MACHINE LEARNING TECHNIQUE

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the academic requirements for the award of the degree.

Bachelor of Technology

in

Computer Science and Engineering

Submitted by

A. Akshith
(19H51A0592)
K. Bharath Raj
(19H51A05A7)
L. Upendra
(19H51A05A9)

Under the esteemed guidance of

Mr. V. Venkataiah
(Associate Professor and
Additional controller of Examinations)

Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING & TECHNOLOGY
(UGC Autonomous)
*Approved by AICTE *Affiliated to JNTUH *NAAC Accredited with A^+ Grade

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2019- 2023

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Major Project report entitled **"IRIS RECOGNITION USING MACHINE LEARNING TECHNIQUE"** being submitted by A. AKSHITH (19H51A0592), K. BHARATH RAJ (19H51A05A7), L. UPENDRA (19H51A05A9) in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out his/her under my guidance and supervision.

The results embodies in this project report have not been submitted to any other University or Institute for the award of any Degree.

Mr. V. Venkataiah
Associate Professor of CSE
& Additional Controller of Examinations

Dr. Siva Skandha Sanagala
Associate Professor and HOD
Dept. of CSE

ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success.

We are grateful to **Mr. V. Venkataiah,** Associate Professor of CSE and Additional Controller of Examinations, Department of Computer Science and Engineering for his valuable technical suggestions and guidance during the execution of this project work.

We would like to thank **Dr. Siva Skandha Sanagala,** Head of the Department of Computer Science and Engineering, CMR College of Engineering and Technology, who is the major driving forces to complete my project work successfully.

We are very grateful to **Dr. Vijaya Kumar Koppula**, Dean-Academic, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Dr. V A Narayana,** Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the Teaching & Non- teaching staff of Department of Computer Science and Engineering for their co-operation

We express our sincere thanks to **Mr. Ch. Gopal Reddy**, Secretary, CMR Group of Institutions, for his continuous care.

Finally, we extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project work.

A. Akshith 19H51A0592
K. Bharath Raj 19H51A05A7
L. Upendra 19H51A05A9

TABLE OF CONTENTS

List of Figures

List of Tables

ABSTRACT

It is indispensable to ensure security biometrically in the most authentication and identification scenario. Iris recognition regarded as the most reliable biometric recognition due to its stable and extraordinary variation in texture. The unique patterns are used in iris recognition to identify individuals in requiring a high level of security. This paper explores an efficient technique that uses convolutional neural network (CNN) and support vector machine (SVM) for feature extraction and classification respectively to increase the efficiency of recognition. The proposed technique has been successfully applied and also clearly demonstrates the performance of the experimental evaluation on iris images from the CASIA database.

CHAPTER 1
INTRODUCTION

CHAPTER 1

INTRODUCTION

The biometric process has been mainly used to recognize individual types of physical aspects and features. For this purpose, a tremendous amount of acknowledgement technologies has been generally provided with the actual fingerprint, iris procedures and voice acknowledgement. The biometric mainly deals with the proper technical and technological fields for the body controls and body dimensions. The authentication system is based on the appropriate biometric security system that has increased the actual importance within all countries. The used system has been shown the proper valid and best impressive performance based on all these procedures and aspects. For this purpose, the fingerprint is the only procedure for providing the proper security techniques to provide the true uniqueness and the strong privacy properties of the entire system. The exceptional fingerprint assurance or the proper kind of imprint approval has been mainly insinuating the automated methods and procedures to ensure similarity between the two people fingerprints. The entire chapter has been generally provided with the actual purpose of the fundamental research that is overall dependent on the research objectives and respective research questions. In this chapter, the research framework of the entire study has also been provided. The fundamental research has described all the factors that are responsible for this recognition process

Biometrics is related to human unique characteristics. The most promising methods for authenticating a user are biometric systems. Biometric authentication can be favored over many conventional strategies, such as smart Cards and passwords since information is hard to steal here in biometrics. A biometric recognition device is used to recognize a person under surveillance and access control. Physiological characteristics and behavioral characteristics are commonly classified as biometric identifiers. Physiological characteristics are associated with the physical properties of the body, such as fingerprint, palm veins, DNA, facial recognition, iris, and so on.

The other category referred to an individual's model behaviour, such as voice, gait, etc. Biometrics offers an important protection platform in both physiological and behavioural ways. Biometrics has been fully integrated into our everyday lives.

Many studies have shown that other biometrics such as the ears, fingerprint, the iris have several advantages. Iris is also approved for precise and accurate biometric systems. It is one of the correct biometric identifications. Iris is the annular eye region between the white sclera and the black pupil, which makes it entirely shielded against environmental conditions.

The texture of the iris provides a high degree of randomness and individuality, which is very unlikely to be unique in any of the two iris patterns, either for identical twins or from a person's left and right eyes. This consistency in iris patterns is primarily due to the richness and differentiation within the iris texture features, including circles, ridges, crypts, freckles, furrows, zigzag patterns. This property makes it a reliable way of recognizing people. Also, the iris pattern remains constant until he died for an individual. This recognition approach is therefore considered safer and less vulnerable to spoofing attacks.

Finally, where an attempt to change its patterns is more risk of surgery, unlike fingerprint that is comparatively easier to manipulate, the most reliable biometric characteristic against fraudulent techniques and spoofing attempts by an impostor. The implementation of iris authentication systems, however, is a difficult job as it acquires eyelashes, reflections, and eyelids that may interfere with the recognition efficiency in a device. There are extensive uses of iris detection today in various protection applications, Such as restricted-area access control, Aadhar card database access, the bank, and other modern-day applications. In every biometric device, there are two ways to use the system. The first is the enrolment mode, which allows the processed prototype to be deposited in the database. The second mode is the identification mode, in which the template is identified by comparing it, if it exists, to one of the templates stored in the database.

Process Involved in Biometric Systems

A biometric system consists mainly of two stages, namely the stage of enrolment and the phase of verification. The images are collected from a single biometric trait in the enrolment phase and are processed to obtain a clear image as well as to correct distortions and to obtain the region of interest for extracting features. The feature extraction module extracts the details from the pre-processed image. The feature vector is created by extracting specific features from the image function and storing them in a database Shown in Fig 1.1.

Enrolment mode is a phase of learning to gather biometric data about whom to recognize. The query image that is to be tested is pre-processed to improve the quality of an image in the recognition phase.

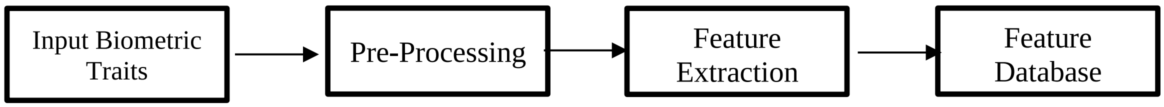


Fig 1.1 Steps involved in enrolling a biometric trait

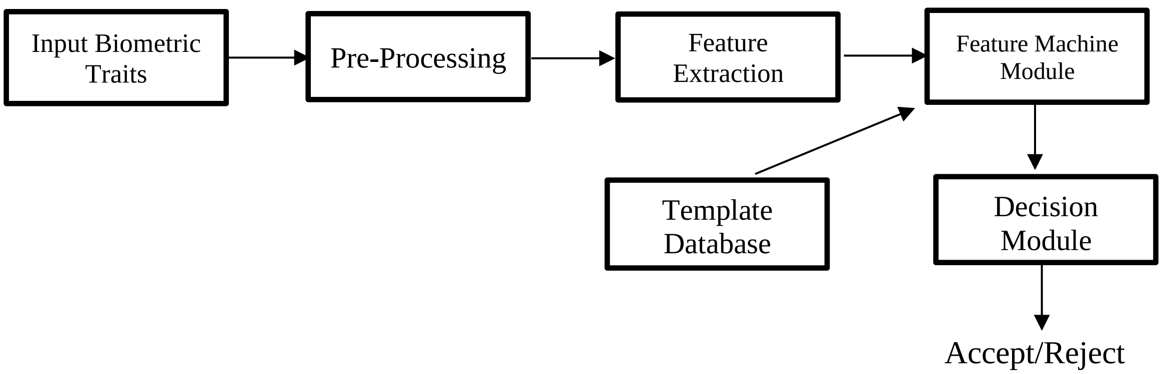


Fig 1.2 Steps Involved in a Biometric System during Recognition

The ROI is the process of highlighting key and interesting features as a smaller region in a biometric characteristic that will further be used in a biometric system as matching features and then extracted features. Thus, extracted features will be compared against the feature database in the matching module to generate a match score and finally by using the match score Decision module will identify the authorized persons. These sequences of steps are shown in Fig.2. A biometric system offers two important functions, one of which is verification and the other is identification. Verification involves a one-to-one search (1:1), while identification is achieved when a one-to-many (1: N) search is performed by a biometric system [14]. A biometric system demonstrates that you are who you claim you are during verification. On the other hand, in identification, an individual attempts to claim that he or she is one of the individuals registered in the database.

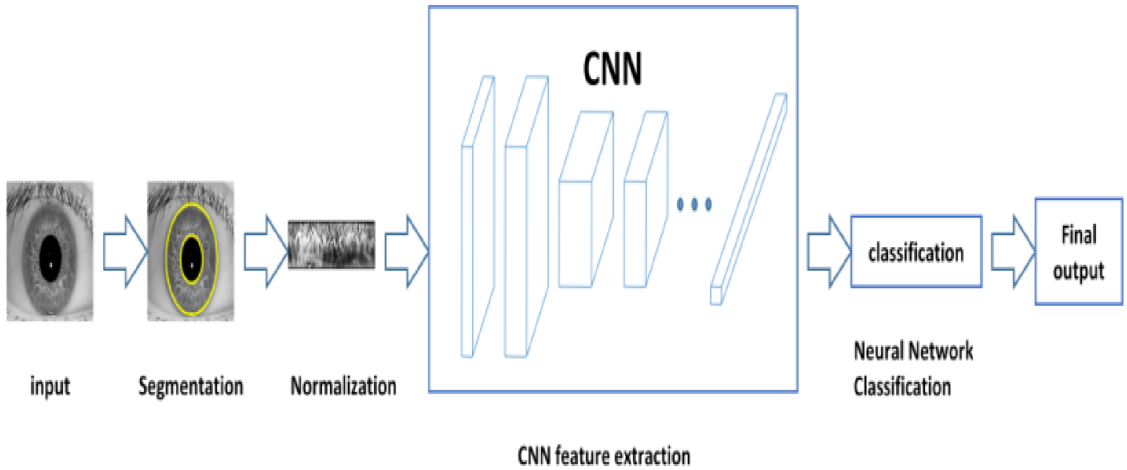


Fig 1.3 Process Involved in Iris Systems

The pre-processing on iris images is implemented by performing three stages: image enhancement, iris segmentation, and iris normalization shown in Fig.3. But before giving the input images to segmentation, the acquired images need to be processed to eliminate the noise during capturing. This work, also involving convert the color image (if the image is color) to the gray scale image, and a median filter is used to improve the quality of the image because brightness is not distributed uniformly and to remove some noise around the pupil as shown in Fig.

1.1 Problem Statement

Conventionally passwords, secret codes and PINs are used for identification which can be easily spleen observed or forgotten. In pattern recognition problems, the key issue is the relation between inter-class and intra-class variability. objects can be reliably classified only the variability among different instances of a given class is less than the variability between different classes. For example, in face recognition difficulties arise from the fact that the face is a changeable social organ displaying a variety of expressions, as well as being an active 3D object whose mage varies with viewing angle, pose, illumination, accoutrements, and age. So as an alternative we propose to use biometrics (iris recognition) system to identify an individual. This system will never allow to change the Biometric.

