FACE DETECTION AND RECOGNITION FOR CRIMINAL IDENTIFICATION SYSTEM

A PROJECT REPORT

Submitted by

SRIDHAYAA A S	[211419104324]
ISWARYA G	[211419104501]
RAMYA P	[211419104322]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2023

BONAFIDE CERTIFICATE

Certified that this project report **"FACE DETECTION AND RECOGNITION FOR CRIMINAL IDENTIFICATION SYSTEM"** is the bonafide work of **"SRIDHAYAA A S (211419104324), ISWARYA G (211419104501), RAMYA P (211419104322)"** who carried out the project work under my supervision.

SIGNATURE

Dr.L.JABASHEELA,M.E.,Ph.D.,
HEAD OF THE DEPARTMENT

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

SIGNATURE

Mrs.V.SATHYA PREIYA,M.E.,(Ph.D.)
ASSOCIATE PROFESSOR
SUPERVISOR

DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NASARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the End Semester Project Viva-Voce Examination held on..........................

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We **SRIDHAYAA A S (211419104324), ISWARYA G (211419104501), RAMYA P (211419104322)** hereby declare that this project report titled **“FACE DETECTION AND RECOGNITION FOR CRIMINAL IDENTIFICATION SYSTEM”**, under the guidance of **Mrs.V.SATHYA PREIYA M.E.,(Ph.D.)** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

SRIDHAYAA A S
ISWARYA G
RAMYA P

ACKNOWLEDGEMENT

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

We express our sincere thanks to our beloved Directors **Tmt.C.VIJAYARAJESWARI**, **Dr.C.SAKTHI KUMAR,M.E.,Ph.D** and **Dr.SARANYASREE SAKTHI KUMAR B.E.,M.B.A.,Ph.D.,** for providing us with the necessary facilities to undertake this project.

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

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

We would like to thank my Project Guide **Mrs.V.SATHYA PREIYA,M.E.,(Ph.D.)** and coordinator **Dr.K.VALARMATHI, M.E., Ph.D.** and all the faculty members of the Department of CSE for their advice and encouragement for the successful completion of the project.

SRIDHAYAA A S

ISWARYA G

RAMYA P

ABSTRACT

The process of identifying and recognizing the criminal is the time consuming and difficult task. There are several ways to identify culprits at the crime site, including fingerprinting, DNA matching, and eyewitness testimony. The criminal face identification system will be built on a existing criminal database. The method for identifying a human face using features extrapolated from an image is presented in this study. This paper presents a methodology for recognizing the human face based on the features derived from the image. As the human face is a complex multidimensional visual representation, it is extremely challenging to create a computational model for identifying it. The video captured by the camera will be translated into frames as part of the suggested process. We proposed an improved texture classification algorithm local binary pattern (LBP)with histograms of oriented gradient HOG descriptor to improve detection accuracy. When a face is found in a frame, it is preprocessed to remove redundant information and minimize noise. The real-time processed image is compared to the trained images that have previously been saved in the database. The technology will send an automatic email notice to the police officials if the surveillance camera detects a criminal.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
DNA	Deoxyribonucleic Acid
NCRB	National Crime Records Bureau
LBPH	Local Binary Path Histogram
CNN	Convolutional Neural Network
OpenCV	Open-Source Computer Vision
UML	Unified Modeling Language
CCTV	Closed-Circuit Television

CHAPTER 1

INTRODUCTION

1.INTRODUCTION

1.1 OVERVIEW

Finding a thief has been a difficult procedure over the years. Formerly, the entire process was dependent solely on leads gleaned from evidence found at the scene of the crime and the search for biological proof is challenging. The ability to hide their tracks and leave no traces of trackable evidence, however, has improved and crooks are wiser than ever. Facial detection and awareness are relevant in this situation. The face is a terrific tool for identifying people, and each one is distinctive since it can be easily distinguished. One unique biometric technology, face recognition for criminal identification, has the advantages of high accuracy and minimal intrusion. It is a method that uses facial recognition to automatically confirm someone's identification in video or image frames. The face recognition tool described in this paper combines the best face detection, characteristic extraction, and classification techniques currently available. The algorithm extracts significant important features and converts them into pixels to create a data vector as part of an approach to improve the accuracy of recognition. These ideas are useful for categorization and performing database fits. This system uses the two important approaches of detection and identification. Two crucial strategies are triggered by face attention: training and evaluation. The facial recognition system's assessment section compares the recently acquired check photo with the database that already exists. If the data is matched with the database a email is triggered to the concern officials.

1.2 PROBLEM DEFINITION

This system is aimed to identify the criminals in any investigation department. In this system, we are storing the images of criminals in our database along with his details and then these images are segmented into four slices- forehead, eyes, nose, and lips. These images are again stored in another database record to make the identification process easier. Eyewitnesses will select the slices that appear on the screen and by using it we retrieve the image of the face from the database. Thus this system provides a very friendly environment for both the operator and the eyewitness to easily identify the criminal, if the criminals record exists in the database. This project is intended to identify a person using the images previously taken.

CHAPTER 2
LITERATURE SURVEY

2. LITERATURE SURVEY

2.1 CRIMINAL FACE DETECTION

Author Name : Prajyot Rupanvar, Vishvajit Kale

Year of Publish : 2022

Criminal Face Detection project aims to build an automated Criminal Face Detection system by levering the human ability to recall minute facial details. Identification of criminals at the scene of a crime can be achieved in many ways like fingerprinting, DNA matching or eye witness accounts. Out of these methods eye witness accounts are preferred because it stands scrutiny in court and it is a cost-effective method. It is possible that witnesses to a crime have seen the criminal though in most cases it may not be possible to completely see the face of the perpetrator. The Criminal Face Detection System will be built of an existing criminal database. Input would be provided in the form of sketch or an image and matched against the existing database and results would be provided. Criminal record generally contains personal information about particular person along with photograph. To identify any Criminal we need some identification regarding person, which are given by eyewitness. The human face is a complicated multidimensional visual model and hence it is very difficult to develop a computational model for recognizing it. The paper presents a methodology for recognizing the human face based on the features derived from the image. The proposed methodology is implemented in two stages. The first stage detects the human face in an image using viola Jones algorithm. In the next stage the detected face in the image is recognized using a fusion of principle.

2.2 CRIMINAL IDENTIFICATION SYSTEM USING FACE DETCTION AND RECOGNITION

Author Name : Rohit Alex Badana, Lohith Morishetty

Year of Publish : 2022

Identifying and Recognizing a criminal is a time-consuming and challenging task. According to the survey of NCRB (National Crime Records Bureau) , 80%of the same criminals do the same crimes repetitively. Criminals are becoming smarter by not leaving any biological evidence or fingerprint impressions at the crime site. The face is a unique and crucial aspect of the human body structure that recognizes a person. This Face recognition from an image may be used to identify criminals or a video frame captured by the cameras that are installed in multiple regions. As a result, we may utilize it to track down a criminal's identification. Face recognition is a biometric based technique that mathematically maps an individual's facial traits and retains the data as a face print. It generates a unique pattern for each face and compares it to other images that are included in the collection. If a match is identified for the input face, the details linked with the relevant image will be displayed. This approach will reduce crime and protect public safety.

2.3 CRIMNINAL IDENTIFICATION SYSTEM USING DEEP LEARNING

Author Name **:** D.Nagamallika , P.Vandana

Year of Publish **:** 2021

In this paper, we have developed a system for detecting criminal faces, for this, we have used deep learning algorithms. Since deep learning is now the most famous technology, it is used in different applications. One such application is crime detection and prevention. This system identifies the criminal face, retrieves the information stored in the database for the identified criminal and a notification is sent to the police personnel with all the details and the location at which the criminal was under the surveillance of the camera.

2.4 ONLINE CRIMINAL IDENTIFICATION USING ML & FACE RECOGNITION TECHNIQUES

Author Name **:** GANTA TEJASWINI , KESAVARAO SEERAPU

**Year of Publish : **2021

In current days identifying criminals is becoming very complicated task for the cybercrime people because it is having a lot of factors need to be analyzed. Criminal record generally contains personal information about particular person along with photograph. To identify any criminal we need some identification regarding person, which are given by eyewitnesses. In most cases the quality and resolution of the recorded image-segments is poor and hard to identify a face. To overcome this sort of problem we are developing many software's by using recent trends to identify the criminals but no method is accurate in identifying the criminal information accurately. Some of the best methods are iris, eyes, biometric, thumb, face and lot more. One of the applications is face identification. The face is our primary focus of attention in social inter course playing a major role in conveying identity and emotion. Although the ability to infer intelligence or character from facial appearance is suspect, the human ability to recognize faces is remarkable. The operator first logs into the system by entering username and password. Then depending on the work allotted he has to select the screens from main menu screen. There are mainly three important function which he can do they are adding details, clipping image and finally construction of the face by using the eyewitness. The face that is finally formed is one the who has done the crime.

2.5 FACE RECOGNITION AND IDENTIFICATION USING DEEP LEARNING APPROACH

**Author Name : **KH Teoh, RC Ismail, SZM Naziri

**Year of Publish : **2021

Human face is the significant characteristic to identify a person. Everyone has their own unique face even for twins. Thus, a face recognition and identification are required to distinguish each other. A face recognition system is the verification system to find a person's identity through biometric method. Face recognition has become a popular method nowadays in many applications such as phone unlock system, criminal identification and even home security system. This system is more secure as it does not need any dependencies such as key and card but only facial image is needed. Generally, human recognition system involves 2 phases which are face detection and face identification. This paper describes the concept on how to design and develop a face recognition system through deep learning using OpenCV in python. Deep learning is an approach to perform the face recognition and seems to be an adequate method to carry out face recognition due to its high accuracy. Experimental results are provided to demonstrate the accuracy of the proposed face recognition system.

2.6 CRIMINAL IDENTIFICATION SYSTEM USING FACIAL RECOGNITION

Author Name **:** Nagnath B. Aherwadi, Deep Chokshi

Year of Publish **:** 2021

We all know that our Face is a unique and crucial part of the human body structure that identifies a person. Therefore, we can use it to trace the identity of a criminal person. With the advancement in technology, we are placed CCTV at many public places to capture the criminal's crime. Using the previously captured faces and criminal's images that are available in the police station, the criminal face recognition system of can be implemented. In this paper, we propose an automatic criminal identification system for Police Department to enhance and upgrade the criminal distinguishing into a more effective and efficient approach. Using technology, this idea will add plus point in the current system while bringing criminals spotting to a whole new level by automating tasks. Technology working behind it will be face recognition, from the footage captured by the CCTV cameras; our system will detect the face and recognize the criminal who is coming to that public place. The captured images of the person coming to that public place get compared with the criminal data we have in our database. If any person’s face from public place matches, the system will display their image on the system screen and will give the message with their name that the criminal is found and present in this public place. This system matching more than 80% of the captured images with database images.

2.7 CRIMINAL IDENTIFICATION BY USING REALTIME IMAGE PROCESSING

Author Name **:** Tiwari Aanchaladevi S, Ghotekar Shubhangi S

Year of Publish **:** 2021

The main goal of this paper is to help in real time for face recognition by using automated face surveillance camera. The proposed system has of 4 steps, it include training of real time images, face detection using Haar based classifier, comparison of trained real time images with images from the surveillance, camera result based on the comparison between them. Main application of interest is automated surveillance, the aim of automated surveillance to acknowledge people around watch list. The main goal of this paper is to compare an image with several images, which is already trained. In this paper, we represent a methodology for face detection strongly in real time environment. Haar cascading is one of the algorithm for face detection. In that we use Haar like classifiers to track faces on OpenCV platform. The correctness of the face recognition is very high. The proposed system can successfully recognize more faces which is useful for quickly searching suspected persons, as the computation time is very low. In India, we have a system to accept citizen called Aadhaar. If we use this as a citizenship database then we can differentiate between citizen and foreigner as well as we will be able to investigate whether the identified person is criminal or not.

2.8 CRIMINAL IDENTIFICATION FOR LOW RESOLUTION SURVEILLANCE

**Author Name : ** Saniya Prashant Patil, Grishma Sunil Yadav

**Year of Publish : ** 2021

A Criminal Identification System allows the user to identify a certain criminal based on their biometrics. With advancements in security technology, CCTV cameras have been installed in many public and private areas to provide surveillance activities. The CCTV footage becomes crucial for understanding of the criminal activities that take place and to detect suspects. Additionally when a criminal is found it is difficult to locate and track him with just his image if he is on the run. Currently this procedure consists of finding such people in CCTV surveillance footage manually which is time consuming. It is also a tedious process as the resolution for such CCTV cameras is quite low. As a solution to these issues, the proposed system is developed to go through real time surveillance footage, detect and recognize the criminals based on reference datasets of criminals. The use of facial recognition for identifying criminals proves to be beneficial. Once the best match is found the real time cropped image of the recognized criminal is saved which can be accessed by authorized officials for locating and tracking criminals or for further investigative use.

2.9 AUTOMATED CRIMINAL IDENTIFICATION SYSTEM USING FACE DETECTION AND RECOGNITION

**Author Name : ** Piyush Chhoriya

**Year of Publish : ** 2019

As the world has seen exponential advancement over the last decade, there is an abnormal increase in the crime rate and also the number of criminals are increasing at an alarming rate, this leads toward a great concern about the security issues. various causes of theft, stealing crimes, burglary, kidnapping, human trafficking etc. are left unsolved because the availability of police personnel is limited, many times there is no identification of the person who was involved in criminal activities. To avoid this situation an automated facial recognition system for criminal identification is proposed using Haar feature-based cascade classifier. This paper presents a real-time face recognition using an automated surveillance camera. This system will be able to detect and recognize face automatically in real-time

2.10 CRIMINAL IDENTIFICATION USING FACIAL RECOGNITION

**Author Name : **Archana Naik, Rohan Basukala

**Year of Publish : **2019

The individualistic characters of the human face can be extracted by face recognition. The human face detection and recognition finds a major role in the application as video surveillance, face image database management. Face recognition is a simple and agile biometric technology. This technology uses the most obvious human identifier to the face. The face recognition finds its application in security, health care, criminal identification, places where human recognition is the necessity. With the advancement in technology, the extracting features of the human face are become simpler. This paper discusses on a different algorithm to recognize the human face. The purpose is to identify the criminal face and retrieve the information stored in the database for the identified criminal. The process is categorized into two major steps. First, the face is extracted from the image, distinguishing factors in the face are extracted and stored in the database. The second step is to compare the resultant image with the existing image and return the data related to that image from the database.

