VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi, Karnataka-590018

Report of

Internship work entitled

"Artificial Intelligence and Machine Learning"

Internship Carried
at
Inventeron Technologies and
Business Solutions LLP,
Bangalore -560017

Submitted by

THEJASGOWDA M R   **4YG20EC021**

Under the Guidance of

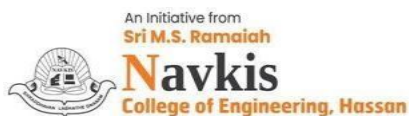| **Internal guide**	**External guide**
Mrs. Siddika Sabahath Anjum	**Mr. Mohammed Misbah**
Assistant Professor	**Director**
NCE, Hassan	Inventeron Technologies,
	Bengaluru

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
NAVKIS COLLEGE OF ENGINEERING, HASSAN -573 255

2023-24

NAVKIS COLLEGE OF ENGINEERING
HASSAN- 573201
DEPARTMENT OF ELECTRONICS &COMMUNICATION ENGINEERING

CERTIFICATE

This is to certify that the Internship Work entitled **"**Artificial Intelligence and Machine Learning**"** is bonafide Internship work carried out by **THEJASGOWDA M R (4YG20EC021),** in partial fulfillment of VIII Semester to award the Bachelor Degree in **Electronics & Communication Engineering** of the Visvesvaraya Technological University, Belagavi during the year **2023-2024**. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the Report and deposited in the department. The Internship Report has been approved as it satisfies all the academic requirements in respect of the Internship Work prescribed for the Bachelor of EngineeringDegree.

Mrs. Siddika Sabahath Anjum   **Ms.Asha J**   **Dr. Babita Kumari Jain**
Internal guide   **Internship coordinator**   **HOD**

Dr. M Venu Gopala Rao
Principal

Name and Signature of Examiners with Date

1. ────────────────   ────────────────

2. ────────────────   ────────────────

inventeron

nothing but smart...

Reg. No : AAB9565

INVENTERON TECHNOLOGIES AND BUSINESS SOLUTIONS LLP

CERTIFICATE OF INTERNSHIP

This is to certify that Mr/Ms. __Thejas gowda. M. R__
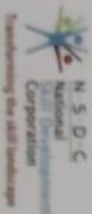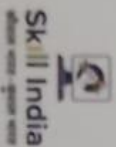
has completed internship on __Artificial Intelligence and Machine Learning__

from __14-AUG-2023__ to __14-SEP-2023__ successfully.

We wish this intern all the best for future endeavours.

For Inventeron Technologies And Business Solutions LLP

Managing Director

Skill India
कौशल भारत – कुशल भारत

N S D C
National
Skill Development
Corporation
Transforming the skill landscape

IoT

Raspberry Pi

Java

www.inventeron.com

ACKNOWLEDGEMENT

DECLARATION

I, Mr. Thejasgowda M R (4YG20EC021) student of Final Semester B.E, Electronics and Communication Engineering, Navkis College of Engineering, Hassan hereby declare that the internship work on "Artificial Intelligence and Machine Learning" has been carried out by me under the guidanceof Ms. SHRUTHI K M, Associate Professor, Department of Electronics and Communication Engineering, in partial fulfillment of requirements for the award of degree, Bachelor of Engineering in Electronics and Communication of the Visvesvaraya Technological University, Jnana sangama, Belagavi – 590018, during the academic year 2023-2024.

Place: Hassan   Thejasgowda M R

Date: (4YG20EC021)

ABSTRACT

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

CONTENTS

LIST OF FIGURES

CHAPTER 1

Company Profile

Inventeron Technologies and Business Solutions LLP, or ITABS, is an Indian-based engineering and electronics company headquartered in Bangalore, Karnataka, India. It is both a product and service-oriented software company having its products in wireless communication Technology and provides quality service to its valuable clients in its domain.

Fig 1.1: Company Logo

1.1 Profile

Even though the company was legally registered in the year 2013, it made its humble beginning in the year 2012 with a team of only six members. In the beginning, the team started designing some protocols for wireless communication with a range of up to 4 to 5 km line of sight. After the successful completion of the project, the team started approaching the clients who were in need. The company got a couple of good clients and started serving them. That is how the company started generating revenue. Since the team members were all engineers were experts in Embedded electronics, Java, Dot net, and Android, the company simultaneously started to develop websites and the few latest apps needed for the clients. The company later registered on 24th December 2013 and took its own space for the office, established well-equipped office space with a good R&D unit and needed infrastructure, hired many freshers for different domains of the company, and started working with new hopes. Presently the company has around 25 employees including all the departments like Embedded, Java, Dot net, android, Testing, PCB design, IOT, and so on. The company is working with many Industrial projects in different domains and working on its products.