1.2 Research Objectives

- First, this project manages to eliminate the iris segmentation phase since in conventional IRS, its' failure can indirectly affect the recognition rate.
- Second, transferability of pre-trained ConvNet model can be tested using support vector machine (SVM) classifier with transfer learning technique.
- Third, a high-performance ConvNet model that able to run IRS task is developed using transfer learning technique.
- To get high performance, High efficiency and Accuracy of Recognition

1.3 Project Scope and Limitations

- While the most common use of iris recognition to date is physical access control in private enterprise and government, the versatility of the technology will lead to its growing use in large sectors of the economy such as transportation, healthcare, and national identification programs. Although security is clearly a prime concern, iris recognition is also being adopted for productivity-enhancing applications like time and attendance.
- Enterprise and government both acknowledge the convergence of physical and information security environments, but there are new security challenges on the horizon – just-in-time inventory control, sophisticated supply chain management, and even a phenomenon called “coopetition”-in which companies that compete in some areas, cooperate in others.
- Can't use a regular camera; requires IR light source and sensor. Visible light must be minimized for highest accuracy required for search.
- Generally, require close proximity to camera, which can cause discomfort for some.
- Less value for criminal investigation (no latents)
- Enterprise and government both acknowledge the convergence of physical and information security environments, but there are new security challenges on the horizon – just-in-time inventory control, sophisticated supply chain management, and even a phenomenon called “coopetition”-in which companies that compete in some areas, cooperate in others.

CHAPTER 2
BACKGROUND WORK

CHAPTER 2

BACKGROUND WORK

2.1 Face Detection by using Viola-Jones algorithm

2.1.1 Introduction

From last few years, the face detection has received important consideration. In few areas the face detection gain extraordinary position due to various reasons such as verification of identities, large range of commercial and law enforcement available for feasible technologies. Face detection is one of many applications in digital image processing. It is concerned with the automatic identification of an individual in a digital image. Face detection is used to identifying and locating the human face irrespective of its size, position and situations. Face detection is an easy task for human brain but it is a very difficult task for computer system [1]. To detect the face easily and accurately computer system needs some training factors so that the computer system can easily identify whether it is face or non- face. For detecting the face some thresholds values are sets based on this value a system can detect the human face. If the image specifies the desired threshold value, then the image is a face otherwise it is a non-face.

2.1.2 Merits and Demerits

Merits of Face Detection by using Viola-Jones algorithm

• Detection is very fast • Simple to understand and implement

• Less data needed for training than other ML models

• No rescaling of images needed (like with CNN’s)

• Much more interpretable than contemporary models

Demerits of Face Detection by using Viola-Jones algorithm

• Restricted to binary classification

• Mostly effective when face is in frontal view

• May be sensitive to very high/low exposure (brightness)

• High true detection rate, but also high false detection rate

• Training time is very slow

2.1.3 Implementation of Face Detection by using Viola-Jones algorithm

Viola Jones algorithm is named after two computer vision researchers who proposed the method in 2001, Paul Viola and Michael Jones in their paper, “Rapid Object Detection using a Boosted Cascade of Simple Features”. This algorithm mainly focuses on the relevant features on our face. For example Eyes, nose, lips, eye brows etc,to identify a face.

This Algorithm follows fours steps to identify a face

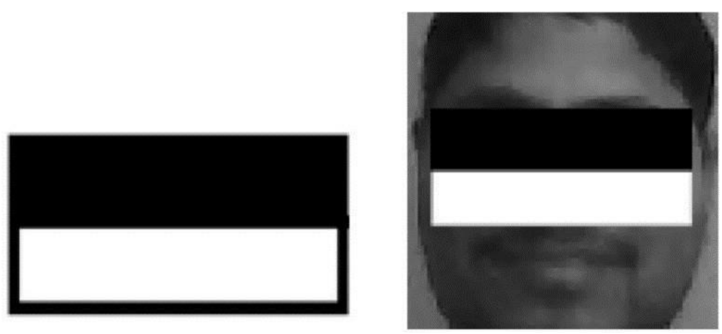
1.Haar Feature

2.Integral Image

3.Boosting with adaboost

4. Cascading

1.Haar Feature



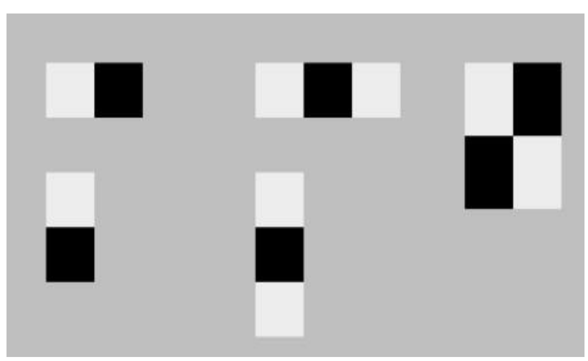


Fig-2.1.3.1 Haar Feature

These features help the machine understand what the image is. Imagine what the edge of a table would look like on a b&w image. One side will be lighter than the other, creating that edge like b&w feature as you can see in the picture above.
In the two important features for Face Detection, the horizontal and the vertical features describe what eyebrows and the nose, respectively, look like to the machine.

2. Integral Image

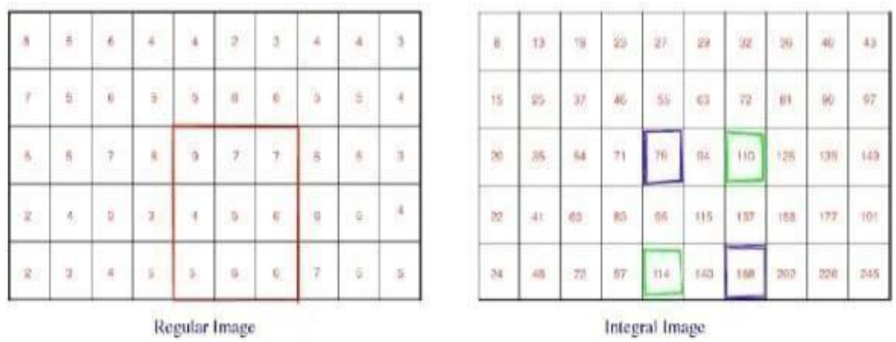


Fig2.1.3.2 Integral Image

The integral image plays its part in allowing us to perform these intensive calculations quickly so we can understand whether a feature of a number of features fit the criteria.

3.Boosting with AdaBoost

The number of features that are present in the 24×24 detector window is nearly 160,000, but only a few of these features are important to identify a face. So we use the AdaBoost algorithm to identify the best features in the 160,000 features.
In the Viola-Jones algorithm, each Haar-like feature represents a weak learner. To decide the type and size of a feature that goes into the final classifier, AdaBoost checks the performance of all classifiers that you supply to it.
To calculate the performance of a classifier, you evaluate it on all subregions of all the images used for training. Some subregions will produce a strong response in the classifier. Those will be classified as positives, meaning the classifier thinks it contains a human face. Subregions that don’t provide a strong response don’t contain a human face, in the classifiers opinion. They will be classified as negatives.

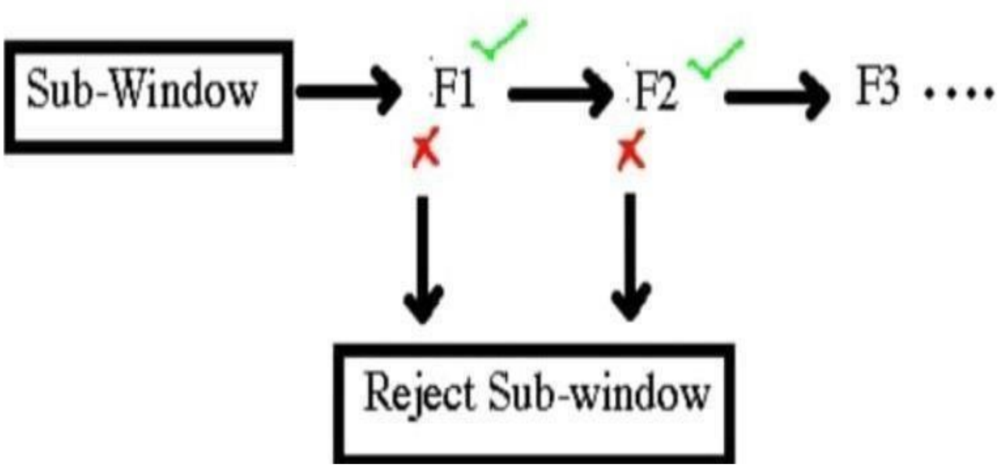
The classifiers that performed well are given higher importance or weight. The final result is a strong classifier, also called a boosted classifier, that contains the best performing weak classifiers.

So when we're training the AdaBoost to identify important features, we're feeding it information in the form of training data and subsequently training it to learn from the information to predict. So ultimately, the algorithm is setting a minimum threshold to determine whether something can be classified as a useful feature or not.

4.Cascading

Cascading is another sort of "hack" to boost the speed and accuracy of our model. So we start by taking a sub window and within this sub window, we take our most important or best feature and see if it is present in the image within the sub window. If it is not in the sub window, then we don't even look at the sub window, we just discard it. Then if it is present,we look at the second feature in the sub window. If it isn't present, then we reject the subwindow. We go on for the number of features have, and reject the sub windows without the feature. Evaluations may take split seconds but since you have to do it for each feature, it could take a lot of time. Cascading speeds up this process a lot, and the machine is able to deliver results much faster.

$$F(x) = a1f1(x) + a2f2(x) + a3f3(x) \ldots\ldots$$



Results

Five studies are analyzed and described the application of the Viola-Jones algorithm for facial recognition

- The first study had a very good accuracy level of 85%-95% in detecting faces.
- The second study had accuracy, precision, recall, and achievement times of 0.74, 0.73, 0.76, and 15 seconds in recognizing a person's emotions through facial expressions.
- Meanwhile, the third study had a very good accuracy level of 94.5% in recognizing faces that are 1 meter away.

2.2 Counterfeit Iris Detection Based on Texture Analysis

2.2.1 Introduction

Personal identification using biometrics has been developing rapidly since the past decade. Biometric systems have already been deployed to border control, access to personal computers and the control of airport. However, biometric systems still have vulnerabilities. Spoofing biometric system may occur in every step from data acquisition to decision lev el, including using fake biometrics, replaying attacks, corrupting match scores and overriding final decision, etc. Iris pattern is considered as the most accurate and stable biometric modality; however, iris recognition system meets new challenge in anti-counterfeit iris as color contact lens become popular recently. Attackers wearing contact lens with artificial textures printed onto them may try to access the system unauthorized, which is one of the potential means to spoof the systems. This paper addresses the issue of detecting fake iris wearing printed color contact lens, with the purpose of making iris recognition system more robust in anti-spoofing. This issue is a liveness detection problem in biometrics, which aims to ensure that images acquired by the camera are real patterns. In previous studies of this issue, Daughman proposed to detect printed iris pattern using spurious energy in 2D Fourier spectra. Lee et al. suggested detecting fake iris based on Purkinje image. He et al. used four features (image mean, variance, contrast and angular second moment) to detect fake iris. In this paper, we propose three anti spoofing measures to prevent printed color contact lens from accessing iris recognition system.

Detecting iris edge sharpness is the first measure proposed. Another measure comes from our previous work for coarse iris classification, using feature named Iris-Texton by learned visual vocabulary to classify iris texture. Lastly, textural features based on co-occurrence matrix (CM) are adopted, and feature selection procedure is applied to find a combination of 3 features from the 28 features proposed by Haralick et al.

2.2.2 Merits and Demerits

Merits of Counterfeit Iris Detection Based on Texture Analysis

- It can be employed in systems with speed requirement given its low computational cost
- Its performance is comparable to the state-of-the-art.
- It can also be deployed by setting the threshold to a low FAR (False Acceptance Rate).
- CCR (Correct Classification Rate) is closely related to the type of contact lens, or the technique of making them.

Demerits of Counterfeit Iris Detection Based on Texture Analysis

- The proposed method highly depends on segmentation accuracy.
- Some datasets were more challenging to segment the iris regions exactly.
- While considering we cannot give assurance that the iris will be segmented exactlyfor all the real time data.
- The False Rejection Rate is also low.