2.11 CRIME DETECTION USING TEXT RECOGNITION AND FACE RECOGNITION

**Author Name : **Shivam Bachhety, Ramneek Singhal

**Year of Publish : **2018

With rapid growth and development in big towns and cities, the crime is also increasing at an alarming rate. Various cases of theft, stealing crimes, burglary, kidnapping, human trafficking etc. are left unsolved because the vehicle involved could not be identified accurately as it is not feasible for human eye to verify characters from license plate of fast moving vehicles. Many a times there is no identification of the person who was involved in criminal activities. To avoid such situation, we have proposed an effective crime detection system using Text and Face recognition technique. Such systems will be very effective at toll tax collection, parking system, airports, and Border crossings. Text Recognition can be used to identify number plate and face recognition can be used for criminal identification. The efficiency of the system depends on the quality, illumination and view of the image captured. This paper aims to provide better result both in terms of speed and accuracy than conventional methods in use relating to the text and face recognition process for criminal identification.

2.12 CRIMINAL IDENTIFICATION SYSTEM USING FACE DETECTION AND RECOGNITION

Author Name **:** Piyush Kakkar, Mr. Vibhor Sharma

Year of Publish **:** 2018

There is an abnormal increase in the crime rate and also the number of criminals is increasing, this leads towards a great concern about the security issues. Crime preventions and criminal identification are the primary issues before the police personnel, since property and lives protection are the basic concerns of the police but to combat the crime, the availability of police personnel is limited. With the advent of security technology, cameras especially CCTV have been installed in many public and private areas to provide surveillance activities. The footage of the CCTV can be used to identify suspects on scene. In this paper, an automated facial recognition system for criminal database was proposed using known Haar feature-based cascade classifier. This system will be able to detect face and recognize face automatically in real time. An accurate location of the face is still a challenging task. Viola-Jones framework has been widely used by researchers in order to detect the location of faces and objects in a given image. Face detection classifiers are shared by public communities, such as OpenCV.

2.13 FACE RECOGNITION FOR CRIMINAL IDENTIFICATION: AN IMPLEMENTATION OF PRINCIPAL COMPONENT ANALYSIS FOR FACE RECOGNITION

Author Name **:** Nurul Azma Abdullaha, Md. Jamri Saidi

Year of Publish **:** 2017

In practice, identification of criminal in Malaysia is done through thumbprint identification. However, this type of identification is constrained as most of criminal nowadays getting cleverer not to leave their thumbprint on the scene. With the advent of security technology, cameras especially CCTV have been installed in many public and private areas to provide surveillance activities. The footage of the CCTV can be used to identify suspects on scene. However, because of limited software developed to automatically detect the similarity between photo in the footage and recorded photo of criminals, the law enforce thumbprint identification. In this paper, an automated facial recognition system for criminal database was proposed using known Principal Component Analysis approach. This system will be able to detect face and recognize face automatically. This will help the law enforcements to detect or recognize suspect of the case if no thumbprint present on the scene. The results show that about 80% of input photo can be matched with the template data.

2.14 CRIMINAL FACE RECOGNITION SYSTEM

Author Name **:** Alireza Chevelwalla , Ajay Gurav

Year of Publish **:** 2015

Face recognition is one of the most challenging topics in computer vision today. It has applications ranging from security and surveillance to entertainment websites Face recognition software are useful in banks, airports, and other institutions for screening customers. Germany and Australia have deployed face recognition at borders and customs for Automatic Passport Control. Human face is a dynamic object having high degree of variability in its appearance which makes face recognition a difficult problem in computer vision. In this field, accuracy and speed of identification is a main issue. Many challenges exist for face recognition. The robustness of the system can be obstructed by humans who alter their facial features through wearing colored contact lenses, growing a mustache, putting on intense make-up, etc. Ethical concerns are also related to the process of recording, studying, and recognizing faces. Many individuals do not approve of surveillance systems which take numerous photographs of people who have not authorized this action. The goal of this paper is to evaluate face detection and recognition techniques and provide a complete solution for image based face detection and recognition with higher accuracy, better response rate and an initial step for video surveillance. Solution is proposed based on performed tests on various face rich databases in terms of subjects, pose, emotions and light

2.15 JOINT FACE DETECTION AND ALIGNMENT USING MULTI-TASK CASCADED CONVOLUTIONAL NETWORKS

Author Name : Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li

Year of Publish : 2015

Face detection and alignment in unconstrained environment are challenging due to various poses, illuminations and occlusions. Recent studies show that deep learning approaches can achieve impressive performance on these two tasks. In this paper, we propose a deep cascaded multi-task framework which exploits the inherent correlation between them to boost up their performance. In particular, our framework adopts a cascaded structure with three stages of carefully designed deep convolutional networks that predict face and landmark location in a coarse-to-fine manner. In addition, in the learning process, we propose a new online hard sample mining strategy that can im-prove the performance automatically without manual sample selection. Our method achieves superior accuracy over the state-of-the-art techniques on the challenging FDDB and WIDER FACE benchmark for face detection, and AFLW benchmark for face alignment, while keeps real time performance.

CHAPTER 3
SYSTEM ANALYSIS

3. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

The Existing System is single face recognition system and a new face detection approach using color base segmentation and morphological operations is presented. The algorithm uses color plane extraction, background subtraction, thresholding, morphological operations (such as erosion and dilation), filtering (to avoid false detection). Then particle analysis is done to detect only the face area in the image and not the other parts of the body. This method given result is poor performance and accuracy.

3.2 PROPOSED SYSTEM

The proposed system comprises of 4 phases, including training of real time photos, multiple face detection, comparison of learned real time images with images from the surveillance camera, outcome based on the comparison. In our suggested system, the video collected from the camera will be translated into frames. When a face is found in a frame, it is preprocessed to remove redundant information and minimize noise. The processed real time image is compared with the processed photos already recorded in the database. The technology will send an automatic email notice to the police officials if the surveillance camera detects a criminal.

3.3 HARDWARE ENVIRONMENT

- Hard Disk : 500GB and Above
- RAM : 4GB and Above
- Processor : I3 and Above
- Webcam – 1

3.4 SOFTWARE ENVIRONMENT

Operating System	: Windows 10 (64 bit)
Software	: Python
Tools	: Anaconda

3.5 TECHNOLOGIES USED

- Python
- Machine Learning

3.5.1 Python

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently. It is used for:

- web development (server-side),
- software development,
- mathematics,
- System scripting.

Python is Interpreted

- Many languages are compiled, meaning the source code you create needs to be translated into machine code, the language of your computer's processor, before it can be run. Programs written in an interpreted language are passed straight to an interpreter that runs them directly.
- This makes for a quicker development cycle because you just type in your code and run it, without the intermediate compilation step.

- One potential downside to interpreted languages is execution speed. Programs that are compiled into the native language of the computer processor tend to run more quickly than interpreted programs. For some applications that are particularly computationally intensive, like graphics processing or intense number crunching, this can be limiting.
- In practice, however, for most programs, the difference in execution speed is measured in milliseconds, or seconds at most, and not appreciably noticeable to a human user. The expediency of coding in an interpreted language is typically worth it for most applications.
- For all its syntactical simplicity, Python supports most constructs that would be expected in a very high-level language, including complex dynamic data types, structured and functional programming, and object-oriented programming.
- Additionally, a very extensive library of classes and functions is available that provides capability well beyond what is built into the language, such as database manipulation or GUI programming.
- Python accomplishes what many programming languages don't: the language itself is simply designed, but it is very versatile in terms of what you can accomplish with it.

3.5.2 Machine Learning

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task.

Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task. Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through learning. In its application across business problems, machine learning is also referred to as predictive analytics.

Machine learning tasks

Machine learning tasks are classified into several broad categories. In supervised learning, the algorithm builds a mathematical model from a set of data that contains both the inputs and the desired outputs. For example, if the task were determining whether an image contained a certain object, the training data for a supervised learning algorithm would include images with and without that object (the input), and each image would have a label (the output) designating whether it contained the object. In special cases, the input may be only partially available, or restricted to special feedback. Semi algorithms develop mathematical models from incomplete training data, where a portion of the sample input doesn't have labels. Classification algorithms and regression algorithms are types of supervised learning. Classification algorithms are used when the outputs are restricted to a limited set of values. For a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email. For an algorithm that identifies spam emails, the output would be the prediction of either "spam" or "not spam", represented by the Boolean values true and false.

CHAPTER 4

SYSTEM DESIGN

4. SYSTEM DESIGN

4.1 UML DIAGRAMS

UML is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML was created by the Object Management Group (OMG) and UML 1.0 specification draft was proposed to the OMG in January 1997.OMG is continuously making efforts to create a truly industry standard. UML stands for Unified Modeling Language. UML is different from the other common programming languages such as C++, Java, COBOL, etc. UML is a pictorial language used to make software blueprints. UML can be described as a general-purpose visual modeling language to visualize, specify, construct, and document software system. Although UML is generally used to model software systems, it is not limited within this boundary. It is also used to model non-software systems as well. For example, the process flow in a manufacturing unit, etc. UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation with object-oriented analysis and design. After some standardization, UML has become an OMG standard.

4.1.1 USE CASE DIAGRAM

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases. A use case diagram doesn't go into a lot of detail—for example, don't expect it to model the order in which steps are performed. Instead, a proper use case diagram depicts a high-level overview of the relationship between use cases, actors, and systems. Experts recommend that use case diagrams be used to supplement a more descriptive textual use case. Use cases are represented with a labeled oval shape. Stick figures represent actors in the process, and the actor's participation in the system is modeled with a line between the actor and use case.

Use case diagram consists of two parts:

Use case: A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

Actor: An actor is a person, organization or external system that plays a role in one or more interaction with the system.

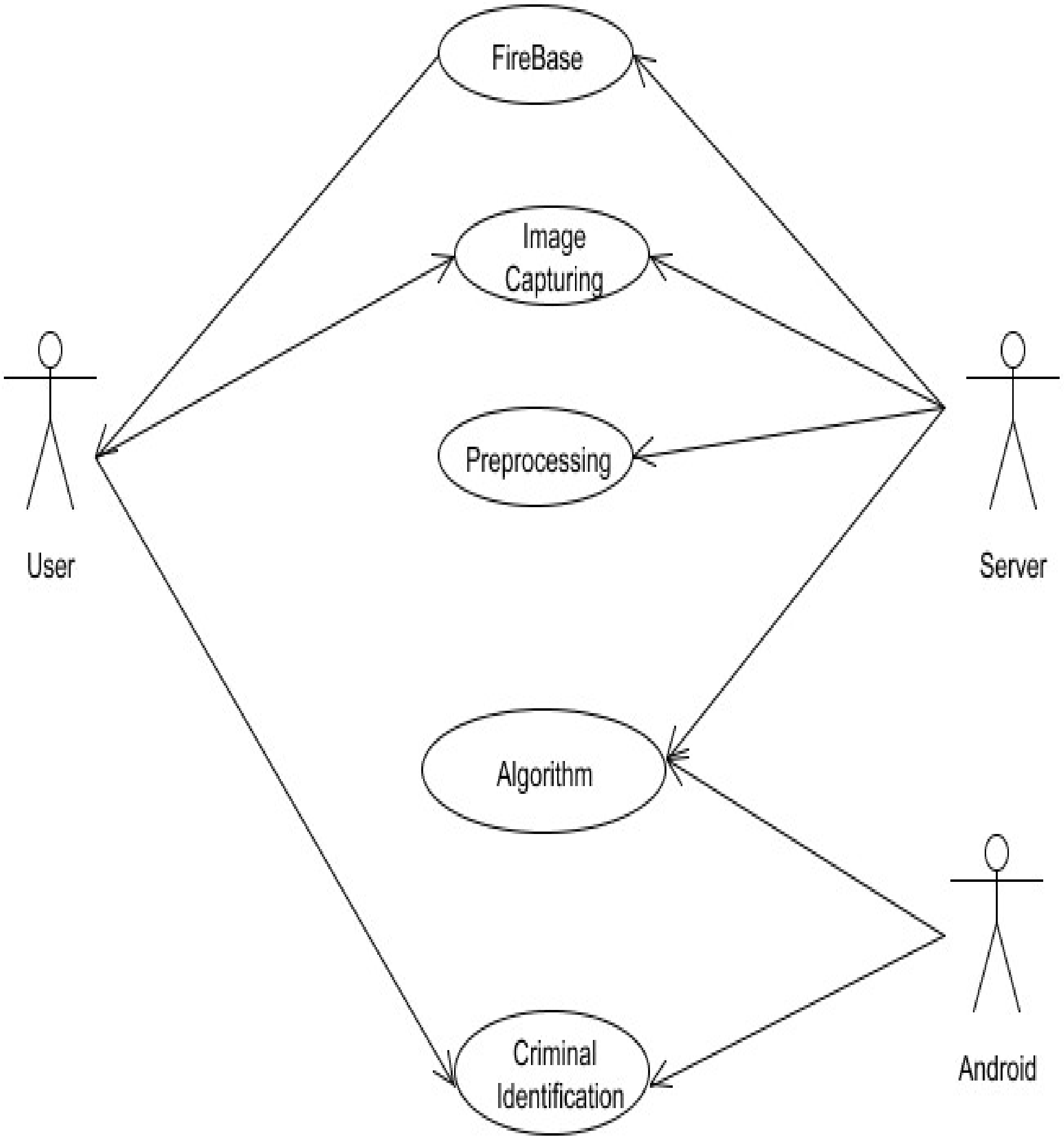


Fig 4.1.1 USE CASE DIAGRAM FOR FACE DETECTION AND RECOGNITION FOR CRIMINAL IDENTIFICATION SYSTEM

4.1.2 SEQUENCE DIAGRAM

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.

