CRIMINAL IDENTIFICATION AND WEAPON DETECTION

Submitted in partial fulfilment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

KARANAM SHIVA SHANKAR (Reg.No - 39110453)
KARRI BHASKAR REDDY(Reg.No - 39110458)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119

APRIL - 2023

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **KARANAM SHIVA SHANKAR (Reg.No - 39110453) and KARRI BHASKAR REDDY (Reg.No - 39110458)** who carried out the Project Phase-2 entitled **"CRIMINAL IDENTIFICATION AND WEAPON DETECTION"** under my supervision from January 2023 to April 2023.

J. Refonaa

Internal Guide

Dr. J.REFONAA, M.E., Ph.D.

[illegible]

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY
DEPARTMENT OF CSE

Submitted for Viva voce Examination held on 25.4.2023

J. Refonaa

[illegible]

Internal Examiner **External Examiner**

DECLARATION

I**, KARANAM SHIVA SHANKAR (Reg.No: 39110453),** hereby declare that the Project Phase-2 Report entitled **"CRIMINAL IDENTIFICATION AND WEAPON DETECTION"** done by me under the guidance of **Dr. J . REFONAA, M.E.,Ph.D.,** is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE: 25-04-2023

[illegible]

PLACE: CHENNAI **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph. D**, **Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.,** Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. J . REFONAA, M.E.,Ph.D.,** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

The primary distinguishing feature of this project is that it aims to moderate crimes by capturing and identifying criminal faces. Crime prevention and criminal identification are the primary issues before the police personnel, but the availability of police personnel is limited. With the advent of security technology, this project aims to complement the programs and software that are used to detect, record, save and store the identity and the facial features of the criminal. Real time footage of the camera can be used to identify suspects on scene. In this project, an automated facial recognition system for criminal databases is implemented using the well-known Haar feature-based cascade classifier. This system will be able to detect and recognize faces automatically in real time, and also map these faces and collect information from the database, displaying the required criminal files as well. The computer language used in this project is python, and the primary library is OpenCV. OpenCV is a set of library functions that are aimed at real time computer vision. Other notable features of this criminal identification system are face detection, face identification, criminal database, graphical user interface, real time video probing and criminal file uploading. The future scope of this project also includes incorporating this along with other notable criminal security softwares and to broaden its range.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Human beings have recognition capabilities that are unparalleled within the modern computing era. These are mainly because of the high degree of interconnectivity, adaptive nature, learning skills and generalization capabilities of the system. The human brain has numerous highly interconnected biological neurons which, on some specific tasks, can outperform supercomputers. Advancements in computing capability over the past few decades have enabled comparable recognition capabilities from such engineered systems quite successfully. Early recognition algorithms used simple geometric models, but recently the popularity process has now matured into a science of sophisticated mathematical representations and matching processes. Major advancements and initiatives have propelled face recognition technology into the spotlight. Recognition technology are often employed in a wide selection of applications. Computers that detect and recognize faces can be applied to a good type of practical applications including criminal identification etc. Face detection and recognition is employed in many places nowadays, verifying websites hosting images and social networking sites. Crime recognition and detection is achieved using technologies associated with computing. Features extracted from a weapon are processed and compared with similarly processed faces present within the database. If a weapon is recognized it's known or the system may show an identical weapon existing in the database else it's unknown. In closed-circuit television if an unknown face appears quite just once then it's stored in a database for further recognition. These steps are very useful in criminal identification. In general, crime recognition techniques are divided into two groups supporting the face representation: they use appearance-based, which uses holistic texture features and is applied to either whole-face or specific face image and feature-based, which uses geometric facial features and geometric relationships between them.

1.1 AIM OF THE PROJECT

The main goal of this project is to identify the criminal and register into the database and also detect the video survillences crime which is going on in real time. The advanced bringing up of security technology, this application is bringing forward to implement the programs and software that are used to detect, record, preserved and store the identities of the person and the facial features of the criminal The system will consist of several basic elements

- Python IDLE
- OpenCV
- Database connectivity with MYSQL
- Kaggle Data set
- GUI interface
- Web cam

1.2 USERS AND STAKEHOLDERS

The design phase goal is to acquire a clear grasp of what the government expects from their project. The overall system design aim is to provide an efficient, modular design that will reduce complexity for the client, facilitate change and result in an easy implementation. This will be accomplished by developing a strong cohesion system with minimal connections. The system necessities and logical description of the entities, relationships and attributes of the info that were documented throughout the need analysis section are additionally refined and allotted into system and info style specifications that are organized during an approach appropriate for implementation at intervals the constraints of a physical setting. Prior to the detailed design of the application, a formal high-level architectural design review is performed to ensure that the design meets the Client's requirements, conforms to the organization's architecture, and meets established design standards, and to raise and resolve any significant issues related to the technical project.

1.2.1 SELF

I will be developing, maintaining, and testing the crime detection and recognition application through its phases of development.

1.2.2 USERS

The users are anyone who has the program for the crime recognition and detection. A user may also be a bystander who happens to surpass. This person's body may have accidentally shown itself within the camera. Nonetheless, he or she's going to even be detected if he or she possesses a weapon.

1.2.3 USE CASES

These are the utilization cases for the client of the criminal identification and detection program. The programmer has access to all or any of those cases likewise.

1.2.3.1 CAMERA WINDOW

The program has code to open the inbuilt camera of a system. It works on any camera, and also on specific camera apps. A resizable window opens and therefore the user’s weapon is seen in color together with the background.

1.2.3.2 TRAINER PROGRAM

The trainer program enables the user to coach his or her face by spotlight before the camera. The python idle is employed to run this code. Half a second is taken for the trainer program to run its course, during which period it takes double the copies of the user's face and stores it within the database folder that has a name of the criminal.

1.2.3.3 USERS

The users are anyone who appears before the camera during execution time.

1.2.3.4 USER PROMPT

The user prompt is that the python is idle. It allows the user to enter into both the modules. It has two buttons. One is for real time crime detection and the other is for entering criminal information in a database.

CHAPTER 2

LITERATURE SURVEY

In the year 2012 the face recognition “A Robust Face Recognition method using edge based features”,Symposium on Computers and Informatics. In this technique and algorithm on face feature extraction and face recognition, the true positive means the portion of face image to be detected by the system is the distance vector of the labeled edge between two vertices.

Rapid Object Detection using a Boosted Cascade of Simple Features by Paul Viola, Michael Jones This object detection was proposed by Paul Viola and Michael Jones. Although it can be trained to detect a variety of object classes, it was motivated primarily by the problem of face detection.

An automated technique for criminal face identification using biometric approach”, on Advances in Communication and Control Systems by Jyoti Nautiyal, Shivali Gahlot and Pawan Kumar Mishra, “.This proposed system was using biometric to detect the criminal .This takes a long process to identify the criminal by using his finger prints. Fingerprints have been used in criminal investigations as a means of identification for centuries. It is one of the most important tools of crime detection because of their robustness and uniqueness. A fingerprint is the pattern of friction ridges and valleys on the surface of a fingertip.

2.1 OPEN PROBLEMS IN EXISTING SYSTEM

Robust – very high detection rate (true-positive rate) & very low false-positive rate always.

Real time – For practical applications at least 2 frames per second must be processed.

Face detection only (not recognition) - The goal is to distinguish faces from non-faces (detection is the first step in the recognition process).

The algorithm has four stages:

- Haar Feature Selection

- Creating an Integral Image
- Adaboost Training
- Cascading Classifiers

The features sought by the detection framework universally involve the sums of image pixels within rectangular areas. As such, they bear some resemblance to Haar basis functions, which have been used previously in the realm of image-based object detection. However, since the features used by Viola and Jones all rely on more than one rectangular area, they are generally more complex. The figure on the right illustrates the four different types of features used in the framework. The value of any given feature is the sum of the pixels within clear rectangles subtracted from the sum of the pixels within shaded rectangles. Rectangular features of this sort are primitive when compared to alternatives such as steerable filters. Although they are sensitive to vertical and horizontal features, their feedback is considerably coarser.

CHAPTER 3

DESCRIPTION OF PROPOSED SYSTEM

3.1 PROBLEM STATEMENT:

This project aims to make a criminal identification and crime detection program which will successfully detect crime along with providing a useful user interface with a database for entering criminal information. This is done by using python and opencv together with deep learning concepts. The database is preprocessed for removal of noise and to reduce redundancy. Feature extraction is done with Haar techniques. Database consists of pictures and details of citizens in the given country.It also has pictures and information regarding criminals belonging to that country. The database also includes information on criminals who are not citizens of the nation. When you enable the footage, it will be converted to frames. As a frame identifies a face, it reduces noise and redundancy. Then feature extraction occurs, which is when Haar cascade comes into play.It allows you to run the identical code on multiple platforms without recompilation. The image obtained is compared to the faces already in the criminal database. If the faces are similar, they are compared to photographs recorded in the local watch list database to determine if the current person is a criminal or not. If he is a criminal, the duration of his appearance in front of the camera will be recorded. If he is not a resident of the area, the data will be compared to that on the global suspects database list. If a match is discovered, the time length is recorded. If no match is discovered on both lists, he is presumed innocent.

3.2 AIM AND SCOPE

The aim of the criminal identification system is to capture and aid police personnel in identifying and effectively detecting weapons and crime. The scope of those programs extends to a variety of individuals who appear within the camera.

3.3 TARGET AUDIENCE

The audience is someone who wants to use a crime recognition program that occupies less space and is user friendly.

3.4 PROGRAMMING LANGUAGE

Python is an interpreted, dynamically-typed, object-oriented scripting language with a number of built-in data types. The Python interpreter works by loading a source file or reading a line typed at the keyboard, parsing it into an abstract syntax tree, compiling the tree into bytecode, and executing the bytecode. While writing a software application, you need to concentrate on the standard of its ASCII text file to simplify maintenance and updates. The syntax rules of Python allow you to specific concepts without writing additional code. At the same time, Python, unlike other programming languages, emphasizes on code readability, and allows you to use English keywords rather than punctuations.

Hence, you'll be able to use Python to make custom applications without writing additional code. The readable and clean code base will facilitate your to keep up and update the software without putting time beyond regulation and energy.Like other modern programming languages, Python also supports several programming paradigms. It supports object oriented and structured programming fully. Also, its language features support various concepts in functional and aspect-oriented programming. At the same time, Python also features a dynamic type system and automatic memory management. The programming paradigms and language features facilitate your use of Python for developing large and complicated software applications.At present, Python supports many operating systems. you'll be able to even use Python interpreters to run the code on specific platforms and tools. Also, Python is an interpreted artificial language. It allows you to run the identical code on multiple platforms without recompilation. Hence, you're not required to recompile the code after making any alteration. you'll run the modified application code without recompiling and check the impact of changes made to the code immediately. The feature makes it easier for you to create changes to the code without increasing development time. Its large and robust standard library makes Python score over other programming languages. the quality library allows you to decide on a large range of modules in keeping with your precise needs. Each module further enables you to feature functionality to the Python application without writing additional code. For example, while writing an internet application in Python, you'll be able to use specific modules to implement web services, perform string operations, manage package interface or work with internet protocols. you'll be able to even gather information about various modules by browsing through the Python Standard Library documentation. As an open source programming language, Python helps you to curtail software development cost significantly. you'll be able to even use several open source Python frameworks, libraries and development tools to curtail development time without increasing development cost. You even have the choice to select from a large range of open source Python frameworks and development tools in step with your precise needs. For example, you'll simplify and speedup web application development by using robust Python web frameworks like Django, Flask, Pyramid, Bottle and Cherrypy. Likewise, you'll be able to accelerate desktop GUI application development using Python GUI frameworks and toolkits like PyQT, PyJs, PyGUI, Kivy, PyGTK and WxPython. Python may be a general purpose artificial language. Hence, you'll be able to use the artificial language for developing both desktop and web applications. Also, you'll be able to use Python for developing complex scientific and numeric applications. Python is intended with features to facilitate data analysis and visualization. you'll benefit from the information analysis features of Python to form custom big data solutions without putting beyond regular time and energy. At the same time, the info visualization libraries and APIs provided by Python facilitate your to visualize and present data in an exceedingly more appealing and effective way. Many python developers even use Python to accomplish AI (AI) and tongue processing tasks. You can use Python to make a prototype of the software application rapidly. Also, you'll build the software application directly from the prototype just by refactoring the Python code. Python even makes it easier for you to perform coding and testing simultaneously by adopting test driven development (TDD) approach. you'll be able to easily write the desired tests before writing code and use the tests to assess the applying code continuously. The tests may also be used for checking if the application meets predefined requirements supporting its ASCII text file. However, Python, like other programming languages, has its own shortcomings. It lacks a number of the built-in features provided by other modern programming languages. Hence, you've got to use Python libraries, modules, and frameworks to accelerate custom software development. Also, several studies have shown that Python is slower than several widely used programming languages including Java and C++. you've got to hurry up the Python application by making changes to the applying code or using custom runtime. But you'll always use Python to hurry up software development and simplify software maintenance.

