Republic of Iraq
Ministry of Higher Education and Scientific Research
Misan University
College of Engineering

Criminal Face Recognition System

A graduation project is submitted to the Electrical Engineering Department
in partial fulfillment of the requirements for the degree
Of Bachelor of Science in
College of Engineering - Electrical Engineering

BY
ALI ADEL MAZID
MOHAMMED MAJEED MATAR
MONTAZER KARIM MANATI
MASOUMA JAWAD ABDUL KADHIM

SUPERVISOR
M.SC: MOHAMED HASSOUNI

Misan, Iraq
2024-2025

SUPERVIOSR CERTIFICATION

I certify that preparation of this project entitled

Criminal Face Recognition System,

Prepared by **ALI ADEL MAZID**

MOHAMMED MAJEED MATAR

MONTAZER KARIM MANATI

MASOUMA JAWAD ABDUL KADHIM

was made under my supervision at General College of Engineering Electrical Engineering Department in partial fulfillment of the Requirements for the Degree of Bachelor of Science in College of Engineering - Electrical Engineering.

Signature:

Name: **M.SC: MOHAMED HASSOUNI** (Supervisor)

Date: / /2025

DEDICATION

Here we conclude our graduation project and academic journey, dedicating our work to the teacher of humanity, our Prophet Muhammad (peace be upon him)**.**

Also, to our honorable family who supported us (my father, mother, brothers, and sisters)**.**

To everyone who has had an impact on our lives**.**

And to those who paved the way for us to reach the peak of knowledge, the teachers of science and knowledge**.**

ACKNOWLEDGMENTS

First, we would like to thank Allah, for letting us through all the difficulties.

We would like to acknowledge and give our warmest thanks to our supervisor (**M.SC: MOHAMED HASSOUNI)** who made this work possible. his guidance and advice carried us through all the stages of writing my project.

We would also like to give special thanks to our family as a whole for their continuous support and understanding when undertaking our research and writing our research. Your support for us was what sustained us this far.

ABSTRACT

Everyone is now quite concerned about the rising crime rate. Catching criminals in time has become quite challenging. For the relevant authorities, timely identification has become a tiresome task. A face recognition system can solve this issue.

The system of face recognition is straightforward. It makes use of the face, a human identity. Technology has advanced to the point that it is now easy to extract facial traits. In addition to identifying the face, the goal here is to share the detected image with the relevant authorities. Automated facial recognition systems can identify offenders' faces on Web cameras and send images of such offenders to the appropriate authorities.

LIST OF CONTENTS

Chapter One

Introduction

1.1. Introduction:

In wealthy countries, law enforcement agencies compile a database of faces, which they then compare to any suspect using facial recognition technology. When a security expert inputs a person's image, a facial recognition system for criminal identification determines whether the suspect has a criminal record. The system pre-processes the image to remove any unwanted noise as a first step in this process. Additionally, the system classifies the image based on features such as jaw length and eye distance. The system then searches for these characteristics in the images stored in the database. If a match is found, the criminal record is displayed, along with output indicating that the image has been recognized. The same procedures are applied to automatically identify criminals, with images of the most wanted criminals stored in the database. A webcam or closed-circuit television (CCTV) camera is used to implement automated facial recognition. The facial recognition process remains the same: the facial features of all passersby on the street near the CCTV or webcams are scanned and matched with those in a database. If the features match, an automated email containing a photo of the identified person can be sent to the relevant authorities. This speeds up the apprehension of the offender [1].

The crime rate is rising abnormally, and the number of criminals is increasing at an unprecedented rate, raising serious security concerns. Since protecting people and property is the primary focus of the police, preventing crime and identifying criminals are top priorities. However,

there is a shortage of police officers to combat crime. With the advancement of security technology, many public and private places are equipped with surveillance cameras, particularly closed-circuit television (CCTV) cameras and Webcams, for surveillance. Using CCTV and Webcam footage, suspects can be identified at the crime scene [2].

Crimes and violations are on the rise, and there are no appropriate standards for identifying, detecting, isolating, and predicting these criminals. Crimes are on an unprecedented rise, regardless of the number of surveillance cameras deployed in various areas. Criminal investigation offices are unable to quickly identify criminals. Initial security developments in many countries were used to identify and identify criminals for public and private security purposes, using fingerprints, fingerprints, and facial recognition, or through other dubious findings from surveillance cameras. However, only limited automated systems exist that can accurately identify criminals and determine the proximity of thieves based on recorded video footage now available in police criminal records [3].

Modern law enforcement agencies rely heavily on technology, and facial recognition has emerged as a vital tool for identifying individuals. This study presents an innovative method for identifying potential criminals by combining their criminal history with facial recognition technology. The goal is to speed up the identification process and improve its accuracy, especially in crowded urban areas where manual identification is difficult.

In our project, we researched and implemented a relatively simple and highly successful face detection algorithm that takes human skin tone into account. Our goal, which we believe we have achieved, is to create a

system that enables law enforcement or investigative agencies to identify criminals based on their faces. Facial recognition technology is fast, reliable, easy to use, and accurate, and its algorithms and techniques are easy to understand.

1.2. Problem Statement:

People are concerned about the increasing incidence of criminals these days, especially those who harm children. Governments and organizations are using cameras and additional personnel to keep them away from places like daycare centers and schools. However, simply asking individuals to scan and view images is not enough. We need an intelligent system capable of quickly analyzing images and determining whether or not they contain criminals. This study discusses a new technical system. This device can recognize criminals' faces in real time by viewing camera footage. It notifies relevant individuals who can help when it detects a criminal's face. How does it work? To detect faces faster, the algorithm first reduces the size of the images. It then tracks these faces over time to ensure they belong to the same person. To determine its confidence in identifying a person's face, the system assigns a number. This number is determined based on how accurately it detects the face. The final number takes into account the system's reliability, as well as how similar the face is to other faces of known criminals. This way, even if security personnel overlook something, the system can detect it early, especially in areas where negative events are likely to occur. One advantage of this system is the speed with which cameras display images, even if they are smaller. By closely observing the faces of perpetrators and avoiding errors, it can accurately identify them. In cases where there are many faces, the system also solves a problem. By accumulating decision scores, the algorithm improves over time. The

system has shown good performance in tests, making correct decisions approximately 90% of the time. It has even outperformed other similar systems. This system could significantly facilitate the detection of criminals and the protection of important sites, preventing negative outcomes.

1.3. Aim of The Project:

This project aims to make it easier for law enforcement to locate criminals from anywhere, without wasting time or money. This program also contributes to the preservation of criminal records. All criminal information is available in our app. As a result, this app benefits law enforcement in many ways.

1.4. Objectives of The Project:

1. The primary goal of the facial recognition-based instant criminal identification application is to assist law enforcement in identifying criminals.
2. This application aims to provide details about a specific criminal we are trying to track down.
3. This program allows police officers to locate criminals anytime, anywhere.
4. Any police officer can access this program online at any time, from anywhere.
5. This application is accurate, fast, reliable, and fairly easy to use.

1.5. Literature Review:

Several methodologies have been proposed for real-time face recognition. Viola-Jones developed a framework that can accurately detect faces in challenging conditions, such as erratic head movement or poor lighting. Ni Kadek et al. proposed an eigenface approach that uses OpenCV library for face recognition. Shreyak Sawhney et al. created a real-time smart attendance system that uses Eigenface values, PCA, and convolutional neural networks. Weihua Sheng et al. established a facial recognition framework for a security system using TensorFlow.

1. **Nurul Azma Abdullah, Md. Jamri Saidi and Nurul Hidayah Ab Rahman:**This undertaking plunges into the subject of facial acknowledgment and facial location in a computerized correspondences framework. Face acknowledgment is an innovation that is broadly utilized today which carries different advantages to society. Facial acknowledgment varies from facial discovery in the perspective that facial identification just finds and recognizes the current face/s in a picture while, in facial acknowledgment, the PC tracks down the face/s present in a subject and can recognize the face from an example of various countenances. This exploration centers around the execution of both a facial acknowledgment framework and a facial identification framework in MATLAB. This examination would utilize the different imaging tool compartments accessible in the program and would be decided on its capacity to distinguish and perceive an example in each data set precisely. Moreover, this framework ought to have the option to make and to peruse an information base of various faces [4].
2. **Apoorva.P, Ramesh.B and Varshitha.M.R:** We understand that a person's face is a unique and essential component of their body that

sets them apart. We can use it to track the personality of a criminal in this way. As a result of technological advancements, CCTV is installed in many public areas to capture criminal activity. The lawbreaker face recognition system can be used with the newly apprehended faces and criminals' photographs that are available at the police headquarters. This article presents a programmed criminal ID framework that the Police Division can use to upgrade and modernize their crook recognition process into a more effective and efficient one. Through inventiveness, this idea will complement the current framework and raise the bar for recognizing thieves. We compare the captured images of the person reaching that public location with the criminal data in our database. The framework will display the person's picture on the framework screen and indicate with their name that the crook has been located and is present in this open spot if it assumes the person's face from the public spot coordinates. This framework synchronizes more than 80% of the captured photographs with those from the data set [5].