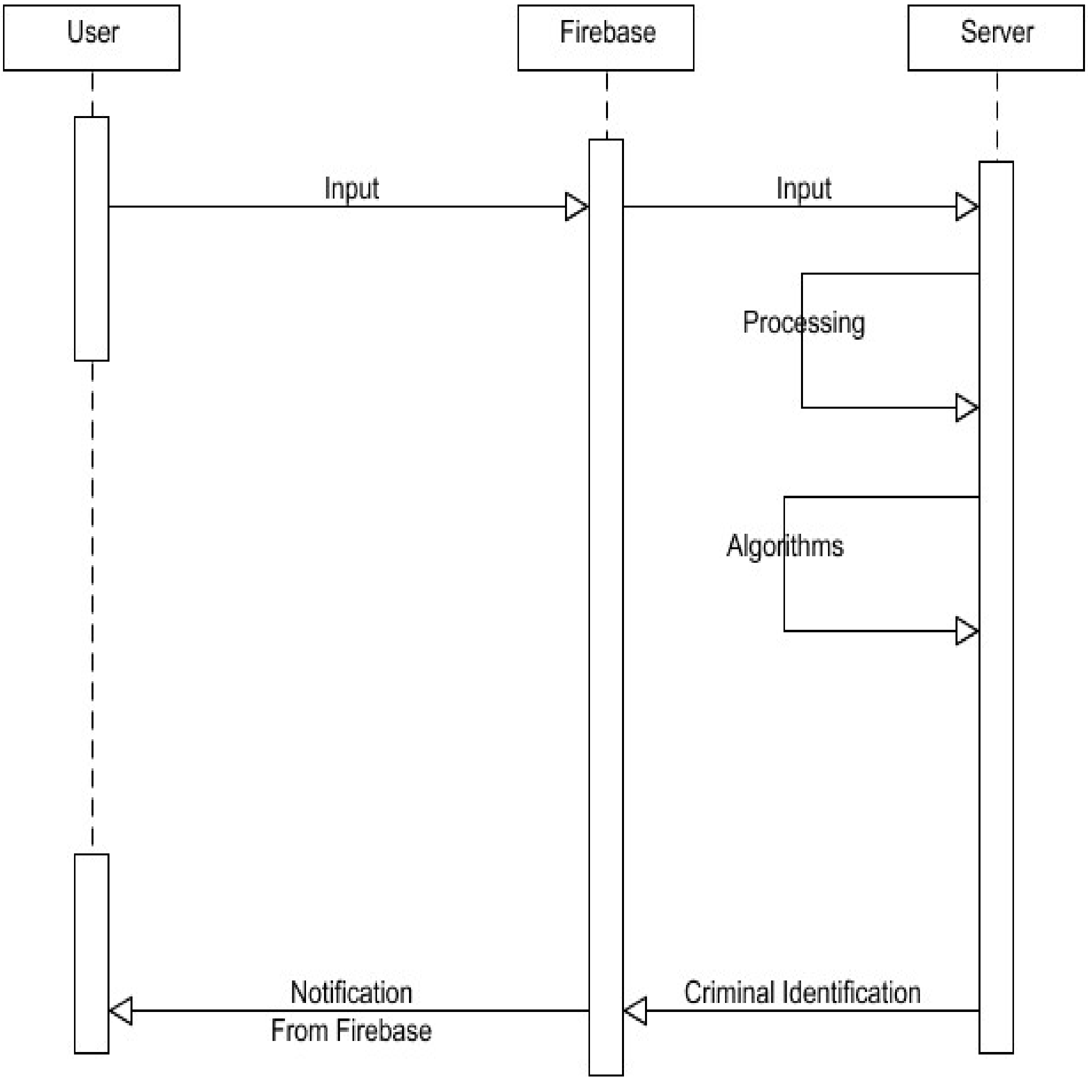


Fig 4.1.2 SEQUENCE DIAGRAM FOR FACE DETECTION AND RECOGNITION FOR CRIMINAL IDENTIFICATION SYSTEM

4.1.3 ACTIVITY DIAGRAM

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control. The most important shape types:

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.

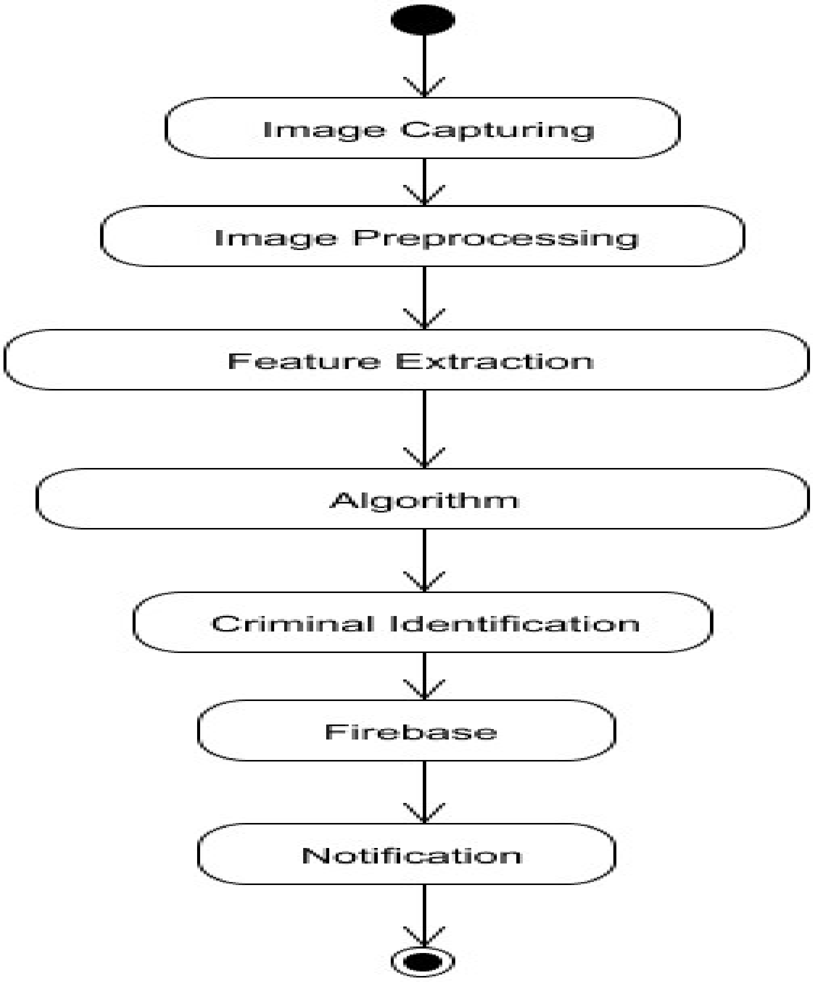


Fig 4.1.3 ACTIVITY DIAGRAM FOR FACE DETECTION AND RECOGNITION FOR CRIMINAL IDENTIFICATION SYSTEM

4.1.4 COLLABORATION DIAGRAM

UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects.

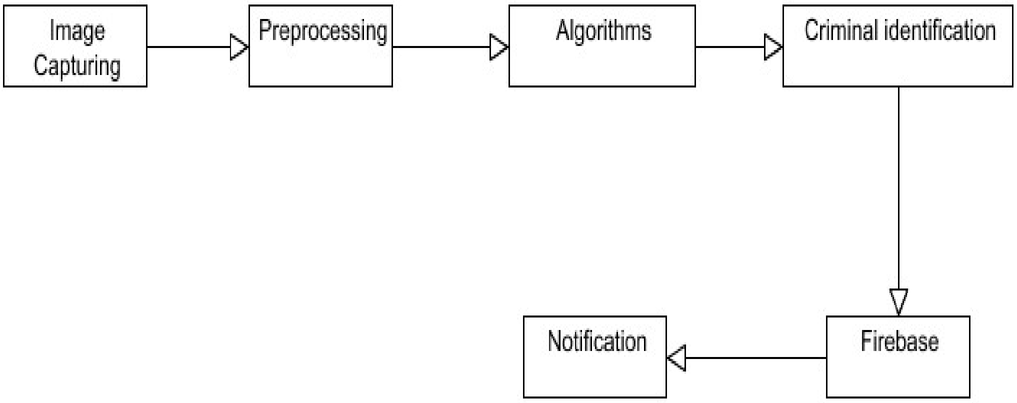


Fig 4.1.4 COLLABORATION DIAGRAM FOR FACE DETECTION AND RECOGNITION FOR CRIMINAL IDENTIFICATION SYSTEM

4.2 DATA FLOW DIAGRAM (DFD)

A data flow diagram (DFD) is a graphical or visual representation using a standardized set of symbols and notations to describe a business's operations through data movement. They are often elements of a formal methodology such as Structured Systems Analysis and Design Method (SSADM). Superficially, DFDs can resemble flow charts or Unified Modeling Language (UML), but they are not meant to represent details of software logic. DFDs make it easy to depict the business requirements of applications by representing the sequence of process steps and flow of information using a graphical representation or visual representation rather than a textual description. When used through an entire development process, they first document the results of business analysis. Then, they refine the representation to show how information moves through, and is changed by, application flows. Both automated and manual processes are represented. The image is received from the camera or from the video only the face image is extracted,preprocessor,grayscale images are generated.The Grayslake images are used for training the model.The trained data are stored as the .yml file.The .yml files are used to compare the faces stored in the database to extract information about the face.

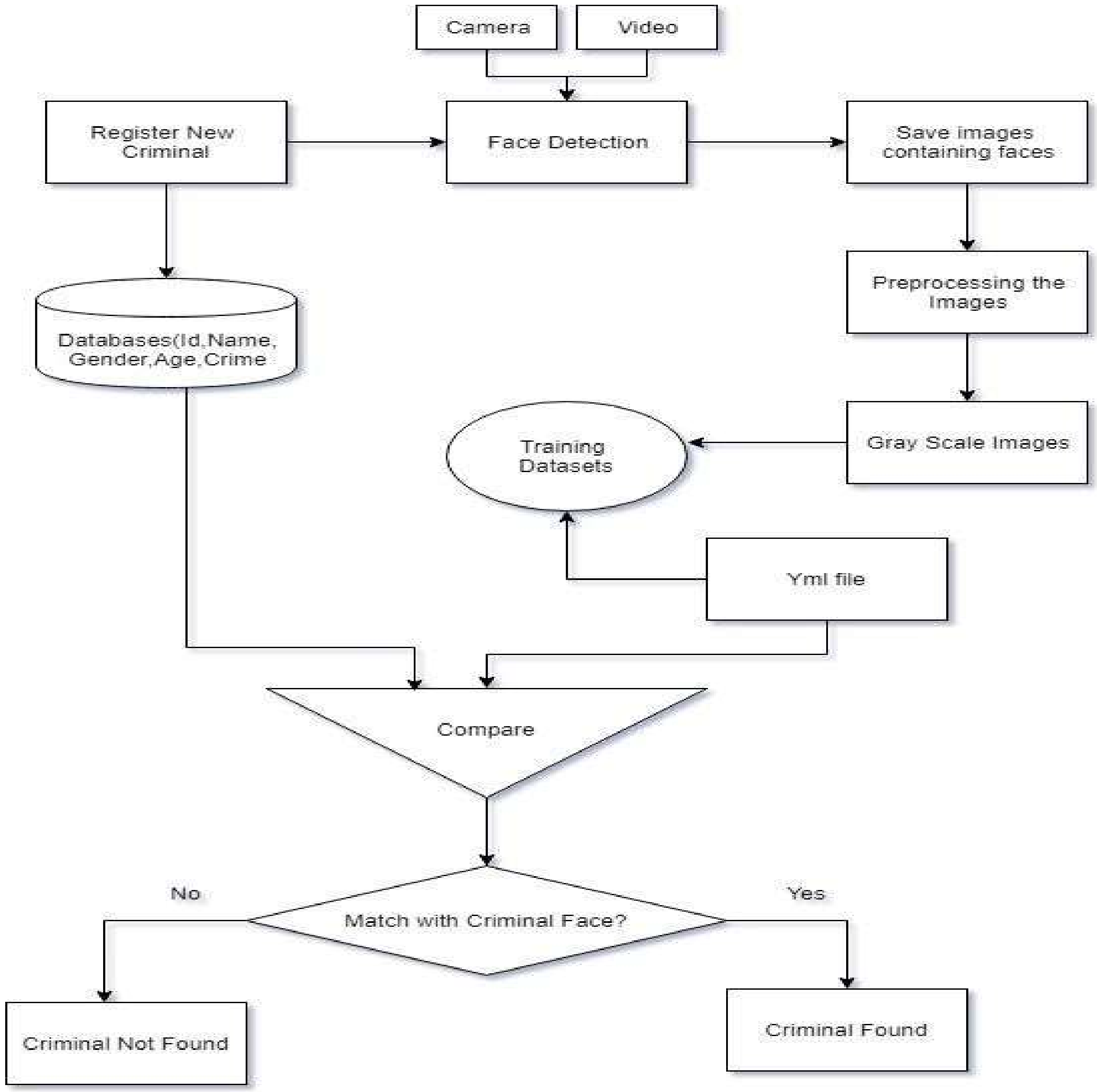


Fig 4.2 DATA FLOW DIAGRAM FOR FACE DETECTION AND RECOGNITION FOR CRIMINAL IDENTIFICATION SYSTEM

4.3 ENTITY-RELATIONSHIP DIAGRAM

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how "entities" such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs.

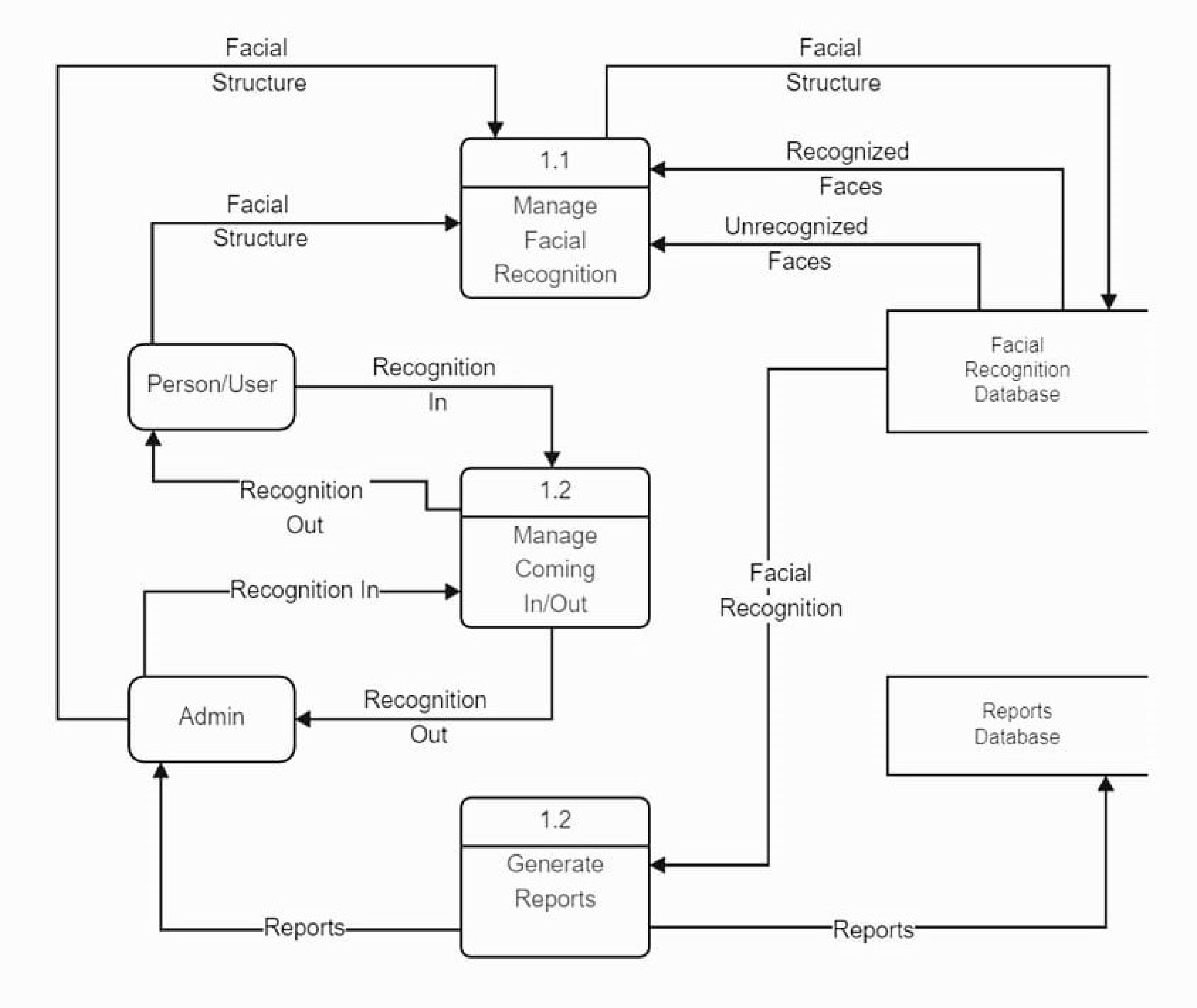


Fig 4.3 ER DIAGRAM FOR FACE DETECTION AND RECOGNITION FOR CRIMINAL IDENTIFICATION SYSTEM

4.4 DATA DICTIONARY DIAGRAM

A data dictionary provides terminology for all relevant data to be used by the developers in a project. It helps in performing analysis based on the impact of some data on the processing activities. It also helps the developers to determine the definition of different data structures in terms of their basic elements while designing activities. In the case of large systems, data dictionaries may become extremely voluminous and difficult to handle. In such case, CASE (Computer-Aided Software Engineering) tools are used, that capture all the data items appearing in the DFD and automatically generate the data dictionary. Using any convention's DFD rules or guidelines, the symbols depict the four components of data flow diagrams.