3.4.1 OPENCV

OpenCV may be a library of programming functions mainly aimed toward real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez. The library is cross-platform and free to be used under the open-source BSD license. OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to supply a standard infrastructure for computer vision applications and to accelerate the employment of machine perception within the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has over 2500 optimized algorithms, which incorporates a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms is accustomed detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to supply a high resolution image of a complete scene, find similar images from a picture database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. OpenCV has over 47 thousand users and an estimated number of downloads exceeding 18 million. The library is employed extensively in companies, research groups and by governmental bodies. Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups like Applied Minds, VideoSurf, and Zeitera, that make extensive use of OpenCV. OpenCV's deployed uses span the range from stitching street view images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and develop objects at Willow Garage, detection of swimming bath drowning accidents in Europe, running interactive art in Spain and the big apple, checking runways for debris in Turkey, inspecting labels on products in factories round the world on to rapid face detection in Japan. It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed at once. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and features a templated interface that works seamlessly with STL containers.

3.4.2 DEEP LEARNING

Deep learning is a synthetic intelligence (AI) function that imitates the workings of the human brain in processing data and creating patterns to be used in higher cognitive processes. Deep learning could be a subset of machine learning in computer science that has networks capable of learning unsupervised from data that's unstructured or unlabeled. Also referred to as deep neural learning or deep neural network.

Deep learning is an AI function that mimics the workings of the human brain in processing data to be used in detecting objects, recognizing speech, translating languages, and making decisions.

Deep learning AI is ready to find out without human supervision, drawing from data that's both unstructured and unlabeled.

Deep learning, a type of machine learning, is accustomed to help detect fraud or hiding, among other functions.

Deep learning has evolved hand-in-hand with the digital era, which has led to an explosion of knowledge in all forms and from every region of the globe. This data, known simply as big data, is drawn from sources like social media, internet search engines, e-commerce platforms, and online cinemas, among others. This enormous amount of information is instantly accessible and may be shared through fintech applications like cloud computing.

However, the data, which normally is unstructured, is so vast that it could take decades for humans to grasp it and extract relevant information. Companies realize the incredible potential that may result from unraveling this wealth of data and are increasingly adapting to AI systems for automated support. Deep learning unravels huge amounts of unstructured data that might normally take humans decades to grasp and process.

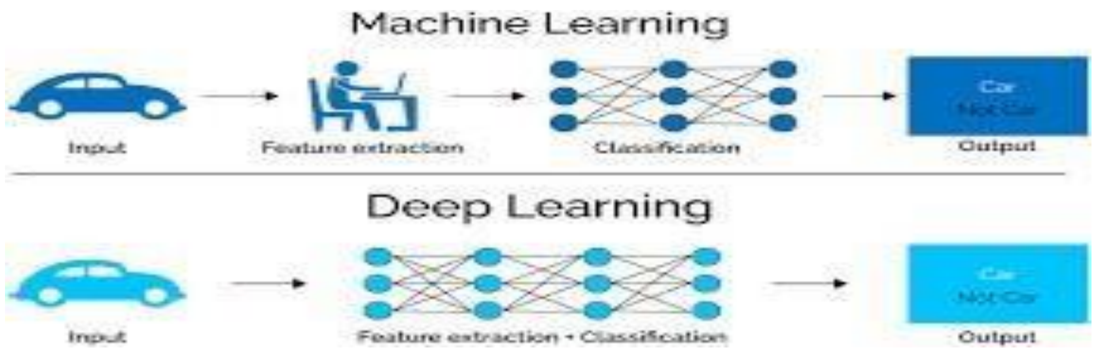


FIGURE 3.1: DEEP LEARNING

3.5 DEEP LEARNING VS MACHINE LEARNING

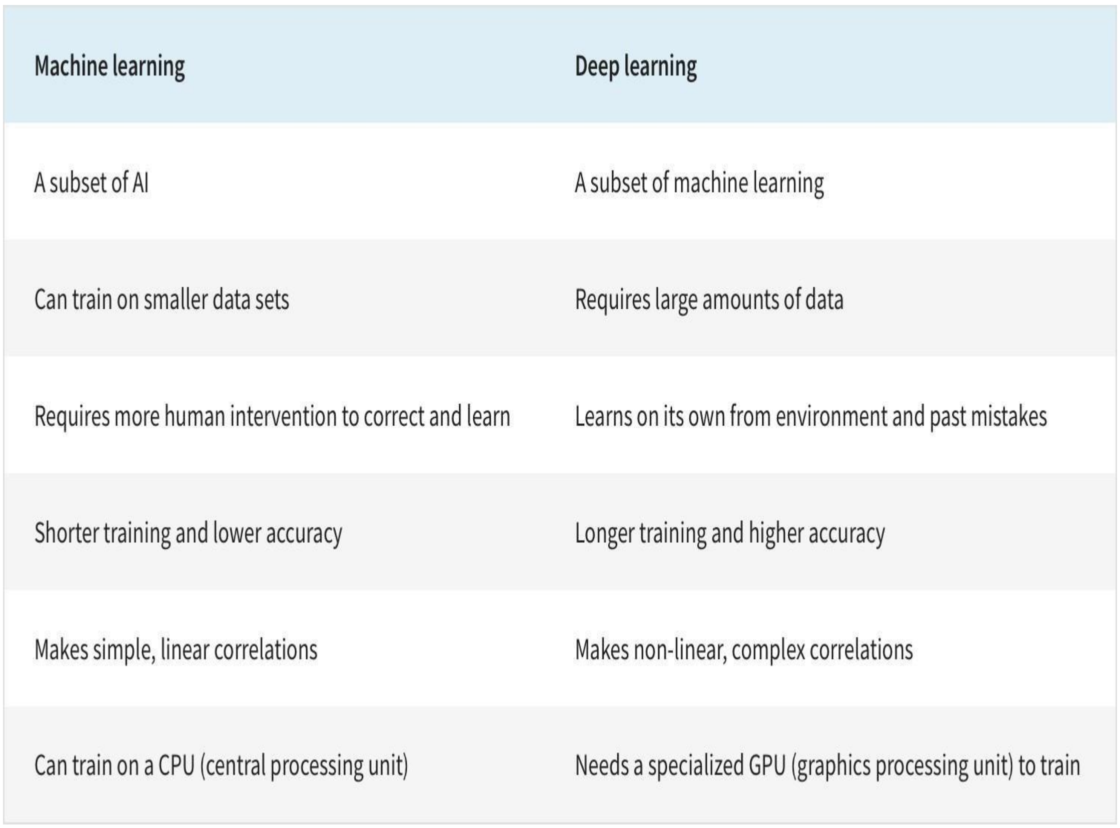
One of the foremost common AI techniques used for processing big data is machine learning, a self-adaptive algorithm that gets increasingly better analysis and patterns with experience or with newly added data. If a digital payments company wanted to detect the occurrence or potential for fraud in its system, it could employ machine learning tools for this purpose. The computational algorithm built into a computer model will process all transactions happening on the digital platform, find patterns within the data set, and imply any anomaly detected by the pattern. Deep learning, a subset of machine learning, utilizes a hierarchical level of artificial neural networks to hold out the method of machine learning. The substitute neural networks are built just like the human brain, with neuron nodes connected together sort of a web. While traditional programs build analysis with data in a very linear way, the hierarchical function of deep learning systems enables machines to process data with a nonlinear approach.Machine learning and deep learning are both types of AI. In short, machine learning is AI that can automatically adapt with minimal human interference. Deep learning is a subset of machine learning that uses artificial neural networks to mimic the learning process of the human brain.

Take a look at these key differences before we dive in further.

Machine learning refers to the study of computer systems that learn and adapt automatically from experience, without being explicitly programmed.

With simple AI, a programmer can tell a machine how to respond to various sets of instructions by hand-coding each “decision.” With machine learning models, computer scientists can “train” a machine by feeding it large amounts of data. The machine follows a set of rules—called an algorithm—to analyze and draw inferences from the data.

The more data the machine parses, the better it can become at performing a task or making a decision.Here’s one example you may be familiar with: Music streaming service Spotify learns your music preferences to offer you new suggestions. Each time you indicate that you like a song by listening through to the end or adding it to your library, the service updates its algorithms to feed you more accurate recommendations. Netflix and Amazon use similar machine learning algorithms to offer personalized recommendations.



Machine learning	Deep learning
A subset of AI	A subset of machine learning
Can train on smaller data sets	Requires large amounts of data
Requires more human intervention to correct and learn	Learns on its own from environment and past mistakes
Shorter training and lower accuracy	Longer training and higher accuracy
Makes simple, linear correlations	Makes non-linear, complex correlations
Can train on a CPU (central processing unit)	Needs a specialized GPU (graphics processing unit) to train

FIGURE 3.2: MACHINE LEARNING VS DEEP LEARNING

CHAPTER 4

REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

This section will cover the functional requirements of the chat application. It deals with the functionalities of the appliance.

4.1.1 IN-BUIT CAMERA

This functional requirement may be a must prerequisite for the face detection program. It is often a camera application or an inbuilt one. The image will appear in colors, only to be changed to grayscale and stored. This is often done in order that the program runs even faster.

4.1.2 DETECTOR

The database prompt, prompts the user to enter the required information. The photographs are stored in .jpg format within the database folder. The detector detects the face/ faces and captures it.

4.1.3 TRAINER

The trainer enables the user to coach his or her images using the algorithm. All the photographs are trained no matter the quantity of it. This image is stored together with the name of the criminal.

4.1.4 RECOGNIZER

The recognizer identifies the weapon shown within the camera. It then displays the positive identification because the required output. Even when multiple weapons are shown, it adheres to it efficiently and displays multiple coloured boxes.

4.1.5 GRAPHICS

A blue blinking square is drawn over the image. This depicts the weapon held and its outline. A message “Weapon Detected” also takes place over the image.

4.2 NON- FUNCTIONAL REQUIREMENTS

These are the nonfunctional requirements of the application. This can be basically the section that deals with the standard of the criminal identification and crime detection application instead of the functionalities of the application.

4.2.1 USER FRIENDLY

The python code itself is user friendly. With only 3 programs spanning over some lines of code, the face recognition and detection program strives to satisfy the users requirements successfully. Upon running the code, the program automatically activates the camera and also the process begins.

4.2.2 VERSATILE

Python and opencv are both open source languages. Although open source is chiefly C++ based, it may be incorporated easily into python. Any system with the subsequent requirements can run the program.

4.3 SYSTEM REQUIREMENTS

4.3.1 SOFTWARE REQUIREMENTS

Software Requirements. The software requirements are descriptions of features and functionalities of the target system. Requirements convey the expectations of users from the software package

• Python version 3.7

• Python IDLE 3.7

• Xampp control panel with Apache and mysql

4.3.2 HARDWARE REQUIREMENTS

In hardware requirements we want all those components which are able to provide us the plat-form for the event of the project. The minimum hardware required for this project is as follows

• CPU configuration :Intel Pentium 3 or later

• RAM capacity 256MB or above

• Storage Minimum 4TB

A good system design is to organize the program modules in such the simplest way that are easy to develop and alter. Structured design techniques help developers to accommodate the scale and complexity of programs. Analysts create instructions for the developers about how code should be written and the way pieces of code should fit together to make a program. Thus the program is split into three different parts.

These are:

• WeaponDetection.py

• Home.py

• Dbconnection.py

`

CHAPTER 5
IMPLEMENTATION DETAILS

5.1 IMPORTANCE

• If any pre-existing code has to be understood, organized and pieced together.

• It is common for the project team to jot down some code and produce original programs that support the appliance logic of the system.

5.2 USE CASE DIAGRAM

A use case diagram at its simplest may be a representation of a user's interaction with the system that shows the connection between the user and therefore the different use cases within which the user is involved.

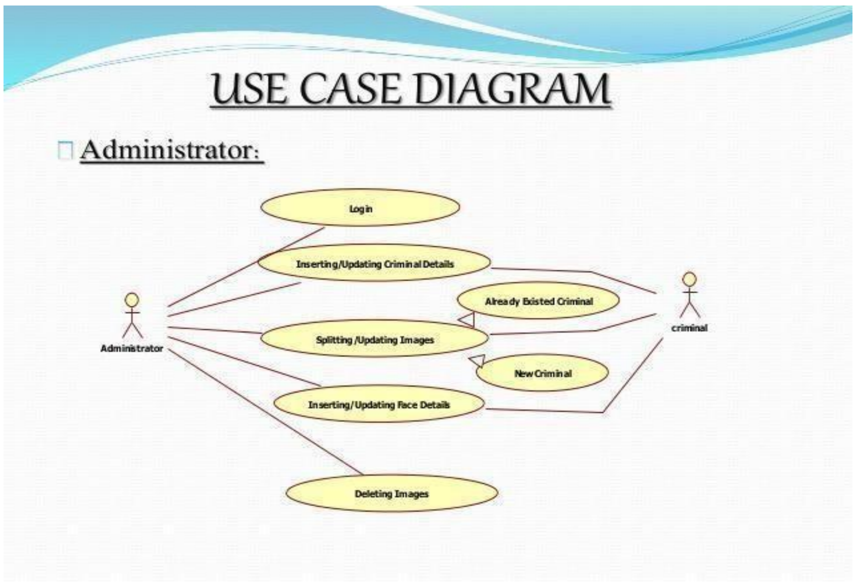
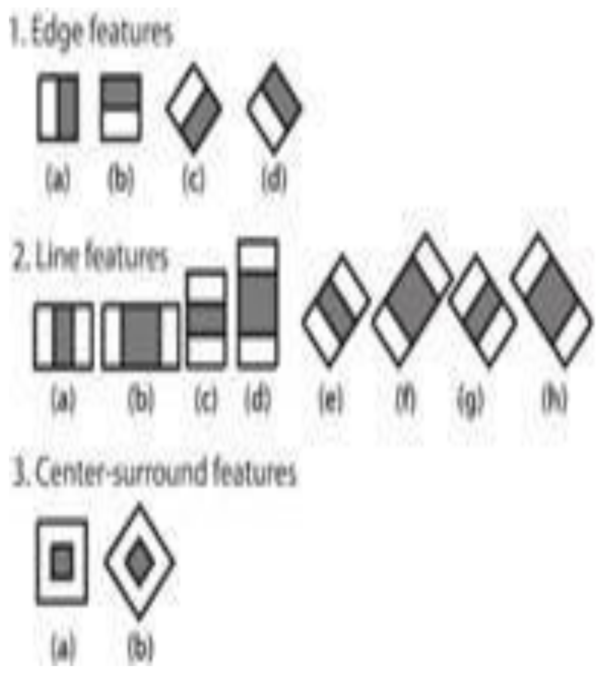