3. **Rasanayagam, K.Kumarasiri, S.D.D, Tharuka, W. A. D. Samaranayake, N. Samarasinghe and P. Siriwardana:** The quick financial improvement in South Korea has brought about increment of wrongdoings. Opportune identification and decrease of violations are essential focal point of cops. Web of Things (IoT) and progressively modest and wearable sensors can be utilized to work with this errand. For the most part, the use of IoT innovations to the fields of savvy urban communities, brilliant strategies and medical services should be visible more regularly. In this paper, we present the plan of IoT based brilliant wrongdoing recognition framework. The proposed framework can distinguish violations continuously by breaking down the human emotions [6].

4. **Mantoro, T., Ayu, M. A., Suhendi.**: Considered as item-based picture examination (OBIA). It is a viable strategy for high spatial goal (HSR) imaging. Order by an unmistakable and instinctive specialized process. In any case, OBIA depends on manual change of the picture. Arrangement capability. This is interesting work. Profound learning (DL) The innovation consequently gains picture highlights from an enormous number of pictures, Accomplishing higher picture characterization exactness than before Procedure. The review utilizes another technique called object scale versatile convolutional brain organizations (OSA-CNN), Consolidate OBIA and CNN, suggested for HSR pictures arrangement. To begin with, OSA-CNN gathers picture masses Head pivot of the item crude taken from the picture division; the size of the previous is resolved naturally by the hub width of the last option. This step produces the info Units expected for CNN arrangement. Second The crush and excitation blocks are extricated from the SE network [7].

Chapter Two

Design of the System

2.1. Overview About the Proposed System:

With the rate of crime and criminals increasing day by day, managing, locating, and tracking these criminals has become a major challenge for police officers. While some applications help police departments store records and data on criminals, they do not help in locating them. Various details were managed using log books or stored as software records in data requirements. Previously, when a criminal was convicted, their photo was taken and stored in a record, but these photos are useless. Current records management only helps in identifying criminals and helps in locating criminals from anywhere.

The proposed system deals with the current trend of security systems, i.e., inclusion of privacy concerns, cost effectiveness, and efficient alarm systems. Surveillance in dynamic scenes attempts to recognize, detect, and track certain objects from image sequences, and more generally to understand object behaviors. The objective is to develop an intelligent visual surveillance system to replace the traditional passive video surveillance system.

In this project, we use a classification method for analyzing the video and detecting anomalies. The user should provide a live webcam feed. The fed video is then converted into frames, and it is analyzed for abnormalities. We first check whether there are some human actions present in the frames. This checking is done with the help of the trained models. These trained models are compared with the actions present in the frame to detect human presence in that frame. This is done using a

program, and the code is written in OpenCV using Python using Visual Studio Code Environment.

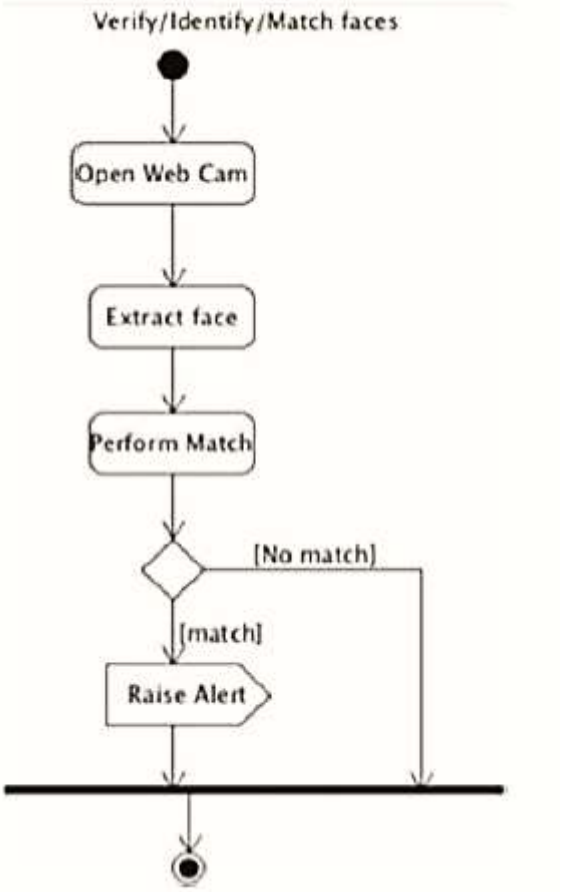


Fig. (2- 1) The workflow of the project.

2.2. Software Requirements:

Since the project is only a software project, we needed a number of software requirements to complete the project, which we will mention in the following sections:

2.2.1. Visual Studio Code:

One of the most common requirements in software development today is building applications and services that run on multiple systems and devices, especially with the continued expansion of cloud and artificial intelligence services. Visual Studio Code is not just another evolved Notepad with syntax colorization and automatic indentation. Instead, it is a very powerful code-focused development environment expressly designed to make it easier to write web, mobile, and cloud applications using languages that are available to different development platforms and to support the application development lifecycle with a built-in debugger and with integrated support for the popular Git version control engine. With Visual Studio Code, Programmers and developers can work with individual code files or with structured file systems based on folders [8].

Visual Studio Code is the first cross-platform development tool in the Microsoft Visual Studio family that runs on Windows, Linux, and macOS. It is free, open source, and it is a code-centric tool, which makes it easier to edit code files and folder-based project systems as well as writing cross-platform web and mobile applications over the most popular platforms, such as Node.js and .NET Core, with integrated support for a huge number of languages and rich editing features such as IntelliSense, finding symbol references, quickly reaching a type definition, and much more [8].

Visual Studio Code is based on Electron, a framework for creating cross-platform applications with native technologies, and combines the simplicity of a powerful code editor with the tools a developer needs to support the application lifecycle development, including debuggers and version control integration based on Git. It is therefore a complete development tool, rather than being a simple code editor [8].

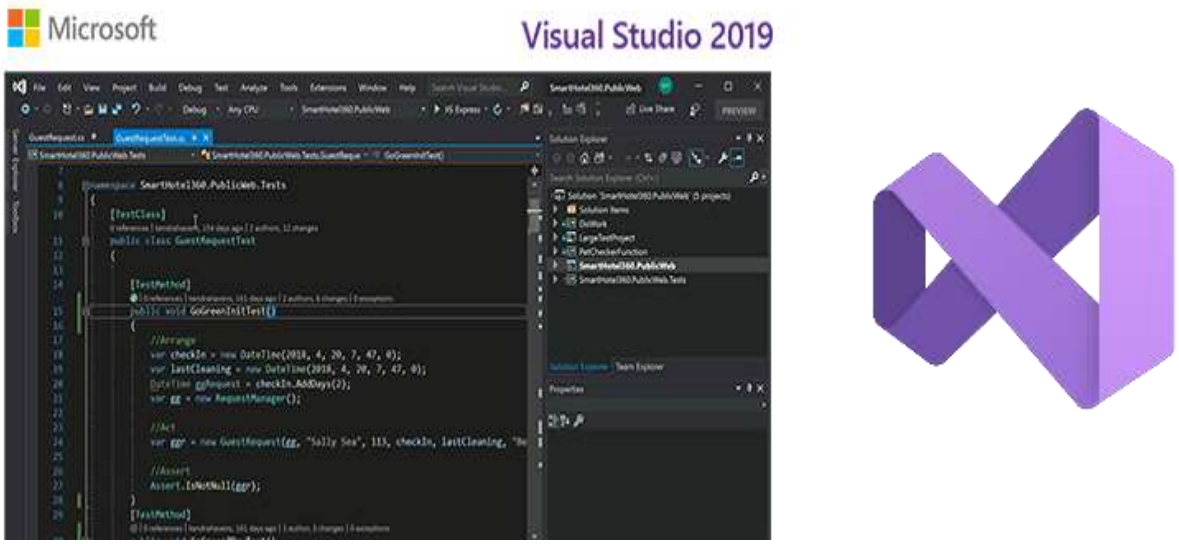


Fig. (2 -2) Visual Studio Code.