1.1.1 Vision

To empower unskilled individuals with knowledge, skills, and technical competencies in the field of information technology and embedded engineering, enabling them to contribute to the growth of their company and nation.

1.1.2 Mission

- Provide cost-effective and reliable solutions to customers across various latest technologies.

- Offer scalable end-to-end application development and management solutions.

- Provide cost-effective highly scalable products for varied verticals.

- Focus on creating sustainable value growth through innovative solutions and unique partnerships.

- Create design and deliver business solutions with high value and innovation by leveraging technology expertise and innovative business models to address long-term business objectives.

- Keep our products and services updated with the latest innovations in the respective requirements and technology.

1.1.3 Objectives

- To develop software and Embedded solutions and services focussing on quality standards and customer values.

- Offer end-to-end embedded solutions that ensure the best customer satisfaction.

- To build a Skilled and Talented manpower pool for global industry requirements.

- To develop software and embedded products which are globally recognized

- To become a global leader in Ottering Scalable and cost-effective Software solutions and services across various domains like E-commerce, Banking, Finance, Healthcare, and much more.

- To generate employment for the skilled and highly talented youth of our Country INDIA

1.2 Company Services offered

1.2.1 Services

- **IT Solutions and Services**

Inventeron Technologies and Business Solutions LLP, or ITABS software-based IT Training and Software Development center with exclusive expertise in the area of IT Services and Solutions Inventeron Technologies and Business Solutions LLP, or ITABS Ltd. is also expertise in Web design and Consulting Services

- **Embedded Design and Development**

Inventeron Technologies and Business Solutions LLP, or ITABS has expertise in the Design and development of embedded products and offers solutions and services in the field of Electronics.

- **Academic Projects**

Inventeron Technologies and Business Solutions LLP, or ITABS helps students in their academics by imparting industrial experience into projects to strive for excellence of students Inventeron Technologies and Business Solutions LLP, or ITABS, encourages students to implement their ideas to projects keeping in mind "A small seed sown upfront will be nourished to become a large tree one day, thereby focusing the future entrepreneurs. They have a wide range of IEEE projects for BE, MTech, MCA, BCA, and DIPLOMA students for all branches in every domain.

- **Inplant Training**

Inventeron Technologies and Business Solutions LLP, or ITABS provide Implant training for students according to the interest of students keeping in mind the current technology and academic benefit one obtains after completing the training. Students will be nourished and will be trained throughout with practical experience. Students will be exposed to industrial standards which boost their career. Students will become acquainted with various

structural partitions such as labs, workshops, assembly units, stores, administrative units, and machinery units. They help students to understand their functions, applications, and maintenance Students will be trained from the initial stage that is from the collection of Project Requirements, Project Planning, Designing, implementation, testing, deployment, and maintenance thereby helping to understand the business model of the industry. The entire project life cycle will be demonstrated with hands-on experience. Students will also be trained in management skills and team-building activities. They assure that by the end of implant training students will Enhance their communication skills and acquire technical skills, employability skills, and start-up skills, and will be aware of risks in the industry, management skills, and many other skills that are helpful to professional engagement.

- **Software Courses**

Inventeron Technologies and Business Solutions LLP, or ITABS provides courses for students according to the interests of students keeping in mind the current technology and assisting them for their further Employment. The company provides various courses such as Python, Data Science, Artificial Intelligence, Machine Learning, Deep Learning, and Full Stack Web Development on the Software front. Embedded Systems and IoT, Robotics, Electric Vehicles, etc. on the Hardware front.

CHAPTER 2

Topics Learned During the Course

The main objective of the internship is to apply theoretical knowledge of "Machine Learning using Python" to solve real-time complex problems, to achieve the following basic concepts learned:

- Python
- Machine Learning

2.1 Python

Python is a multiparadigm, general-purpose, interpreted, high-level programming language. Python allows programmers to use different programming styles to create simple or complex programs, get quicker results, and write code almost as if speaking in a human language.

Fig 2.1: Python Logo

The topics learned in Python are as follows:

- Installation of Python.
- Use of variables to store, retrieve, and calculate information.
- Utilization of core programming operations such as functions and loops.
- Operation on strings, Python-supported libraries for Machine Learning.