External entity: an outside system that sends or receives data, communicating with the system being diagrammed.

Process: any process that changes the data, producing an output.

Data store: files or repositories that hold information for later use, such as a database table or a membership form.

Data flow: the route that data takes between the external entities, processes and data stores.

Attribute Name	Data Type	Description
Criminal_id	int	Unique criminal identifier
name	varchar	Criminal's full name
Date_of_birth	date	Criminal's date of birth
nationality	varchar	Criminal's nationality
height	decimal	Criminal's height in meters
weight	decimal	Criminal's weight in kilograms

Table 4.4.1 CRIMINAL ENTITY TABLE

Attribute Name	**Data Type**	**Description**
Record_id	int	Unique criminal record identifier
Criminal_id (FK)	int	Foreign key to Criminal entity
Face_id (FK)	int	Foreign key to Face_data entity
notes	text	Additional notes or comments
date	date	Date when the face was matched with a criminal record
time	time	Time when the face was matched with a criminal record
location	varchar	Location where the face was matched with a criminal record

Table 4.4.2 CRIMINAL RECORD ENTITY TABLE

Attribute Name	**Data Type**	**Description**
face_id	int	Unique face identifier
cctv_id (FK)	int	Foreign key to CCTV entity
image	blob	Image data of the face
Captured_date	date	Date when the face was captured

Table 4.4.3 FACE DATA ENTITY TABLE

CHAPTER 5

SYSTEM ARCHITECTURE

5. SYSTEM ARCHITECTURE

5.1 ARCHITECTURE OVERVIEW

An architecture diagram is a graphical representation of a set of concepts that are part of an architecture, including their principles, elements and components. It is also defined as a visual representation that maps out the physical implementation for components of a software system. It shows the general structure of the software system and the associations, limitations, and boundaries between each element.

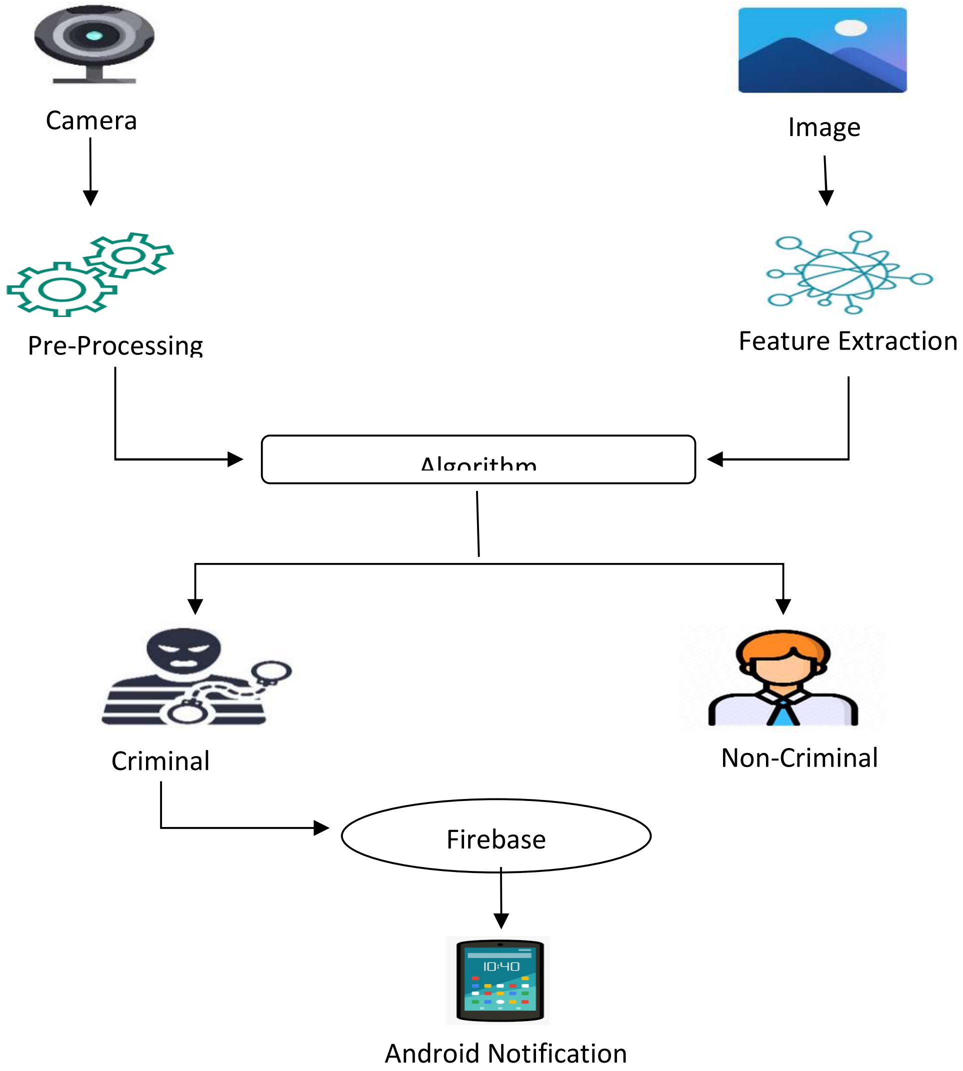


Fig 5.1 ARCHITECTURE DIAGRAM FOR FACE DETECTION AND RECOGNITION FOR CRIMINAL IDENTIFICATION SYSTEM

5.2 MODULE DESIGN SPECIFICATION

MODULES

- Face Detection
- Creation of Dataset
- Training Faces
- Face Recognition

5.2.1 FACE DETECTION

- Face detection just mean that a system is able to identify that there is a human face present in an image and video, And this is possible using Haar cascade classifier.
- A Haar classifier or Haar cascade classifier, is a machine learning object detection program that identifies objects in an image and video.
- We can detect different parts of the human body using Haar cascade classifier like eyes, nose, face, etc.
- To detect the any part of the human body, there is different xml files are available, for eg. To detect the face we use Haarcascade_frontalface_default.xml file, OpenCv already contains this file.
- After that set the border using rectangle() to show the detected faces in zero or more bounding boxes.

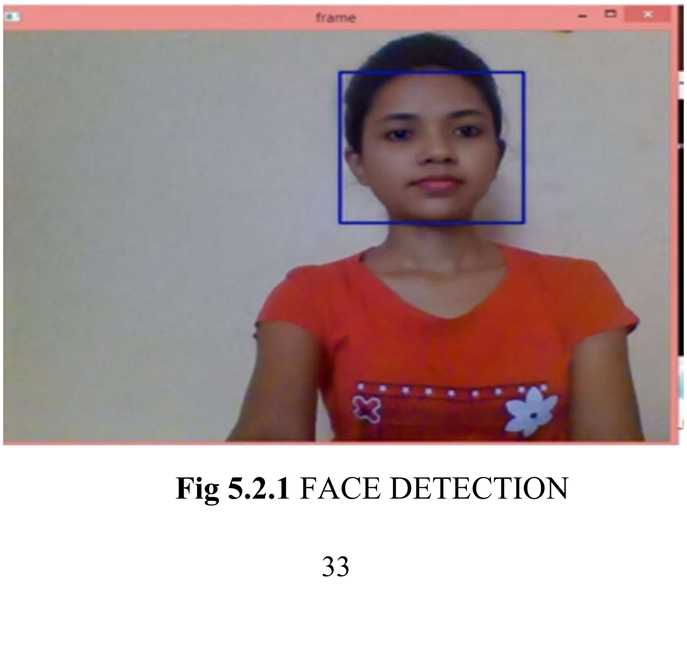


Fig 5.2.1 FACE DETECTION

5.2.2 CREATION OF DATASET

- Create a dataset to add information to recognize the person in the image or video.
- For this first set the id for that person whose face is detected, then using this id we can set the other information while face recognition.
- Here we can create our own dataset or start with the available face databases.
- The input for the dataset is user id, it can be different for different faces.
- After capturing the image from webcam, set the id for that image and save it into a particular folder for use of another program.

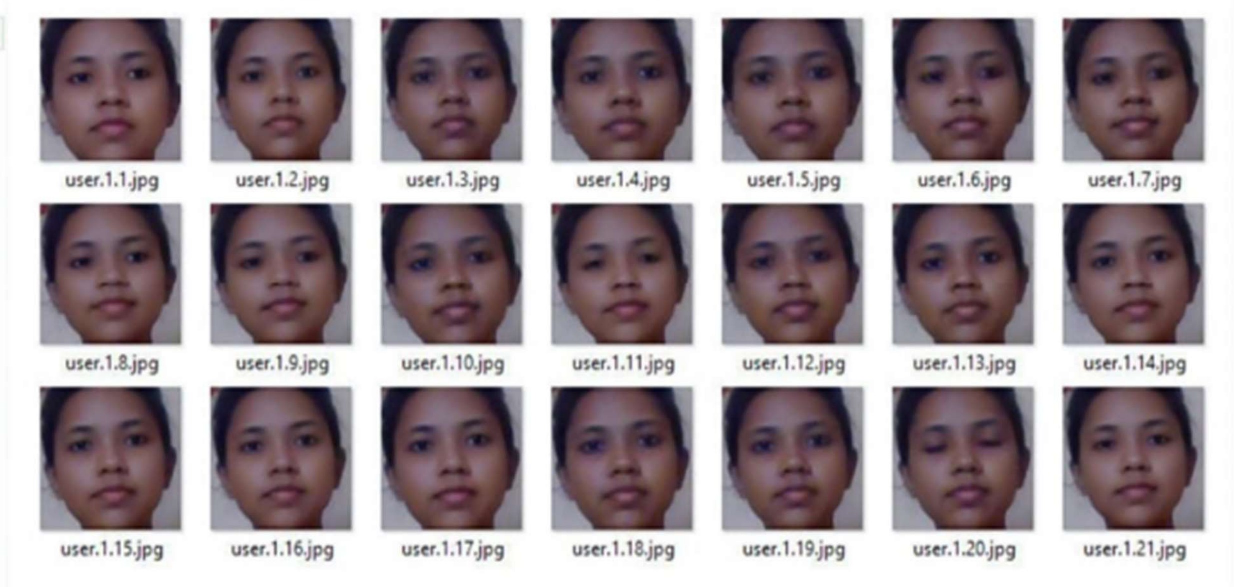


Fig 5.2.2 DATABASE IMAGES

5.2.3 TRAINING FACES

- Using the training faces, all face dataset converted into a single .yml file.
- For this we have trainingdata.yml file saved in a particular folder .
- Create the function to prepare the training set: Now, we will define a function **getImageWithId(path)**
- That takes the absolute path to the image dataset as input argument and returns tuple of two list, one containing the detected faces and the other containing the corresponding lable for that face.

- For example, if the ith index in the list of faces represents the 5th individual in the database, then the corresponding ith location in the list of labels has value equal to 5.
- Then covert the dataset faces (which is created in Creation of dataset) into .yml file.

Fig 5.2.3 TRAINED FACE

5.2.4 FACE RECOGNITION

This is the final step of the program that the face recognition process.

In this we are going through following two steps:

1)Capturing the video from Webcam

2)Compare it with .yml file

a) Capturing video from Webcam

Here, we can capture the image through Webcam as an input.

b) Compare it with .yml file

After capturing the image from Webcam, The image will compare with .yml file. And then finally we got an output as an image with their identity like, the name of the person, age, etc....

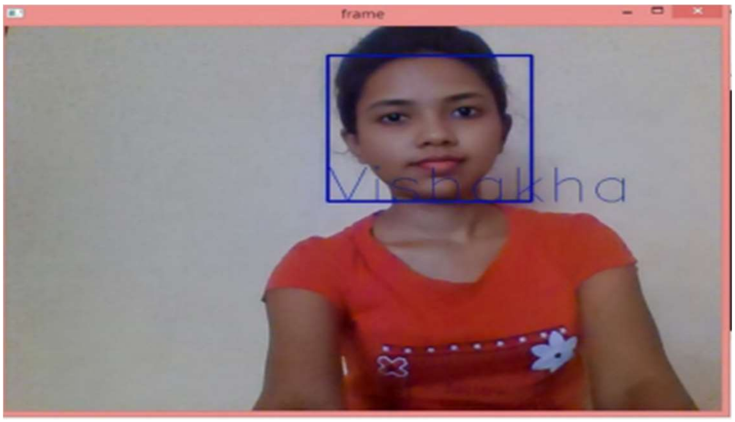


Fig 5.2.4 FACE RECOGNITION

5.3 ALGORITHMS

5.3.1 LBPH ALGORITHM

Local Binary Pattern (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number. It was first described in 1994 (LBP) and has since been found to be a powerful feature for texture classification. It has further been determined that when LBP is combined with histograms of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets. Using the LBP combined with histograms we can represent the face images with a simple data vector.