2.2.2. Python:

Python is an interpreted high-level programming language for general-purpose programming that emphasizes code readability. It provides constructs that enable clear programming of both small- and large-scale software applications. Python features a dynamic type system and automatic memory management. The Python language supports multiple programming paradigms, including object-oriented, functional, and procedural, and has a large and comprehensive runtime library [9].

Python is an open source and cross-platform programming language, that has become increasingly popular over the last ten years. It was first released in 1991. Python is an object-oriented programming

language (OOP), but it's can use Python in basic application without the need to know about or use the object-oriented features in Python [10].

Python is a popular programming language, and it is one of the most used programming languages today. Python works on all the main platforms and operating systems used today, such Windows, macOS, and Linux. Python is a multi-purpose programming language, which can be used for simulation, creating web pages, communicate with database systems, etc [10].

Fig. (2 -3) Python.

2.2.2.1. Dlib:

Dlib is a powerful open-source toolkit that provides a range of machine learning and computer vision functionalities, accessible through its Python API. It is widely used for tasks such as face detection, facial landmark estimation, and face recognition [11].

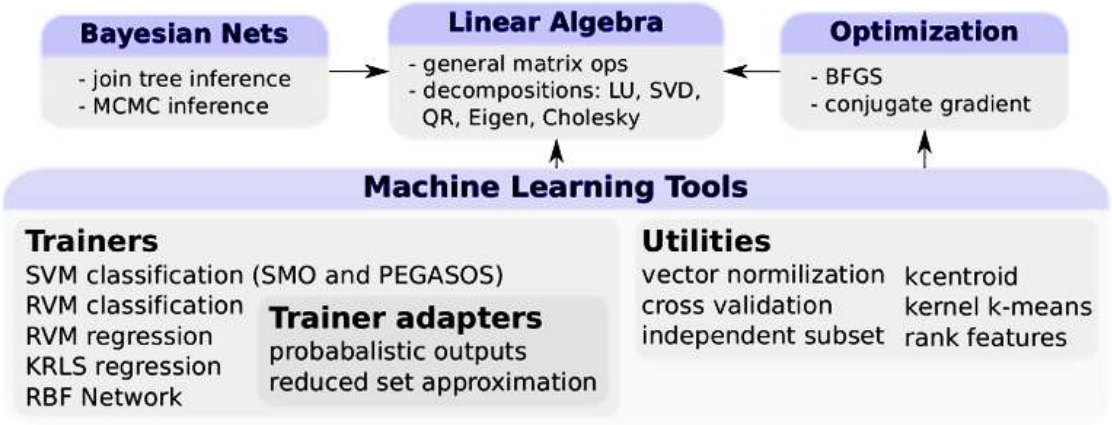


Fig. (2- 4) Elements of dlib-ml. Arrows show dependencies between components.

Face detection methods in Dlib [12]:

1. **HOG + Linear SVM Detector**: This method utilizes Histogram of Oriented Gradients (HOG) features combined with a linear Support Vector Machine (SVM) classifier. It is efficient and performs well under standard conditions but may struggle with faces at extreme angles or under varying lighting conditions.
2. **CNN-Based MMOD Detector:** Employing a Max-Margin Object Detection (MMOD) Convolutional Neural Network (CNN) model, this approach offers higher accuracy and robustness, especially in challenging scenarios. It requires more computational resources and benefits significantly from GPU acceleration.

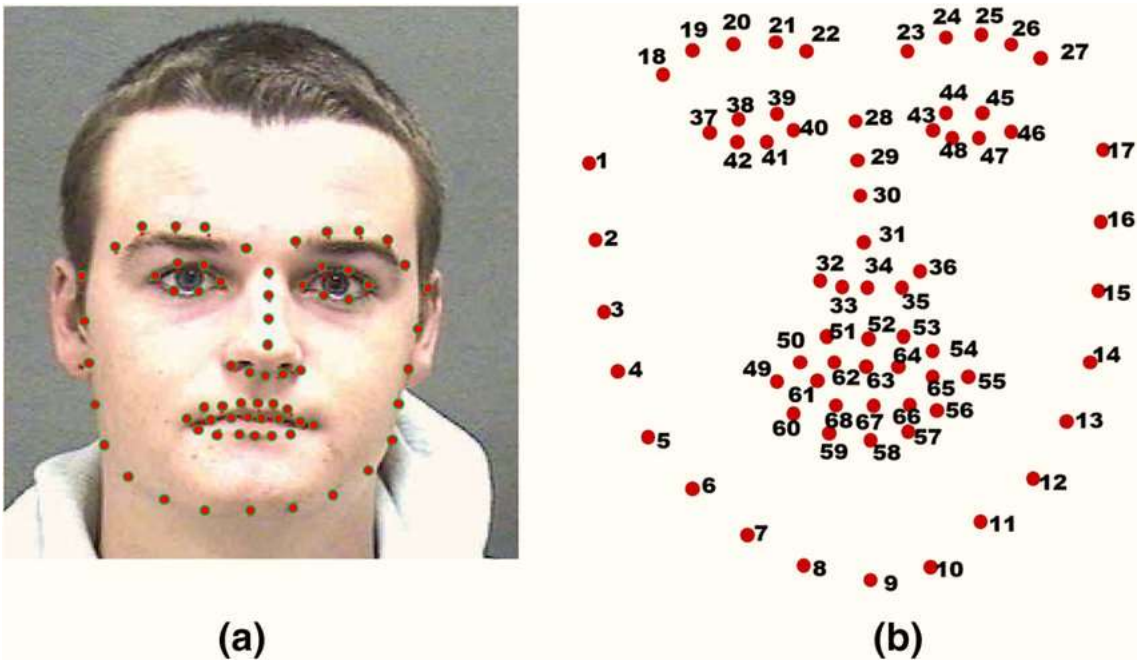


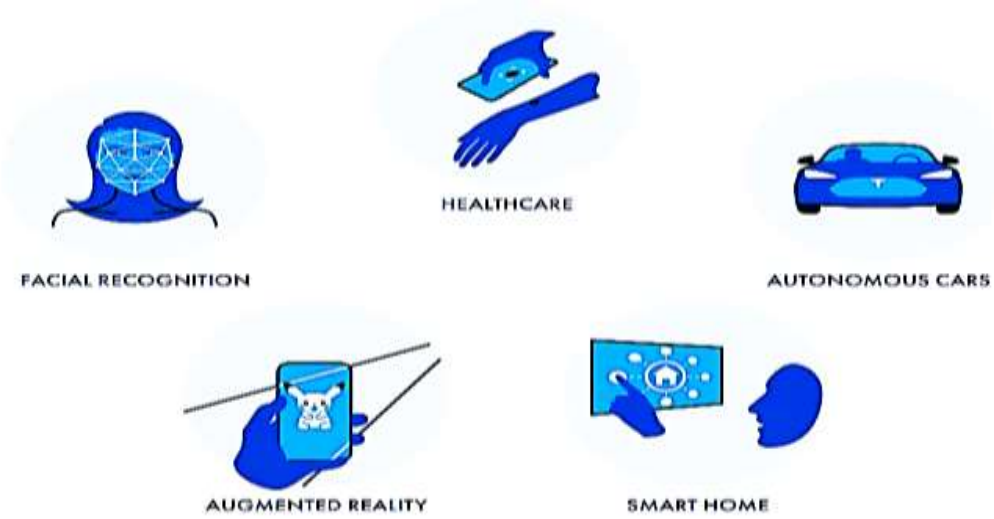
Fig. (2- 5) Identification of facial landmarks using Dlib. (a) Facial landmarks. (b) The position and order of 68 points on the face.

2.2.2.2. OpenCv:

Computer vision is a rapidly growing field, partly as a result of both cheaper and more capable cameras, partly because of affordable processing power, and partly because vision algorithms are starting to mature. OpenCV itself has played a role in the growth of computer vision by enabling thousands of people to do more productive work in vision. With its focus on real-time vision, OpenCV helps students and professionals efficiently implement projects and jump-start research by providing them with a computer vision and machine learning infrastructure that was previously available only in a few mature research labs [13] .

OpenCV-Python is a Python wrapper around the C++ implementation of the OpenCV library. It makes use of the NumPy library for numerical operations and is a rapid prototyping tool for

computer vision problems. OpenCV-Python is a cross-platform library, available for use on all Operating systems (OS) platforms, including Windows, Linux, macOS, and Android. OpenCV also supports the Graphics Processing Unit (GPU) acceleration. Prior knowledge of Python and the NumPy library is essential to understand the functionality



of OpenCV-Python [14] .