2.1.1 Features of Python

- Extensive support libraries (NumPy for numerical calculations, Pandas for data analytics).

- Open source and community development.

- Dynamically typed language (No need to mention data type based on value assigned, it takes data type).

- Object-oriented language, Portable and Interactive across Operating systems.

2.2 Python Libraries for Data Analytics

Machine Learning, as the name suggests, is the science of programming a computer by which they can learn from different kinds of data. A more general definition given by Arthur Samuel is - "Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed." They are typically used to solve various types of life problems.

In the older days, people used to perform Machine Learning tasks by manually coding all the algorithms and mathematical and statistical formulas. This made the process time-consuming, tedious, and inefficient. But in the modern days, it has become very easy and efficient compared to the olden days with various Python libraries, frameworks, and modules. Python libraries that are used in Machine Learning are:

- NumPy
- Pandas
- Matplotlib

2.2.1 NumPy

NumPy is a basic package for scientific computing. It is the Python language implementation which includes powerful N-dimensional array structure, sophisticated functions, Tools that can be integrated into C/C++ and Fortran code, Linear algebra, Fourier transform, and Random number features. Besides its obvious scientific uses, NumPy can also be used as an efficient multidimensional container of generic data.

The main aspect of NumPy is the NumPy array, on which you can do various operations. The key is that a NumPy array isn't just a regular array you'd see in a language like Java or C++, instead, it is like a mathematical object as a vector or a matrix. That means you can do vector and matrix operations like addition, subtraction, and multiplication. The most important aspect of NumPy arrays is that they are optimized for speed.

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.

2.2.2 Basic Array Operations

NumPy, arrays allow a wide range of operations that can be performed on a particular array or a combination of Arrays. These operations include some basic Mathematical operations as well as Unary and Binary operations.

```
# Python program to demonstrate

# Basic operations on a single array

import NumPy as np

import NumPy as np

a = np.array ([[1, 2], [3, 4]])                  # Defining Array 1

b = np.array ([[4, 3], [2, 1]])                  # Defining Array 2

print ("Adding 1 to every element:", a + 1)      #Adding 1 to every element

print ("\n Subtracting 2 from each element:", b - 1)   # Subtracting 2 from each
                                                           element

# sum of the array element
Print("\n Sum of all array elements:", a.sum())   # Performing unary operations
Print("\n array sum :\n",a+b)                     # Performing binary operations
```

2.2.3 Pandas

Pandas Data Frame is a two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). A Data frame is a two-dimensional data structure, i.e., data is aligned in a tabular fashion in rows and columns.

Pandas Data Frame consists of three principal components, the data, rows, and columns.

The basic operations which can be performed on Pandas Data Frame are:

- Creating a Data Frame
- Dealing with Rows and Columns
- Indexing and Selecting Data
- Working with Missing Data

2.2.4 Matplotlib

Matplotlib is a very popular Python library for data visualization. Like Pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. A module named plot makes it easy for programmers to plot as it provides features to control line styles, font properties, formatting axes, etc. It provides various kinds of graphs and plots for data visualization, viz., histograms, error charts, bar charts, etc,

```
#Python program using Matplotlib for forming a linear plot

import NumPy as np

import matplotlib.pyplot as pltx = [0, 10, 100]        # Prepare the data

plt.plot (x, x, label = 'linear')

plt.legend()            # Add a legend

plt.show()              # Show the plot
```

2.3 Software tool used

Anaconda is a free and open-source distribution of the Python programming language for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. Package versions are managed by the package management system conda Anaconda distribution comes with more than 1,500 packages as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command line interface (CLI). Anaconda Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages, and update them. There are many applications available by default in Navigator, among them is the Spyder.

Spyder is an open-source cross-platform integrated development environment (IDE) for scientific programming in the Python language. Spyder integrates with several prominent packages. Some of the features of Spyder are:

- An editor with syntax highlighting, introspection, and code completion.

- Support for multiple I Python consoles.

- The ability to explore and edit variables from a GUL.

- A Help pane able to retrieve and render rich text documentation on functions, classes, and methods automatically or on-demand.

- A debugger linked to IP dB, for step-by-step execution.\

- A run-time Profiler, to benchmark code.

- Project support, allowing work on multiple development efforts simultaneously.

- A built-in file explorer, for interacting with the filesystem and managing projects.

- A "Find in Files" feature, allowing full regular expression search over a specified scope.

- An online help browser, allowing users to search and view Python and package documentation inside the IDE.