1.Step-by-Step: Now that we know a little more about face recognition and the LBPH, let’s go further and see the steps of the algorithm:

Parameters: the LBPH uses 4 parameters:

Radius: the radius is used to build the circular local binary pattern and represents the radius around the central pixel. It is usually set to 1.

Neighbors: the number of sample points to build the circular local binary pattern. Keep in mind: the more sample points you include, the higher the computational cost. It is usually set to 8.

Grid X: the number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector.

It is usually set to 8.

Grid Y: the number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. It is usually set to 8.

2. Training the Algorithm: First, we need to train the algorithm. To do so, we need to use a dataset with the facial images of the people we want to recognize. We need to also set an ID (it may be a number or the name of the person) for each image, so the algorithm will use this information to recognize an input image and give you an output. Images of the same person must have the same ID. With the training set already constructed, let's see the LBPH computational steps.

3. Applying the LBP operation: The first computational step of the LBPH is to create an intermediate image that describes the original image in a better way, by highlighting the facial characteristics. To do so, the algorithm uses a concept of a sliding window, based on the parameters radius and neighbors.

The image below shows this procedure:

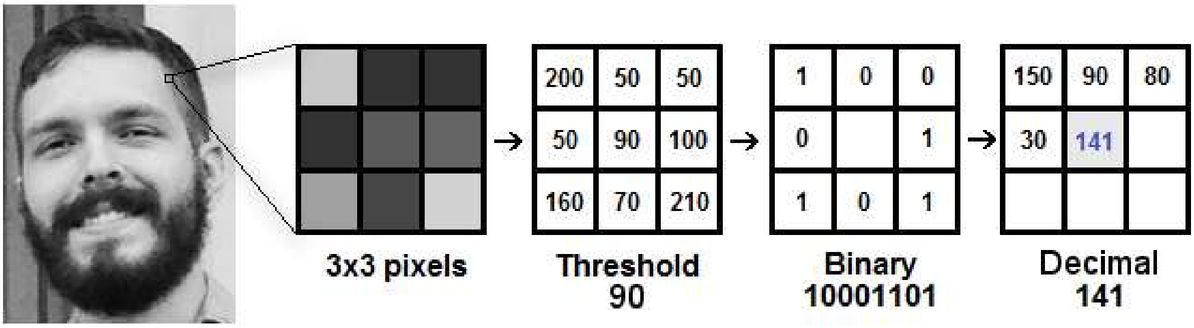


Fig 5.3.1 LBPH ALGORITHM WORKING

5.3.2 CASCADING CLASSIFIER

Cascading Classifier Another way by which Viola Jones ensured that the algorithm performs fast is by employing a cascade of classifiers The cascade classifier essentially consists of stages where each stage consists of a strong classifier. This is beneficial since it eliminates the need to apply all features at once on a window. Rather, it groups the features into separate sub-windows and the classifier at each stage determines whether or not the sub-window is a face. In case it is not, the sub-window is discarded along with the features in that window. If the sub-window moves past the classifier, it continues to the next stage where the second stage of features is applied. The process can be understood with the help of the diagram below.

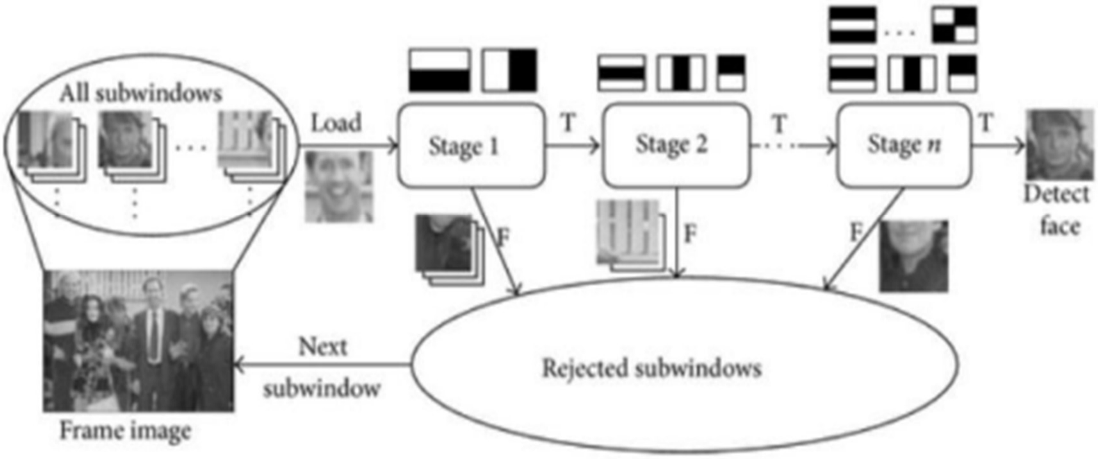


Fig 5.3.2 WORKING OF CASCADING CLASSIFIER

5.3.3 OPEN CV

OpenCV (Open-Source Computer Vision Library) is a library of programming functions mainly aimed at real-time computer vision. Classifier gives the differences between positive and negative image where the positive image is for face and the negative image is for non-face image. OpenCV trains the classifier on any desired face as set in the program and provides two pre-trained and ready for implementing face detection classifier. Two files haarcascade_frontalface_alt.xml and haarcascade_eye is used for detecting face and eye respectively. The other feature provided by OpenCV is LBP (Local Binary Pattern) cascade classifier which is local binary patterns that train the grayscale image of hundreds to thousands. LBP looks at every 9 pixels of 3 × 3 window. Compare the central pixel with the value of surrounding 8 pixels. For each pixel that is greater than the central pixel value, is replaced by 1 and for smaller, it sets the value to 0. Finally, it creates the block histogram to form one image and one feature image converts it into a yml format for further recognizing the face.

This Figure gives the layers of the learning and detecting the features of the face.

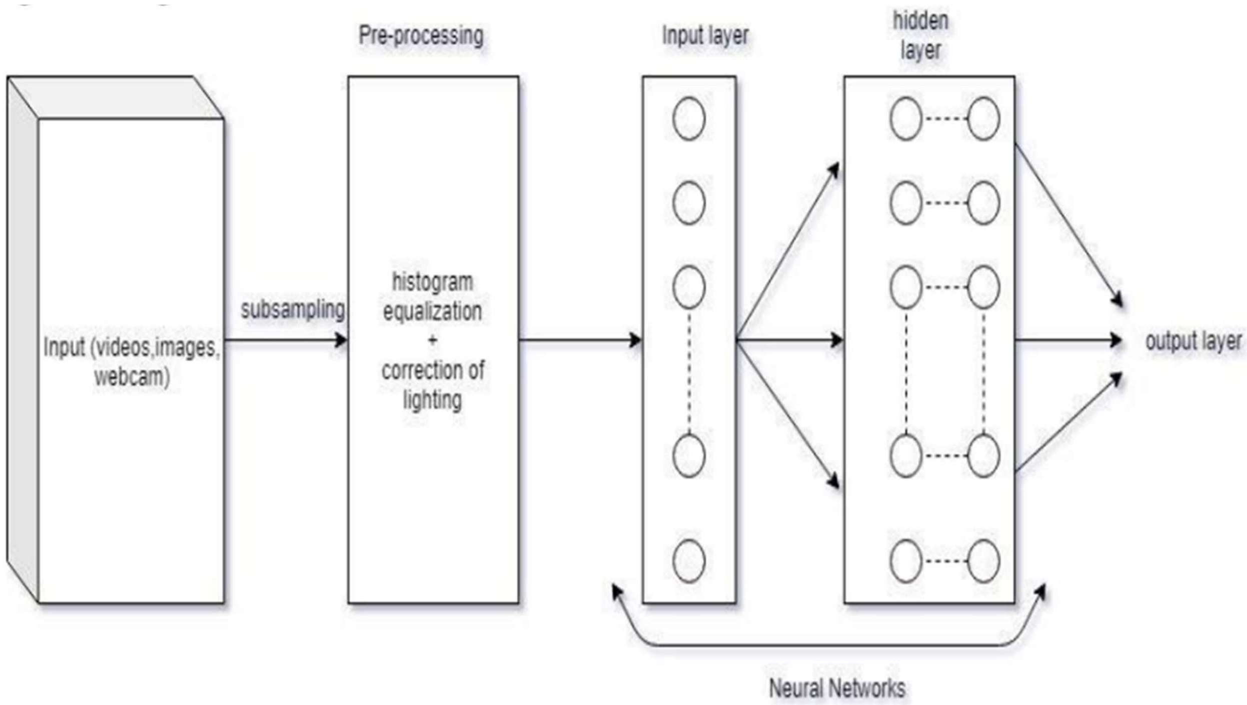

Pre-processing
Input layer
hidden layer
Input (videos,images, webcam)
subsampling
histogram equalization + correction of lighting
output layer
Neural Networks