Fig. (2- 6) OpenCV Applications.

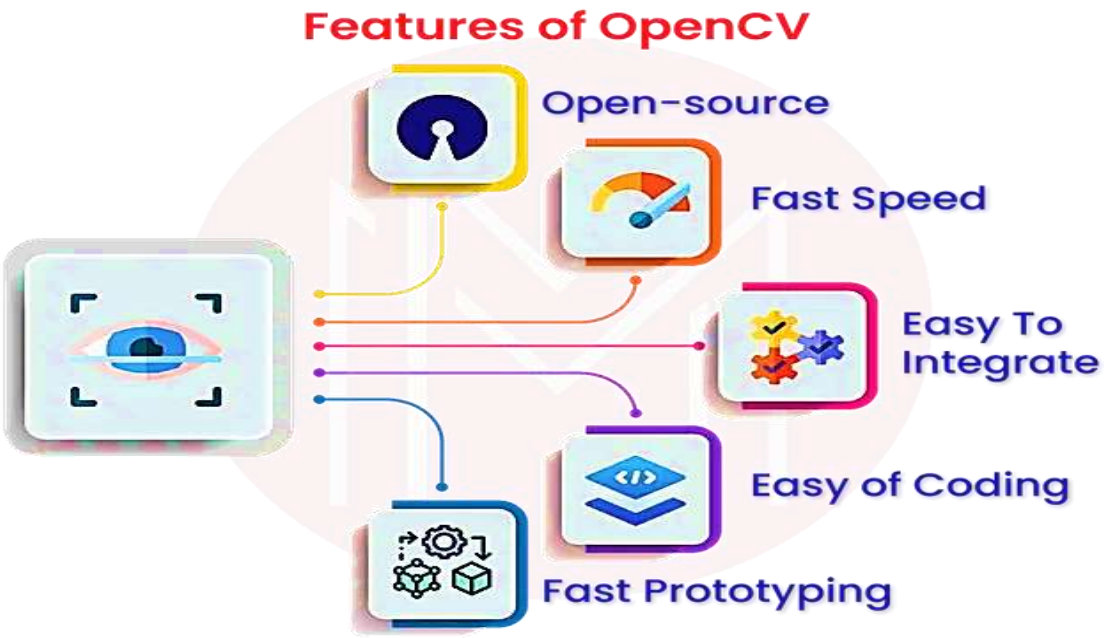


Fig. (2- 7) Features of OpenCV.

2.2.2.3. Numpy:

Numpy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data. Array in Numpy is a table of elements (usually numbers), all of the same type, indexed by a tuple of positive integers. In Numpy, number of dimensions of the array is called rank of the array. A tuple of integers giving the size of the array along each dimension is known as shape of the array. An array class in Numpy is called as ndarray. Elements in Numpy arrays are accessed by using square brackets and can be initialized by using nested Python Lists. Arrays in Numpy can be created by multiple ways, with various number of Ranks, defining the size of the Array. Arrays can also be created with the use of various data types such as lists, tuples, etc. The type of the resultant array is deduced from the type of the elements in the sequences [15] .

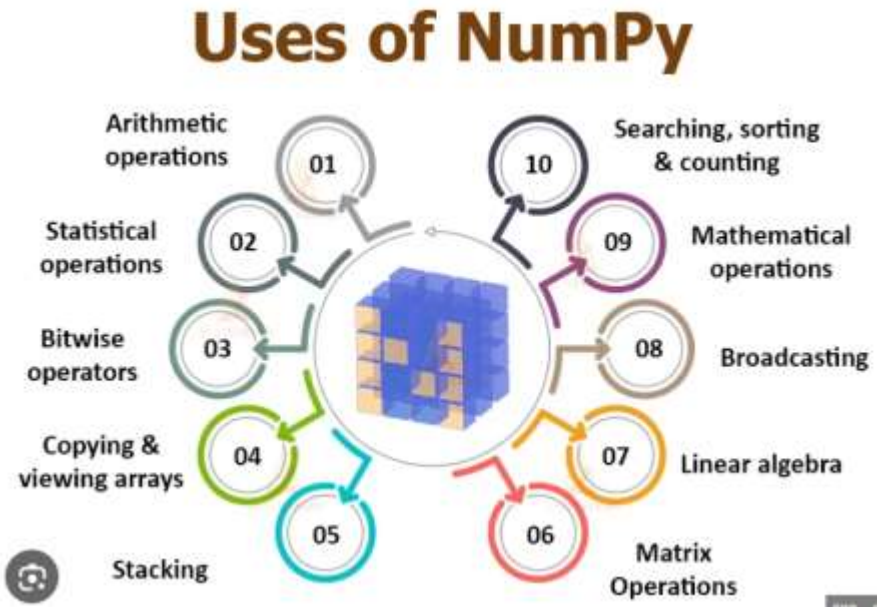


Fig. (2- 8) Uses of Numpy.

2.2.3. SQLite:

SQLite is a highly efficient, server less, and self-contained SQL database engine that stands out for its simplicity and ease of integration. Designed to be embedded within applications, SQLite eliminates the need for separate database server processes and complex configurations.

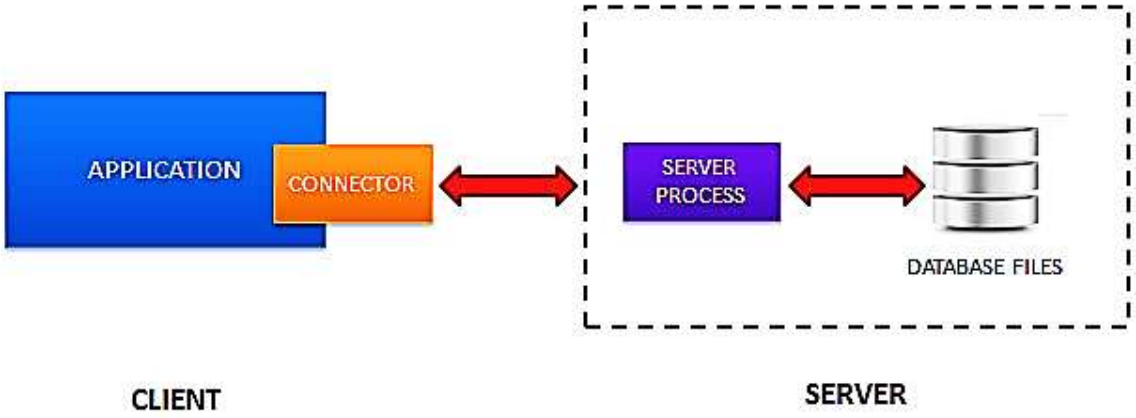


Fig. (2- 9) The client/server architecture of a relational database management system.

2.3. Proposed System:

The proposed project is a comprehensive criminal face recognition system, designed using PyQt5 for the GUI and OpenCV, along with Dlib for face detection and recognition. Upon startup, the program checks for the availability of the machine learning models needed to detect and recognize facial features. The user interface is divided into three main tabs: Face Recognition, Crime Management, and Search. In the Face Recognition tab, the program activates the computer's camera and processes each video frame to detect and recognize faces. If a recognized criminal is found (based on previously stored facial codes), the program highlights the face in red and displays their name, confidence level, and crime details directly on the video. Civilians (not identified as criminals) are highlighted in green only, with no additional details displayed.

The Crime Management tab allows the user to add, edit, or delete criminal records from the database. Each criminal record contains the criminal's name, crime type, crime details, facial code (obtained through image analysis), and the image itself. When adding a new criminal, the system ensures that the selected image contains only one face to avoid misidentification. The user can search for an image, enter its details, and save the record, which updates both the database and the internal identification system. Editing or deleting existing criminals is equally streamlined. The program also handles cases where the image is missing or incorrect, providing clear error or success messages for user feedback. A "Refresh Data" button is available to reload criminal data into the identification system without having to restart the entire application.

The "Search" tab also offers a quick search feature that enables users to search the criminal database by name, crime type, or details. Matching results are displayed in a list, and upon selecting an item, full details about the criminal are displayed, such as name, crime type, detailed crime description, and date added. The system ensures smooth operation throughout the application by properly handling camera on/off, seamlessly managing exceptions, and ensuring the validity of user input. Overall, this software creates a powerful and easy-to-use tool for real-time crime detection, criminal database management, and record searching all integrated into an intuitive and professional application.