- A history log, recording every user command entered in each console.

- An internal console, allowing for introspection and control over Spyder's own operation.

2.4 Open CV

OpenCV is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python. C, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human When it is integrated with various libraries, such as NumPy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal te., whatever operations one can do in NumPy can be combined with OpenCV.

2.4.1 Definition

OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

CHAPTER 3

Criminal Face Detection Using Python

3.1 Introduction

Images uploaded to platforms like Picasa, Photobucket, and Facebook now have the added feature of automatic tagging. This feature allows people in the picture to be identified and gives others an idea about who is in the image. We have studied and implemented a simple yet effective face detection algorithm that takes human skin color into account. Our goal was to develop a system that can be used by police or investigation departments to recognize criminals from their faces. Our face recognition method is fast, robust, reasonably simple, and accurate with relatively simple and easy-to-understand algorithms and techniques.

3.2 Motivation

The motivation behind this project is to provide ease to police personnel to find criminals from anywhere without wasting time and cost. This application also helps to maintain records of criminals, one can also find all the details of the criminal in our application. Therefore this application helps police in many ways.

3.3 Problem Definition

To develop an application that will serve as a way to register and track criminals remotely with the help of criminal data. This application provides two ways to identify criminals. One is by manually providing the photos of criminals and the other way is by using live CCTV cameras.

3.4 Objective of the Project

- The main objective of Real-time criminal identification based on face recognition The application is to help police personnel identify criminals.
- The objective of this application is to provide information about a particular criminal which we are finding.
- Police personnel can use this application anytime, anywhere to find a criminal

- Any police personnel can access this application using the internet from anywhere and anytime.

- We can also find criminals from live CCTV surveillance cameras.

- This application is fast, robust, reasonably simple, and accurate with a relatively simple and easy-to-understand GUI.

3.5 Existing System

As the crime rate and criminals are increasing day by day managing, finding, and tracking these criminals is a major issue for police personnel. Some applications will help police departments to store the records and data about a criminal but these applications won't help in finding those criminals. Criminal details were mainly managed using records books or stored as software records in the database. Previously when a criminal is found guilty the picture of the criminal is being taken and stored in records but these pictures serve no purpose. The existing methods will only help in managing criminal records and those methods will not find criminals from any location.
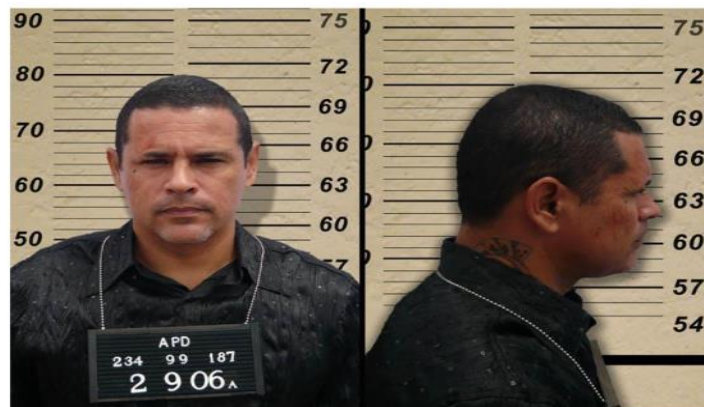
Fig 3.5 : Mugshots Of Criminals Can Be Used To Track And Monitor Criminals

3.6 Disadvantages of Existing System

- It is not possible to detect criminals from any location

- Existing methods only provide data storage and security for data but not live tracking. No application would find criminals from CCTV footage.

- Criminal details were stored manually in a record and it requires a lot of work.

- Information can be lost or manipulated in records easily.

- Previous applications were not 100 percent accurate and this leads to inaccurate information about criminals.

3.7 Proposed System

This project is aimed at developing an application called Real-Time criminal identification system based on face recognition. We can detect and recognize faces of the criminals in an image and in a video stream obtained from a camera in real-time. We have used Haar feature-based cascade classifiers in the OpenCV approach for face detection. It is a machine learning-based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. Also, we have used Local Binary Patterns Histograms (LBPH) for face recognition. This application helps police personnel in many ways.

In our application, we can register a criminal, once it is successfully done we can track and find criminals using CCTV footage or by manually giving images as input. Data is each criminal is managed through a dataset. When a criminal is detected at any time on camera (CCTV) criminal details will be displayed. In this way, a lot of time is saved and this is a highly secure process and one can detect criminals easily. Our application is 95 percent accurate and it is fast, robust, reliable, and easy to use.