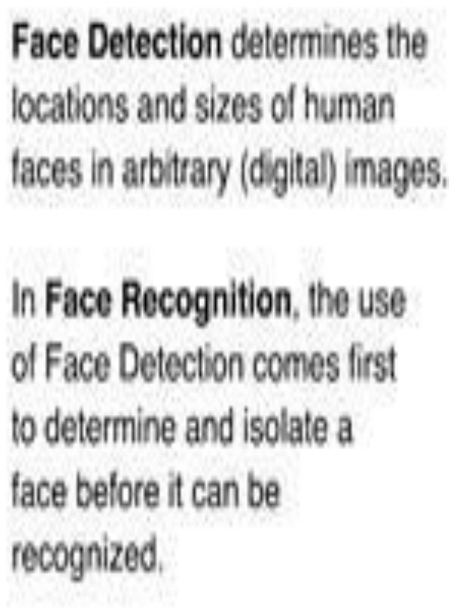
FIGURE 5.1: USE CASE DIAGRAM

5.3 ALGORITHM USED

5.3.1 HAAR CASCADE CLASSIFIER

It is supported by the Haar Wavelet technique to investigate pixels within the image into squares by function. This uses machine learning techniques to induce a high degree of accuracy from what's called “training data”. This uses “integral image” concepts to compute the “features” detected. Haar Cascades use a learning algorithm which selects atiny low number of important features from an outsized set to present efficient results of classifiers.




Face Detection determines the locations and sizes of human faces in arbitrary (digital) images.

In **Face Recognition**, the use of Face Detection comes first to determine and isolate a face before it can be recognized.

FIGURE 5.2 : FACE DETECTION VS RECOGNITION

Sample haar features traverse in window-sized across the picture to compute and match features.Haar cascade works as a classifier. It classifies positive data points → that are part of our detected object and negative data points → that don’t contain our object.

Haar cascades are fast and can work well in real-time.

Haar cascade is not as accurate as modern object detection techniques are.

Haar cascade has a downside. It predicts many false positives.

Simple to implement, less computing power required.

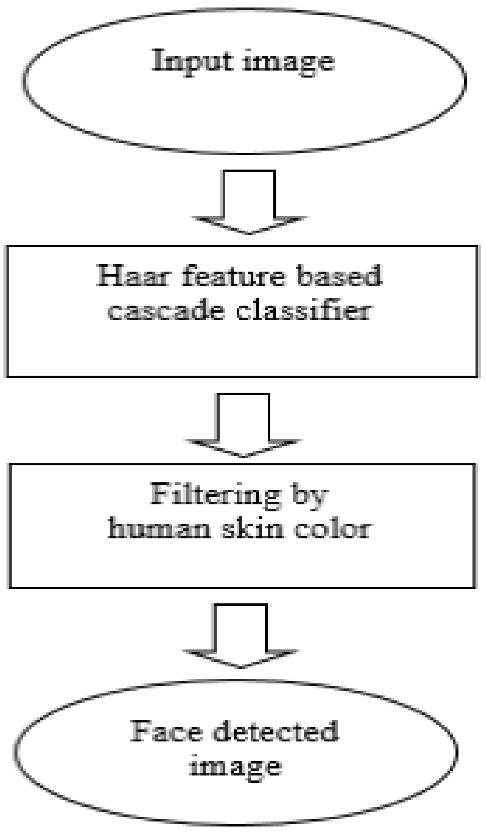

Input image
Haar feature based cascade classifier
Filtering by human skin color
Face detected image

FIGURE 5.3 : HAAR CASCADE CLASSIFIER

5.3.2 FEATURE EXTRACTION

Haar Cascades use machine learning techniques during which a function is trained from plenty of positive and negative images. This process within the algorithm is feature extraction. Here we'll work with face detection. Initially, the algorithm needs lots of positive images (images of faces) and negative images (images without faces) to coach the classifier. Then we want to extract features from it.

The process of feature extraction is useful when you need to reduce the number of resources needed for processing without losing important or relevant information. Feature extraction can also reduce the amount of redundant data for a given analysis. Also, the reduction of the data and the machine's efforts in building variable combinations (features) facilitate the speed of learning and generalization steps in the machine learning process.

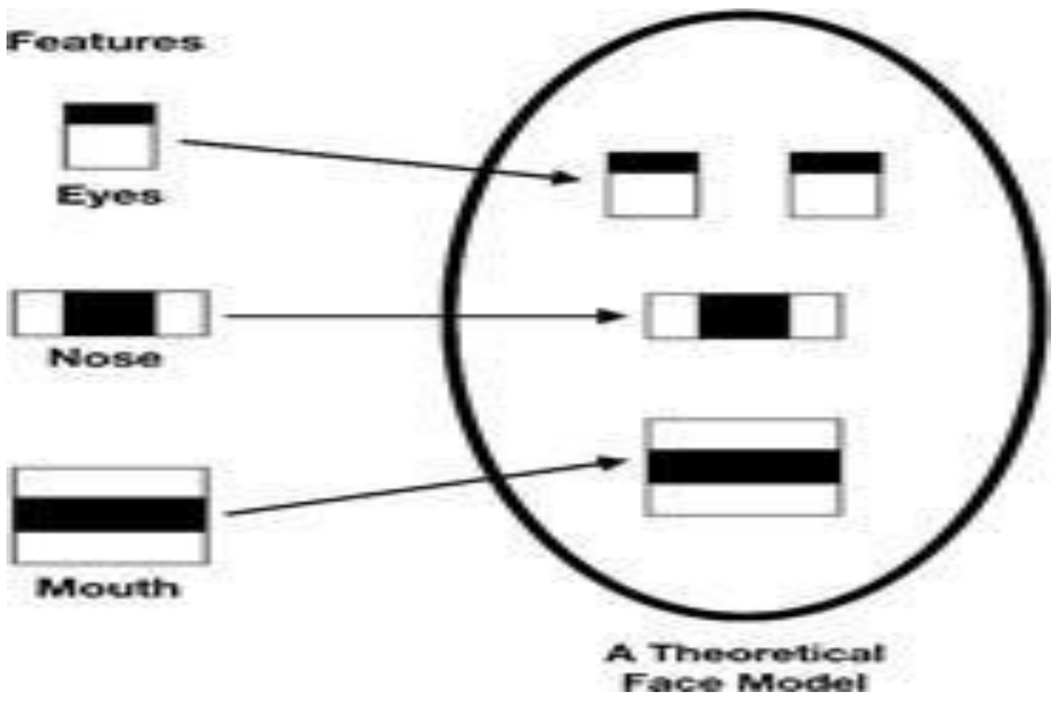


FIGURE 5.4: FEATURE EXTRACTION

Feature extraction can be accomplished manually or automatically:
Manual feature extraction requires identifying and describing the features that are relevant for a given problem and implementing a way to extract those features. In many situations, having a good understanding of the background or domain can help make informed decisions as to which features could be useful. Over decades of research, engineers and scientists have developed feature extraction methods for images, signals, and text. An example of a simple feature is the mean of a window in a signal.
Automated feature extraction uses specialized algorithms or deep networks to extract features automatically from signals or images without the need for human intervention. This technique can be very useful when you want to move quickly from raw data to developing machine learning algorithms. Wavelet scattering is an example of automated feature extraction.
With the ascent of deep learning, feature extraction has been largely replaced by the first layers of deep networks – but mostly for image data. For signal and time-series applications, feature extraction remains the first challenge that requires significant expertise before one can build effective predictive models.

Traditional feature extractors can be replaced by a convolutional neural network(CNN), since CNN’s have a strong ability to extract complex features that express the image in much more detail, learn the task specific features and are much more efficient. Multiple works have been done on this. Few of them are listed below:

SuperPoint

Self-Supervised Interest Point Detection and Description (paper) — The authors suggest a fully convolutional neural network that computes SIFT like interest point locations and descriptors in a single forward pass. It uses an VGG-style encode for extracting features and then two decoders, one for point detection and the other for point description.

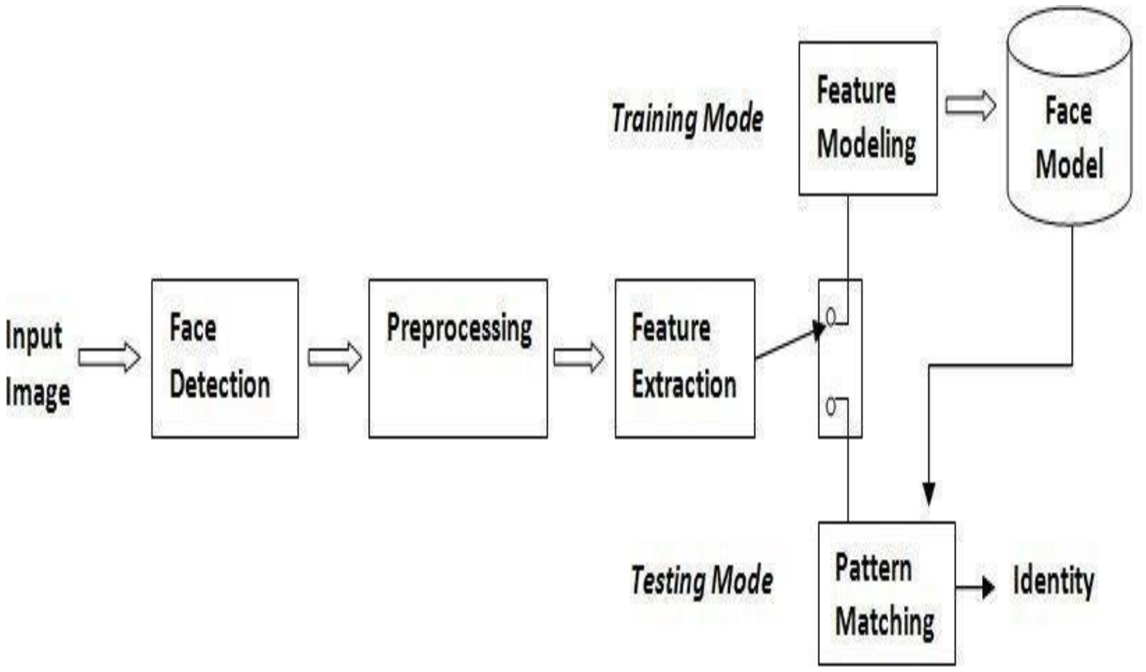


FIGURE 5.5 : FACE RECOGNITION SYSTEM-MODEL

5.4 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 this 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.

5.4.1 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.

5.4.2 Functional Tests

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

- Performance Test
- Stress Test
- Structure Test
- Performance Test

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.
Stress Test
Stress Test is those test designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.
Structure Test

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.
Exercise all logical decisions on their true or false sides.
Execute all loops at their boundaries and within their operational bounds.
Exercise internal data structures to assure their validity.
Checking attributes for their correctness.
Handling end of file condition, I/O errors, buffer problems and textual errors in output information

5.4.3 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. . An alarm will sound if security has been breached and this alarm will sound only for two seconds. Two alarms may overlap if there is multiple weaponry.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.

5.4.4 TESTING TECHNIQUES / TESTING STRATEGIES

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. A good test case design is one that as a probability of finding an yet undiscovered error System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding an yet undiscovered error. A successful test is one that uncovers an yet undiscovered error. Any engineering product can be tested in one of the two ways:

5.4.5 White box testing

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

5.4.6 Basis path testing:

Flow graph notation

Cyclometric complexity

Deriving test cases

Graph matrices Control

5.4.7 Black box testing

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software. The steps involved in black box test case design are:

Graph based testing methods

Equivalence partitioning

Boundary value analysis

Comparison testing

5.5 SOFTWARE TESTING STRATEGIES:

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

The developer of the software and an independent test group conducts testing.

Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

5.5.1 Integration Testing

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when we put them together. There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

5.5.2 Program Testing

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output.

Condition testing exercises the logical

conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

Security Testing

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

Validation Testing

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

The function or performance characteristics confirm to specifications and are accepted.