2.2.3 Implementation of Counterfeit Iris Detection Based on Texture Analysis

- This paper addresses the issue of counterfeit iris detection, which is a liveness detection problem in biometrics.
- Fake iris mentioned here refers to iris wearing color contact lens with textures printed onto them.
- We propose three measures to detect fake iris: measuring iris edge sharpness, applying Iris-Texton feature for characterizing the visual primitives of iris textures and using selected features based on co-occurrence matrix (CM).
- This paper addresses the issue of counterfeit iris detection, which is a liveness detection problem in biometrics.

- This paper addresses the issue of counterfeit iris detection, which is a liveness detection problem in biometrics.
- Fake iris mentioned here refers to iris wearing color contact lens with textures printed onto them.
- We propose three measures to detect fake iris: measuring iris edge sharpness, applying Iris-Texton feature for characterizing the visual primitives of iris textures and using selected features based on co-occurrence matrix (CM).
- Extensive testing is carried out on two datasets containing different types of contact lens with totally 640 fake iris images, which demonstrates that IrisTexton and CM features are effective and robust in anticounterfeit iris.
- Detailed comparisons with two state of-the-art methods in literatures are also presented, showing that the proposed iris edge sharpness measure acquires a comparable performance with these two methods, while Iris-Texton and CM features outperform the state-of-the-art.
- The two data sets used in this Analysis are CASIA and BATH.

Result

	DB1			DB2		
	CCR	FAR	FRR	CCR	FAR	FRR
IES	97.8	1.87	2.5	76.8	27.1	19.4
ITF	98.3	1.25	1.87	95.8	4.17	5.83
CMF	100	0	0	94.1	8.13	3.75
He[7]	96.7	4.4	2.5	78.1	24.6	19.2
D [5]	96.3	7.5	0	/	/	/

Table 2.2.1 Result

CHAPTER 3
PROPOSED SYSTEM

CHAPTER 3

PROPOSED SYSTEM

3.1 Objective of Proposed System

Our proposed system is iris recognition s using machine learning. In this system we are going to to use the Convolutional neural network(CNN) for feature extraction and classification to increase the efficiency of the recognition.
The proposed technique has been successfully applied and also clearly demonstrates the performance of the experimental evaluation on iris images from the CASIA database.

In this project there are few step to implement the system
1. Image Acquisition
2. Segmentation
3. Normalization
4. Feature Extraction
5. Matching
6. Result

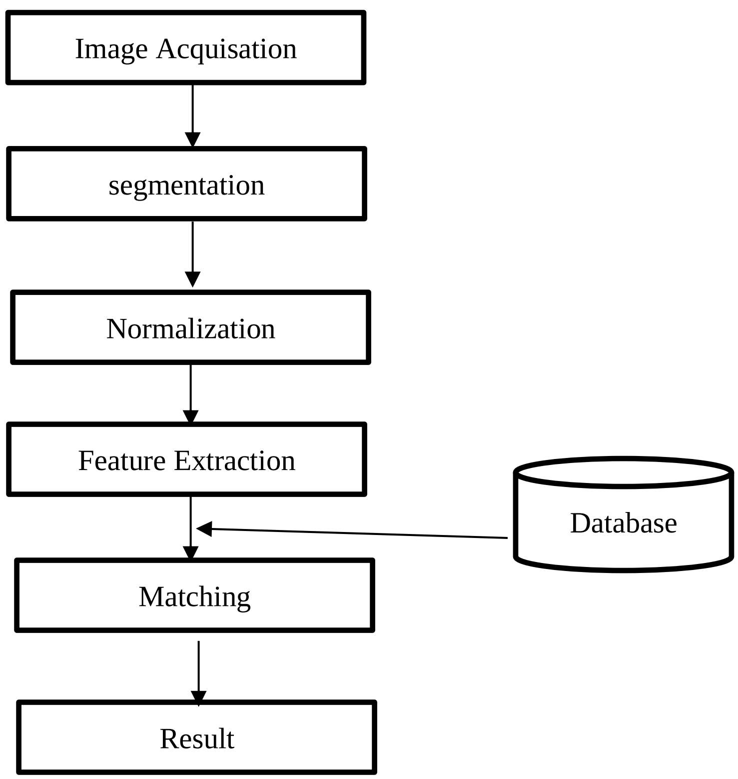


Fig 3.1.1 Flow Chart

- **Image Acquisition**

- The first step of iris recognition is image acquisition deals with capturing sequence of high-quality iris images from the subject using cameras and sensors.
- Instead of capturing images we use CASIA dataset.
- CASIA dataset contain 108 Irises and 7 images per iris

3.2 Algorithms Used

- **Iris Segmentation**

- The next step of iris recognition is iris segmentation, is a process to isolate the actual iris region in a digital eye image.
- We use Hough transform algorithm for this iris segmentation
- Segmentation using Hough Transforms

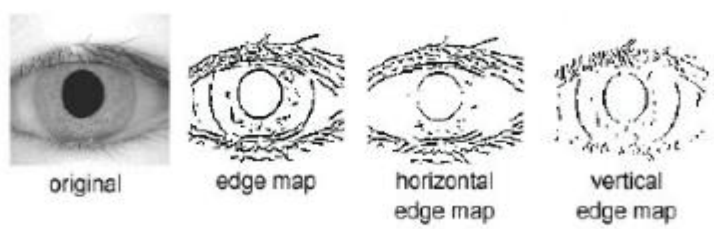


Fig3.2.1 Iris Segmentation

- The system looks for the Iris-pupil boundary within that circle after determining the Iris-sclera boundary. The Eq.(1) is the generalized expression for Circular Hough Transform, whereas the expansion for Eq.(1) is stated in Eq.(2) having the circle information
- H (a, b, r) = h (xi, yi , a, b, r) n i=1 Eq .(1)
- Here (xi, yi) is an edge pixel and i is the index of the edge pixel
- Where h (xi , yi , a, b, r) = (xi − a) 2 + (yi −b) Eq.(2)
- The Circular Hough transformation is used to determine the iris and pupil's center and radius. The radius is denoted by r and the center of the iris is represented by (a, b) and the coordinates of the circle are(xi , yi). The pair of equations at Eq.(3) results in the expression of the circle equation in parametric polar form.
    - a = x + rcos θ b = y+ rsin(θ) Eq. (3)

- The edge image in Hough space is used to cast votes for the parameters, with the central coordinates (xc, yc,) and the radius 'r' of circles passing through each edge point. They are capable of describing any circle by such criteria. In general, it is difficult for edge detectors to adapt to various situations.
- The quality of edge detection is heavily dependent on the lighting conditions, the presence of similar intensity objects, the density of the edges in the scene, and also the noise.
- To minimize the search space, the eye image must be improved both before and after the edge map is created. A 2D median filter is used in this work to smooth the eye image and reduce noise.

➢ **Normalization**

- After successfully segmented the iris region from an eye image, the circular iris region is transformed into a fixed size rectangular block.
- Daughman Rubber sheet model is used here to normalize iris image.

$$I[x(r,\theta), y(r,\theta)] \rightarrow I(r,\theta)$$

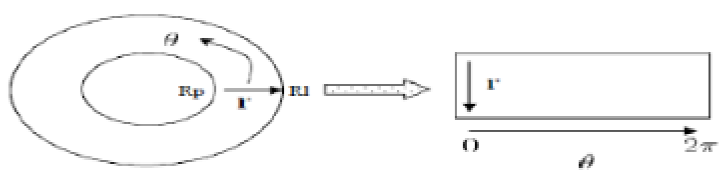


Fig 3.2.2 Daughman Rubber Sheet Model

- Normalization creates a 2-D array with horizontal angular resolution dimensions and vertical radial resolution dimensions from the iris region as shown in Fig.3.1.2
- After successfully segmented the iris region from an eye image, the circular iris region is transformed into a fixed size rectangular block.

The angular resolution is the number of radial vectors that travel across the iris field, while the radial resolution is the number of data points selected for each radial vector . Remapping region of the iris to the polar coordinates from the Cartesian coordinates using Mapping and remapping from Cartesian to Polar coordinate system and the vice-versa is performed using the Eq. (4), Eq. (5), and Eq. (6)

$$x(r, \theta) = r \times \cos(\theta) \quad \text{Eq. (4)}$$

$$y(r, \theta) = r \times \sin(\theta) \quad \text{Eq. (5)}$$

$$l(x(r, \theta), y(r, \theta)) \rightarrow I(r, \theta) \quad \text{Eq .(6)}$$

Feature Extraction

The most important step in iris recognition is feature extraction. Feature extraction is a process of finding the most discriminating information present in an iris pattern. The recognition rate and run time of matching of two iris templates mostly depends on feature extraction technique. In this paper, CNN is used for extracting feature of the normalized iris image.

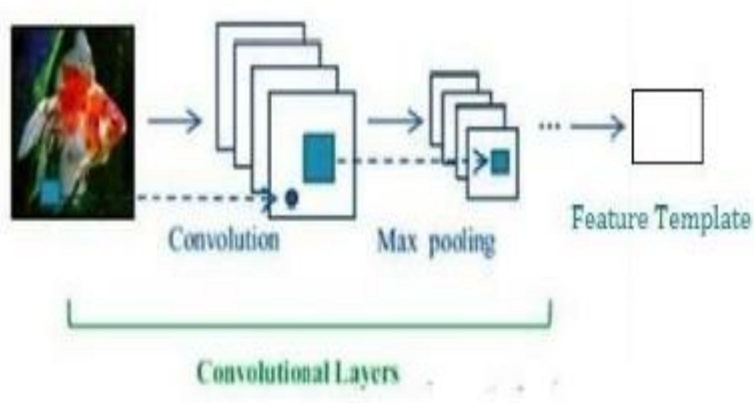
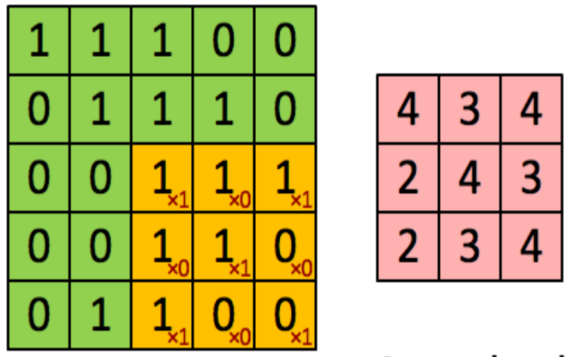


Fig 3.2.3 Convolutional Neural Layers

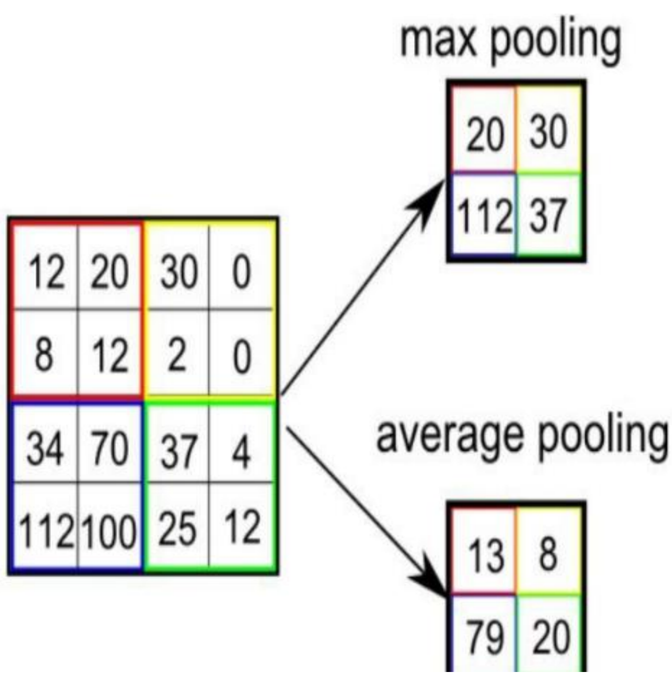
Convolution



One of the main reasons for CNN's success in computer vision is that these deep networks are highly adapted to capture and encrypt complex imaging functions with more layers and millions of parameters with greater efficiency. The structure of the proposed CNN includes Convolutional layers, max pooling, and fully connected layers.