Fig 5.3.3 FACE DETECTION PROCESS

CHAPTER 6

SYSTEM IMPLEMENTATION

6. SYSTEM IMPLEMENTATION

6.1 CODING

```
#!/usr/bin/env python
# coding: utf-8
# In[ ]:
import tkinter as tk
from tkinter import Message ,Text
from tkinter import *
import cv2,os
import shutil
import csv
import numpy as np
from PIL import Image, ImageTk
import pandas as pd
import datetime
import time
import tkinter.ttk as ttk
import tkinter.font as font
from firebase import firebase
fixefixed_interval = 3
firebase = firebase.FirebaseApplication('https://my-first-project-75585-default-rtdb.firebaseio.com/', None)
window = tk.Tk()
#helv36 = tk.Font(family='Helvetica', size=36, weight='bold')
window.title("AUTOMATED CRIMINAL IDENTIFICATION ")
dialog_title = 'QUIT'
dialog_text = 'Are you sure?'
```

```
#answer = messagebox.askquestion(dialog_title, dialog_text)
window.geometry('1368x768')
#window.configure(background='#496E7C')
#window.attributes('-fullscreen', True)
window.grid_rowconfigure(0, weight=1)
window.grid_columnconfigure(0, weight=1)
global key
img = ImageTk.PhotoImage(Image.open("./images/9.jpg"))
panel = Label(window, image = img)
panel.place(x = 0, y = 0)
message = tk.Label(window, text="AUTOMATED CRIMINAL
IDENTIFICATION SYSTEM" ,bg="#496E7C"  ,fg="white"  ,width=50
,height=2,font=('times', 30, 'bold'))
message.place(x=80, y=10)
lbl = tk.Label(window, text="Enter ID",width=20  ,height=1  ,fg="white"
,bg="#496E7C" ,font=('times', 15, ' bold ') )
lbl.place(x=100, y=200)
txt = tk.Entry(window,width=20  ,bg="#ced5db" ,fg="black",font=('times', 15, '
bold '))
txt.place(x=400, y=200)
lbl2 = tk.Label(window, text="Enter Name",width=20  ,fg="white"  ,bg="#496E7C"
,height=1 ,font=('times', 15, ' bold '))
lbl2.place(x=100, y=250)
txt2 = tk.Entry(window,width=20  ,bg="#ced5db"  ,fg="black",font=('times', 15, '
bold ')  )
txt2.place(x=400, y=250)
lbl3 = tk.Label(window, text="Enter Age",width=20  ,height=1  ,fg="white"
,bg="#496E7C" ,font=('times', 15, ' bold ') )
```

```
lbl3.place(x=100, y=300)
txt3 = tk.Entry(window,width=20  ,bg="#ced5db"  ,fg="black",font=('times', 15,'
bold ')  )
txt3.place(x=400, y=300)
lbl4 = tk.Label(window, text="Enter Gender",width=20  ,height=1  ,fg="white"
,bg="#496E7C" ,font=('times', 15, ' bold ') )
lbl4.place(x=100, y=350)
txt4 = tk.Entry(window,width=20  ,bg="#ced5db"  ,fg="black",font=('times', 15, '
bold ')  )
txt4.place(x=400, y=350)
lbl3 = tk.Label(window, text="Notification",width=15  ,fg="white"  ,bg="#496E7C"
,height=2 ,font=('times', 15, ' bold'))
lbl3.place(x=200, y=475)
message = tk.Label(window, text="" ,bg="#ced5db"  ,fg="black"  ,width=50
,height=2, activebackground = "yellow" ,font=('times', 15, ' bold '))
message.place(x=400, y=475)
lbl3 = tk.Label(window, text="Criminal_Reports",width=15  ,fg="white"
,bg="#496E7C"  ,height=2 ,font=('times', 15, ' bold'))
lbl3.place(x=200, y=550)
message2 = tk.Label(window, text="" ,fg="black"
,bg="#ced5db",activeforeground = "green",width=50  ,height=3  ,font=('times', 15, '
bold '))
message2.place(x=400, y=550)
def clear():
   txt.delete(0, 'end')
   txt2.delete(0, 'end')
   txt3.delete(0, 'end')
   txt4.delete(0, 'end')
```

```
    res = ""
message.configure(text= res)

'''def clear2():
    txt2.delete(0, 'end')
    res = ""
    message.configure(text= res) '''

def is_number(s):
    try:
        float(s)
        return True
    except ValueError:
        pass

    try:
        import unicodedata
        unicodedata.numeric(s)
        return True
    except (TypeError, ValueError):
        pass

    return False

def TakeImages():
    Id=(txt.get())
    name=(txt2.get())
    age=(txt3.get())
    gender=(txt4.get())
    if(is_number(Id) and name.isalpha()):
        cam = cv2.VideoCapture(0)
        harcascadePath = "haarcascade_frontalface_default.xml"
        detector=cv2.CascadeClassifier(harcascadePath)
        sampleNum=0
        while(True):
            ret, img = cam.read()
            gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
            faces = detector.detectMultiScale(gray, 1.3, 5)
            for (x,y,w,h) in faces:
                cv2.rectangle(img,(x,y),(x+w,y+h),(255,0,0),2)
                #incrementing sample number
                sampleNum=sampleNum+1
                #saving the captured face in the dataset folder TrainingImage
                cv2.imwrite("Capturing_Images\ "+name +"."+Id +'.'+ str(sampleNum) +
".jpg", gray[y:y+h,x:x+w])
                #display the frame
                cv2.imshow('frame',img)
            #wait for 100 miliseconds
            if cv2.waitKey(100) & 0xFF == ord('q'):
                break
            # break if the sample number is morethan 100
            elif sampleNum>60:
                break
        cam.release()
        cv2.destroyAllWindows()
        res = "Images Saved for ID : " + Id +" Name : "+ name +"Age :" + age +
"Gender:" +gender
        row = [Id , name,age,gender]
        with open('Wanted_List\Wanted_List.csv','a+') as csvFile:
            writer = csv.writer(csvFile)
            writer.writerow(row)
csvFile.close()
        message.configure(text= res)
    else:
        if(is_number(Id)):
            res = "Enter Alphabetical Name"
            message.configure(text= res)
        if(name.isalpha()):
            res = "Enter Numeric Id"
            message.configure(text= res)
    def TrainImages():
    recognizer = cv2.face_LBPHFaceRecognizer.create()#recognizer =
cv2.face.LBPHFaceRecognizer_create()#$cv2.createLBPHFaceRecognizer()
    harcascadePath = "haarcascade_frontalface_default.xml"
    detector =cv2.CascadeClassifier(harcascadePath)
    faces,Id = getImagesAndLabels("Capturing_Images")
    recognizer.train(faces, np.array(Id))
    recognizer.save("Models\Trainner.yml")
    res = "Image Trained"#+",".join(str(f) for f in Id)
    message.configure(text= res)
def getImagesAndLabels(path):
    #get the path of all the files in the folder
    imagePaths=[os.path.join(path,f) for f in os.listdir(path)]
    #print(imagePaths)
```

```
     #create empth face list
    faces=[]
    #create empty ID list
    Ids=[]
    #now looping through all the image paths and loading the Ids and the images
    for imagePath in imagePaths:
   #loading the image and converting it to gray scale
        pilImage=Image.open(imagePath).convert('L')
        #Now we are converting the PIL image into numpy array
        imageNp=np.array(pilImage,'uint8')
        #getting the Id from the image
        Id=int(os.path.split(imagePath)[-1].split(".")[1])
        # extract the face from the training image sample
        faces.append(imageNp)
        Ids.append(Id)
    return faces,Ids
def TrackImages():
    recognizer =
cv2.face.LBPHFaceRecognizer_create()#cv2.createLBPHFaceRecognizer()
    recognizer.read("Models\Trainner.yml")
    harcascadePath = "haarcascade_frontalface_default.xml"
    faceCascade = cv2.CascadeClassifier(harcascadePath);
    df=pd.read_csv("Wanted_List\Wanted_List.csv")
    cam = cv2.VideoCapture(0)
    font = cv2.FONT_HERSHEY_SIMPLEX
    col_names =  ['Id','Name','Date','Time','Location']
    attendance = pd.DataFrame(columns = col_names)
    while True:
```

```
    ret, im =cam.read()
    gray=cv2.cvtColor(im,cv2.COLOR_BGR2GRAY)
    faces=faceCascade.detectMultiScale(gray, 1.2,5)
    for(x,y,w,h) in faces:
      cv2.rectangle(im,(x,y),(x+w,y+h),(225,0,0),2)
      Id, conf = recognizer.predict(gray[y:y+h,x:x+w])
      if(conf < 50):
       Location="chennai_Airport"
        ts = time.time()
        date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
        timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
        aa=df.loc[df['Id'] == Id]['Name'].values
        tt=str(Id)+"-"+aa+"-"+"WANTED"
attendance.loc[len(attendance)] = [Id,aa,date,timeStamp,Location]
 else:
       Id='Non-Criminal'
        tt=str(Id)

      if(conf > 75):
        noOfFile=len(os.listdir("Database"))+1
        cv2.imwrite("Database\Image"+str(noOfFile) + ".jpg", im[y:y+h,x:x+w])
    cv2.putText(im,str(tt),(x,y+h), font, 1,(255,255,255),2)
    attendance=attendance.drop_duplicates(subset=['Id'],keep='first')

    cv2.imshow('Face_Recognize',im)

    if (cv2.waitKey(1)==ord('q')):
      break
    ts = time.time()
    date = datetime.datetime.fromtimestamp(ts).strftime('%Y-%m-%d')
    timeStamp = datetime.datetime.fromtimestamp(ts).strftime('%H:%M:%S')
    Hour,Minute,Second=timeStamp.split(":")
    fileName="Criminal_Reports\criminals_"+date+"_"+Hour+"-"+Minute+"-
"+Second+".csv"
   attendance.to_csv(fileName,index=False)
    cam.release()
    cv2.destroyAllWindows()
     res=attendance
    message2.configure(text= res)
    data =  {"Id":Id,"Name":aa,"Date":date,"Time":timeStamp,"Location":Location}
    print(data)
  n=str(aa)
    l=str(Location)
    t=str(timeStamp)
    date=str(date)
 firebase.put('/', '/criminal/Name', n)
    firebase.put('/', '/criminal/Location', l)
    firebase.put('/', '/criminal/time', t)
    firebase.put('/', '/criminal/date', date)
    count=0
clearButton = tk.Button(window, text="Clear", command=clear  ,fg="white"
,bg="#496E7C"  ,width=10  ,height=1 ,activebackground = "white" ,font=('times',
15, ' bold '))
clearButton.place(x=300, y=400)
takeImg = tk.Button(window, text="Take Images", command=TakeImages
,fg="white"  ,bg="#496E7C"  ,width=15  ,height=2, activebackground = "white"
,font=('times', 15, ' bold '))
takeImg.place(x=750, y=200)
trainImg = tk.Button(window, text="Train Images", command=TrainImages
,fg="white"  ,bg="#496E7C"  ,width=15  ,height=2, activebackground = "white"
,font=('times', 15, ' bold '))
trainImg.place(x=1000, y=200)
trackImg = tk.Button(window, text="Track Images", command=TrackImages
,fg="white"  ,bg="#496E7C"  ,width=15  ,height=2, activebackground = "white"
,font=('times', 15, ' bold '))
trackImg.place(x=750, y=300)
quitWindow = tk.Button(window, text="Quit", command=window.destroy
,fg="white"  ,bg="#496E7C"  ,width=15  ,height=2, activebackground = "white"
,font=('times', 15, ' bold '))
quitWindow.place(x=1000, y=300)
copyWrite = tk.Text(window, background=window.cget("background"),
borderwidth=0,font=('times', 30, 'italic bold underline'))
copyWrite.tag_configure("superscript", offset=10)
copyWrite.insert("insert", "Python","", "TEAM", "superscript")
copyWrite.configure(state="disabled",fg="red"  )
copyWrite.pack(side="left")
copyWrite.place(x=800, y=750)
window.mainloop()
# In[ ]:

# In[ ]:
```

CHAPTER 7

SYSTEM TESTING

7. SYSTEM TESTING

7. SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example, the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough. Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

7.1 UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

7.2 INTEGRATION TESTING

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected. The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

7.3 FUNCTIONAL TESTING

Functional testing is a type of software testing whereby the system is tested against the functional requirements specifications. Functions or features are tested by feeding them input and examining the output. Functional testing ensures that the requirements are properlysatisfied by the application. This type of testing is not concerned with how processing occurs but rather with the results of processing. It simulates actual system usage but does not make any system structure assumptions. During functional testing, Black box testing technique is used in which the internal logic of the system being tested is not known to the tester. Functionaltesting is normally performed during the levels of system testing and acceptance testing. Typically, Functional testing involves the following steps:

- Identify functions that the software is expected to perform.
- Create input data based on the function's specification.
- Determine the output based on the function's specification.

7.4 STRUCTURAL TESTING

Structural testing is the type of testing carried out to test the structure of code. It is also knownas testing or Glass box testing. This type of testing requires knowledge of the code, so it is mostly done by the developers. It is more concerned with how system does it rather than the functionality of the system. It provides more coverage to the testing. For example, to test certainerror message in application, we need to test the trigger condition for it, but there must be many triggers for it. It is possible to miss out one while testing the requirements drafted in SRS. Butusing this testing, the trigger is most likely to be covered since structural testing aims to coverall the nodes and paths in the structure of code. It is concerned with exercising the internal logicof a program and traversing particular execution.

7.5 OUTPUT TESTING

Output of test cases compared with the expected results created during design of test cases.Asking the user about the format required by them tests the output generated or displayed by the system under consideration. Here, the output format is considered into two was, oneis on screen and another one is printed format. The output on the screen is found to be correctas the format was designed in the system design phase according to user need.

7.6 USER ACCEPTANCE TESTING

Final Stage, before handling over to the customer which is usually carried out by the customer where the test cases are executed with actual data. The system under considerationis tested for user acceptance and constantly keeping touch with the prospective system userat the time of developing and making changes whenever required. It involves planning andexecution of various types of tests in order to demonstrate that the implemented software system satisfies the requirements stated in the requirement document. Two set of acceptance test to be run

a) Those developed by quality assurance group
b) Those developed by customer

7.7 TESTCASES

**TEST REPORT : **01

**USECASE : **DETECT CRIMINAL

Req_Id	Tkt_Id	Req_description	Actual_o/p	Expected_o/p	Tkt_status
1	201	Upload image from pictures and if criminal is found	Criminal details will be displayed	Criminal details will be displayed	PASS
	202	Upload image from pictures and if criminal is not found	No criminal recognized message	No criminal recognized message	PASS

Table 7.3.1 TEST CASE FOR DETECT CRIMINAL

TEST REPORT: 02

**USECASE : **VIDEO SURVEILLANCE

Req_Id	Tkt_Id	Req_description	Actual_o/p	Expected_o/p	Tkt_status
2	301	Web cam will be opened for live surveillance	Web cam will open successfully and will start recognizing criminals	Web cam will open successfully and will start recognizing criminals	PASS
	302	Web cam will be opened for live surveillance	Web cam won’t open due to some access privileges	Web cam will open successfully and will start recognizing criminals	FAIL

Table 7.3.2 TEST CASE FOR VIDEO SURVEILLANCE

CHAPTER 8

CONCLUSION

8. CONCLUSION

8.1 CONCLUSION

In conclusion, face detection and recognition using a combination of LBPH, cascading classifiers, and OpenCV is a powerful approach for criminal identification systems. LBPH is a robust method for extracting facial features and recognizing faces, while cascading classifiers are effective at detecting faces in images and video streams. The use of OpenCV, an open-source computer vision library, provides an easy-to-use framework for implementing these techniques and integrating them into a larger system. The combined use of these techniques has several advantages, including high accuracy and speed, robustness to varying lighting conditions and facial expressions, and scalability to large databases of criminal records. Overall, face detection and recognition using LBPH, cascading classifiers, and OpenCV is a powerful approach for criminal identification systems that can provide high accuracy and robustness in a scalable and efficient manner.

In the present world, almost all people are aware of the importance of CCTV footage, but in most cases, this footage is being used for investigation purposes after a crime/incident has happened. The proposed model has the benefit of identifying and catching criminals before they commit another crime. The real-time CCTV footage is being tracked and analyzed. The result of the analysis is a command to the respective authority to take any action if in case the system identifies the criminal. Hence this can be stopped. In addition to offering the Police great convenience in identifying criminals, this enhanced version of the criminal detection system also saves them time because operations are automated. This project is unique in that it uses face encodings to detect faces.

8.2 FUTURE WORK

For future work, we can add the Alarms to the criminal detection system.

It will only sound when a match is made, letting anyone who isn't watching the CCTV room know that someone from the database has been located in that public area. It presents a surveillance system that will give us alerts when any controversy, fight, or intruder is detected by using CCTV footages**.** However, there are some limitations to consider. For example, the accuracy of face recognition may be affected by factors such as occlusions, changes in appearance over time, and variations in pose or facial expressions. Additionally, the performance of cascading classifiers may depend on the quality of the training data and the tuning of the classifier parameters. There are several avenues for future work to improve the performance of face detection and recognition for criminal identification systems using LBPH, cascading classifiers, and OpenCV. Here are a few suggestions:

Integration with deep learning techniques: While LBPH and cascading classifiers are effective for face detection and recognition, deep learning-based approaches such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have shown promising results in recent years. Integrating these techniques with LBPH and cascading classifiers could potentially improve the accuracy of face recognition in criminal identification systems.

Handling occlusions: One of the limitations of LBPH and cascading classifiers is their inability to handle occlusions, where part of the face is covered. Future work could explore ways to improve the robustness of these techniques to occlusions, such as using additional sensors or fusion with other modalities such as voice or gait recognition.

Real-time performance: In criminal identification systems, it is often critical to have real-time performance to enable quick identification of suspects. Future work could focus on improving the speed of face detection and recognition techniques by optimizing the algorithms and using hardware acceleration.

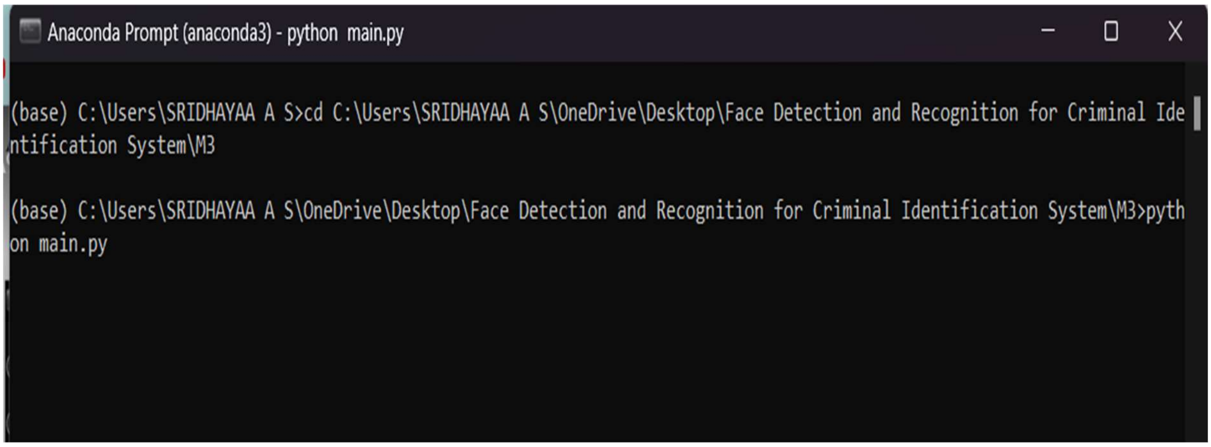
Privacy and ethical considerations: With the increased use of face recognition technologies in law enforcement, it is important to consider the ethical and privacy implications of these systems. Future work could explore ways to incorporate privacy-preserving techniques such as differential privacy and federated learning to protect the privacy of individuals in criminal identification systems.

Dataset diversity: Finally, to improve the accuracy and robustness of face detection and recognition systems, it is important to have diverse and representative datasets that include individuals from different ethnicities, genders, and ages. Future work could focus on developing more diverse and inclusive datasets for training and testing face recognition systems.