A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation testing and found tobe working satisfactorily. Though there were deficiencies in the system they were not catastrophic

5.6 CRIME IDENTIFICATION AND DETECTION

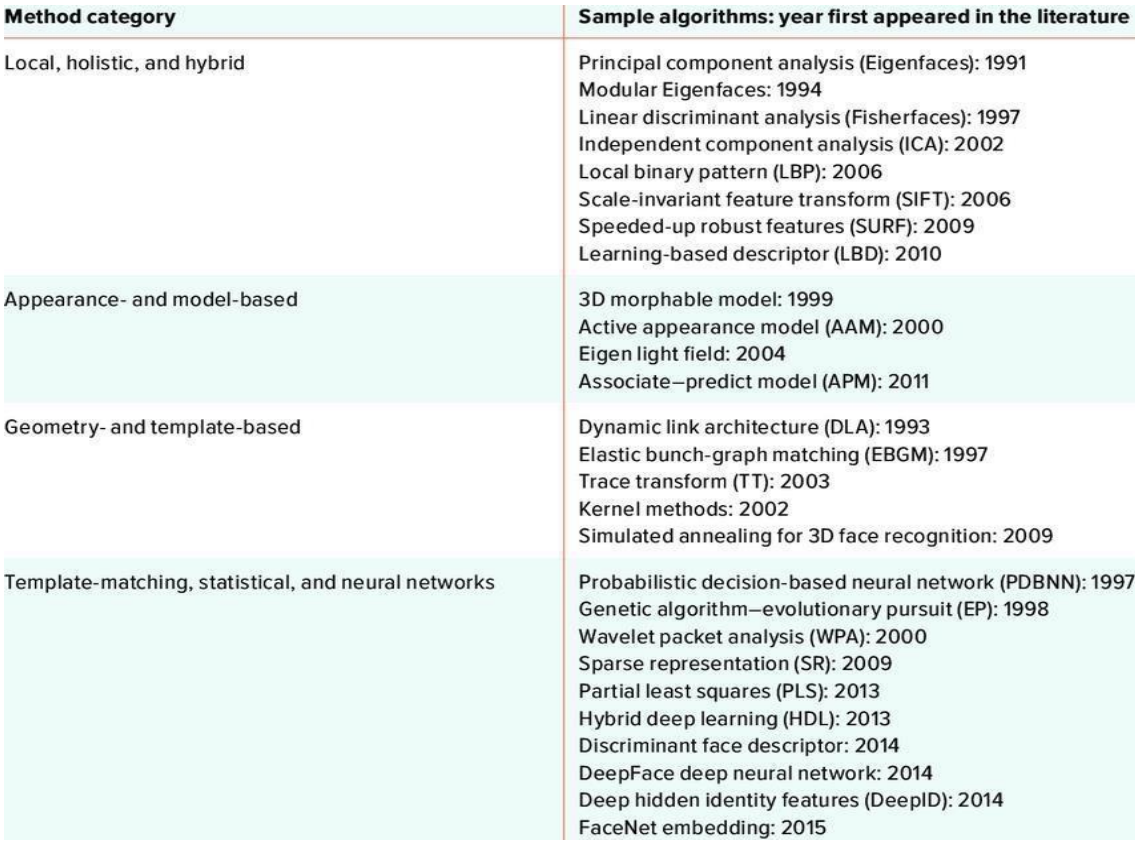
Here two folders are made and inserted into the Cascade classifier GUI.

These are the n folder consisting of negative images and the p folder consisting of positive images. A .xml file is then created that has the dataset coded into it. This is then used to detect weapons like knives and guns.

The intended criminal recognition system is implemented here using the Haar cascade approach. This system contains an image processing capability that allows the input picture to be rendered less fuzzy so that the system can recognise faces on poorer quality photos using black and white pixels. Although if input image does not match with stored image of database then the input image will be classified by using LBPH (Local Binary Pattern Histogram). This proposed system also have a video surveillance where crime can be detected. Testing begins at the module level and works “outward” toward the integration of the entire computer based system.

Different testing techniques are appropriate at different points in time.

This works on training the system with the crime data sets taken from kaggle which is stored in form of XML file. The future scope of this project also includes incorporating this along with other notable criminal security software, and to broaden its range and scope. Advanced MachineLearning algorithms can contribute to a more accurate output



Method category	Sample algorithms: year first appeared in the literature
Local, holistic, and hybrid	Principal component analysis (Eigenfaces): 1991 Modular Eigenfaces: 1994 Linear discriminant analysis (Fisherfaces): 1997 Independent component analysis (ICA): 2002 Local binary pattern (LBP): 2006 Scale-invariant feature transform (SIFT): 2006 Speeded-up robust features (SURF): 2009 Learning-based descriptor (LBD): 2010
Appearance- and model-based	3D morphable model: 1999 Active appearance model (AAM): 2000 Eigen light field: 2004 Associate–predict model (APM): 2011
Geometry- and template-based	Dynamic link architecture (DLA): 1993 Elastic bunch-graph matching (EBGM): 1997 Trace transform (TT): 2003 Kernel methods: 2002 Simulated annealing for 3D face recognition: 2009
Template-matching, statistical, and neural networks	Probabilistic decision-based neural network (PDBNN): 1997 Genetic algorithm–evolutionary pursuit (EP): 1998 Wavelet packet analysis (WPA): 2000 Sparse representation (SR): 2009 Partial least squares (PLS): 2013 Hybrid deep learning (HDL): 2013 Discriminant face descriptor: 2014 DeepFace deep neural network: 2014 Deep hidden identity features (DeepID): 2014 FaceNet embedding: 2015

FIGURE 5.6: TYPES OF RECOGNITION ALGORITHM

Pattern Recognition has been attracting the attention of scientists across the world. In the last decade, it has been widespread among various applications in medicine, communication systems, military, bioinformatics, businesses, etc. Pattern recognition can be defined as the recognition of surrounding objects artificially. Naturally, the process of recognition is complex task artificially. This is accomplished in machines via machine learning and pattern recognition specific algorithms. Pattern Recognition gives the solution to problems like facial expressions recognition, speech recognition, classification, healthcare, GIS, remote sensing, image analysis, etc

STATISTICAL MODEL

In this model, the pattern is termed in the form of features. These Features are selected in a way that different patterns take space without overlapping. It is able to predict and recognize the probabilistic nature. It works so nicely that the selected features are helping the formation of clusters. It analyses the probability distribution, decision boundaries, etc., for the patterns. The machine learns and adapts accordingly. Then these patterns are projected to further processing, training. Then we apply testing patterns for recognition of patterns. This leads to further classification methods. The various schemes used in it are Baye’s Decision Rule, PCA, etc.

TEMPLATE-MATCHING

The model of Template matching is simplest. It is most primitive of all the models. The model is used to determine similarity among two images. The pattern matched is being stored in templates, and the templates are given flexibility for scalar and rotational changes. The competence of this model relies upon the already stored templates in the database. We take correlation function to be the function of recognition in this case, and later it is optimized according to the availability of the training set. The only problem with this model is that this approach is not as efficient while working in distorted patterns.

NEURAL NETWORKS

Neural networks are the most widely used model. They are composed of parallel structures or subunits called neurons. They are efficiently used in Classification. The property of changing the weights repeatedly on iteration patterns, learning abilities, gives this model an edge of competence over other existing models. Perceptron is one of the oldest neuron models. It is basically a two-layered structure. If we find the output function as a step, then it does classification problems. If this is linear, then it is supposed to solve the regression proble.The very commonly used is Feed-Forward Backpropagation neural networks, also acronym as FFBPNN. The variety of neural networks is used for different tasks in recognition of patterns and requirement function. The performance of the neural networks improves as the numbers increase for hidden layers. The number of neurons should also be large to be able to represent the problem and find the patterns hidden in them. Thus there is a requirement of the trade-off between size and complexity of the network.

5.6.1 FLOWCHART

Given below is the flowchart for the entire process.

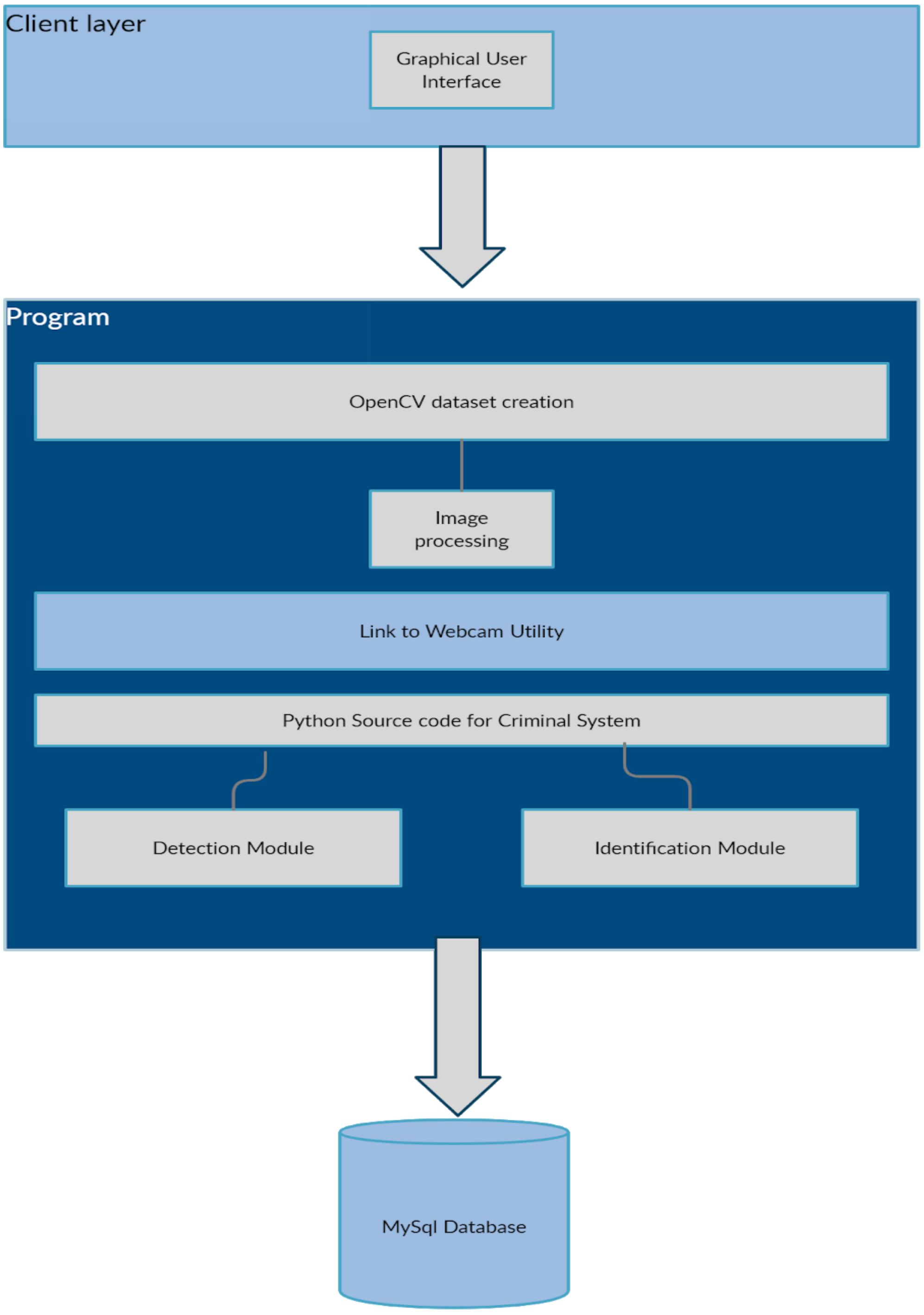


FIGURE 5.7 FLOWCHART

CHAPTER - 6

RESULTS AND DISCUSSION

6.1 RESULTS AND DISCUSSION

The final results of this crime detection software program are shown clearly. Any object that may be or closely resemble a weapon is detected by a red square and is shown along with the graphics. Here, a weapon is defined as an object that can cause harm or injury to any of the persons involved in close proximity to it. A gun, a pistol, or a hand knife may be included with definition. Guns are black or brown coloured in nature and knives are stainless steel or silver coloured. The red square will take the size of the entire weapon, and if the weapon is a hand held object, then portions of a human hand may or may not be included in the preview. One or many weapons will be detected real time. Multiple gadgets can be identified and can also remain overlapping. In this case, multiple red squares will be there and all the red squares will be shown on top of each other. An alarm will sound if security has been breached and this alarm will sound only for two seconds. Two alarms may overlap if there is multiple weaponry.

A video in full color will be shown as it is a real time motion capture. Multiple objects may stand in front of it and objects here may define any human, animal, sky, land or other inanimate objects. These are found in the negative n folder of our dataset and are used to get the required result. Other than the crime identification, the results of this crime detection and identification project solely lean on displaying criminal details and this is done using the Graphical user interface. This is a supplement to the already displayed results, i.e, live monitoring of criminal activities. Various columns and rows are displayed that are understandable to the user. This enables the user to freely scan through and check for criminal records. The window can be minimized and maximized according to the user’s will and the sub-results are displayed. Name, gender, age, criminal activity and other important details pertaining to the criminal can be found at ease and are displayed accurately and smoothly. Five images, if previously uploaded by the user, are displayed and can be moved back and forth for ease of access. Various minor details about the user can also be found.

For example, an identification mark for the criminal can be mentioned during datainput. The graphical user interface has many features and these are given here- buttons, labels, scrollable bars, window, and text input areas. These features greatly enhance the results of the program as a whole and even complement the main criminal detection module. Thus the results of crime identification and detection software are explained in detail.

6.2 PERFORMANCE ANALYSIS

This application could even perform better if given more time duration to do and it can also be further improved for the sake of Police, which adds new features and functionalities to this application.I want to improve The project for the future with the added features like we can even create some alarm or some signal while crime is going on, with the added features and security The application could perform better in any condition with high accuracy.