Pooling Layer:

Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data by reducing the dimensions.



Classification

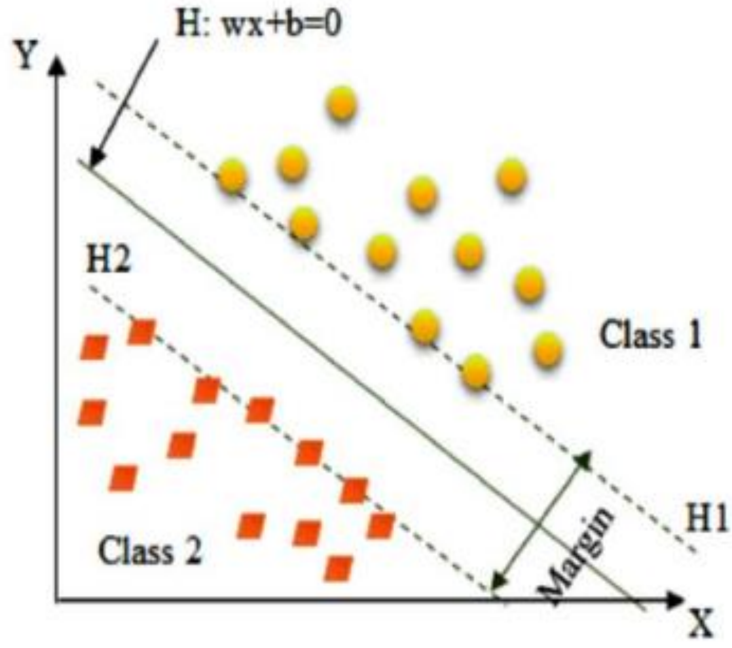
The template generated in the feature extraction stage needs a matching metric to measure the similarity between two iris templates. This metric gives one range of values when templates generated from the same eye are compared and another range of values when templates generated from different persons eye are compared; so that we can decide as to whether the two templates belong to the same or different persons.

Support Vector Machine

Pattern recognition is performed by Support vector machine (SVM) that works on the basis of the principle of structural risk minimization. SVM is treated as binary classifier that separates the two classes of data optimally. The two major aspects of developing SVM as a classifier is:

- to determine the optimal hyperplane in between two separate classes of data and
- to transform the non-linearly separable classification problem into linearly separable problem.

Linearly separable classification problem is shown in as
for an example.



SVM with Linear separable data

- Let x is a set of input feature vector and y is the class label. The input feature vectors and the class label can be represented as{xi,yi}, where i = 1, 2 . . . N and y = ± 1. The separating hyperplane can be represented as follows,

  w.x+b=0

  This implies,

  Yi(w.xi+b)>=1 where i=1,2,3,..
- (w,b) can have numerous possible values which create separating hyperplane. It is believed that points often lie between two data classes in such a way that there is always some margin in between them. SVM maximizes this margin by considering it as a quadratic problem [22]. The SVM is used to make two possible decisions during iris recognition; acceptance or rejection of a person.

3.3 Designing

3.3.1 UML Diagram

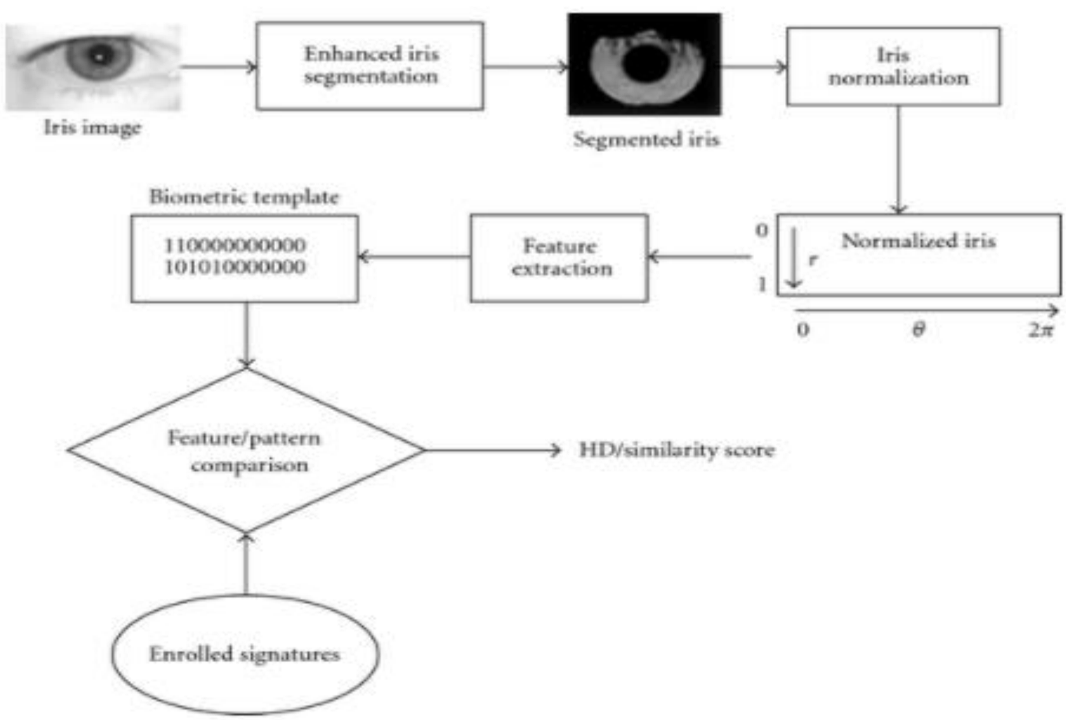


Fig 3.3.1.1 UML Diagram

3.4 Step wise Implementation and code

Code for Iris recognition Using machine learning Technique

```
from tkinter import messagebox
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
from tkinter.filedialog import askopenfilename
import numpy as np
import matplotlib.pyplot as plt
import os
from keras.utils.np_utils import to_categorical
from keras.layers import  MaxPooling2D
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Convolution2D
from keras.models import Sequential
from keras.models import model_from_json
import pickle
import cv2
from keras.preprocessing import image
from skimage import data, color
from skimage.transform import hough_circle, hough_circle_peaks
from skimage.feature import canny
from skimage.draw import circle_perimeter
from skimage.util import img_as_ubyte
main = tkinter.Tk()

main.title("Iris Recognition using Machine Learning Technique") #designing main screen
main.geometry("1300x1200")

global filename
global model

def getIrisFeatures(image):
    global count
    img = cv2.imread(image,0)
    img = cv2.medianBlur(img,5)
    cimg = cv2.cvtColor(img,cv2.COLOR_GRAY2BGR)
    circles                                                                                    =
cv2.HoughCircles(img,cv2.HOUGH_GRADIENT,1,10,param1=63,param2=70,minRadius=0,
maxRadius=0)
    if circles is not None:
        height,width = img.shape
```

```
    if circles is not None:
        height,width = img.shape
        r = 0
        mask = np.zeros((height,width), np.uint8)
        for i in circles[0,:]:
            cv2.circle(cimg,(i[0],i[1]),int(i[2]),(0,0,0))
            cv2.circle(mask,(i[0],i[1]),int(i[2]),(255,255,255),thickness=0)
            blank_image = cimg[:int(i[1]),:int(i[1])]

            masked_data = cv2.bitwise_and(cimg, cimg, mask=mask)
            _,thresh = cv2.threshold(mask,1,255,cv2.THRESH_BINARY)
            contours                                                    =
cv2.findContours(thresh,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
            x,y,w,h = cv2.boundingRect(contours[0][0])
        crop = img[y:y+h,x:x+w]
            r = i[2]
        cv2.imwrite("test.png",crop)
    else:
        count = count + 1
        miss.append(image)
    return cv2.imread("test.png")

def uploadDataset():
    global filename
    filename = filedialog.askdirectory(initialdir=".")
    text.delete('1.0', END)
    text.insert(END,filename+" loaded\n\n");

def loadModel():
    global model
    text.delete('1.0', END)
    X_train = np.load('model/X.txt.npy')
    Y_train = np.load('model/Y.txt.npy')
    print(X_train.shape)
    print(Y_train.shape)
    text.insert(END,'Dataset contains total '+str(X_train.shape[0])+' iris images from
'+str(Y_train.shape[1])+"\n")
    if os.path.exists('model/model.json'):
        with open('model/model.json', "r") as json_file:
            loaded_model_json = json_file.read()
            model = model_from_json(loaded_model_json)
        model.load_weights("model/model_weights.h5")
   model._make_predict_function()
```

```
with open('model/model.json', "r") as json_file:
        loaded_model_json = json_file.read()
        model = model_from_json(loaded_model_json)
    model.load_weights("model/model_weights.h5")
 model._make_predict_function()
    print(model.summary())
    f = open('model/history.pckl', 'rb')
    data = pickle.load(f)
    f.close()
    acc = data['accuracy']
    accuracy = acc[59] * 100
    text.insert(END,"CNN Model Prediction Accuracy = "+str(accuracy)+"\n\n")
    text.insert(END,"See Black Console to view CNN layers\n")
  else:
    model = Sequential()
    model.add(Convolution2D(32, 3, 3, input_shape = (64, 64, 3), activation = 'relu'))
    model.add(MaxPooling2D(pool_size = (2, 2)))
    model.add(Convolution2D(32, 3, 3, activation = 'relu'))
    modeladd(MaxPooling2D(pool_size = (2, 2)))
    model.add(Flatten())
    model.add(Dense(output_dim = 256, activation = 'relu'))
    model.add(Dense(output_dim = 108, activation = 'softmax'))
    print(model.summary())
    model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics =
['accuracy'])
    hist = model.fit(X_train, Y_train, batch_size=16, epochs=60, shuffle=True, verbose=2)
    model.save_weights('model/model_weights.h5')
    model_json = classifier.to_json()
    with open("model/model.json", "w") as json_file:
        json_file.write(model_json)
    f = open('model/history.pckl', 'wb')
    pickle.dump(hist.history, f)
    f.close()
    f = open('model/history.pckl', 'rb')
    data = pickle.load(f)
    f.close()
    acc = data['accuracy']
    accuracy = acc[59] * 100
    text.insert(END,"CNN Model Prediction Accuracy = "+str(accuracy)+"\n\n")
    text.insert(END,"See Black Console to view CNN layers\n")
    img = np.asarray(im2arr)
   img = img.astype('float32')
   img = img/255
  preds = model.predict(img)
 predict = np.argmax(preds) + 1
```

```
 def predictChange():
   filename = filedialog.askopenfilename(initialdir="testSamples")
   image = getIrisFeatures(filename)
   img = cv2.resize(image, (64,64))
   im2arr = np.array(img)
    im2arr = im2arr.reshape(1,64,64,3)
   img = np.asarray(im2arr)
   img = img.astype('float32')
   img = img/255
   preds = model.predict(img)
   predict = np.argmax(preds) + 1
   print(predict)
   img = cv2.imread(filename)
   img = cv2.resize(img, (600,400))
   img1 = cv2.imread('test.png')
   img1 = cv2.resize(img1, (400,200))
cv2.putText(img, 'Person ID Predicted from Iris Recognition is : '+str(predict), (10, 25),
cv2.FONT_HERSHEY_SIMPLEX,0.7, (255, 0, 0), 2)
   cv2.imshow('Person ID Predicted from Iris Recognition is : '+str(predict), img)
   cv2.imshow('Iris features extacted from image', img1)
   cv2.waitKey(0)
def graph():
   f = open('model/history.pckl', 'rb')
   data = pickle.load(f)
   f.close()

   accuracy = data['accuracy']
   loss = data['loss']
   plt.figure(figsize=(10,6))
   plt.grid(True)
   plt.xlabel('Epoch')
   plt.ylabel('Accuracy/Loss')
   plt.plot(loss, 'ro-', color = 'red')
   plt.plot(accuracy, 'ro-', color = 'green')
   plt.legend(['Loss', 'Accuracy'], loc='upper left')
      img = cv2.resize(image, (64,64))
   im2arr = np.array(img)
    im2arr = im2arr.reshape(1,64,64,3)
   img = np.asarray(im2arr)
   img = img.astype('float32')
   img = img/255
   preds = model.predict(img)
   predict = np.argmax(preds) + 1
   print(predict)
```

```
    #plt.xticks(wordloss.index)
    plt.title('GoogLeNet Accuracy & Loss Graph')
    plt.show()

def close():
    main.destroy()

font = ('times', 16, 'bold')
title = Label(main, text='Iris Recognition using Machine Learning Technique')
title.config(bg='goldenrod2', fg='black')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)

font1 = ('times', 12, 'bold')
text=Text(main,height=20,width=150)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=50,y=120)
text.config(font=font1)


font1 = ('times', 13, 'bold')
uploadButton = Button(main, text="Upload Iris Dataset", command=uploadDataset,
bg='#ffb3fe')
uploadButton.place(x=50,y=550)
uploadButton.config(font=font1)

modelButton = Button(main, text="Generate & Load CNN Model", command=loadModel,
bg='#ffb3fe')
modelButton.place(x=240,y=550)
modelButton.config(font=font1)

graphButton = Button(main, text="Accuracy & Loss Graph", command=graph, bg='#ffb3fe')
graphButton.place(x=505,y=550)
graphButton.config(font=font1)

predictButton = Button(main, text="Upload Iris Test Image & Recognize",
command=predictChange, bg='#ffb3fe')
predictButton.place(x=730,y=550)
predictButton.config(font=font1)
exitButton = Button(main, text="Exit", command=close, bg='#ffb3fe')
exitButton.place(x=1050,y=550)
exitButton.config(font=font1)
main.config(bg='magenta')
```

main.config(bg='magenta')
main.mainloop()