We developed the project as a desktop application with a user-friendly interface. The development was carried out using Visual Studio Code and the Python programming language, alongside several essential libraries, including face recognition, dlib, OpenCV, NumPy as np, os, sqlite3, PyQt5, and cv2.

Using Visual Studio Code, we established the project's file structure, downloaded the necessary models, and created a main.py file to serve as the core of the application. Once the environment was set up, we proceeded to implement the code within main.py, ensuring that the application, upon execution, would automatically generate a criminal record file and a dynamic database. This database is designed to allow for updates, modifications, and efficient management of criminal profiles and individuals. This phase covered the overall architecture, design, and foundational setup of the project.

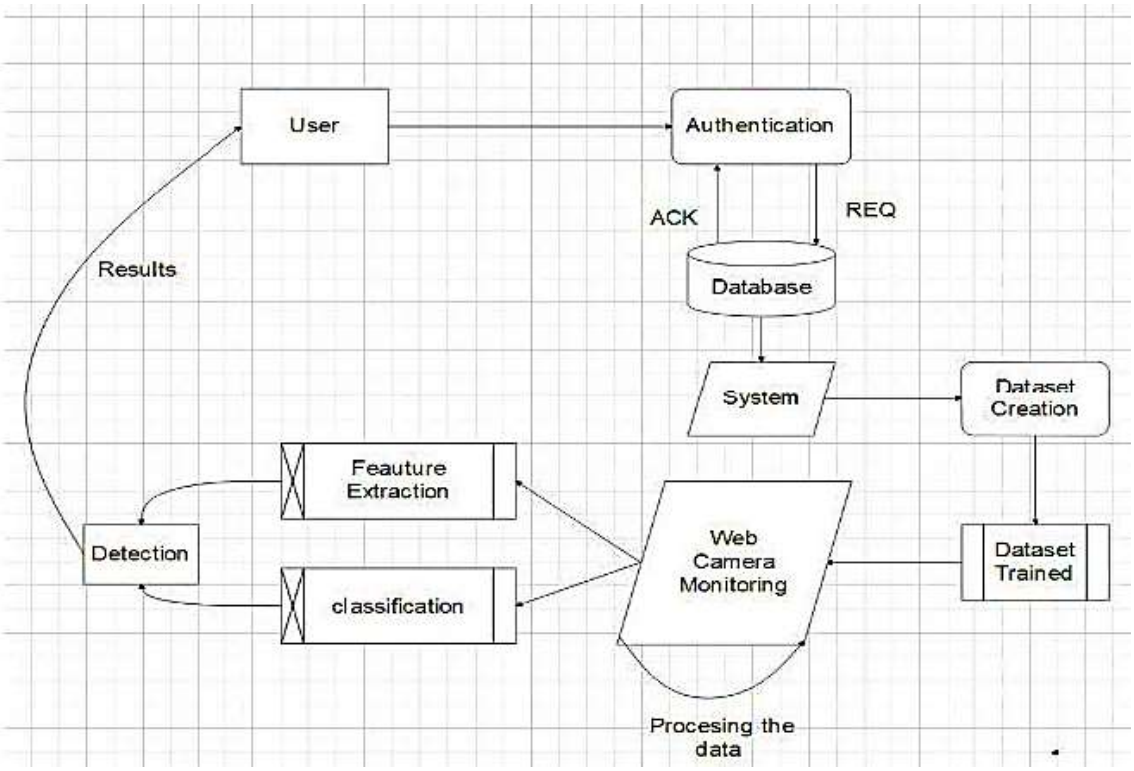


Fig. (2- 10) The architecture of the project.

Chapter Three

Methodology

3.1. Overview:

In this chapter, we will discuss the program we designed to detect criminals through web camera, and the algorithms we used to identify the criminal's face and details, we will go into details, such as office algorithms like HOG and SVM, basically, and how to recognize faces, identify faces, read faces, determine their shape, and compare faces (in short, how the project works in detail) and how the code works.

3.2. Principle of Face Recognition Algorithm:

A common technique is face recognition, which uses computers to extract facial features from people and use them to confirm their identity. Often, a specific image is used, and one or more faces are then identified from the scanned image using a currently available face database. Figure 3-1 illustrates a complete flowchart of face recognition technology. This project uses photo capture as an approach to capturing facial images.

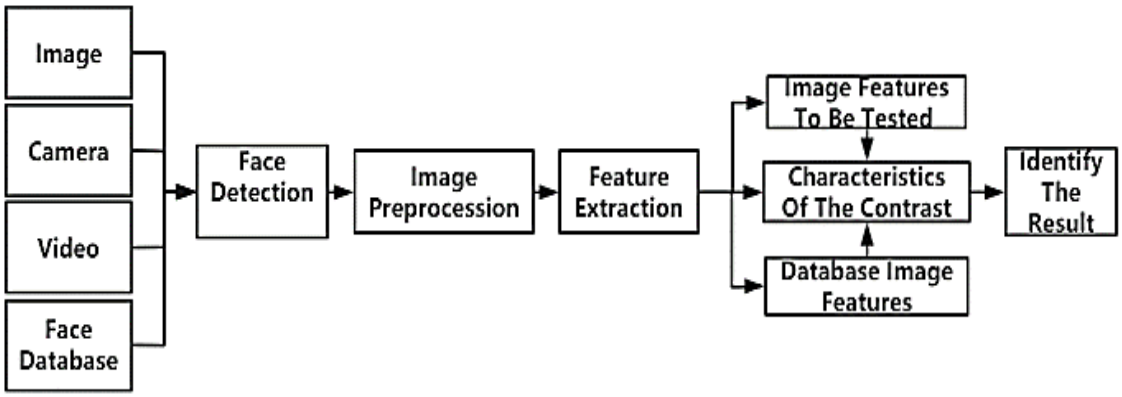


Fig. (3-1) Schematic Diagram of Face Recognition Algorithm.

Face detection is used to locate a face in an image after it has been input. However, excessive noise can cause distortion when the image is input into the computer. Therefore, image preprocessing is the first step before feature extraction. Features can only be extracted after the image has been standardized. Ultimately, the face is recognized, and the recognition process produces these results.

3.3. Histogram of Oriented Gradient (HOG):

HOG (histogram of oriented gradient) means directional gradient histogram, the essence of which is the statistical gradient information, and this gradient mainly exists in the edge. The process of its implementation is to obtain the pixels (x, y) in the original image and gradient component in X direction and Y direction respectively. Then, the horizontal gradient and vertical gradient at the input image pixel can be calculated by using formula (1) and (2).

$$G_x(x,y) = H(x+1,y) - H(x-1,y) \quad (1)$$

$$G_y(x,y) = H(x, y+1) - H(x, y-1) \quad (1)$$

In the above formula, $G_x(x,y)$ is the horizontal gradient at (x,y), $G_y(x,y)$ is the vertical gradient at (x,y) and H (x, y) is the pixel value of (x, y). Thus it can be seen that the gradient amplitude G(x,y), and gradient direction α at the pixel (x, y) are as follows:

$$G(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \quad (3)$$

$$\alpha = \tan^{-1}\left(\frac{G_y(x,y)}{G_x(x,y)}\right) \quad (4)$$

Then the image is divided into small cells and the gradient histogram of each cell is counted to form the descriptor of each cell unit, namely HOG feature. After that, several cell units are formed into a block (e.g. 3 * 3 cell units per block), and the descriptors of each cell unit are connected in series to form the HOG feature of the block.

3.4. Support vector machines (SVMs):

are designed to solve a traditional two-class pattern recognition problem. We adapt SVMs to face recognition by modifying the interpretation of their classifier and creating a representation of face images that matches the two-class problem. Traditional SVMs return a binary value, the object's class. To train our SVM algorithm, we formulate the problem in a different space, one that explicitly identifies the differences between two face images. This represents a departure from the traditional face-space or view-based approach, which encodes each face image as a separate face view.

In a different space, we are interested in the following two classes: the dissimilarities between images of the same individual, and the dissimilarities between images of different people. These two classes are the input to an SVM algorithm. An SVM algorithm generates a decision surface separating the two classes. For face recognition, we re-interpret the decision surface to produce a similarity metric between two facial images. This allows us to construct face-recognition algorithms.

3.5. The Workflow of the Algorithms:

In this project, an intelligent criminal face detection system was developed, similar to a system for detecting specific object classes. Facial recognition systems have also been used to develop effective security systems for many independent services, expanding their scope from banking and securities surveillance to counterterrorism systems, where criminals can be identified at airports and border crossings. Many social media sites use surveillance systems to provide security for their users, such as tagging and locking profiles.