CHAPTER 7

SUMMARY AND CONCLUSIONS

The proposed system on criminal recognition is based on the Haar cascade technique implemented here. This system has an image processing feature where the input image can be made less blurry so that the system can detect faces on lower quality images based on the pixels of the image in black and white. Although if the input image does not match with the stored image of the database then the input image will be classified by using LBPH (Local Binary Pattern Histogram).This proposed system also has video surveillance where crime can be detected. This works on training the system with the crime data sets taken from kaggle which is stored in the form of an XML file. The future scope of this project also includes incorporating this along with other notable criminal security software, and to broaden its range and scope. Advanced Machine Learning algorithms can contribute to a more accurate output.

I had successfully implemented this project **“CRIMINAL IDENTIFICATION AND WEAPON DETECTION”** finally made a successful ending.

REFERENCES

[1] A. J. Phillips, H. Moon, P. Rauss and S. A. Rizvi, "The FERET evaluation methodology for face-recognition algorithms", Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 137-143, Jun. 1997.

[2] B. Sim, S. Baker and M. Bsat, "The CMU pose illumination and expression (PIE) database", Proc. 5th IEEE Int. Conf. Autom. Face Gesture Recognit., pp. 46-51, May 2002.

[3] B. Viola and M. J. Jones, "Robust real-time face detection", Int. J. Comput. Vis., vol. 57, no. 2, pp. 137-154, May 2004.

[4] B. Wiskott, J.-M. Fellous, N. Kruger and C. von der Malsburg, "Face recognition by elastic bunch graph matching", IEEE Trans. Pattern Anal. Mach. Intell., vol. 19, no. 7, pp. 775-779, Jul. 1997.

[5] L. R. Arashloo and J. Kittler, "Energy normalization for pose-invariant face recognition based on MRF model image matching", IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 6, pp. 1274-1280, Jun. 2011.

[6] P. Wolf, T. Hassner and Y. Taigman, "Effective unconstrained face recognition by combining multiple descriptors and learned background statistics", IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 10, pp. 1978-1990, Oct. 2011.

[7] P. Redmon, S. Divvala, R. Girshick et al., "You only look once: Unified real-time object detection", Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779-788, 2016.

[8] R. Shrivastava, A. Gupta and R. Girshick, "Training region-based object detectors with online hard example mining", 2016 IEEE Conference on Computer Vision and Pattern Recognition, pp. 761-769, 2016.

[9] R.Y. Fu, W. Liu, A. Ranga et al., DSSD: Deconvolutional Single Shot Detector, 2017, [online] Available.

[10] Reza Moradi Rad, Abdolrahaman Attar, Reza Ebrahimi atani , “A Robust Face Recognition method using edge based features”, 2012 IEEE Symposium on Computers and Informatics.

[11] Y L. Galvez, G. E. U. Faelden, J. M. Z. Maningo, R. C. S. Nakano, E. P. Dadios, A. A. Bandala, et al., "Obstacle Avoidance Algorithm for Swarm of Quadrotor Unmanned Aerial Vehicle Using Artificial Quadrotor Unmanned Aerial Vehicle Using Artificial", 2020 IEEE Region 10 Conference (TENCON), 2020.

[12] Y. Graves, Alex, et al. “A novel connectionist system for unconstrained handwriting recognition.” Pattern Analysis and Machine Intelligence, IEEE Transactions on 31.5

[13] Yongqiang Li,ShangFei Wang, Yongpin hao and Qiang Ji, “Simultaneous facial feature tracking and facial expression recognition”, IEEE Transactions on image processing, vol.22, no.7, July 2013.

[14] Yansang Park, Hyun-Cheol Choi, Anil.K Jain and Seong-Whan Lee, “Face tracking and recognition at a distance: A coaxial and concentric PTZ camera system”, IEEE transactions on information forensics and security, vol.8, no.10, October 2013.

APPENDIX

A. SOURCE CODE:

HOME PAGE:

```
import tkinter as tk
from tkinter import filedialog
from tkinter import messagebox
from PIL import Image
from PIL import ImageTk
import threading
import shutil
from facerec import *
from register import *
from dbHandler import *

active_page = 0
thread_event = None
left_frame = None
right_frame = None
heading = None
webcam = None
img_label = None
img_read = None
img_list = []
slide_caption = None
slide_control_panel = None
current_slide = -1

root = tk.Tk()
root.geometry("1500x900+200+100")

# create Pages
pages = []
for i in range(4):
    pages.append(tk.Frame(root, bg="#202d42"))
    pages[i].pack(side="top", fill="both", expand=True)
    pages[i].place(x=0, y=0, relwidth=1, relheight=1)


def goBack():
    global active_page, thread_event, webcam

    if (active_page==3 and not thread_event.is_set()):
        thread_event.set()
        webcam.release()
    for widget in pages[active_page].winfo_children():
        widget.destroy()

    pages[0].lift()
    active_page = 0


def basicPageSetup(pageNo):
    global left_frame, right_frame, heading

    back_img  = tk.PhotoImage(file="back.png")
    back_button          =          tk.Button(pages[pageNo],
image=back_img,          bg="#202d42",          bd=0,
highlightthickness=0,
           activebackground="#202d42", command=goBack)
    back_button.image = back_img
    back_button.place(x=10, y=10)

    heading    =    tk.Label(pages[pageNo],    fg="white",
bg="#202d42", font="Arial 20 bold", pady=10)
    heading.pack()

    content   =   tk.Frame(pages[pageNo],   bg="#202d42",
pady=20)
    content.pack(expand="true", fill="both")

    left_frame = tk.Frame(content, bg="#202d42")
    left_frame.grid(row=0, column=0, sticky="nsew")

    right_frame  =  tk.LabelFrame(content,  text="Detected
Criminals", bg="#202d42", font="Arial 20 bold", bd=4,
                        foreground="#2ea3ef", labelanchor="n")
    right_frame.grid(row=0,     column=1,     sticky="nsew",
padx=20, pady=20)

    content.grid_columnconfigure(0,                weight=1,
uniform="group1")
    content.grid_columnconfigure(1,                weight=1,
uniform="group1")
    content.grid_rowconfigure(0, weight=1)


def showImage(frame, img_size):
    global img_label, left_frame

    img = cv2.resize(frame, (img_size, img_size))
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    img = Image.fromarray(img)
    img = ImageTk.PhotoImage(img)
    if (img_label == None):
        img_label     =     tk.Label(left_frame,     image=img,
bg="#202d42")
        img_label.image = img
        img_label.pack(padx=20)
    else:
        img_label.configure(image=img)
        img_label.image = img


def getNewSlide(control):
    global img_list, current_slide

    if(len(img_list) > 1):
        if(control == "prev"):
            current_slide = (current_slide-1) % len(img_list)
        else:
            current_slide = (current_slide+1) % len(img_list)

        img_size = left_frame.winfo_height() - 200
        showImage(img_list[current_slide], img_size)

        slide_caption.configure(text     =     "Image     {}     of
{}".format(current_slide+1, len(img_list)))


def selectMultiImage(opt_menu, menu_var):
    global      img_list,      current_slide,      slide_caption,
slide_control_panel

    filetype = [("images", "*.jpg *.jpeg *.png")]
    path_list   =   filedialog.askopenfilenames(title="Choose
atleast 5 images", filetypes=filetype)

    if(len(path_list) < 5):
        messagebox.showerror("Error",  "Choose  atleast  5
images.")
    else:
        img_list = []
        current_slide = -1

        # Resetting slide control panel
        if (slide_ control _panel != None):
            slide_ control_ panel. destroy()

        # Creating Image list
        for path in path_list:
            img_list.append(cv2.imread(path))

        # Creating choices for profile pic menu
        menu_var.set("")
        opt_menu['menu'].delete(0, 'end')

        for i in range(len(img_list)):
            ch = "Image " + str(i+1)
            opt_menu['menu'].add_command(label=ch,
command= tk._setit(menu_var, ch))
            menu_var.set("Image 1")


        # Creating slideshow of images
        img_size = left_frame.winfo_height() - 200
        current_slide += 1
        showImage(img_list[current_slide], img_size)

        slide_control_panel          =          tk.Frame(left_frame,
bg="#202d42", pady=20)
        slide_control_panel.pack()

        back_img = tk.PhotoImage(file="previous.png")
        next_img = tk.PhotoImage(file="next.png")

        prev_slide        =        tk.Button(slide_control_panel,
image=back_img,              bg="#202d42",              bd=0,
highlightthickness=0,
                            activebackground="#202d42",
command=lambda : getNewSlide("prev"))
        prev_slide.image          =          back_img
        prev_slide.grid(row=0, column=0, padx=60)

        slide_caption       =       tk.Label(slide_control_panel,
text="Image 1 of {}".format(len(img_list)), fg="#ff9800",
                              bg="#202d42", font="Arial 15 bold")
        slide_caption.grid(row=0, column=1)

        next_slide        =        tk.Button(slide_control_panel,
image=next_img,              bg="#202d42",              bd=0,
highlightthickness=0,
                            activebackground="#202d42",
command=lambda : getNewSlide("next"))
        next_slide.image          =          next_img
        next_slide.grid(row=0, column=2, padx=60)
```

```
def register(entries, required, menu_var):
    global img_list

    # Checking if no image selected
    if(len(img_list) == 0):
        messagebox.showerror("Error", "Select Images first.")
        return

    # Fetching data from entries
    entry_data = {}
    for i, entry in enumerate(entries):
        val = entry[1].get()

        if (len(val) == 0 and required[i] == 1):
            messagebox.showerror("Field    Error",    "Required
field missing :\n\n%s" % (entry[0]))
            return
        else:
            entry_data[entry[0]] = val.lower()


    # Setting Directory
    path = os.path.join('face_samples', "temp_criminal")
    if not os.path.isdir(path):
        os.mkdir(path)

    no_face = []
    for i, img in enumerate(img_list):
        # Storing Images in directory
        id = registerCriminal(img, path, i + 1)
        if(id != None):
            no_face.append(id)

    # check if any image doesn't contain face
    if(len(no_face) > 0):
        no_face_st = ""
        for i in no_face:
            no_face_st += "Image " + str(i) + ", "
        messagebox.showerror("Registration              Error",
"Registration failed!\n\nFollowing images doesn't contain"
            "    face    or    Face    is    too
small:\n\n%s"%(no_face_st))
        shutil.rmtree(path, ignore_errors=True)
    else:
        # Storing data in database
        rowId = insertData(entry_data)
        if(rowId > 0):
            messagebox.showinfo("Success",            "Criminal
Registered Successfully.")
            shutil.move(path,        os.path.join('face_samples',
entry_data["Name"]))

            # save profile pic
            profile_img_num = int(menu_var.get().split(' ')[1]) -
1
            if not os.path.isdir("profile_pics"):
                os.mkdir("profile_pics")
            cv2.imwrite("profile_pics/criminal %d.png"%rowId,
img_list[profile_img_num])

            goBack()
        else:
            shutil.rmtree(path, ignore_errors=True)
            messagebox.showerror("Database Error", "Some
error occured while storing data.")


## update scrollregion when all widgets are in canvas
def on_configure(event, canvas, win):
    canvas.configure(scrollregion=canvas.bbox('all'))
    canvas.itemconfig(win, width=event.width)

## Register Page ##
def getPage1():
    global active_page, left_frame, right_frame, heading,
img_label
    active_page = 1
    img_label = None
    opt_menu = None
    menu_var = tk.StringVar(root)
    pages[1].lift()

    basicPageSetup(1)
    heading.configure(text="Register Criminal")
    right_frame.configure(text="Enter Details")

    btn_grid = tk.Frame(left_frame, bg="#202d42")
    btn_grid.pack()

    tk.Button(btn_grid,           text="Select           Images",
command=lambda:               selectMultiImage(opt_menu,
menu_var), font="Arial 15 bold", bg="#2196f3",
           fg="white",  pady=10,  bd=0,  highlightthickness=0,
activebackground="#091428",
        activeforeground="white").grid(row=0,     column=0,
padx=25, pady=25)


   # Creating Scrollable Frame
   canvas    =    tk.Canvas(right_frame,    bg="#202d42",
highlightthickness=0)
   canvas.pack(side="left",    fill="both",    expand="true",
padx=30)
   scrollbar              =              tk.Scrollbar(right_frame,
command=canvas.yview,                              width=20,
troughcolor="#202d42", bd=0,
                    activebackground="#00bcd4",
bg="#2196f3", relief="raised")
   scrollbar.pack(side="left", fill="y")

   scroll_frame    =    tk.Frame(canvas,    bg="#202d42",
pady=20)
   scroll_win      =      canvas.create_window((0,      0),
window=scroll_frame, anchor='nw')

   canvas.configure(yscrollcommand=scrollbar.set)
   canvas.bind('<Configure>',          lambda          event,
canvas=canvas,    win=scroll_win:    on_configure(event,
canvas, win))


   tk.Label(scroll_frame,    text="*    Required    Fields",
bg="#202d42", fg="yellow", font="Arial 13 bold").pack()
   # Adding Input Fields
   input_fields = ("Name", "Father's Name", "Mother's
Name", "Gender", "DOB(yyyy-mm-dd)", "Blood Group",
               "Identification Mark", "Nationality", "Religion",
"Crimes Done", "Profile Image")
   ip_len = len(input_fields)
   required = [1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0]

   entries = []
   for i, field in enumerate(input_fields):
      row = tk.Frame(scroll_frame, bg="#202d42")
      row.pack(side="top", fill="x", pady=15)

      label     =     tk.Text(row,     width=20,     height=1,
bg="#202d42",       fg="#ffffff",       font="Arial       13",
highlightthickness=0, bd=0)
      label.insert("insert", field)
      label.pack(side="left")
```

```
        if(required[i] == 1):
            label.tag_configure("star", foreground="yellow",
font="Arial 13 bold")
            label.insert("end", " *", "star")
        label.configure(state="disabled")

        if(i != ip_len-1):
            ent = tk.Entry(row, font="Arial 13",
selectbackground="#90ceff")
            ent.pack(side="right", expand="true", fill="x",
padx=10)
            entries.append((field, ent))
        else:
            menu_var.set("Image 1")
            choices = ["Image 1"]
            opt_menu = tk.OptionMenu(row, menu_var,
*choices)
            opt_menu.pack(side="right", fill="x", expand="true",
padx=10)
            opt_menu.configure(font="Arial 13", bg="#2196f3",
fg="white", bd=0, highlightthickness=0,
activebackground="#90ceff")
            menu =
opt_menu.nametowidget(opt_menu.menuname)
            menu.configure(font="Arial 13", bg="white",
activebackground="#90ceff", bd=0)

    tk.Button(scroll_frame, text="Register",
command=lambda: register(entries, required, menu_var),
font="Arial 15 bold",
           bg="#2196f3", fg="white", pady=10, padx=30, bd=0,
highlightthickness=0, activebackground="#091428",
           activeforeground="white").pack(pady=25)


def  showCriminalProfile(name):
    top = tk.Toplevel(bg="#202d42")
    top.title("Criminal Profile")
    top.geometry("1500x900+%d+%d"%(root.winfo_x()+10,
root.winfo_y()+10))

    tk.Label(top, text="Criminal Profile", fg="white",
bg="#202d42", font="Arial 20 bold", pady=10).pack()

    content = tk.Frame(top, bg="#202d42", pady=20)
    content.pack(expand="true", fill="both")
    content.grid_columnconfigure(0, weight=3,
uniform="group1")
```

```
    content.grid_columnconfigure(1, weight=5,
uniform="group1")
    content.grid_rowconfigure(0, weight=1)

    (id, crim_data) = retrieveData(name)

    path = os.path.join("profile_pics", "criminal %d.png"%id)
    profile_img = cv2.imread(path)

    profile_img = cv2.resize(profile_img, (500, 500))
    img = cv2.cvtColor(profile_img,
cv2.COLOR_BGR2RGB)
    img = Image.fromarray(img)
    img = ImageTk.PhotoImage(img)
    img_label = tk.Label(content, image=img,
bg="#202d42")
    img_label.image = img
    img_label.grid(row=0, column=0)

    info_frame = tk.Frame(content, bg="#202d42")
    info_frame.grid(row=0, column=1, sticky='w')

    for i, item in enumerate(crim_data.items()):
        tk.Label(info_frame, text=item[0], pady=15,
fg="yellow", font="Arial 15 bold",
bg="#202d42").grid(row=i, column=0, sticky='w')
        tk.Label(info_frame, text=":", fg="yellow", padx=50,
font="Arial 15 bold", bg="#202d42").grid(row=i, column=1)
        val = "---" if (item[1]=="") else item[1]
        tk.Label(info_frame, text=val.capitalize(), fg="white",
font="Arial 15", bg="#202d42").grid(row=i, column=2,
sticky='w')


def startRecognition():
    global img_read, img_label

    if(img_label == None):
        messagebox.showerror("Error", "No image selected.
")
        return

    crims_found_labels = []
    for wid in right_frame.winfo_children():
        wid.destroy()

    frame = cv2.flip(img_read, 1, 0)
    gray_frame = cv2.cvtColor(frame,
cv2.COLOR_BGR2GRAY)

    face_coords = detect_faces(gray_frame)

    if (len(face_coords) == 0):
        messagebox.showerror("Error", "Image doesn't
contain any face or face is too small.")
    else:
        (model, names) = train_model()
        print('Training Successful. Detecting Faces')
        (frame, recognized) = recognize_face(model, frame,
gray_frame, face_coords, names)

        img_size = left_frame.winfo_height() - 40
        frame = cv2.flip(frame, 1, 0)
        showImage(frame, img_size)

        if (len(recognized) == 0):
            messagebox.showerror("Error", "No criminal
recognized.")
            return

        for i, crim in enumerate(recognized):
            crims_found_labels.append(tk.Label(right_frame,
text=crim[0], bg="orange",
                                        font="Arial 15 bold", pady=20))
            crims_found_labels[i].pack(fill="x", padx=20,
pady=10)
            crims_found_labels[i].bind("<Button-1>", lambda e,
name=crim[0]:showCriminalProfile(name))


def selectImage():
    global left_frame, img_label, img_read
    for wid in right_frame.winfo_children():
        wid.destroy()

    filetype = [("images", "*.jpg *.jpeg *.png")]
    path = filedialog.askopenfilename(title="Choose a
image", filetypes=filetype)

    if(len(path) > 0):
        img_read = cv2.imread(path)

        img_size = left_frame.winfo_height() - 40
        showImage(img_read, img_size)
```

```
## Detection Page ##
def getPage2():
   global active_page, left_frame, right_frame, img_label,
heading
   img_label = None
   active_page = 2
   pages[2].lift()

   basicPageSetup(2)
   heading.configure(text="Detect Criminal")
   right_frame.configure(text="Detected Criminals")

   btn_grid = tk.Frame(left_frame, bg="#202d42")
   btn_grid.pack()

   tk.Button(btn_grid,            text="Select            Image",
command=selectImage,  font="Arial  15  bold",  padx=20,
bg="#2196f3",
         fg="white",  pady=10,  bd=0,  highlightthickness=0,
activebackground="#091428",
         activeforeground="white").grid(row=0,    column=0,
padx=25, pady=25)
   tk.Button(btn_grid,                         text="Recognize",
command=startRecognition, font="Arial 15 bold", padx=20,
bg="#2196f3",
        fg="white",  pady=10,  bd=0,  highlightthickness=0,
activebackground="#091428",
        activeforeground="white").grid(row=0,     column=1,
padx=25, pady=25)


def videoLoop(model, names):
   global thread_event, left_frame, webcam, img_label
   webcam = cv2.VideoCapture(0)
   old_recognized = []
   crims_found_labels = []
   img_label = None

   try:
      while not thread_event.is_set():
         # Loop until the camera is working
         while (True):
            # Put the image from the webcam into 'frame'
            (return_val, frame) = webcam.read()
            if (return_val == True):
               break
            else:
               print("Failed to open webcam. Trying again...")

         # Flip the image (optional)
         frame = cv2.flip(frame, 1, 0)
         # Convert frame to grayscale
         gray_frame          =          cv2.cvtColor(frame,
cv2.COLOR_BGR2GRAY)

         # Detect Faces
         face_coords = detect_faces(gray_frame)
         (frame,   recognized)   =   recognize_face(model,
frame, gray_frame, face_coords, names)

         # Recognize Faces
         recog_names = [item[0] for item in recognized]
         if(recog_names != old_recognized):
            for wid in right_frame.winfo_children():
               wid.destroy()
            del(crims_found_labels[:])

            for i, crim in enumerate(recognized):

crims_found_labels.append(tk.Label(right_frame,
text=crim[0], bg="orange",
                                         font="Arial     15     bold",
pady=20))
               crims_found_labels[i].pack(fill="x",    padx=20,
pady=10)
               crims_found_labels[i].bind("<Button-1>",
lambda e, name=crim[0]: showCriminalProfile(name))

            old_recognized = recog_names

         # Display Video stream
         img_size       =       min(left_frame.winfo_width(),
left_frame.winfo_height()) - 20

         showImage(frame, img_size)

   except RuntimeError:
      print("[INFO]Caught Runtime Error")
   except tk.TclError:
      print("[INFO]Caught Tcl Error")


## video surveillance Page ##
def getPage3():
global      active_page,      video_loop,      left_frame,
right_frame,thread_event, heading
active_page = 3
pages[3].lift()

basicPageSetup(3)
heading.configure(text="Video Surveillance")
right_frame.configure(text="Detected Criminals")
left_frame.configure(pady=40)

(model, names) = train_model()
print('Training Successful. Detecting Faces')

thread_event = threading.Event()
thread        =        threading.Thread(target=videoLoop,
args=(model, names))
thread.start()


#########################################    Home
Page ######################################
tk.Label(pages[0], text="Criminal Identification System",
fg="white", bg="#202d42",
     font="Arial 35 bold", pady=30).pack()

logo = tk.PhotoImage(file = "logo.png")
tk.Label(pages[0], image=logo, bg="#202d42").pack()

btn_frame = tk.Frame(pages[0], bg="#202d42", pady=30)
btn_frame.pack()

tk.Button(btn_frame,         text="Register         Criminal",
command=getPage1)
tk.Button(btn_frame,        text="Detecting        Criminal",
command=getPage2)
tk.Button(btn_frame,        text="Video        Surveillance",
command=getPage3)

for btn in btn_frame.winfo_children():
btn.configure(font="Arial  20",  width=17,  bg="#2196f3",
fg="white",
      pady=15,            bd=0,            highlightthickness=0,
activebackground="#091428", activeforeground="white")
btn.pack(pady=30)



pages[0].lift()
```

```
root.mainloop()
```