3.8 Advantages of the Proposed System

- It helps police personnel to track and find criminals easily.
- It requires very less men power and the cost of the operation is much less.
- In this application the information cannot be manipulated or lost and data will be safe.
- This system can be used by police or investigation departments to recognize criminals from their faces.
- This application can recognize faces in different lighting conditions with high accuracy
- The application is fast, robust, reasonably simple, and accurate with relatively simple and easy-to-understand algorithms and techniques.

3.9 Architecture
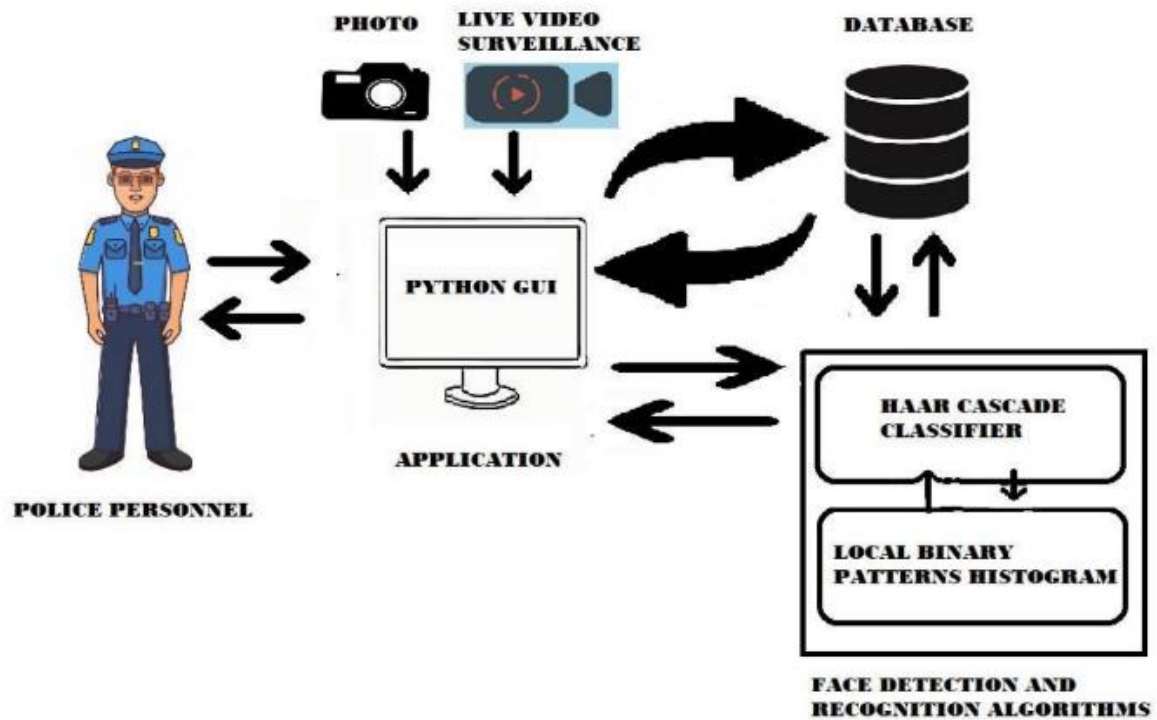
Fig 3.9: Architecture of Real-Time Criminal Identification Using Face Recognition