The main goal is to create a Criminal face recognition system capable of recognizing any type of image and making continuous improvements and changes autonomously. Another important task is saving time and operating the system in real time by removing noise from images, such as light, facial hair, and so on.

The first step in creating a facial recognition system involves comparing two images that are similar in their 128-dimensional values, which are accepted by the system or network, while other images that are not similar are ignored.

The image is then converted to RGB color order, the input image is parsed, faces are identified, and their 128-D relative encoding is calculated. Matching is performed by comparing each face with the current encoding dataset. The function returns a true/false value for each query.

The OpenCV Face Recognizer uses local binary pattern (LBP) to texture an image by thresholding the neighborhood of each pixel and

generating a binary number as a result. The four parameters of LBPH are radius, neighbor, X-grid, and Y-grid. Each image in the training set must have an identifier. First, the image is enhanced using a sliding window based on parameters such as radius and neighbors.

A grayscale image is a 3x3 pixel window matrix with a central threshold value. This value represents the neighboring values of the eight neighboring values. For each threshold value, all values below the threshold are assigned a value of 0, and all values above it is assigned a value of 1. The matrix contains only 0s and 1s, which are then converted to a decimal value. Using this image, a histogram is extracted using the X-grid and Y-grid parameters, by dividing the image into multiple grids.

Histograms are compared using Euclidean distance, chi-square, and absolute value. The calculated distance is called confidence. Lower the confidence the closer you get to the histograms.

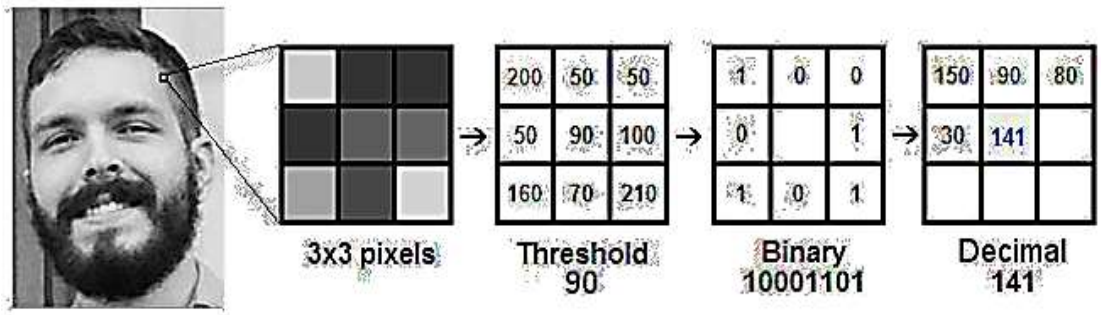


Fig. (3-2) Various conversions of an image from pixels to decimal.

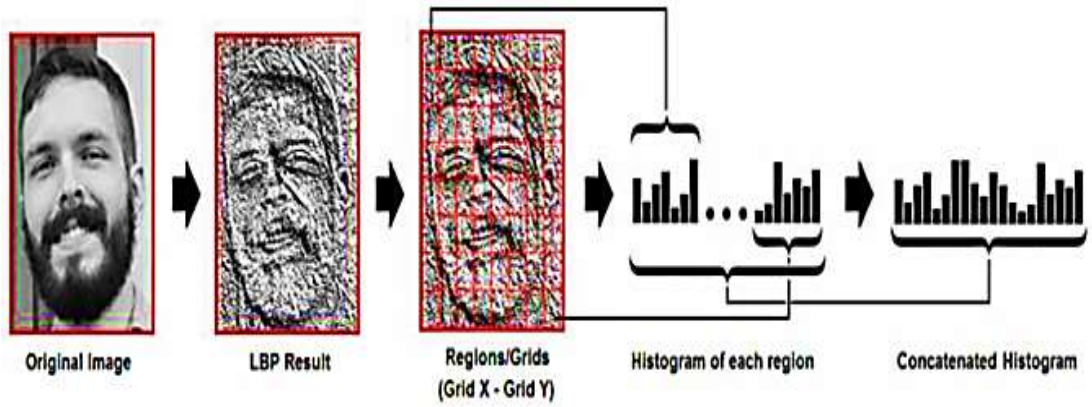


Fig. (3-3) Concatenated and individual histograms of the original image.

A grayscale image is a 3x3 pixel window matrix with a central threshold value. This value represents the neighboring values of the eight neighboring values. For each threshold value, all values below the threshold are assigned a value of 0, and all values above it is assigned a value of 1. The matrix contains only 0s and 1s, which are then converted to a decimal value. Using this image, a histogram is extracted using the X-grid and Y-grid parameters, by dividing the image into multiple grids.

The main steps consist of face detection, feature extraction, data comparison, and finally face recognition. Facial landmarks are detected by Dlib and OpenCV. Using Dlib's advanced facial recognition methodology, built on deep learning techniques, 68 defined points can be applied to any of the following, from the top of the chin to the outer edge of the eye, etc., as shown in Figure 3-4. Here, only basic transformations, such as rotation and scaling, that preserve parallel lines are used, called affine transformations.

Fig. (3-4) Depiction of 68 facial points.

Figure 4 illustrates the complete process of face detection, feature detection, and transformation to obtain a fully centered image. The image is encrypted, capturing multiple images of a single person and extracting facial features. A total of 2,278 distances are calculated for 68 given points. For recognition, the image with the minimum distance, which has the maximum similarity to the same person, is considered the real image as output. To distinguish between unknowns, an upper bound is set, and if the difference is less than that value, the minimum value is set to represent the person. HOG Feature Vector Descriptor The feature vector transforms an image of a region distributed across three channels, i.e., length, width, and height, into a feature vector of a matrix of length n.

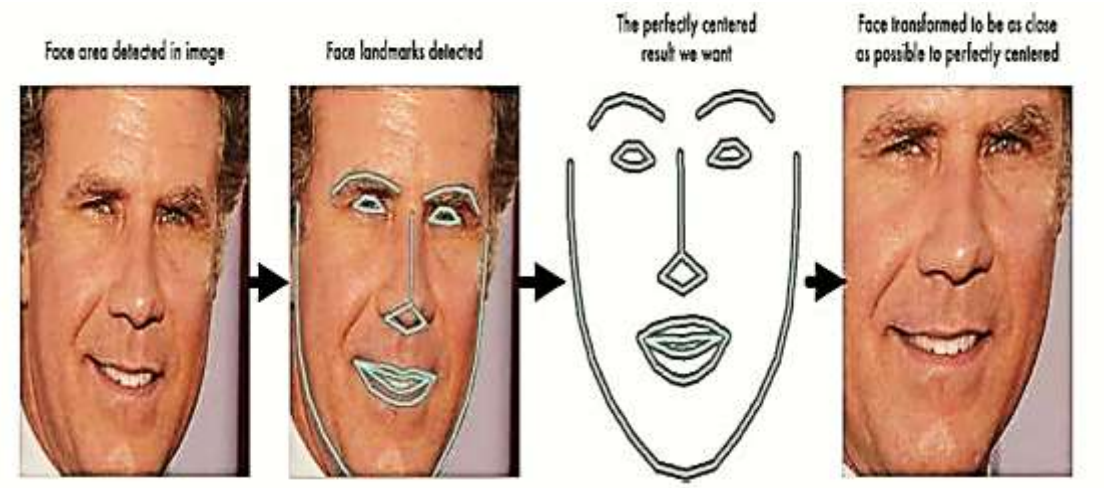


Fig. (3-5) Facial landmarks extraction.

In this case, the input image size is 64 × 128 × 3, and the resulting feature vector has a length of 3780. The horizontal and vertical gradients can be calculated as follows: gx(x gradient) = df(x,y)/dx and gy(y gradient) = df(x,y)/dy, where g is the magnitude and θ is the direction of the gradient. The image is divided into 8 × 8 unit cells, and a gradient is calculated for each cell. An 8 × 8 image patch contains 8 × 8 × 3 = 192 pixels. The gradient of this patch contains two values (magnitude and direction), resulting in 8 × 8 × 2 = 128 values. Here, the representation becomes more compact and less susceptible to noise. HOG was used for person detection. Initially, the 8 × 8 cells in a 64 × 128 pedestrian image were suitable for detecting active features such as the face, the top of the head, etc.