DATA HANDLER:

```
import pymysql

def insertData(data):
    rowId = 0

    db = pymysql.connect("localhost", "criminaluser", "",
"criminaldb")
    cursor = db.cursor()
    print("database connected")

    query = "INSERT INTO criminaldata VALUES(0, '%s',
'%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s', '%s');" % \
            (data["Name"],          data["Father's          Name"],
data["Mother's Name"], data["Gender"],
             data["DOB(yyyy-mm-dd)"],   data["Blood   Group"],
data["Identification Mark"],
             data["Nationality"], data["Religion"], data["Crimes
Done"])

    try:
        cursor.execute(query)
        db.commit()
        rowId = cursor.lastrowid
        print("data stored on row %d" % rowId)
    except:
        db.rollback()
        print("Data insertion failed")


    db.close()
    print("connection closed")
    return rowId

def retrieveData(name):
    id = None
    crim_data = None

    db = pymysql.connect("localhost", "criminaluser", "",
"criminaldb")
    cursor = db.cursor()
    print("database connected")

    query   =   "SELECT   *   FROM   criminaldata WHERE
name='%s'"%name
```

```
try:
    cursor.execute(query)
    result = cursor.fetchone()

    id=result[0]
    crim_data = {
        "Name" : result[1],
        "Father's Name" : result[2],
        "Mother's Name" : result[3],
        "Gender" : result[4],
        "DOB(yyyy-mm-dd)" : result[5],
        "Blood Group" : result[6],
        "Identification Mark" : result[7],
        "Nationality" : result[8],
        "Religion" : result[9],
        "Crimes Done" : result[10]
    }

    print("data retrieved")
except:
    print("Error: Unable to fetch data")

db.close()
print("connection closed")

return (id, crim_data)
```