3.10 Program

3.10.1 Program to take pictures of a person using a Laptop dashcam.

```python
# import the necessary packages
from imutils.video import VideoStream
import argparse
import imutils
# import time
import cv2
import os

# import urllib.request
# import numpy as np

# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
```

```python
ap.add_argument("-c", "--cascade", required=True,
        help="path to where the face cascade resides")
ap.add_argument("-o", "--output", required=True,
        help="path to output directory")
args = vars(ap.parse_args())

# load OpenCV Haar cascade for face detection from disk
detector = cv2.CascadeClassifier(args["cascade"])
# initialize the video stream, allow the camera sensor to warm up,
# and initialize the total number of example faces written to disk
# thus far
print("[INFO] starting video stream...")
# url='http://192.168.0.110/capture'
vs = VideoStream(src=0).start()
# vs = VideoStream(usePiCamera=True).start()
# time.sleep(2.0)
total = 0

# loop over the frames from the video stream
while True:
    # grab the frame from the threaded video stream, clone it, (just
    # in case we want to write it to disk), and then resize the frame
    # So we can apply face detection faster
    # imgResp = urllib.request.urlopen(url)
    # imgNp = np.array(bytearray(imgResp.read()),dtype=np.uint8)
    # frame = cv2.imdecode(imgNp,-1)
    frame = vs.read()
    orig = frame.copy()
    frame = imutils.resize(frame, width=400)
    # detect faces in the grayscale frame
    rects = detector.detectMultiScale(
        cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY), scaleFactor=1.1,
        minNeighbors=5, minSize=(30, 30))
    # loop over the face detections and draw them on the frame
```

```python
for (x, y, w, h) in rects:
    cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)


# show the output frame
cv2.imshow("Frame", frame)
key = cv2.waitKey(1) & 0xFF


# if the `k` key was pressed, write the *original* frame to disk
# So we can later process it and use it for face recognition
if key == ord("k"):
    p = os.path.sep.join([args["output"], "{}.png".format(
        str(total).zfill(5))])
    cv2.imwrite(p, orig)
    total += 1
# if the `q` key was pressed, break from the loop
elif key == ord("q"):
    break


# print the total faces saved and do a bit of cleanup
print("[INFO] {} face images stored".format(total))
print("[INFO] cleaning up...")
cv2.destroyAllWindows()
vs.stop()
```

- Install the required libraries.
- Press key K to take a picture of the person using the system dashcam.
- Train the system by taking multiple pictures.
- To store the photos, we have to create the folder in PyCharm, calling it a database.
- And in the database, we create another folder, calling it criminal detected, in which the photos are stored.
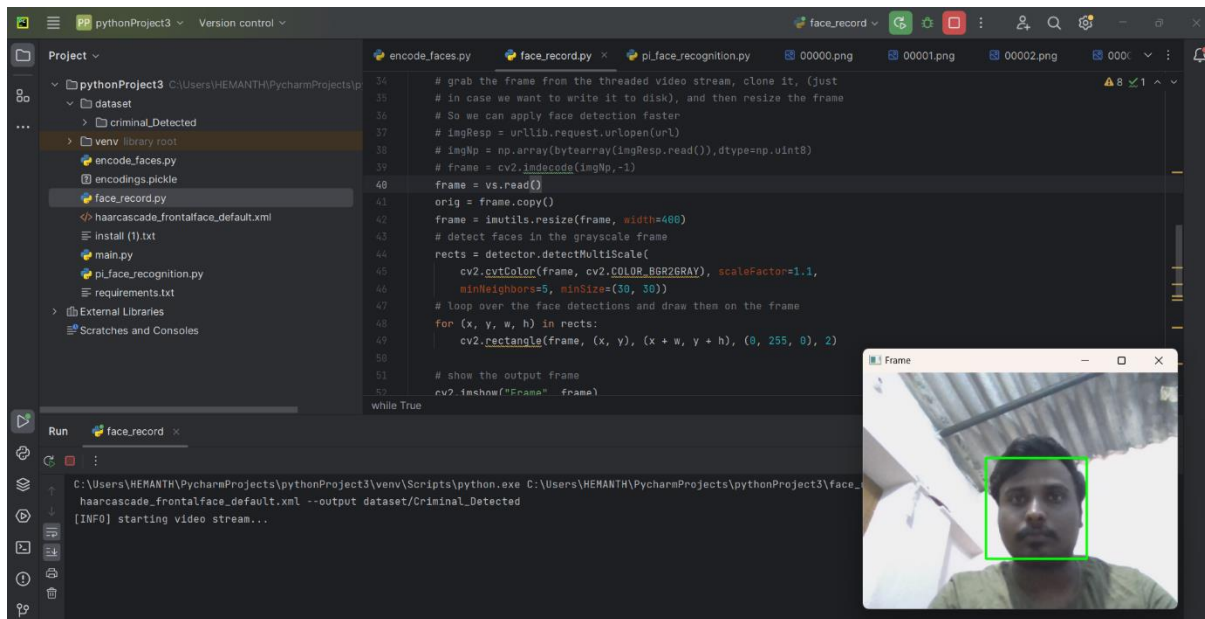
Fig 3.10.1: Taking a picture of a person