The central image with arrows shows the gradient direction, and the length shows the magnitude. The image to the far right shows the digital representation of the gradients in these 8x8 cells, with one end changing the angles between 0 and 180 degrees instead of 0 and 360 degrees.

A histogram of the gradients in these 8x8 cells is generated. These cells consist of nine bins associated with the angles 0, 20, 40, and 160. In Figure 3-6, the bin is chosen based on the direction. The angle is 80 degrees and the magnitude is 2. Therefore, 2 is added to the fifth bin. The pixel circled in red represents the angle of 10 degrees and the magnitude is 4. The pixel volume is divided between the middle of the pixels 0 and 20 bins.

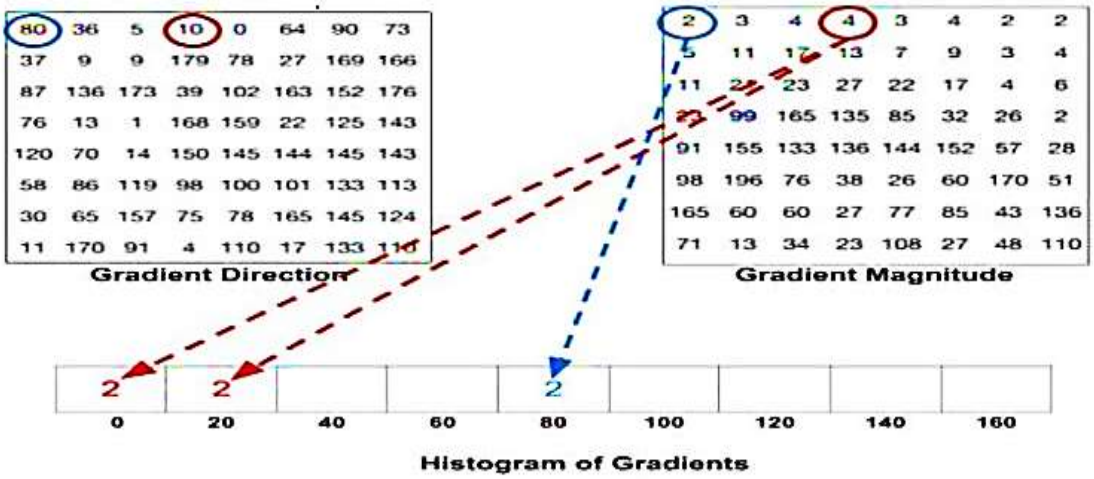


Fig. (3-6) Bin selection.

Fig. (3-7) Trained Face.

Figure represents a standard face that has been trained with 3000 images by Dlib library.

3.6. The Proposed System:

This system matches face images captured from cameras or photos with a stored image database, this technology is highly effective in identifying known criminals. This system provides a reliable, interpretable, and efficient solution for face recognition applications using traditional machine learning techniques such as SVMs and HOG, especially in structured or controlled contexts.

In feature extraction, the HOG descriptor is fundamental. It calculates the gradient or edge direction within each small cell that makes up a face image. The face's structure and shape are then represented by a global feature vector, created by concatenating these directions into distribution plots. This technique is suitable for capturing important visual elements while ignoring unnecessary information such as color changes or background noise. The HOG descriptor is suitable for real-time applications where accuracy and speed are critical, due to its ability to tolerate subtle variations in lighting and location.

After extracting HOG features, they are fed into a supervised learning classification technique called a Support Vector Machine (SVM). Based on their feature vectors, the SVM determines the best hyperplane to segment different classes, in this example, distinct individuals. The SVM is trained to classify these descriptors based on their respective identities. In face recognition technology, each face is represented by its own HOG descriptor. Similarly, an unknown face is processed, and the trained SVM model compares it with the learned features to estimate its identity. If the face matches a face in a criminal database, the model can then notify the police.

Collecting a face image dataset is the first step in the overall process of this system. To improve consistency, the images are preprocessed using resizing, alignment, and normalization. The data is used to train an SVM after feature extraction using HOG. Face detection techniques are used to find faces in video frames while the system is running in real time. The trained SVM model classifies each detected face after undergoing the HOG transformation. The technology sends an alert if a match to a known criminal is detected, enabling security personnel to act quickly and efficiently.

3.7. The Workflow of the System:

- Data Collection: Facial images, such as those of arrested individuals, are collected and stored in a database.
- Preprocessing includes image alignment, resizing, and normalization to reduce illumination variations.
- HOG feature extraction: A HOG description vector is generated from each facial image.
- SVM model training: An SVM model is trained using the HOG vectors and their associated identities (labels).
- Instant Face Detection: Instant face detection, such as HOG with sliding windows, is used in video surveillance systems to identify faces in images.
- Feature extraction from identified faces: Each identified face is converted to HOG format.
- Trained SVM prediction: Using the acquired crime database, the SVM model classifies the face.
- Alert generation: An alert is generated for further action if a match to a criminal identity is detected.

3.8. Results:

The provided content describes the implementation and testing of a prototype system, as depicted in the Figure, to validate the functionality of a proposed model and platform. This prototype consisted of a desktop application, an administrator's page, and a server computer with a deep learning model linked to the webcam.

Through this prototype, real-time monitoring of images from the surveillance camera was enabled, as depicted in the accompanying figures. A red square is placed on the criminal, and under the pictures in the program his name, the violence, the crime, and the details of the crime are written. As for the non-criminal, a green square is placed on him, and he usually appears as unknown. Additionally, the prototype included a push notification capability upon identifying faces associated with criminal activity, thus increasing the efficacy of the system in managing and preventing crime.

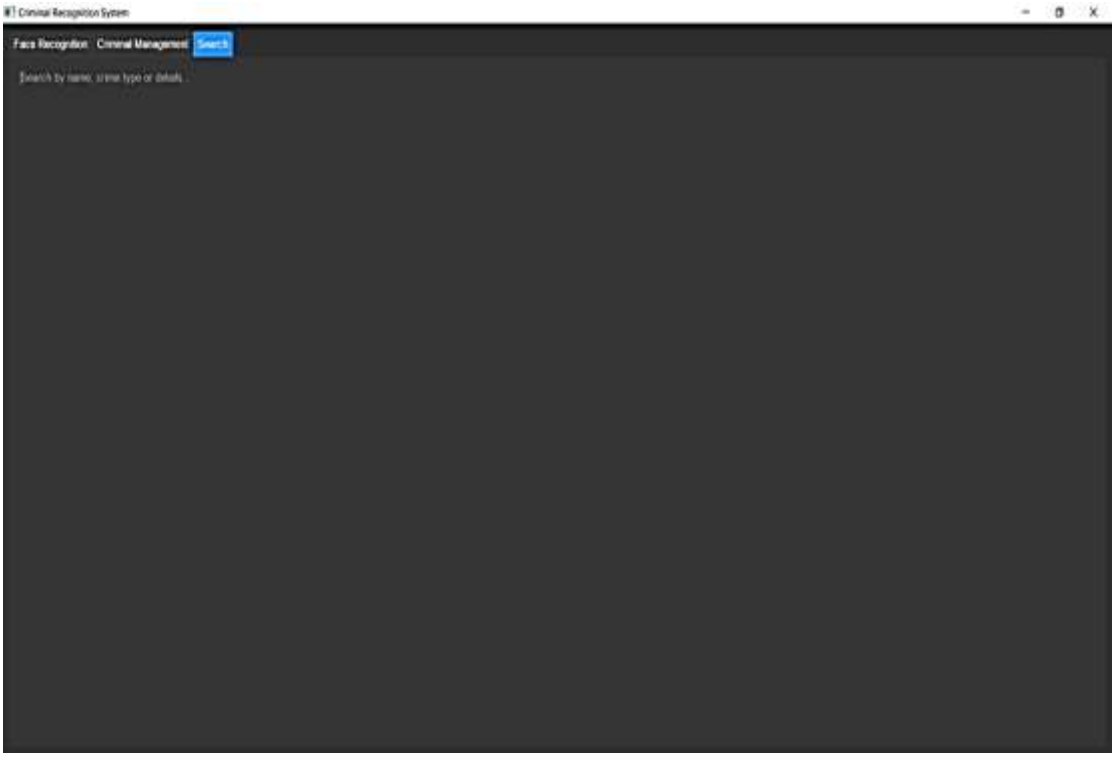


Fig. (3-8) Homepage.