Code for Checking the model

```
from skimage import io
from skimage import data, color
from skimage.transform import hough_circle, hough_circle_peaks
from skimage.feature import canny
from skimage.draw import circle_perimeter
from skimage.util import img_as_ubyte
import cv2
import matplotlib.pyplot as plt
import numpy as np
import os
from keras.utils.np_utils import to_categorical
from keras.layers import  MaxPooling2D
from keras.layers import Dense, Dropout, Activation, Flatten
from keras.layers import Convolution2D
from keras.models import Sequential
from keras.models import model_from_json
import pickle
count = 0
miss = []
def getIrisFeatures(image):
    global count
    img = cv2.imread(image,0)
    img = cv2.medianBlur(img,5)
    cimg = cv2.cvtColor(img,cv2.COLOR_GRAY2BGR)
    circles                                                                                    =
cv2.HoughCircles(img,cv2.HOUGH_GRADIENT,1,10,param1=63,param2=70,minRadius=0,
maxRadius=0)
    status = 'failed'
    crop = None
    if circles is not None:
        height,width = img.shape
        r = 0
        mask = np.zeros((height,width), np.uint8)
        for i in circles[0,:]:
            cv2.circle(cimg,(i[0],i[1]),int(i[2]),(0,0,0))
            cv2.circle(mask,(i[0],i[1]),int(i[2]),(255,255,255),thickness=0)
            blank_image = cimg[:int(i[1]),:int(i[1])]
            masked_data = cv2.bitwise_and(cimg, cimg, mask=mask)
            _,thresh = cv2.threshold(mask,1,255,cv2.THRESH_BINARY)
     Contours                                                                                  =
cv2.findContours(thresh,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
            x,y,w,h = cv2.boundingRect(contours[0][0]
```

```
 cv2.findContours(thresh,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_SIMPLE)
            x,y,w,h = cv2.boundingRect(contours[0][0])
            crop = img[y:y+h,x:x+w]
            cv2.imwrite('test.png',crop)
            crop = cv2.imread('test.png')
            r = i[2]
            status = 'success'

    return status,crop

path = 'CASIA1'

labels = []
X_train = []
Y_train = []

def getID(name):
    index = 0
    for i in range(len(labels)):
        if labels[i] == name:
            index = i
            break
    return index


for root, dirs, directory in os.walk(path):
    for j in range(len(directory)):
        name = os.path.basename(root)
        if name not in labels:
            labels.append(name)
print(labels)

for root, dirs, directory in os.walk(path):
    for j in range(len(directory)):
        name = os.path.basename(root)
        if 'Thumbs.db' not in directory[j]:
            status,img = getIrisFeatures(root+"/"+directory[j])
            if status =='success':
                img = cv2.resize(img, (64,64))
                im2arr = np.array(img)
                im2arr = im2arr.reshape(64,64,3)
                labels = []
               X_train = []
```

```
    X_train.append(im2arr)
            ids = getID(name)
            Y_train.append(int(name)-1)
            print(str(ids)+" "+str(name))

X_train = np.asarray(X_train)
Y_train = np.asarray(Y_train)
print(Y_train)

X_train = X_train.astype('float32')
X_train = X_train/255

test = X_train[3]
cv2.imshow("aa",test)
cv2.waitKey(0)
indices = np.arange(X_train.shape[0])
np.random.shuffle(indices)
X_train = X_train[indices]
Y_train = Y_train[indices]
Y_train = to_categorical(Y_train)
np.save('model/X.txt',X_train)
np.save('model/Y.txt',Y_train)

X_train = np.load('model/X.txt.npy')
Y_train = np.load('model/Y.txt.npy')
print(Y_train)
if os.path.exists('model/model.json'):
    with open('model/model.json', "r") as json_file:
        loaded_model_json = json_file.read()
        classifier = model_from_json(loaded_model_json)
    classifier.load_weights("model/model_weights.h5")
    classifier._make_predict_function()
    print(classifier.summary())
    f = open('model/history.pckl', 'rb')
    data = pickle.load(f)
    f.close()
    acc = data['accuracy']
    accuracy = acc[9] * 100
    print("Training Model Accuracy = "+str(accuracy))
else:
    classifier = Sequential()
    classifier.add(Convolution2D(32, 3, 3, input_shape = (64, 64, 3), activation = 'relu'))
    classifier.add(MaxPooling2D(pool_size = (2, 2)))
    classifier.add(Convolution2D(32, 3, 3, activation = 'relu'))
    classifier.add(MaxPooling2D(pool_size = (2, 2)))
```

```
classifier.add(Flatten())
classifier.add(Dense(output_dim = 256, activation = 'relu'))
classifier.add(Dense(output_dim = 108, activation = 'softmax'))
print(classifier.summary())
classifier.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])
hist = classifier.fit(X_train, Y_train, batch_size=16, epochs=60, shuffle=True, verbose=2)
classifier.save_weights('model/model_weights.h5')
model_json = classifier.to_json()
with open("model/model.json", "w") as json_file:
    json_file.write(model_json)
f = open('model/history.pckl', 'wb')
pickle.dump(hist.history, f)
f.close()
f = open('model/history.pckl', 'rb')
data = pickle.load(f)
f.close()
acc = data['accuracy']
accuracy = acc[9] * 100
print("Training Model Accuracy = "+str(accuracy))
```

CHAPTER 4
RESULTS AND DISCCUSION

CHAPTER 4
RESULTS AND DISCUSSION

4.1 Results

In this project to recognize person from IRIS we are using CASIA IRIS dataset which contains images from 108 peoples and by using this dataset we are training CNN model and then we can use this CNN model to predict/recognize persons. To train CNN model we are extracting IRIS features by using HoughCircles algorithm which extract IRIS circle from eye images. Below screen shots showing dataset with person id and this dataset saved inside ‘CASIA1’ folder

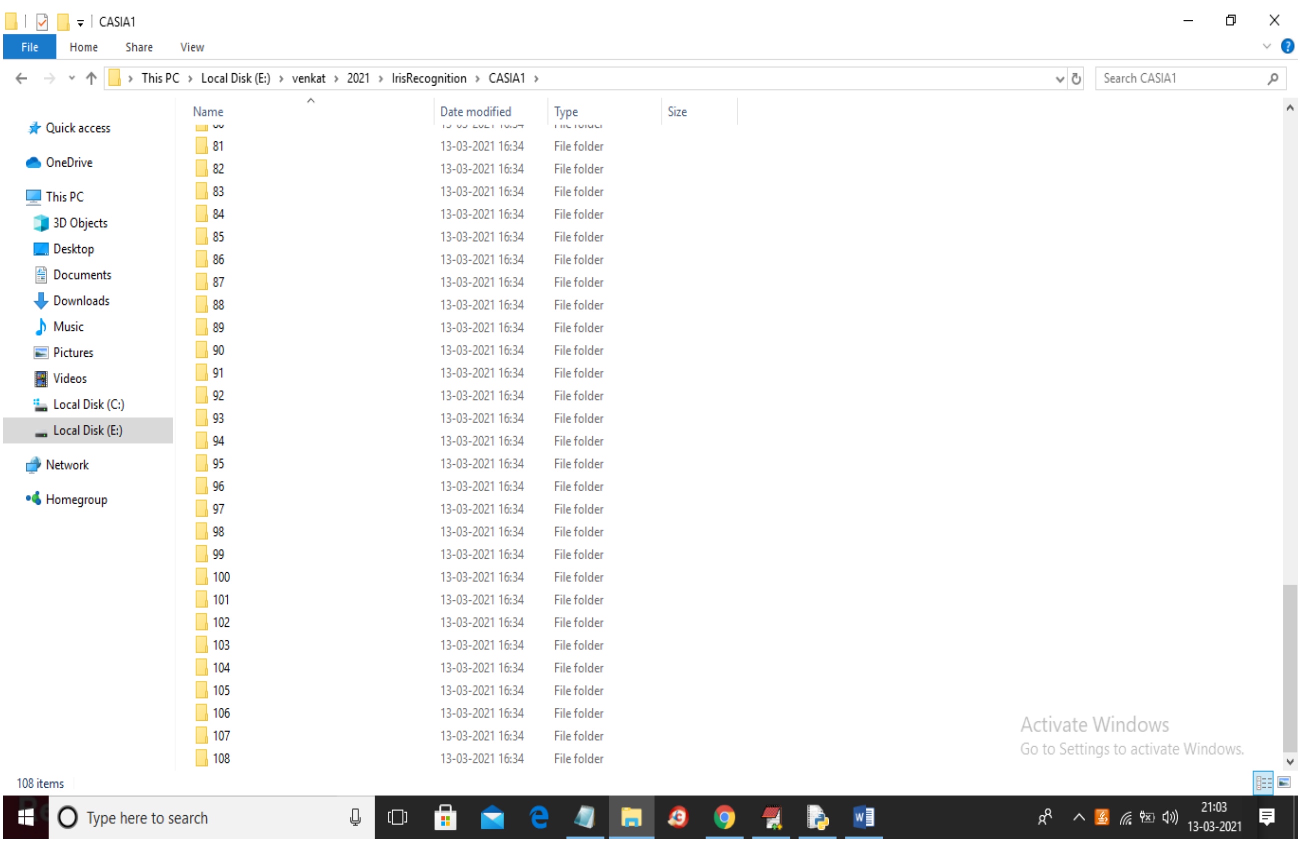


Fig 4.1.1 CASIA DataSet

In above screen we have IRIS images from 108 peoples and just go inside any folder to get that person IRIS images like below screen. And each sample have 7 images. And the total of 108 images will go under the training of CNN. And after the vagarious training completed.