3.10.2 Program to process and store the photos in the created location.

```python
# import the necessary packages
from imutils import paths
import face_recognition
import argparse
import pickle
import cv2
import os
# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "—dataset", required=True,
    help="path to input directory of faces + images")
ap.add_argument("-e", "—encodings", required=True,
    help="path to serialized db of facial encodings")
ap.add_argument("-d", "—detection-method", type=str, default="cnn",
    help="face detection model to use: either `hog` or `cnn`")
args = vars(ap.parse_args())


# grab the paths to the input images in our dataset
print("[INFO] quantifying faces…")
```

```python
imagePaths = list(paths.list_images(args["dataset"]))
# initialize the list of known encodings and known names
knownEncodings = []
knownNames = []


# loop over the image paths
for (i, imagePath) in enumerate(imagePaths):
    # extract the person name from the image path
    print("[INFO] processing image {}/{}".format(i + 1,
        len(imagePaths)))
    name = imagePath.split(os.path.sep)[-2]
    # load the input image and convert it from BGR (OpenCV ordering)
    # to dlib ordering (RGB)
    image = cv2.imread(imagePath)
    rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    # detect the (x, y)-coordinates of the bounding boxes
    # corresponding to each face in the input image
    boxes = face_recognition.face_locations(rgb,
        model=args["detection_method"])
    # compute the facial embedding for the face
    encodings = face_recognition.face_encodings(rgb, boxes)
    # loop over the encodings
    for encoding in encodings:
        # add each encoding + name to our set of known names and
        # encodings
        knownEncodings.append(encoding)
        knownNames.append(name)


# dump the facial encodings + names to disk
print("[INFO] serializing encodings…")
data = {"encodings": knownEncodings, "names": knownNames}
f = open(args["encodings"], "wb")
f.write(pickle.dumps(data))
f.close()
```

3.10.3 Program to recognize the photo.

```python
# import the necessary packages
from imutils import paths
import face_recognition
import argparse
import pickle
import cv2
import os
# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument("-i", "—dataset", required=True,
    help="path to input directory of faces + images")
ap.add_argument("-e", "—encodings", required=True,
    help="path to serialized db of facial encodings")
ap.add_argument("-d", "—detection-method", type=str, default="cnn",
    help="face detection model to use: either `hog` or `cnn`")
args = vars(ap.parse_args())


# grab the paths to the input images in our dataset
print("[INFO] quantifying faces…")
imagePaths = list(paths.list_images(args["dataset"]))
# initialize the list of known encodings and known names
knownEncodings = []
knownNames = []


# loop over the image paths
for (i, imagePath) in enumerate(imagePaths):
    # extract the person name from the image path
    print("[INFO] processing image {}/{}".format(i + 1,
        len(imagePaths)))
    name = imagePath.split(os.path.sep)[-2]
    # load the input image and convert it from BGR (OpenCV ordering)
    # to dlib ordering (RGB)
```

```python
image = cv2.imread(imagePath)
rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
# detect the (x, y)-coordinates of the bounding boxes
# corresponding to each face in the input image
boxes = face_recognition.face_locations(rgb,
    model=args["detection_method"])
# compute the facial embedding for the face
encodings = face_recognition.face_encodings(rgb, boxes)
# loop over the encodings
for encoding in encodings:
    # add each encoding + name to our set of known names and
    # encodings
    knownEncodings.append(encoding)
    knownNames.append(name)


# dump the facial encodings + names to disk
print("[INFO] serializing encodings…")
data = {"encodings": knownEncodings, "names": knownNames}
f = open(args["encodings"], "wb")
f.write(pickle.dumps(data))
f.close()
```

- Install the required libraries.

- When we run the program, if the person whose photo is stored in the database is in front of the dashcam, then the system will recognize that person and show criminal is detected.