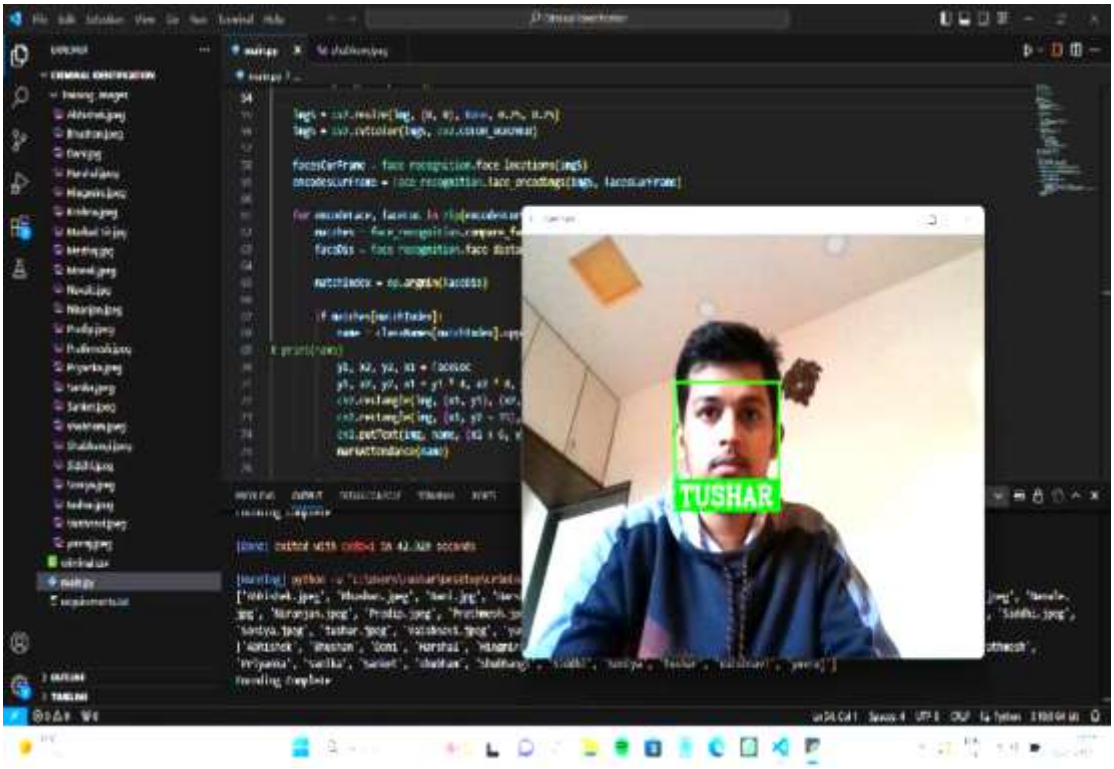


Fig. (3-9) Non-criminal Face.

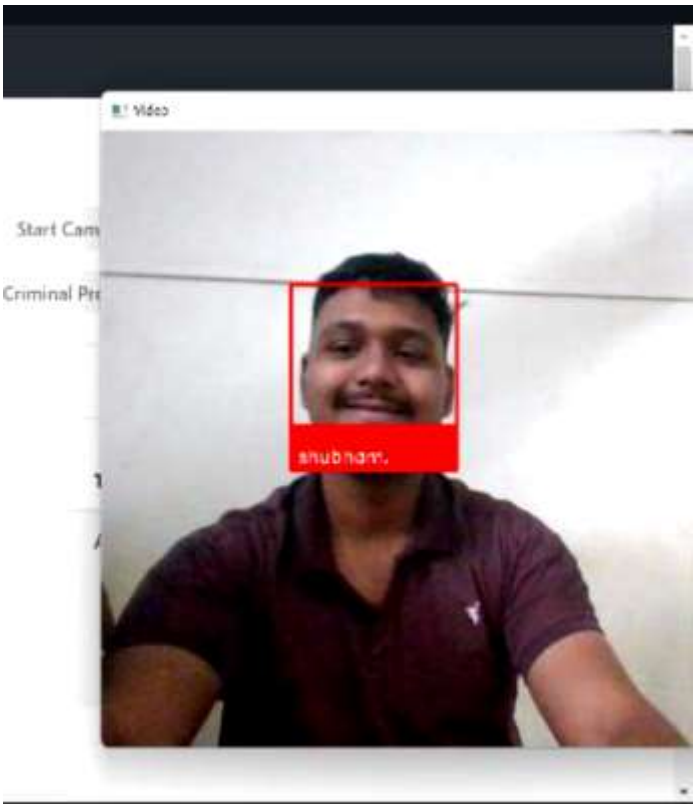


Fig. (3-10) Non-criminal Face.

Chapter Four

Conclusion & Future Works

4.1. Conclusion:

The project represents a vital and proactive response to the escalating challenges of criminal activities and security threats in our society. By harnessing advanced security technologies and innovative approaches, it aspires to significantly enhance crime prevention, detection, and identification. The application of predictive analytics and face recognition not only empowers law enforcement agencies to operate more efficiently but also fosters a sense of public and private security. Continuous technological advancements and global collaboration further underline the commitment to staying ahead of evolving criminal tactics. Ultimately, the project's success will lead to a safer and more orderly environment, reducing crime rates and improving the overall quality of life for the community it serves.

4.2. Future Works:

This system's capabilities could be expanded to simultaneously identify multiple faces, even from blurry or cropped images. Moreover, it could provide precise details regarding the exact location where a criminal was spotted, leveraging the camera's location data. The database could also be enhanced to include additional information such as age, previous criminal activity, associated individuals, and last known whereabouts, offering comprehensive insights into the individual's background.

In the future, advanced face recognition techniques can be used to improve the results, and a login page must be created so that any police personnel can access this application remotely. The application that is developed is simple and user-friendly; the interface of the application can be developed further according to user requirements.

REFERENCES

[1] Udgaonkar, Swaroop, Criminal Identification Using Face Recognition (August 9, 2021). Available at SSRN: https://ssrn.com/abstract=3901729 or http://dx.doi.org/10.2139/ssrn.3901729

[2] Kumar MR, Majeed A, Pasha F, Sujith A. Real-time criminal identification system based on face recognition. Advanced Science Letters. May 2020; 26(5): 320–328.

[3] Markad, D.A., Dongare, M.S., Mundada, M.P., Kulange, M.S., Argade, M.T., & Kale, M.H. Real-Time Criminal Face Detection: The Next Era of High-Precision Surveillance. International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences. 12(2). 2024.

[4] Nurul Azma Abdullah, Md. Jamri Saidi and Nurul Hidayah Ab Rahman "Face recognition for criminal identification: An implementation of principal component analysis for face recognition"The 2nd International Conference on Applied Science and Technology 2017 (ICAST'17).

[5] Apoorva.P, Ramesh.B and Varshitha.M.R "Automated criminal identification by face recognition using open computer vision classifiers" Third International Conference on Computing Methodologies and Communication (ICCMC 2019).

[6] Rasanayagam, K.Kumarasiri, S.D.D, Tharuka, W. A. D. Samaranayake, N. Samarasinghe and P. Siriwardana "CIS: An Automated Criminal Identification System". 2018 IEEE International Conference on Information and Automation for Sustainability (ICIAfS)R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[7] Mantoro, T., Ayu, M. A., Suhendi. (2018).” Multi-Faces Recognition Process Using Haar Cascades and Eigenface Methods” 2018 6th International Conference on Multimedia Computing and Systems (ICMCS).

[8] Del Sole, A. (2019). Introducing Visual Studio Code. In: Visual Studio Code Distilled. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-4224-7_1

[9] python.org, python.org- https://www.python.org, 2018.

[10] Hans-Petter Halvorsen. 2019. Python Programming. availabel on: https://www.halvorsen.blog/

[11] Python Package Index. (2025). dlib 19.24.8. PyPI. https://pypi.org/project/dlib/

[12] Rosebrock, A. (2021). Face detection with dlib (HOG and CNN). PyImageSearch. https://pyimagesearch.com/2021/04/19/face-detection-with-dlib-hog-and-cnn/

[13] Gary Bradski and Adrian Kaehler (2008). Learning OpenCV, [Computer Vision with OpenCV Library; software that sees]. O`Reilly Media. ISBN: 978-0-596-51613-0

[14] Joseph Howse and Prateek Joshi and Michael Beyeler, OpenCV: Computer Vision Projects with Python, 2016, Packt Publishing, ISBN: 978-1-78712-549-0

[15] Geeks for Geeks, Python NumPy, An article on Geeks for Geeks website, 2025, available on the next link: https://www.geeksforgeeks.org/python-numpy/