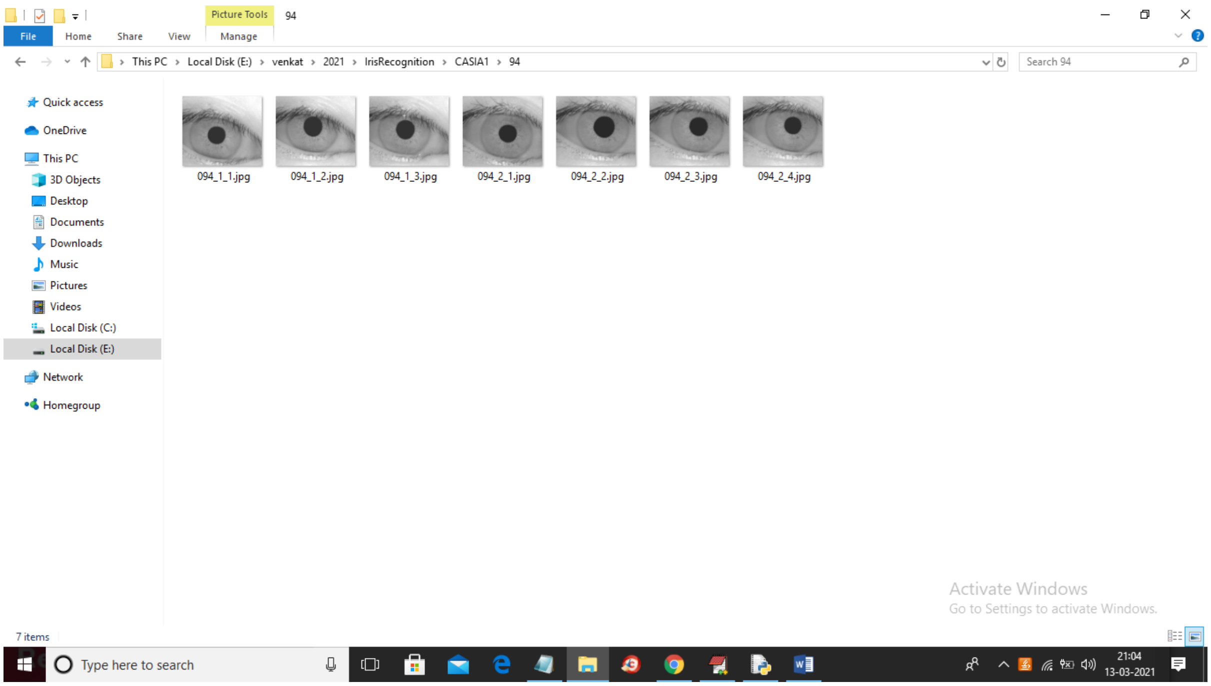


Fig 4.1.2 Irises Images

To run project double click on 'run.bat' file to get below screen

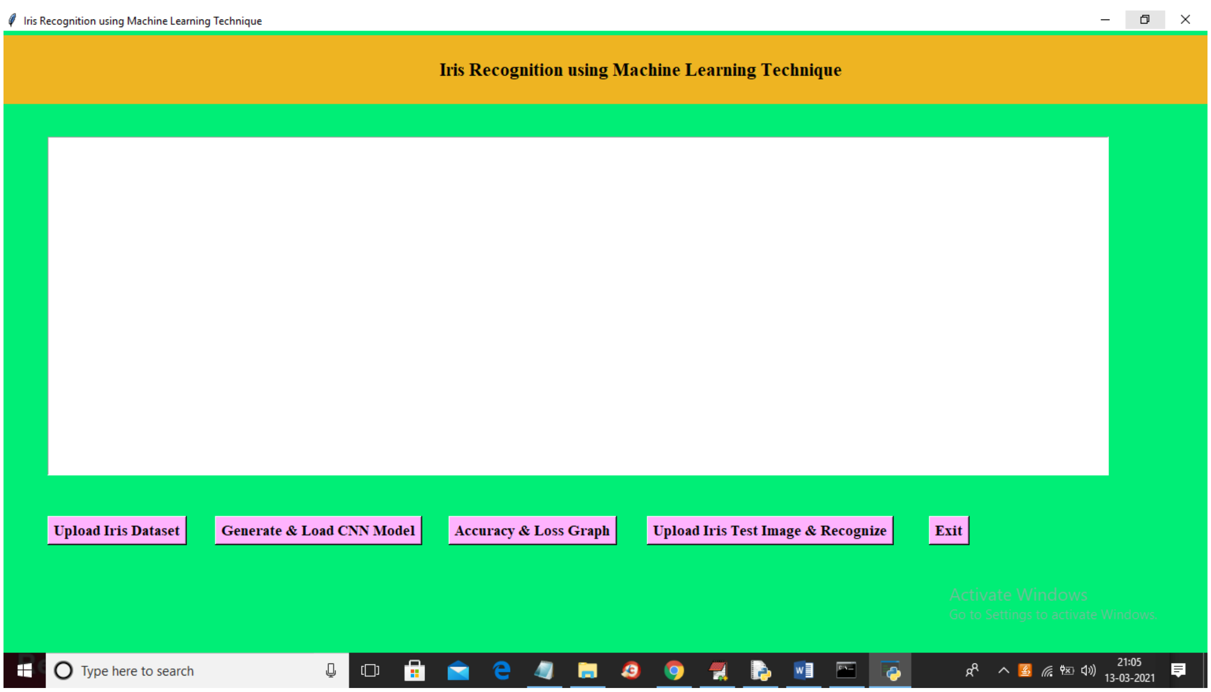


Fig 4.1.3 Interface

In above screen click on 'Upload Iris Dataset' button and upload dataset folder

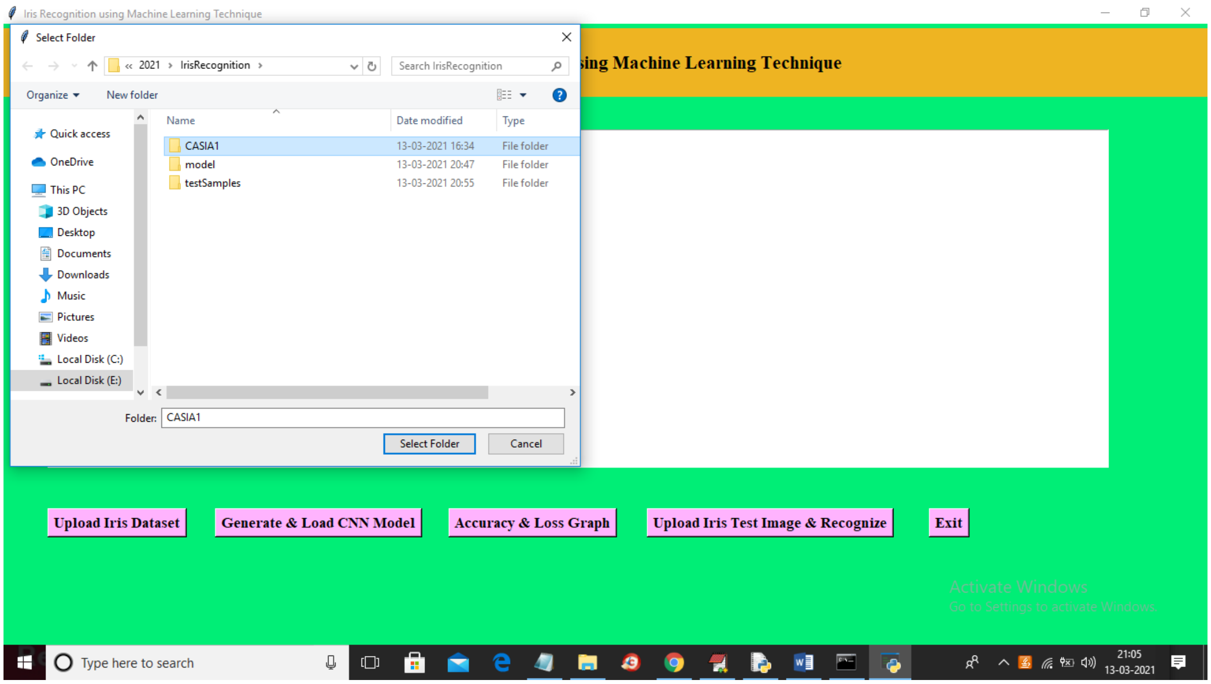


Fig 4.1.4 Dataset Loading

In above screen selecting and uploading 'CASIA1' folder and then click on 'Select Folder' button to load dataset and to get below screen

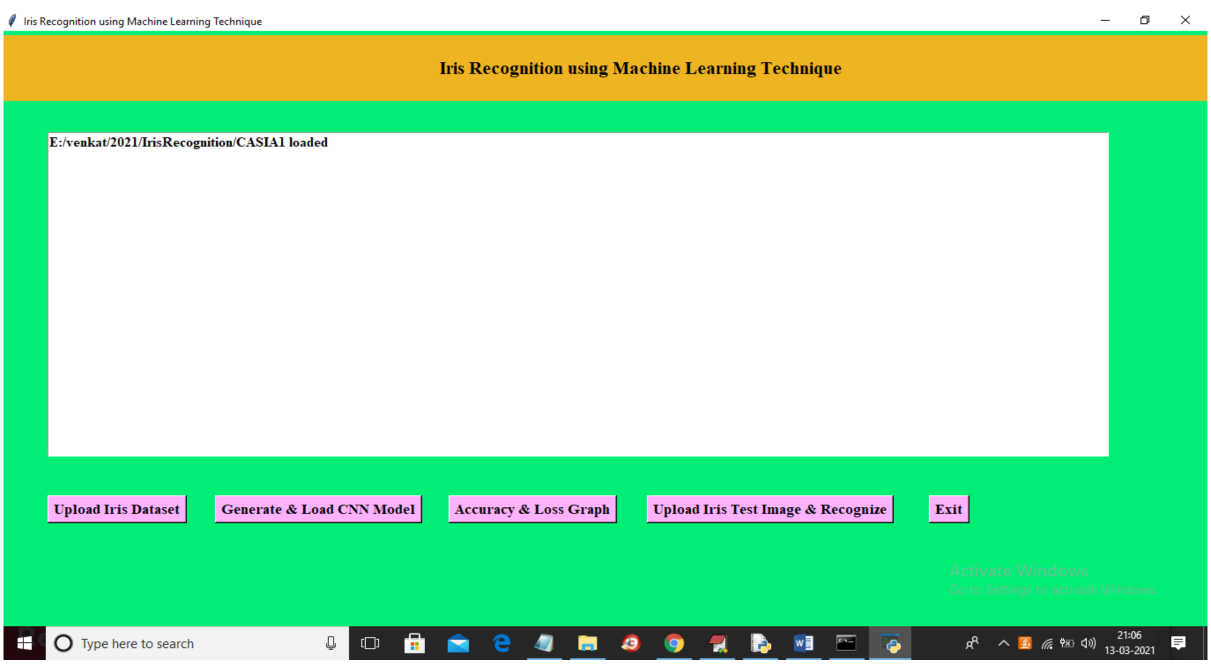


Fig 4.1.5 Data Set Loaded

In above screen dataset loaded and now click on 'Generate & Load CNN Model' button to generate CNN model from loaded dataset

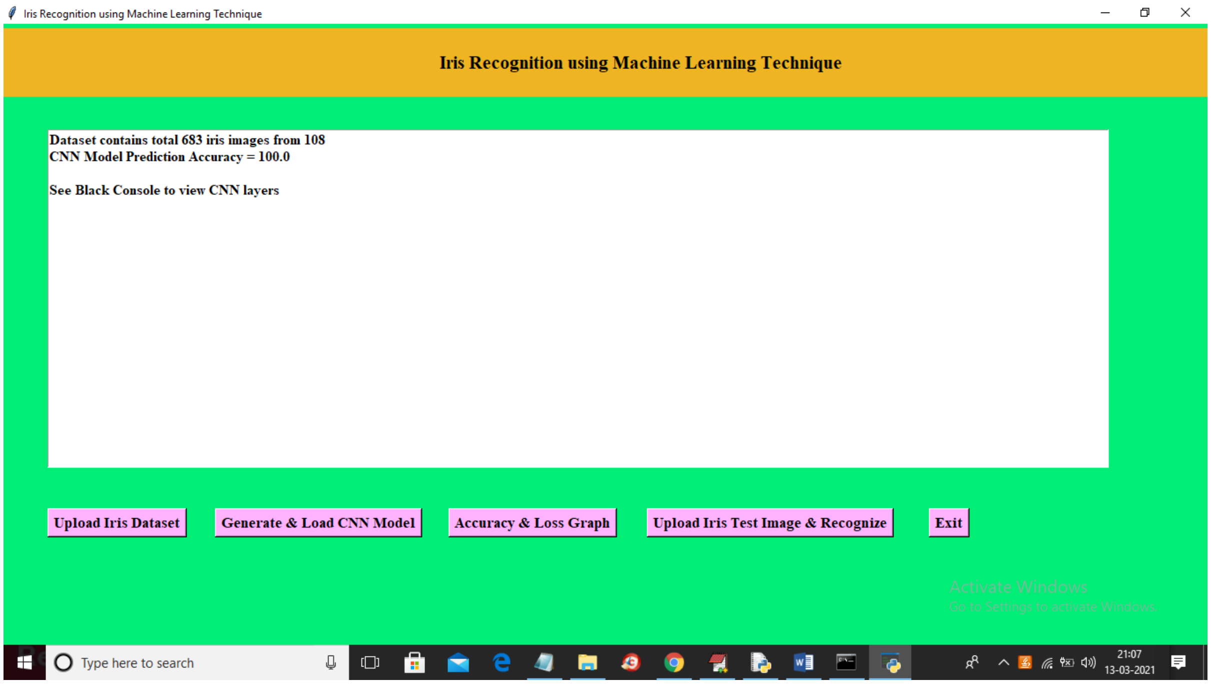


Fig 4.1.6 After Generating CNN

In above screen 683 images loaded from different 108 peoples and we got it prediction accuracy as 100%. Now model is ready and now click on 'Accuracy & Loss Graph' button to get below graph