APPENDICES

A.1 Sample Screens

A1.1 Open anaconda prompt and give cd and locate the project's file and execute as C:\Users\SRIDHAYAA A S\OneDrive\Desktop\Face Detection and Recognition for Criminal Identification System\M3

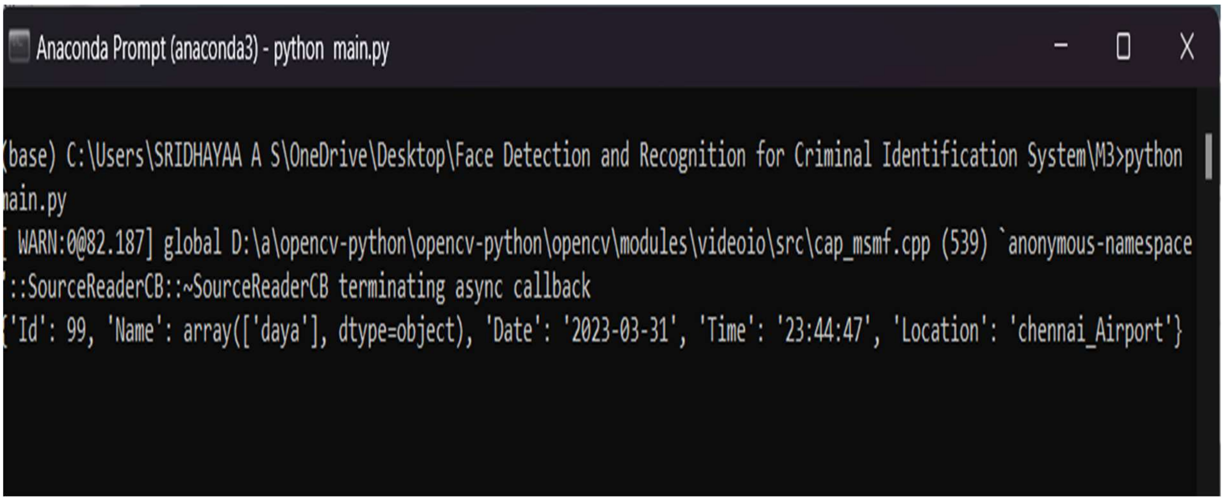


```
(base) C:\Users\SRIDHAYAA A S>cd C:\Users\SRIDHAYAA A S\OneDrive\Desktop\Face Detection and Recognition for Criminal Ide
ntification System\M3

(base) C:\Users\SRIDHAYAA A S\OneDrive\Desktop\Face Detection and Recognition for Criminal Identification System\M3>pyth
on main.py
```

Fig A.1.1 VIRTUAL ENVIRONMENT ACTIVATION

A1.2 After locating the file, type python main.py and once the execution is once need to give quit, after quit if any criminal detected it will callback and display the details such as id, name, date, time, location.



```
(base) C:\Users\SRIDHAYAA A S\OneDrive\Desktop\Face Detection and Recognition for Criminal Identification System\M3>python
main.py
[ WARN:0@82.187] global D:\a\opencv-python\opencv-python\opencv\modules\videoio\src\cap_msmf.cpp (539) `anonymous-namespace
'::SourceReaderCB::~SourceReaderCB terminating async callback
{'Id': 99, 'Name': array(['daya'], dtype=object), 'Date': '2023-03-31', 'Time': '23:44:47', 'Location': 'chennai_Airport'}
```

Fig A.1.2 EXECUTING THE INPUT

A 1.3 This is the home screen of the automated criminal identification system. In this we can give details and train the criminal images, id, name, age and gender.

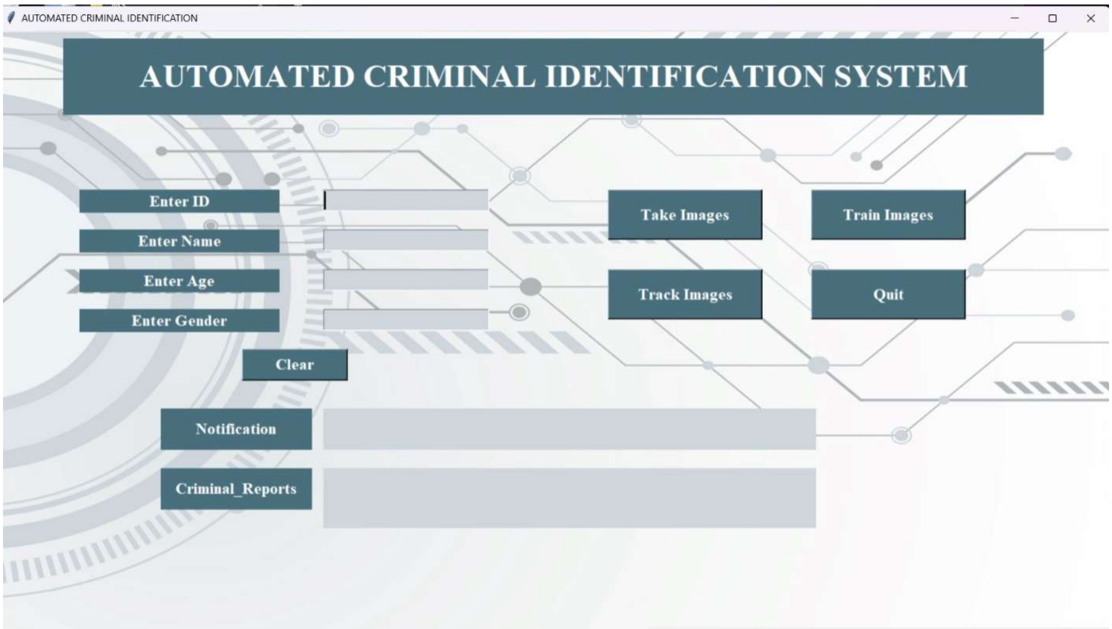
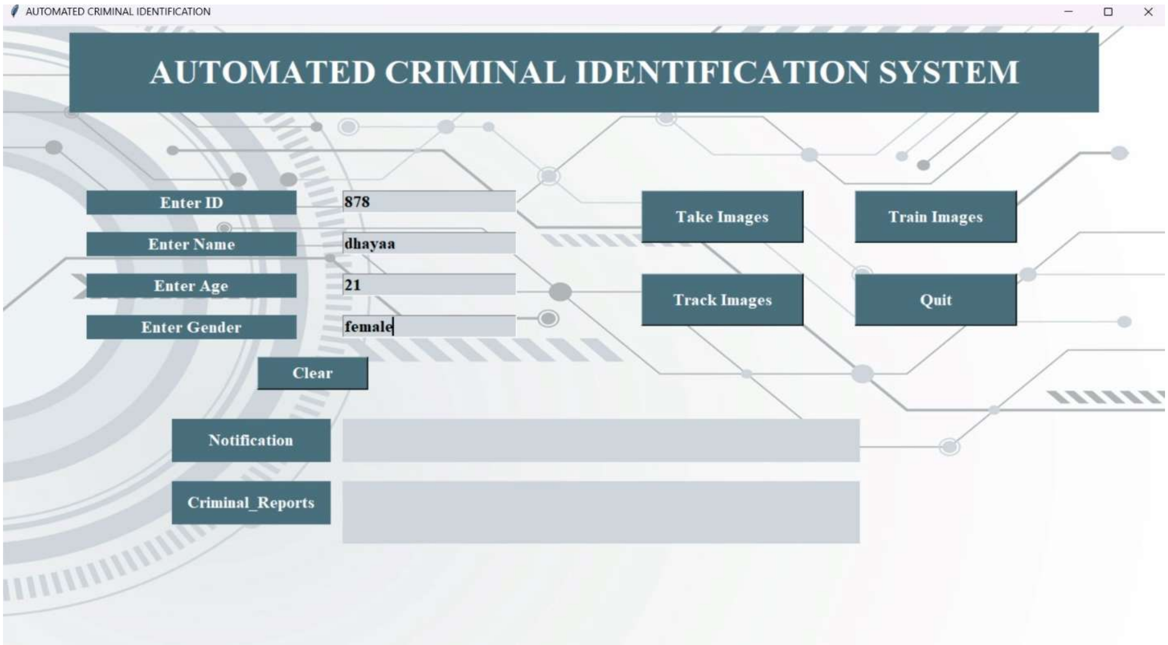


Fig A.1.3 HOME SCREEN OF THE AUTOMATED CRIMINAL IDENTIFICATION SYSTEM

A.1.4 after entering the details such as criminal images, id, name, age and gender, we can take images to train the criminal.


AUTOMATED CRIMINAL IDENTIFICATION
AUTOMATED CRIMINAL IDENTIFICATION SYSTEM
Clear
Take Images
Train Images
Track Images
Quit
Notification
Criminal_Reports

Enter ID 878

Enter Name dhayaa

Enter Age 21

Enter Gender female

Fig A.1.4 ENTERING THE DETAILS OF THE CRIMINAL

A.1.5 when we select the take images the face will be capture in rectangle box and the face will converted into grey scale of multiple images in local database. Below fig A.1.5 shows the face detection of the criminal.

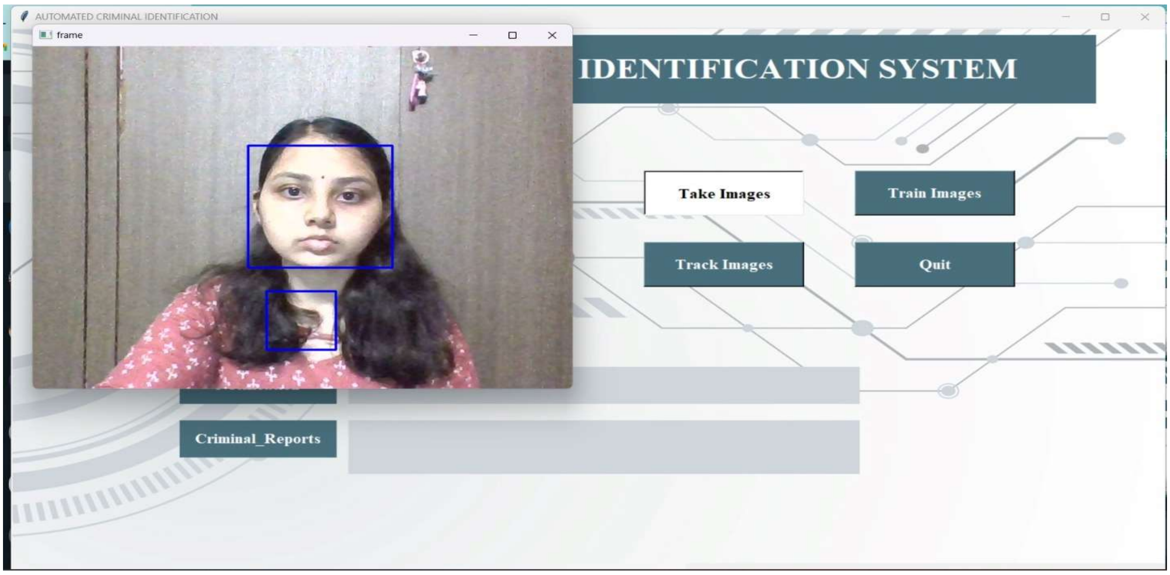
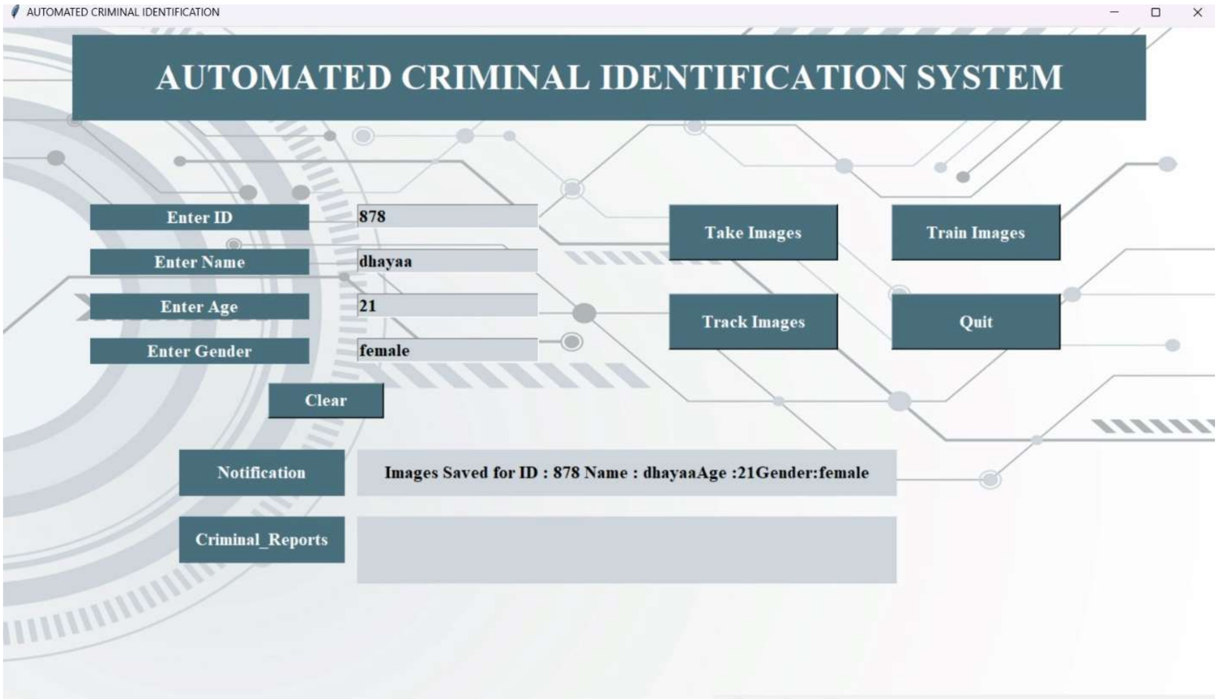


Fig A.1.5 FACE DETECTION OF THE CRIMINAL

A.1.6 once the image is trained the image will be stored in local database and in the notification box we will be notified that details are saved.