- If the person whose photo is not stored in the database is in front of the dashcam, then the system will not recognize that person and show unknown.
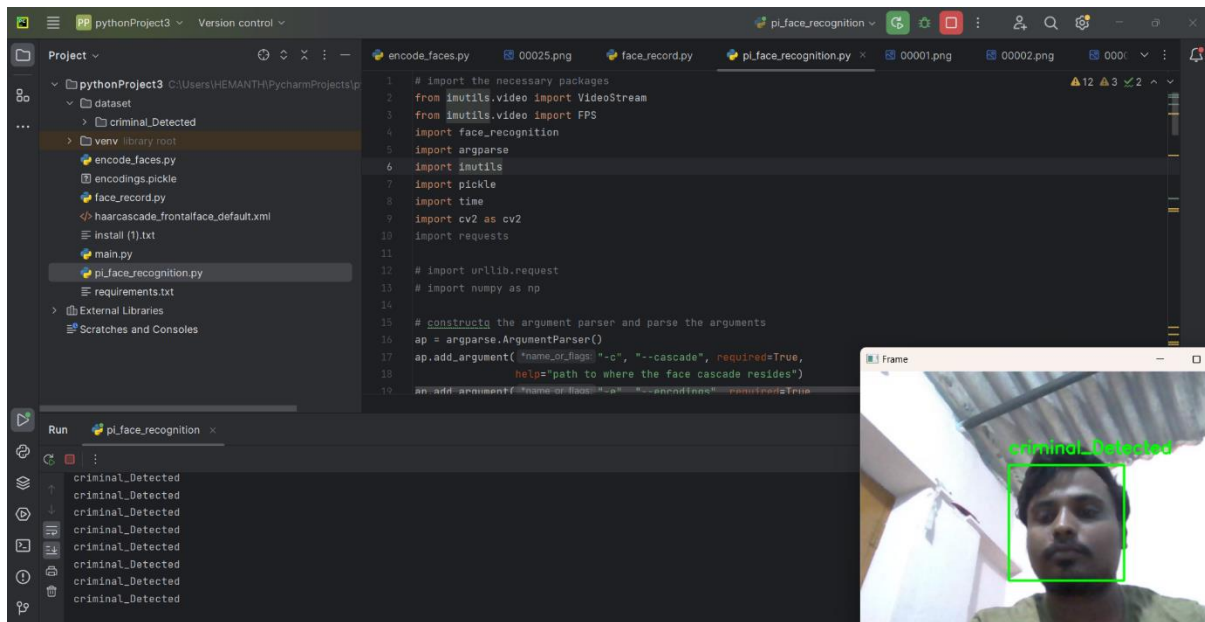
Fig 3.10.3: screenshot of system recognising the person

3.11 Scope of the Project

Scope of the Research Scope of the system is the complete identification of the face. Within the allocated time completing the system with the specified user requirements. One system is for the administrator and the other one is for the users. It can be used in many fields there are Bank, Hotel and Police Station. A throughout survey has revealed that various methods and combinations of these methods can be applied in the development of a new face recognition system. Among the many possible approaches, we have decided to use a combination of knowledge-based methods for the face detection part and a neural network approach for the face recognition part.

3.12 Contribution in the Project

Facial recognition can aid forensic investigations by automatically recognizing individuals in security footage or other videos. Face recognition software can also be used to identify dead or unconscious individuals at crime scenes. Facial recognition is the process of identifying or verifying the identity of a person using their face. It captures, analyzes, and compares patterns based on the person's facial details. The face detection process is an essential step as it detects and locates human faces in images and videos.

CHAPTER 4

Reflection

Throughout our internship, we delved into the fundamental concepts of Python programming and its libraries, which culminated in a project centered on criminal detection using Python. This experience provided us with invaluable insights into the application of machine learning techniques in law enforcement. Our project gave us a deeper understanding of how the police use machine learning algorithms to identify and apprehend criminals. It not only enhanced our programming skills but also highlighted the significance of data-driven approaches in tackling real-world challenges. Overall, this internship equipped us with practical knowledge and skills that are essential for success in the field of data science and law enforcement technology.

CHAPTER 5

Conclusion

After considering all the facts present in the introduction section, we did research on different applications and came up with a solution. A real-time criminal identification system will help police to control the crime rate. This application helps them in many different ways. With the advancement in security technology and the installation of cameras throughout public areas, it will become easier for police personnel to monitor, track, and find criminals from the police control room using this application.

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. Moreover, if a criminal is found in a particular zone then alert messages should be sent to nearby police stations. The application that is developed is simple and user-friendly. By using advanced CSS styles and different front-end technologies, the interface of the application can be developed more according to user requirements.

CHAPTER 6

References

1. Alireza Chevelwalla, Ajay Gurav, Sachin Desai, Prof. Sumitra Sadhukhan "Criminal Face Recognition System" International Journal of Engineering Research & Technology (IJERT) Vol. 4 Issue 03, March-2015.

2. Bornet, O. (2005, May 19). Learning-Based Computer Vision with Intel's Open Source Computer Vision Library. Retrieved April 2007, 2007, from Intel.com Website: http://www.intel.com/technology/itj/2005/volume09issue02/art03_learning_vis ion/p04_face_dete ction.htm.

3. Viola, P. and Jones, M. Rapid object detection using a boosted cascade of simple features. IEEE Conference on Computer Vision and Pattern Recognition, 2001.