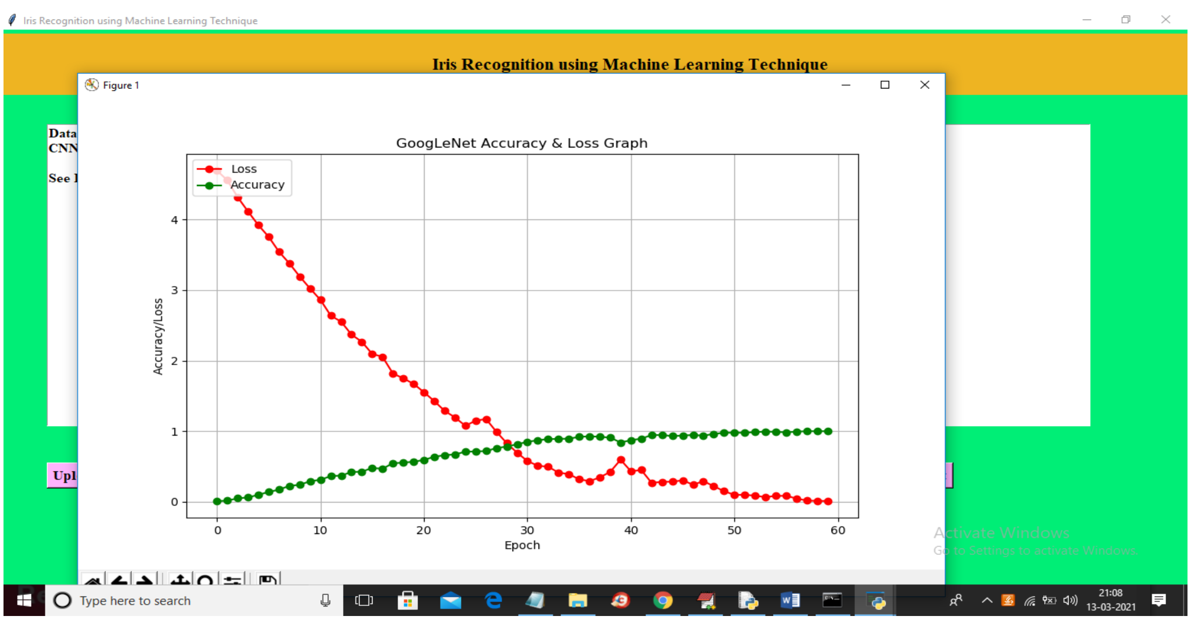


Fig 4.1.7 Graph

In above graph red line represents CNN model loss value and we can see at first iteration loss was more than 4% and when epoch increases then LOSS value reduces to 0

And green line represents accuracy and at first iteration accuracy was 0% and when epoch/iterations of model increases then accuracy reached to 100% and in above graph x-axis represents EPOCH and y-axis represents accuracy and loss values. Now click on ‘Upload Iris Test Image & Recognize’ button and upload any test image and then CNN will recognize person ID from that IRIS image. If you want you can upload test image from CASIA folder also and you will see prediction will be 100% correct

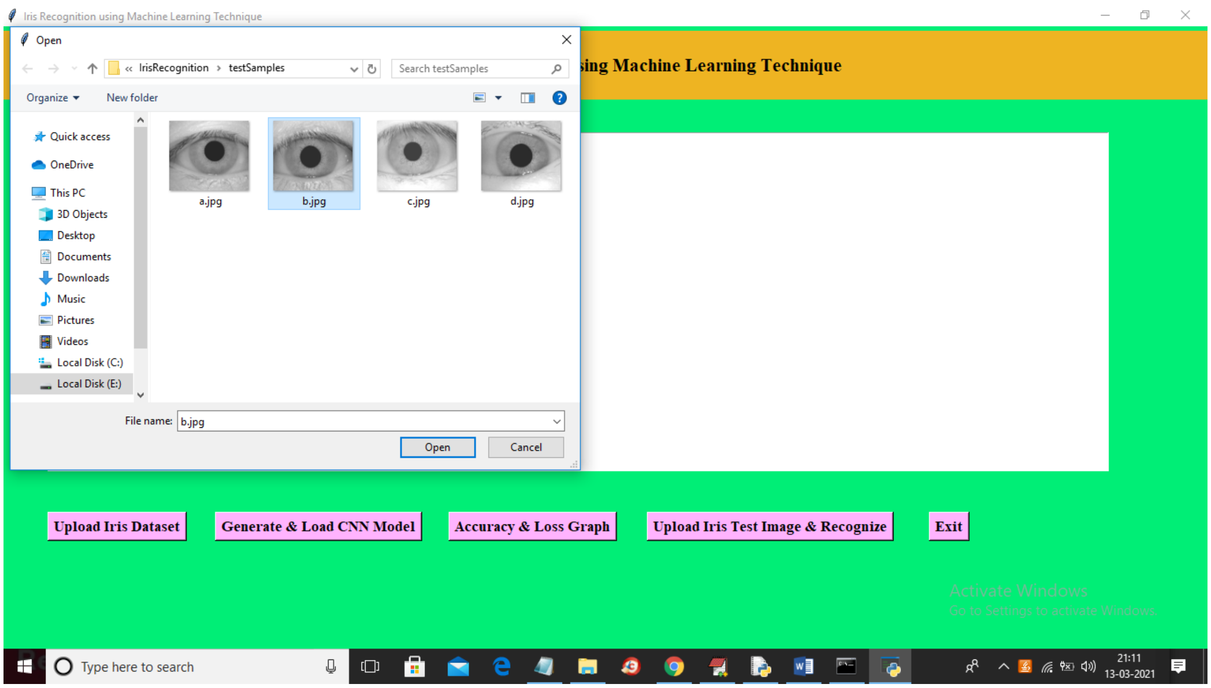


Fig 4.1.8 Uploading of Iris Test Image

In above screen selecting and uploading ‘b.jpg’ file and then click on ‘Open’ button to get below screen. Now click on ‘Upload Iris Test Image & Recognize’ button and upload any test image and then CNN will recognize person ID from that IRIS image. If you want you can upload test image from CASIA folder also and you will see prediction will be 100% correct.Now click on ‘Upload Iris Test Image & Recognize’ button and upload any test image and then CNN will recognize person ID from that IRIS image. If you want you can upload test image from CASIA folder also and you will see prediction will be 100% correct.

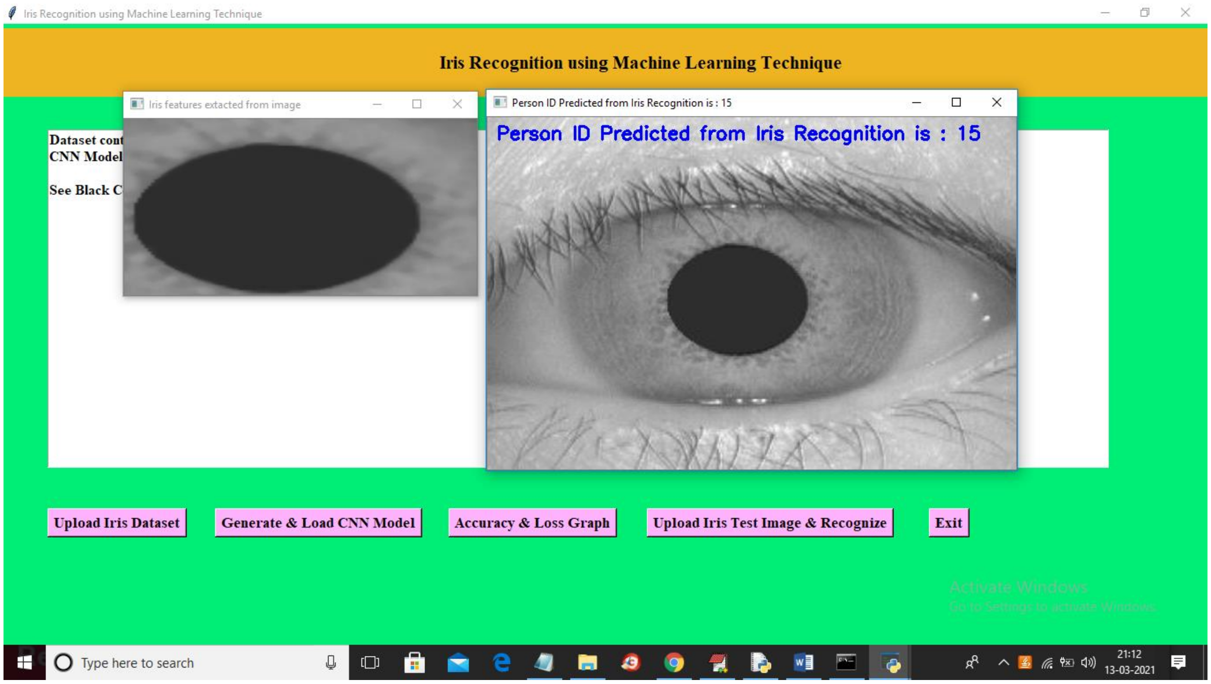


Fig 4.1.9 Extracting Iris Feature

In above screen from uploaded image we extract IRIS features which is displaying in first image and then this image feeds to CNN and then CNN predicted that IRIS belong to person ID 15. Now I will upload one image from CASIA folder and then test whether CNN will predict correctly or not

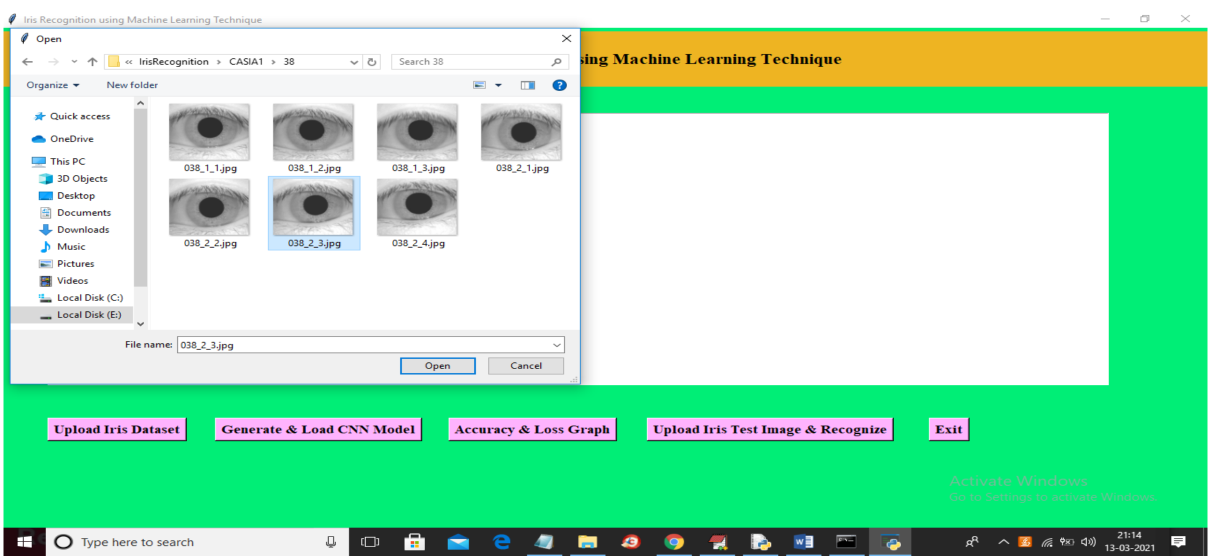


Fig 4.1.10 Uploading Iris

In above screen from CASIA folder I am uploading IRIS of person ID 38 and then click ‘Open’ button to get below result