B. SCREENSHOTS:

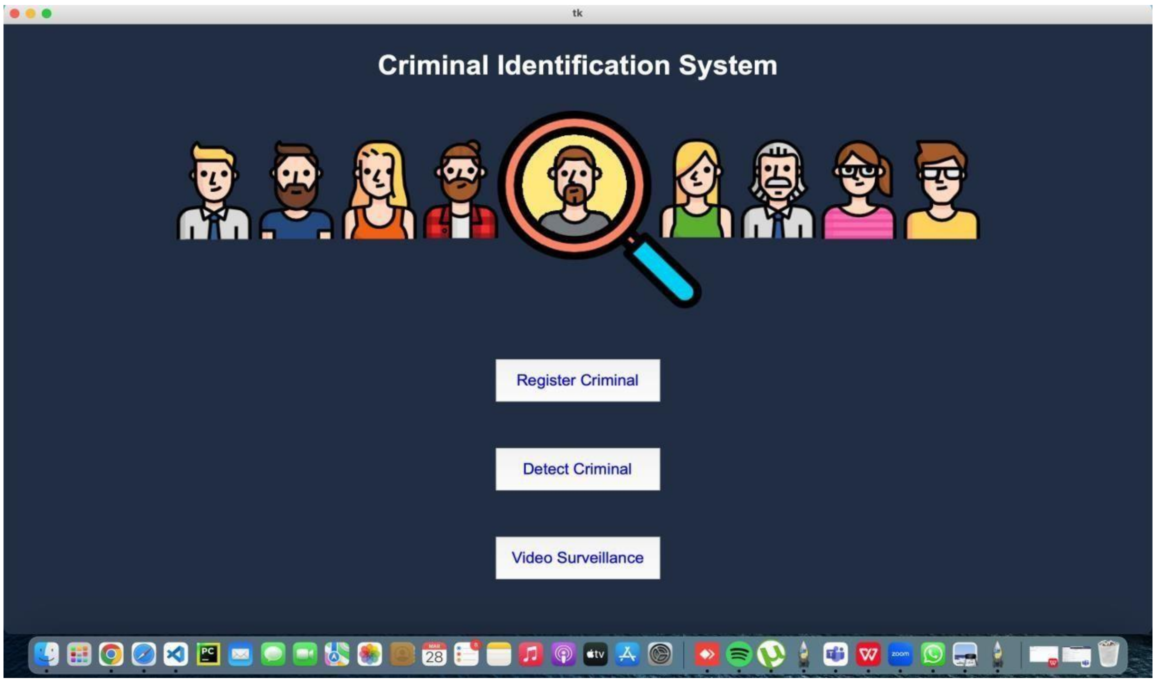
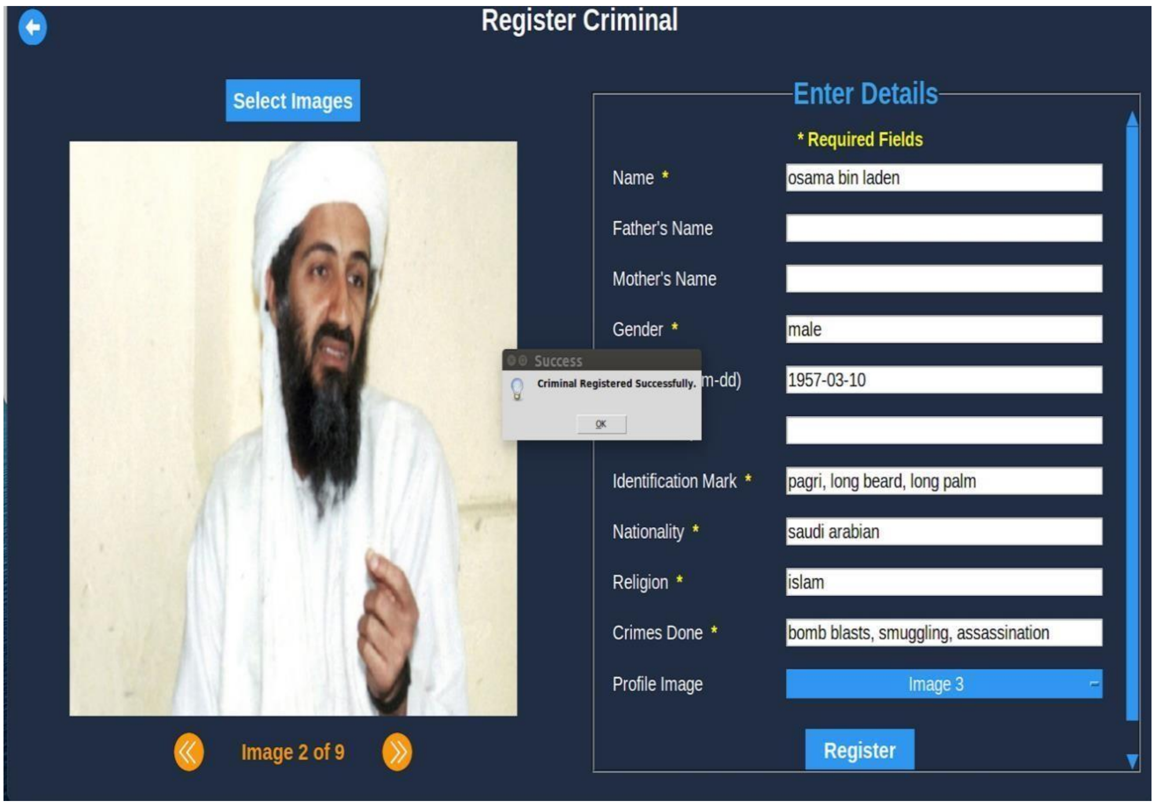


Fig. No 7.1 GUI Interface



Criminal Registered Successfully.

Name *	osama bin laden
Father's Name	
Mother's Name	
Gender *	male
m-dd)	1957-03-10
Identification Mark *	pagri, long beard, long palm
Nationality *	saudi arabian
Religion *	islam
Crimes Done *	bomb blasts, smuggling, assassination
Profile Image	Image 3

Fig. 7.2 Registering criminal details

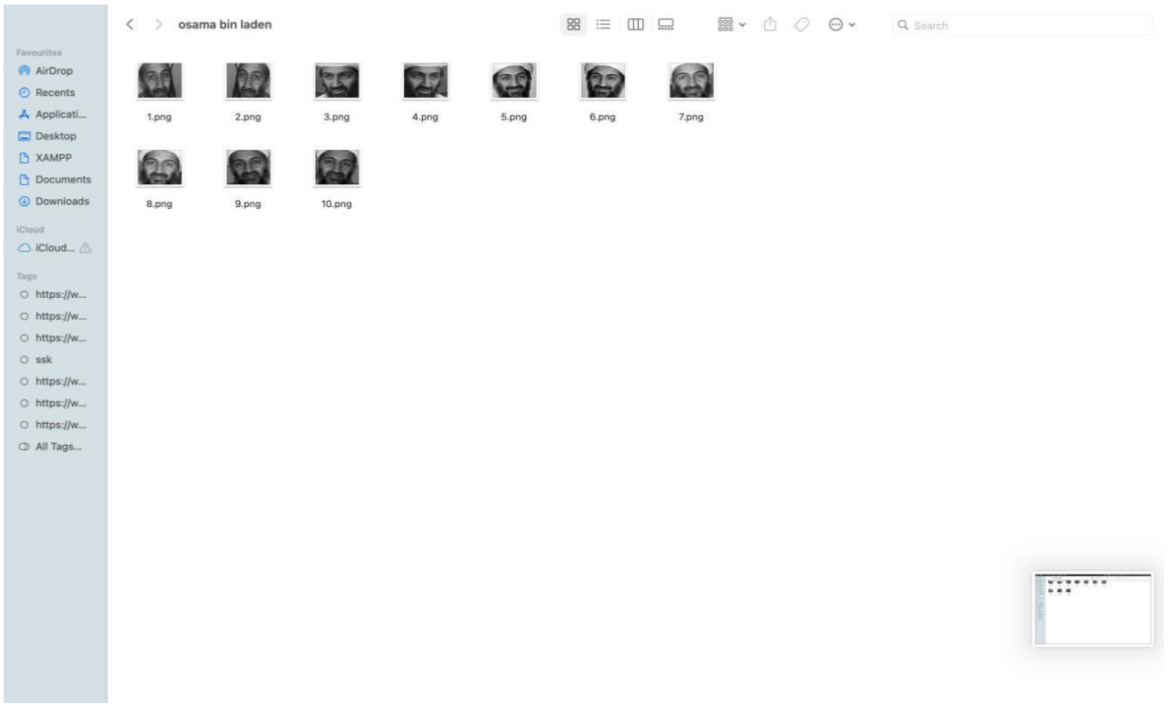


Fig. 7.3 Face Samples of criminal

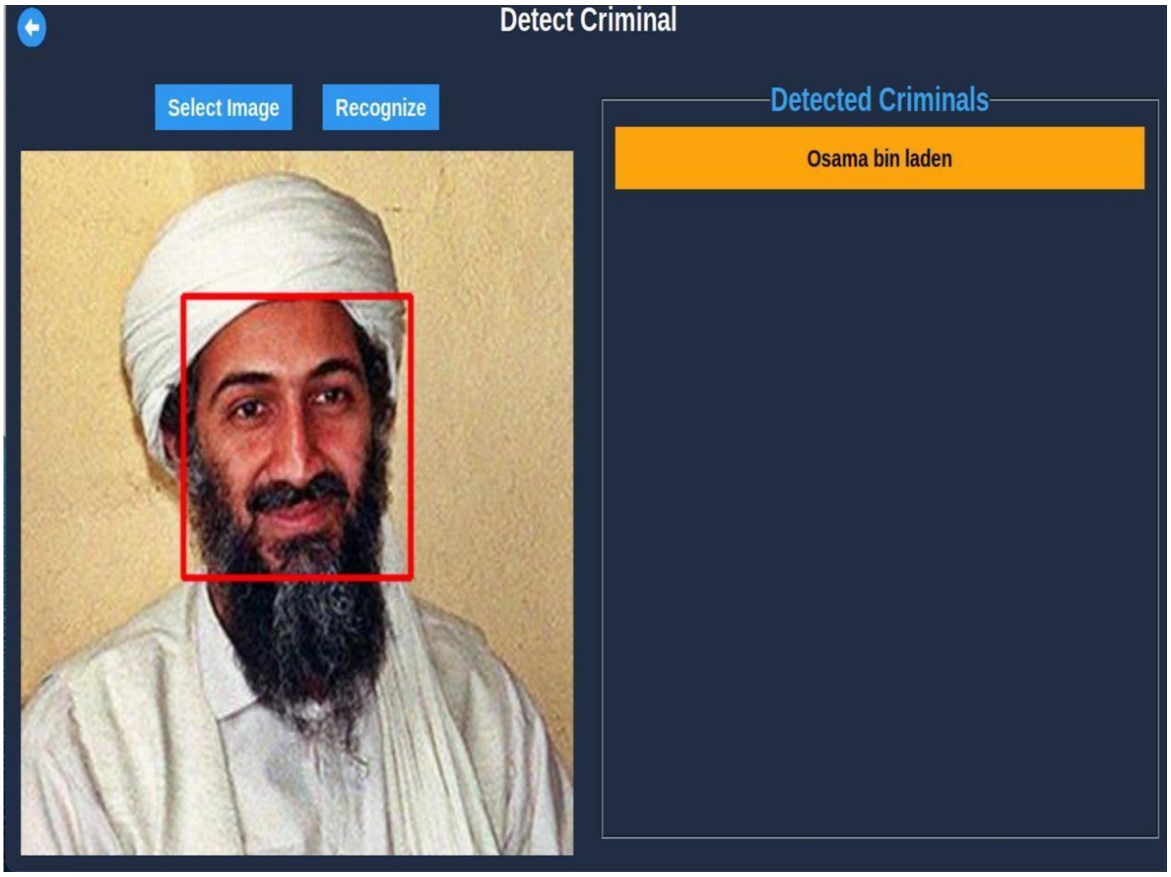


Fig. 7.4 Criminal Detected

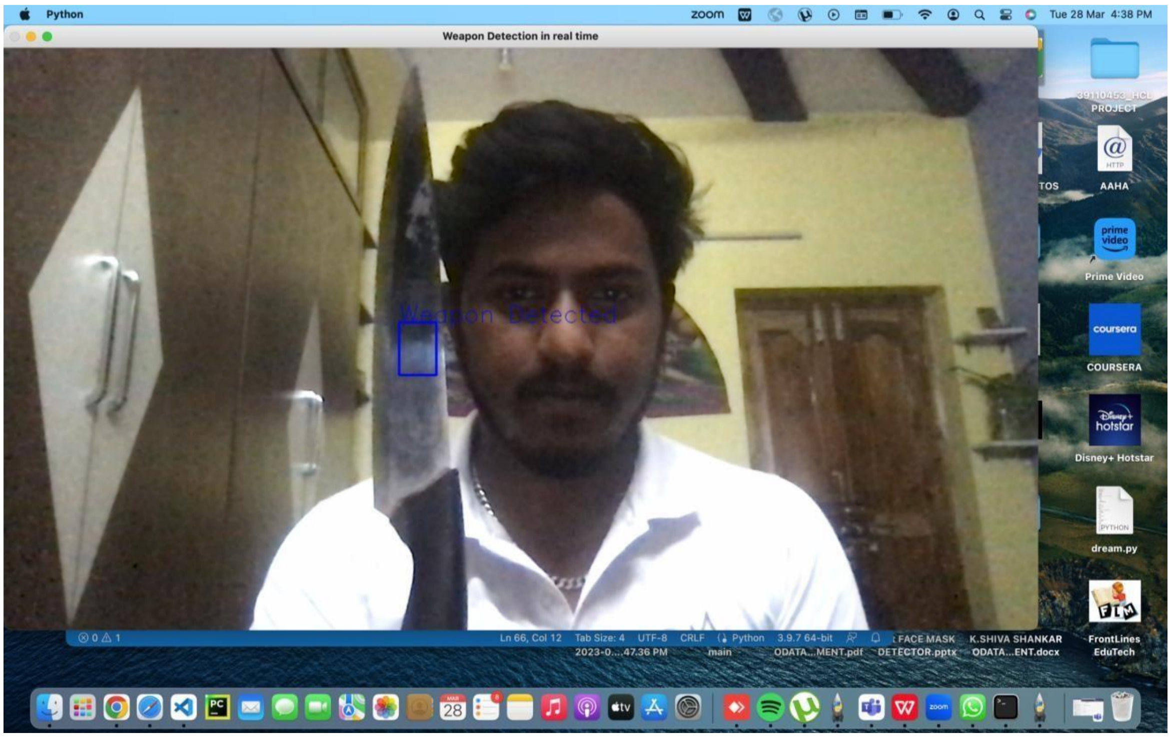


Fig 7.5: Output : It shows weapon detection

C.RESEARCH PAPER

CRIMINAL IDENTIFICATION AND WEAPON DETECTION

Karanam Shiva Shankar
UG Student,
Department of CSE
Sathyabama Institute of Science and Technology, Chennai, India
shivashankarkaranam8@gmail.com

Dr. J. Refonaa
Assistant Professor,
Department of CSE
Sathyabama Institute of Science and Technology, Chennai, India

Dr. S.L.Jany Shabu,
Assistant Professor,
Department of CSE
Sathyabama Institute of Science and Technology, Chennai, India

Karri Bhaskar Reddy
UG Student,
Department of CSE
Sathyabama Institute of Science and Technology, Chennai, India
Bhaskarreddykarri47@gmail.com

Dr. S.Dhamodaran
Assistant Professor,
Department of CSE
Sathyabama Institute of Science and Technology, Chennai, India

Dr. Viji Amutha Mary
Assistant Professor,
Department of CSE
Sathyabama Institute of Science and Technology, Chennai, India

Abstract – The primary distinguishing feature of this project is that it aims to moderate crimes by capturing and identifying criminal faces. Crime prevention and criminal identification are the primary issues before the police personnel, but the availability of police personnel is limited. . Real time footage of the camera can be used to identify suspects on scene. In this project, an automated facial recognition system for criminal databases is implemented using the well-known Haar feature-based cascade classifier. This system will be able to detect and recognize faces automatically in real time, and also map these faces and collect information from the database, displaying the required criminal files as well. The computer language used in this project is python, and the primary library is OpenCV. OpenCV is a set of library functions that are aimed at real time computer vision. Other notable features of this criminal identification system are face detection, face identification, criminal database, graphical user interface, real time video probing and criminal file uploading.