AUTOMATED CRIMINAL IDENTIFICATION
AUTOMATED CRIMINAL IDENTIFICATION SYSTEM
Take Images
Train Images
Track Images
Quit
Clear
Criminal_Reports

Enter ID 878

Enter Name dhayaa

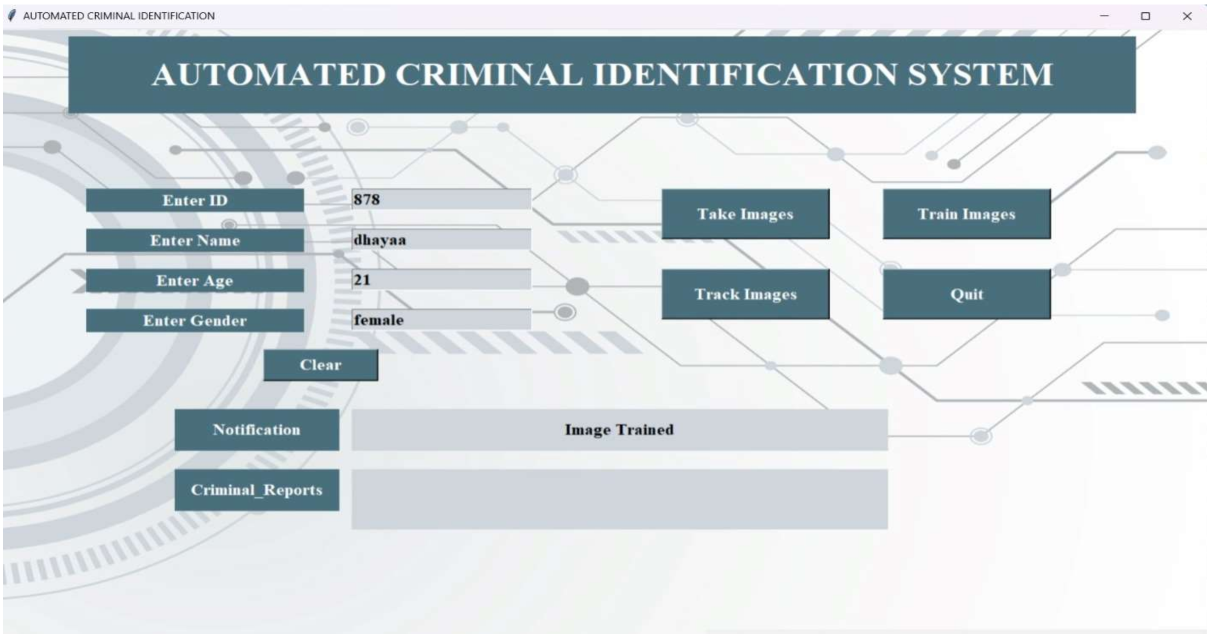
Enter Age 21

Enter Gender female

Notification Images Saved for ID : 878 Name : dhayaaAge :21Gender:female

Fig A.1.6 NOTIFICATION OF IMAGE SAVED

A.1.7 After taking images we can train the faces and once it trained we will get notified


AUTOMATED CRIMINAL IDENTIFICATION

AUTOMATED CRIMINAL IDENTIFICATION SYSTEM

Enter ID: 878

Enter Name: dhayaa

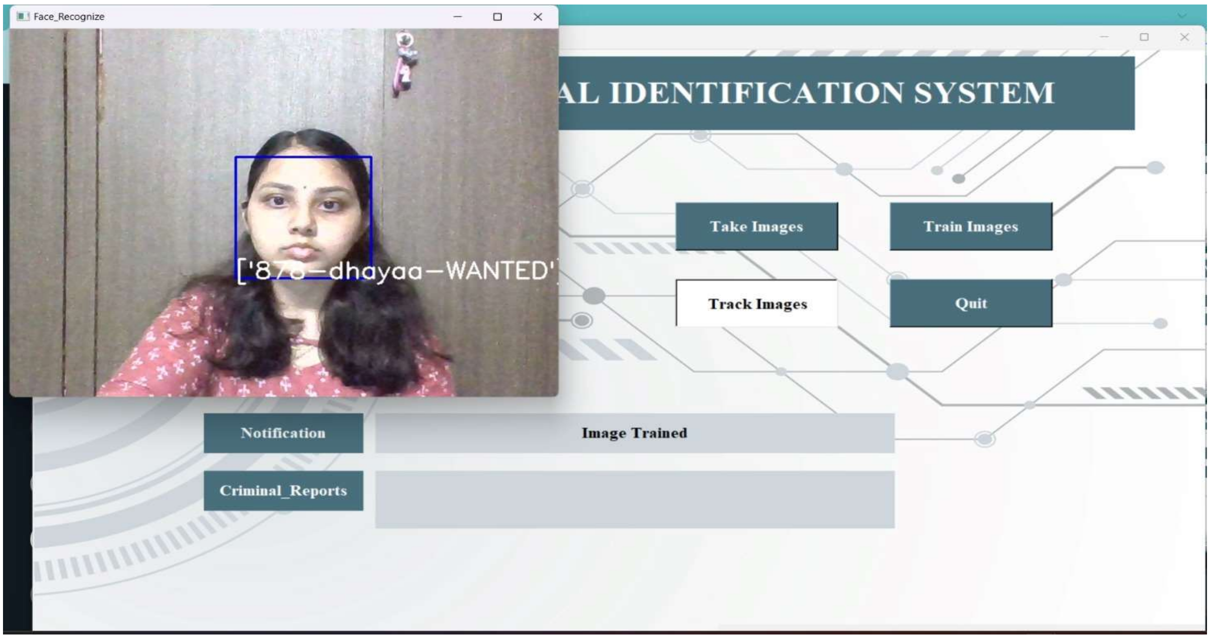
Enter Age: 21

Enter Gender: female

Notification: Image Trained

Fig A.1.7 NOTIFICATION OF IMAGE TRAINED

A.1.8 when the face is trained, next step is tracking images. If the criminal is detected in the rectangle frame capturing, it shows the details like id number, name and as WANTED


Face_Recognize

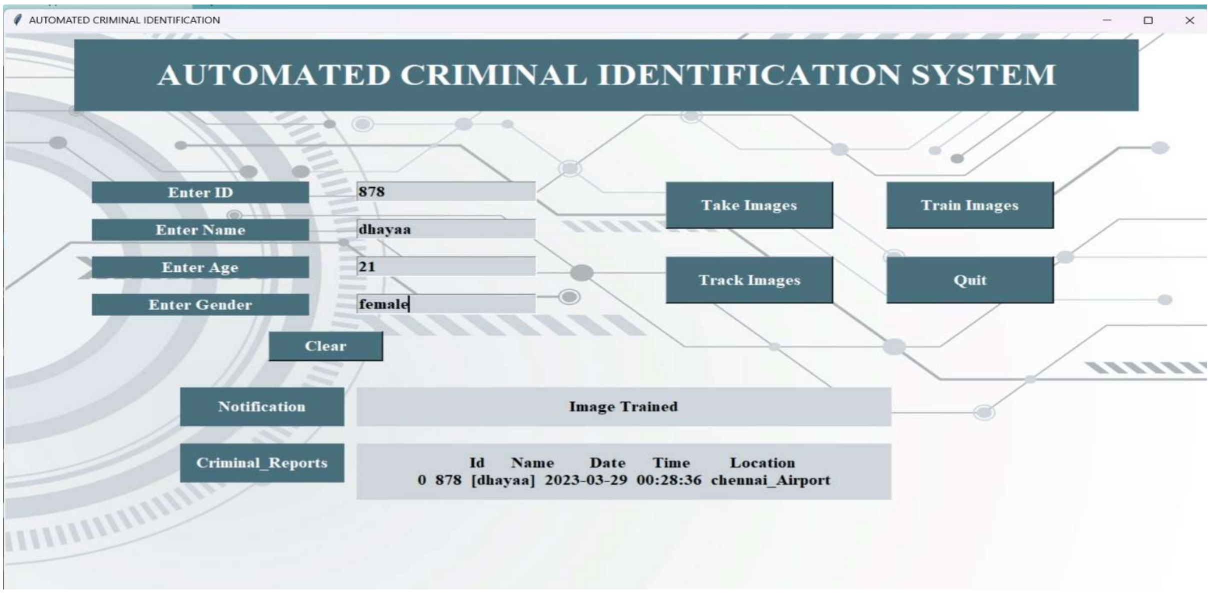
['878—dhayaa—WANTED'

AL IDENTIFICATION SYSTEM

Notification: Image Trained

Fig A.1.8 FACE RECOGNITION OF THE CRIMINAL

A.1.9 if the criminal is detected in criminal reports box it will be notified in detailed



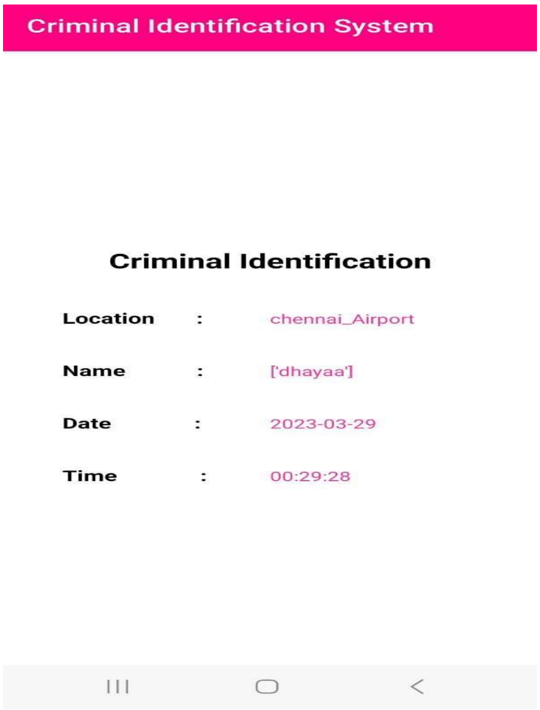
Enter ID: 878
Enter Name: dhayaa
Enter Age: 21
Enter Gender: female

Image Trained

```
   Id    Name         Date        Time      Location
0  878  [dhayaa]  2023-03-29  00:28:36  chennai_Airport
```

Fig **A.1.9** DETAILS MESSAGE OF CRIMINAL REPORTS

A.1.10 If the criminal is detected we receive notification in android application and in criminal reports box



Criminal Identification

Location	:	chennai_Airport
Name	:	['dhayaa']
Date	:	2023-03-29
Time	:	00:29:28

Fig A.1.10 NOTIFICATION OF CRIMINAL IDENTIFICATION THROUGH ANDROID APP

A.1.11 non-trained images will be displayed as non-criminal and following fig is the example of non-criminal

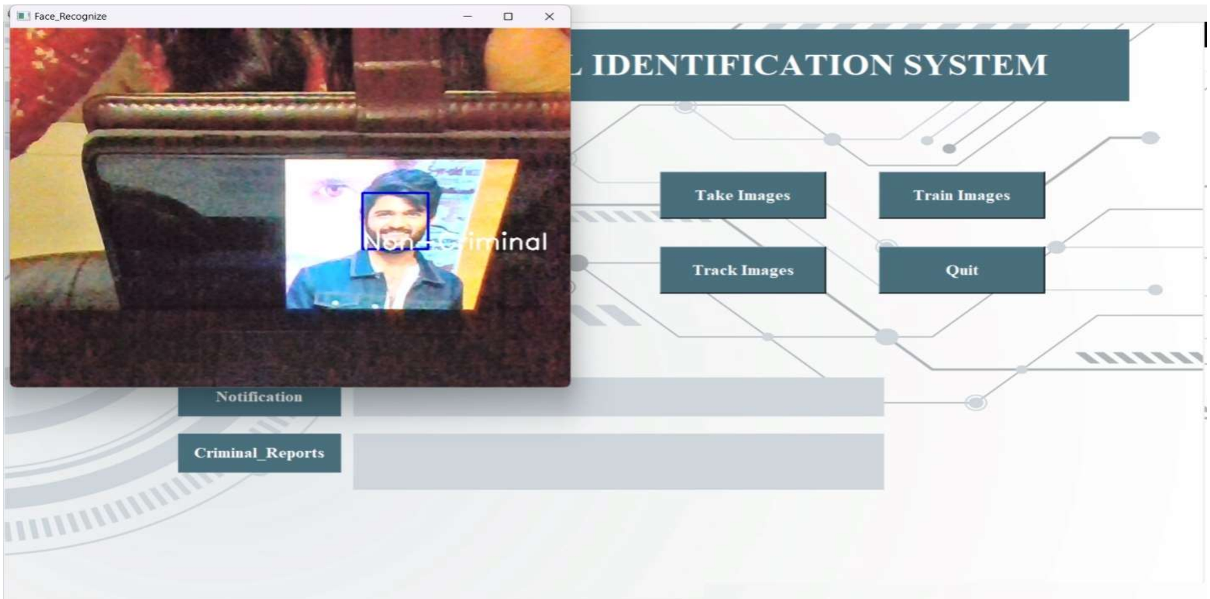


Fig A.1.11 FACE DETECTION OF THE NON-CRIMINAL

REFERENCES

[1] Alireza Chevelwalla," Criminal Face Recognition System", International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 IJERTV4IS030165 ,Vol. 4 Issue 03, March-2015.

[2] Kaipeng Zhang, "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks" ,in IEEE Signal Processing Letters (Volume:23, Issue:10,October 2016), DOI**:** 10.1109/LSP.2016.2603342.

[3] Nurul Azma Abdullah, "Face Recognition for Criminal Identification: An implementation of principal component analysis for face recognition", in the 2nd International Conference on Applied Science and Technology 2017 (ICAST'17) AIP Conf. Proc. 1891, 020002-1–020002-6; AIP Publishing.978-0-7354-1573-7.

[4] Piyush Kakkar, "Criminal Identification System Using Face Detection and Recognition", in the International Journal of Advanced Research in Computer and Communication Engineering ISO 3297:2007 Certified Vol. 7, Issue 3, March 2018, DOI 10.17148/IJARCCE.2018.7346.

[5] Archana Naik, "Criminal identification using facial recognition", in the International Journal of Advance Research, Ideas and Innovations in Technology, ISSN: 2454-132X Impact factor: 4.295 (Volume 5, Issue 3), 2019.

[6] Piyush Chhoriya, "Automated Criminal Identification System using Face Detection and Recognition", international Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 06 Issue: 10, Oct 2019.

[7] Aakriti Singhal, "Criminal Face Detection System", International Journal of Advance Research and Innovation, Volume 9 Issue 2 (2021) 188-191.

[8] D.Nagamallika, “ Criminal Identification System Using Deep Learning”, JETIR July 2021, Volume 8, Issue 7, JETIR2107217.

[9] Ganta Tejaswini, ”ONLINE CRIMINAL IDENTIFICATION USING ML & FACE RECOGNITION TECHNIQUES”, JETIR December 2021, Volume 8, Issue 12, JETIR2112098.

[10] KH Teoh, “Face Recognition and Identification using Deep Learning Approach”, 5th International Conference on Electronic Design (ICED) 2020 Journal of Physics: Conference Series 1755 (2021) 012006 IOP Publishing doi:10.1088/1742-6596/1755/1/012006.

[11] Nagnath B. Aherwadi, “Criminal Identification System using Facial Recognition”, Aherwadi, (July 12, 2021). Proceedings of the International Conference on Innovative Computing & Communication (ICICC) 2021.

[12] Saniya Prashant Patil, “CRIMINAL IDENTIFICATION FOR LOW RESOLUTION SURVEILLANCE”, VIVA Institute of Technology 9 th National Conference on Role of Engineers in Nation Building – 2021 (NCRENB-2021), Volume 1, Issue 4 (2021).