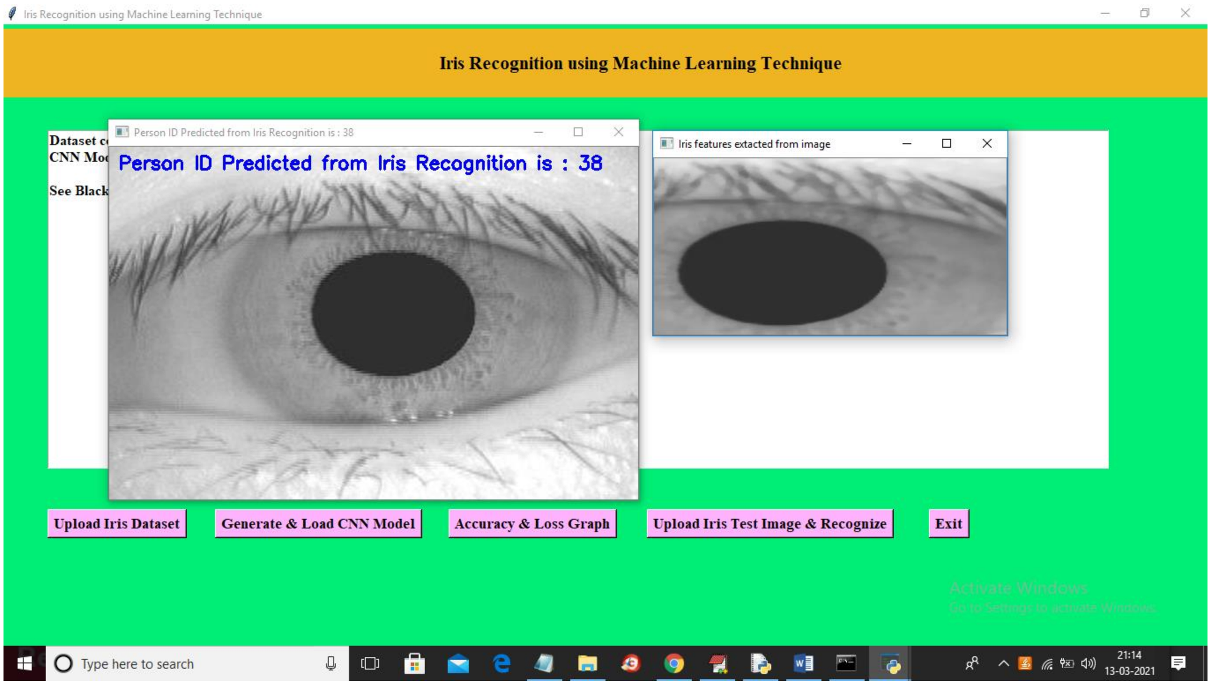


Fig 4.1.11 Output

In above screen CNN predicted ID is 38 which is 100% correct

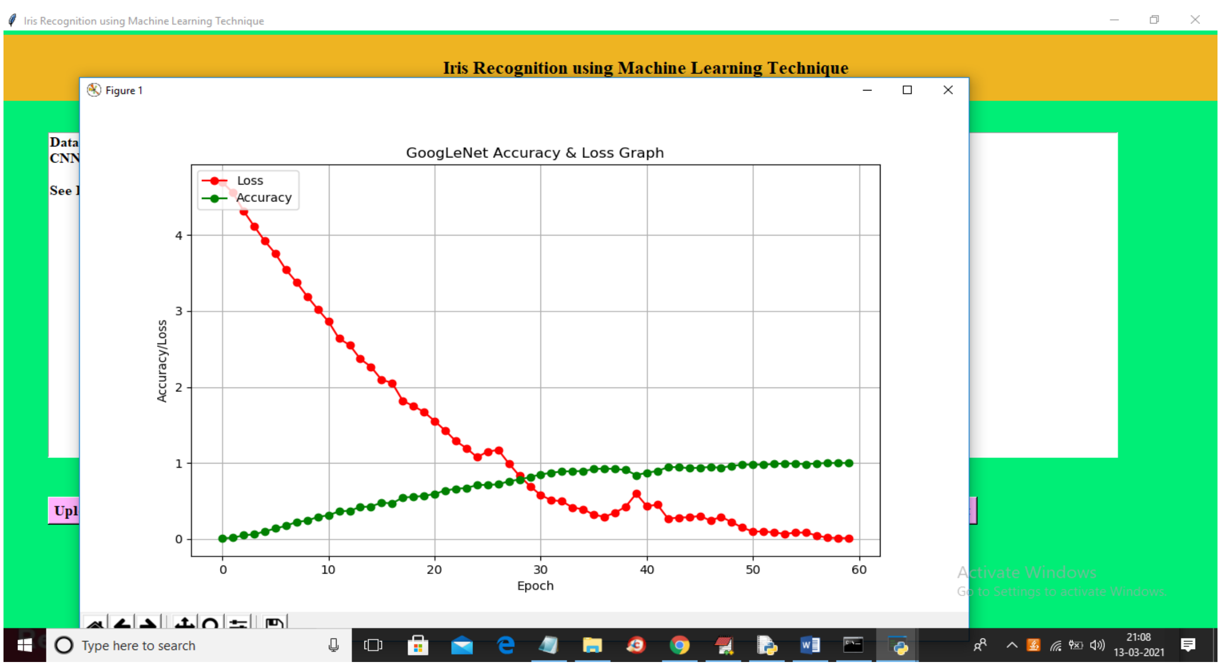


Fig 4.1.12 Accuracy and Loss Graph

In above graph red line represents CNN model loss value and we can see at first iteration loss was more than 4% and when epoch increases then LOSS value reduce to 0 and green line represents accuracy and at first iteration accuracy was 0% and when epoch/iterations of model increases then accuracy reached to 100% and in above graph x-axis represents EPOCH and y-axis represents accuracy and loss values. Now click on ‘Upload Iris Test Image & Recognize’ button and upload any test image and then CNN will recognize person ID from that IRIS image. If you want you can upload test image from CASIA folder also and you will see prediction will be 100% correct

4.1 Performance Matrix

Sl. No	**Name of project**	**Accuracy**
1	Face detection using viola jones algorithm	**94.5%**
2	Counterfeit iris detection based on texture analysis	**97.8%**
3	Iris recognition using machine learning technique	**100%**

Table 4.1.1 Performance Metrics

In the existing solutions we found that the human identity recognition is not accurate and the segmentation and normalization is also not perfect. So we have come up with the new solution and with that we have got 100% accuracy and high performance and efficiency. So we can sy that our model is is performing very well because we have used the popular and fastest algorithms and we have used CASIA dataset to achieve this.with this we also conclude that our system is trained with 108 samples of images and 756 images.we have achieved some of disadvantages from previous projects.

APPLICATIONS:

Applications in Security

- Iris recognition can be used in various security applications, such as access control and border security
- Iris recognition provides a high level of security and accuracy, making it ideal for applications where security is a top priority

  Iris recognition can also be used in combination with other biometric technologies, such as facial recognition and fingerprint scanning

Applications in Healthcare

- Iris recognition can be used in healthcare settings, such as patient identification and tracking medication dispensation
- Iris recognition provides a high level of accuracy and can help to reduce errors and improve patient safety
- Iris recognition can also be used to improve the efficiency of healthcare operations, such as reducing wait times and improving patient flow

Applications in Banking

- Iris recognition can be used in banking settings, such as verifying customer identities and preventing fraud
- Iris recognition provides a high level of security and accuracy, making it ideal for applications where security is a top priority
- Iris recognition can also be used to improve the efficiency of banking operations, such as reducing wait times and improving customer flow

CHAPTER 5
CONCLUSION

CHAPTER 5

CONCLUSION

Iris recognition is a promising field of security concern that uses human iris to identify. Each individual in a population is possible to identify by calculating the iris feature. The reason why iris recognition is an attractive field is due to the fact that iris feature cannot be forgotten or lost, they are difficult to copy, share and distribute and they require the person to be present at the time of authentication. However, the enhancement of accuracy mostly depends on feature extraction and classification techniques. So that feature extraction and classification are emphasized in this study. CNN and SVM are relatively new and good performing machine learning techniques for feature extraction and classification. We have chosen this technique and our experimental results have demonstrated that the proposed technique achieved good performance in accuracy. This confirms that the proposed strategy of feature extraction and classification is suitable for increasing accuracy of iris recognition.

Future Work

As the future work, we would like to simplify our algorithm for mobile devices (as smartphones) or embedded systems (e.g., based on ARM microcontrollers). On the other hand, we would like to precisely test some other classification algorithms as deep convolutional recursive neural networks. However, to deal with such task we have to enlarge our database. The authors are working under collection of much more samples—at least 1000 of additional images have to be added to the database. In the future, we will also try to implement multimodal biometrics system with our iris algorithm. This experiment will provide us an evidence whether the multimodal solution can guarantee better recognition in short time (or comparable to the time needed in the case of iris).

CHAPTER 6
REFERENCES

CHAPTER 6
REFERENCES

1. Sabhanayagam, T., Venkatesan, V. P. & Senthamaraikannan, K. (2018). A comprehensive survey on various biometric systems. Int. J. App. Eng. Res., 13, 2276– 2297.

2. Masek, L. (2003). Recognition of human iris patterns for biometric identification. BEng diss., University of Western Australia.

3. Cruz, F. R. G., Hortinela, C. C., Redosendo, B. E., Asuncion, B. K. P., Leoncio, C. J. S., Linsangan, N. B. & Chung, W. -Y. (2016). Iris recognition using Daugman algorithm on Raspberry Pi. 2016 IEEE region 10 conference (TENCON). New Jersey, USA: Institute of Electrical and Electronics (IEEE), 2126–2129, https://doi.org/10.1109/TENCON.2016.7848401.

4. Kunik, Z., Bykowski, A., Marciniak, T. & Dąbrowski, A. (2017). Raspberry Pi based complete embedded system for iris recognition. 2017 signal processing: algorithms, architectures, arrangements, and applications (SPA). New Jersey, USA: IEEE, 263–268, https://doi.org/10.23919/SPA.2017.8166876

5. Wang, Z., Li, C., Shao, H. & Sun, J. (2018). Eye recognition with mixed convolutional and residual network (MiCoRe-Net). IEEE Access, 6, 17905–17912. https://doi.org/10.1109/ACCESS.2018.2812208.

6. Klein, A., Falkner, S., Bartels, S., Hennig, P. & Hutter, F. (2016). Fast bayesian optimization of machine learning hyperparameters on large datasets. arXiv preprint arXiv:1605.07079.

7. Bazrafkan, S., Thavalengal, S. & Corcoran, P. (2018). An end to end Deep Neural Network for iris segmentation in unconstrained scenarios. Neural Networks, 106, 79–95

8.https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8197830/

9.https://www.mygreatlearning.com/blog/viola-jones-algorithm/

10.https://ieeexplore.ieee.org/document/9265389

11. Daugman, J.G.: High confidence visual recognition of personals by a test of statistical independence. IEEE Trans. PAMI 15(11), 1148–1160 (1993)

GitHub Link

https://upendra0033.github.io/Iris-recognitio/