Keywords: Realtime criminal identification,weapon detection, Haar cascade classifier, tkinter Gui, LBPH pattern.

1. INTRODUCTION

Facial recognition application have been seen larger using in recent days in smartphones and other technologies. Now a days face recognition has become one of the best application for the security.As we know that in many schools, colleges, companies have their own biometric systems for taking attendance or presence of the employ. Although we have biometric systems face recognition application has the accuracywhen compared to biometric technology which has lower iris recognition and fingerprint recognition. Face recognition models are being developed as a technology where human can identify other people through a software.This is used by several organizations in their CCTVs, access controls and many more. But the accuracy it takes when is very less. So in order to overcome this project is proposed here we have used Haar classifer. As researches are less in face recognition using edge-based detection these edge is easy to be processed on face recognition. The Viola–Jones object detection can be trained to detect a various objects, proposed especially to face detection.

2. LITERATURE REVIEW

AUTHOR: Jyoti Nautiyal, Shivali Gahlot and Pawan Kuman Mishra, 2013
An automated technique for criminal face identification using biometric approach. This uses thumbprints as biometric device for recognization. This may take too long time to identify the culprit also the accuracy is less.
AUTHOR: nsang Park, Hyun-Cheol Choi, Anil.K Jain, Seong-Whan Lee,2013
Face Tracking and recognition at a distance. Person recognization is done quickly by using this advanced PTZ cameras advanced biometric system. Once definitive drawback of criminal profiling is making personal assumptions about the perpetrator
AUTHOR: Paul Viola, Michael Jones, 2001
Rapid object detection using a Boosted cascade of simple features. Object recognition framework that allows the detection image features in real time.Not effective on detecting the faces which are turned. AUTHOR: Faizan Ahmad, Aaima Najam and Zeeshan Ahmed,201
Image based face recognition system evaluate various face detection and recognition methods, provide complete solution for image based face

detection and recognition with higher accuracy, better response rate as an initial step for video surveillance. Solution isproposed based on performed tests on various face rich databases in terms of subjects, pose, emotions, race and light

3. ARCHITECTURE DIAGRAM

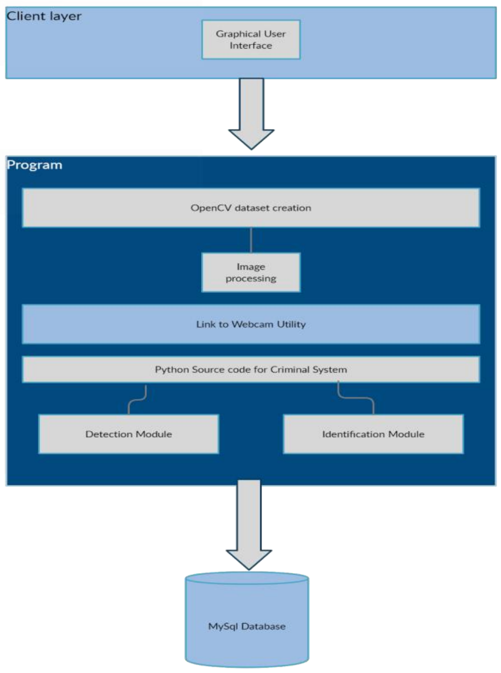


Fig.1 Flowchart of proposed methodology

4. PROPOSED SYSTEM

In this proposed paper, we are using single database. The database is preprocessed for removal of noise and to reduce redundancy. Feature extraction is done with Haar techniques. Database consists of pictures and details of citizens in the given country.It also has pictures and information regarding criminals belonging to that country.The database also contains information regarding criminals who are not the citizens of the country. After enabling the video it will be converted to frames. Noise and redundancy is reduced when a frame recognizes a face. Then feature extraction takes place and this is where Haar cascade comes to play. The obtained image is compared with the faces that are already there in the criminal database. If the faces are similar, it is again compared with the images stored in the local watch list database to f i n d if the present person is a criminal or not. If he is criminal the time duration of his presence under the camera will be noted. If he is not a local citizen, it will be compared with the information in the global suspects list database. If a match is found, time duration will be noted. If match is not found in both the lists, he is innocent.

5. METHODOLOGY

Haar casading techniques:

It is One of the most common technical tool for face recognization.The proposed method is designed to detect by using Haar cascade features, which reduces illumination effects by extracting structural information of objects. The proposed model renders highly accurate face detection backed up by the highly reliable optimized learning data through Adaboost learning algorithm.This manuscript deals with two different resolutions to surpass the problem.

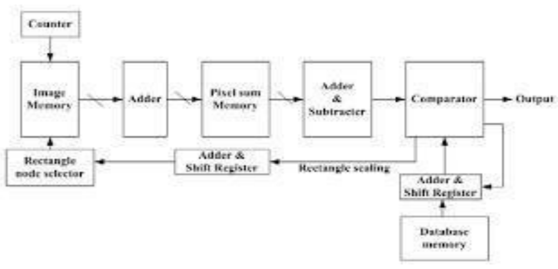
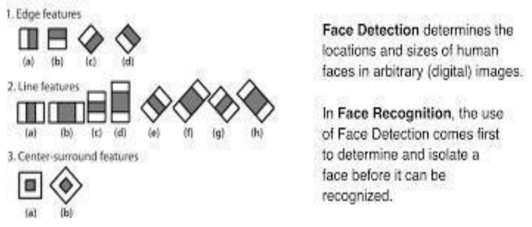


Fig.2 Flow Diagram of face detection

Here, the solutions are described the Python language which contains the syntax that shows ease and simple.The 1st solution is considered as using the more number of classifiers at the time.s. Instead of using complex calculation, MCT technique classifiers which is used to produce the decision. Using of Haar techniques is easy to implemented as it is a part of Open-CV library, which is most popularly used open source with all updated libraries. Open-CV also provides several XML files. Each XML has its own recognize objects such as face, eyes,mouth,guns,knifes etc. Even though we have these XML files for our favourable, but they don't recognize rotated- objects . In order to recognize these rotated-objects, we need to train our own Haar classifier to produce exact XML file. This process takes more time and it is complicated.



Face Detection determines the locations and sizes of human faces in arbitrary (digital) images.

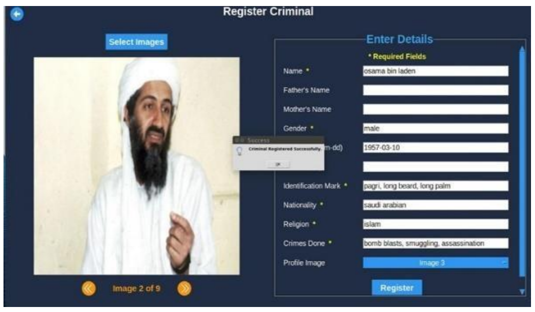
In **Face Recognition**, the use of Face Detection comes first to determine and isolate a face before it can be recognized.

Fig.3 Haar features

In this proposal, we appraise two different approach for rotated object recognition. First approach is by rotating the image into different - different angles, and Haar classifiers will recognition image by using the originalXML file. second approach is by creating various XML files based on the original. Now new XML files will identify the objects and can be used in several classifiers.

6. RESULTS

The final results of this crime detection software program is shown clearly. Any object that may be or closely resemble a weapon is detected by a blue square and is shown along with the graphics. Here, a weapon is defined as an object that can cause harm or injury to any of the persons involved in close proximity to it. A gun, a pistol, or a hand knife may be included with definition. Guns are black or brown coloured in nature and knives are stainless stell or silver coloured. The red square will take size of entire weapon, and if the weapon is a hand held object, then portions of a human hand may or may not be included in the preview. One or many weapons will be detected real time. Multiple gadgets can be identified and can also remain overlapping. In this case, multiple red squares will be there and all the red squares will be shown on top of each other. An alarm will sound if security has been breached and this alarm will sound only for two seconds. Two alarms may overlap if there are multiple weaponry.



Register Criminal

Enter Details

* Required Fields

Name *	osama bin laden
Father's Name	
Mother's Name	
Gender *	male
m-dd)	1957-03-10
Identification Mark *	pagri, long beard, long palm
Nationality *	saudi arabian
Religion *	islam
Crimes Done *	bomb blasts, smuggling, assassination
Profile Image	Image 3

Success

Criminal Registered Successfully.

Fig.4 Criminal registration

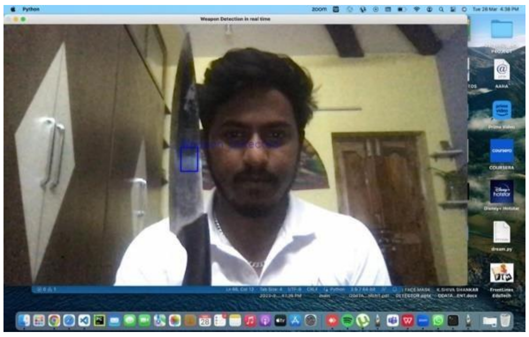


Fig.5 weapon detection

7. EPILOGUE AND FUTURE ESTABLISHMENTS

The proposed system on criminal recognition is based on Haar cascade technique is implementation here. This system has a image processing feature where the input image can be made less blurry so that system can detect face on lower quality images based on the pixels of image in black and white. Although if input image does not match with stored image of database then the input image will be classified by using LBPH (Local Binary Pattern Histogram). This proposed system also have a video surveillance where crime can be detected. This works on training the system with the crime data sets taken from kaggle which is stored in form of XML file. The future scope of this project also includes incorporating this along with other notable criminal security software, and to broaden its range and scope. Advanced Machine Learning algorithms can contribute to a more accurate output.

8. CONCLUSION

The proposed system on criminal recognition is based on Haar cascade technique is implementation here. This system has a image processing feature where the input image can be made less blurry so that system can detect face on lower quality images based on the pixels of image in black and white. Although if input image does not match with stored image of database then the input image will be classified by using LBPH (Local Binary Pattern Histogram).
This proposed system also have a video surveillance where crime can be detected. This works on training the system with the crime data sets taken from kaggle which is stored in form of XML file. The future scope of this project also includes incorporating this along with other notable criminal security software, and to broaden its range and scope. Advanced Machine Learning algorithms can contribute to a more accurate output.

REFERENCES

1. Jyoti Nautiyal, Shivali Gahlot and Pawan Kumar Mishra, “An automated technique for criminal face identification using biometric approach”,Conference on Advances in Communication and Control Systems 2013(CAC2S 2013)

2.Reza Moradi Rad,Abdolrahaman Attar,Reza Ebrahimi atani, “A Robust Face Recognition method using edge based features”,2012 IEEE Symposium on Computers and Informatics.

3.Paul Viola, Michael Jones, “rapid Object Detection using a Boosted Cascade of Simple Features”.

4.nsang Park, Hyun-Cheol Choi, Anil.K Jain and Seong-Whan Lee, “Face tracking and recognition at a distance: A coaxial and concentric PTZ camera system”, IEEE transactions on information forensics and security, vol.8, no.10, October 2013

5.Yongqiang Li,ShangFei Wang, Yongping Zhao and Qiang Ji, “Simultaneous facial feature tracking and facial expression recognition”, IEEE Transactions on image processing, vol.22, no.7, July 2013.

6. Sheu, Jia-Shing; Hsieh, Tsu-Shien; Shou, Ho-Nien. "Automatic Generation of Facial Expression Using Triangular Geometric Deformation". Journal of Applied Research and Technology, 1 December 2014.

7. P. M. Kumar, U. Gandhi, R. Varatharajan, G. Manogaran, R. Jidhesh, and T. Vadivel, “Intelligent face recognition and navigation system using neural learning for smart security in internet of things,” Cluster Computing, vol. 22, no. S4, pp. 7733–7744, 2019.

8. S. Y. Nikouei, Y. Chen, S. Song, R. Xu, B.-Y. Choi, and T. Faughnan, “Smart surveillance as an edge network service: from harr-cascade, SVM to a lightweight CNN,” in Proceedings of the 2018 IEEE 4th International Conference on Collaboration and Internet Computing (CIC), pp. 256–265, Philadelphia, PA, USA, April 2018.

9. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: unified, real-time object detection,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788, Las Vegas, NV, USA, June 2016.

10.M. Grega, A. Matiolański, P. Guzik, and M. Leszczuk, “Automated detection of firearms and knives in a CCTV image,” Sensors, vol. 16, no. 1, p. 47, 2016.

11.G. K. Verma and A. Dhillon, “A handheld gun detection using faster r-cnn deep learning,” in Proceedings of the 7th International Conference on Computer and Communication Technology, pp. 84–88, Kurukshetra, Haryana, November 2017

12. Redmon, Joseph (2016). "You only look once: Unified, real-time object detection". Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

13.Wang, Chien-Yao (2021). "Scaled-YOLOv4: Scaling Cross Stage Partial Network". Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition