

”Detection of Human Entrance From Video Surveillance Streaming Image”

A project report

submitted by

Abdullah Al Mahdi Jahin (ID: 19202103439)

Kaiser Mahmud (ID: 19202103427)

Md. Mostafizur Rahman (ID: 19202103247)

Md. Shaidur Rahman (ID: 19202103437)

MD. Tousif Zaman Fahim (ID: 19202103416)

Submitted in partial fulfillment of the requirements of the degree of
Bachelor of Science in (CSE)



Department of Computer Science and Engineering
Bangladesh University Of Business & Technology(BUBT)

ABSTRACT

This project introduces a "Detection of Human Entrance From Video Surveillance Streaming Image" employing for accurate human presence detection at entrances. The system, utilizing high-resolution cameras and real-time processing, enhances security by swiftly identifying and tracking individuals, providing a valuable addition to existing surveillance infrastructure. The project also incorporates hardware components, including high-resolution cameras and a processing unit, to capture and process video feeds in real-time. An efficient communication mechanism facilitates the seamless integration of the system with existing security infrastructure. A robust model capable of accurately identifying and localizing human presence in video frames. Focus on entrance zones to enhance security measures, with alerts triggered upon unauthorized access or suspicious behavior. Its implementation offers improved security, reduced response times, and enhanced situational awareness. The project findings and insights contribute to the ongoing development of intelligent systems for human detection and monitoring applications.

DECLARATION

We do hereby declare that the research works presented in this thesis entitled, "Detection of Human Entrance From Video Surveillance Streaming Image" are the results of our own works. We further declare that the thesis has been compiled and written by us and no part of this thesis has been submitted elsewhere for the requirements of any degree, award or diploma or any other purposes except for publications. The materials that are obtained from other sources are duly acknowledged in this project.

Member

Abdullah Al Mahdi Jahin

Signature

Abdullah Al Mahdi (Jahin)

kaiser Mahmud

Signature

kaiser Mahmud

Signature

'Shaider Rahman Shuvo

Signature

MD. Shaider Rahman

Mostafizur Rahman Fuad

Md. Mostafizur Rahman

Md Tousif Zaman Fahim

Md. Tousif zaman Fahim

APPROVAL

We do hereby declare that the research works presented in this project entitled, Detection of human entrance from video surveillance streaming image, are the outcome of the original works carried out by Abdullah Al Mahdi, Kaiser Mahmud, Md. Mostafizur Rahman, Md. Shaidur Rahman / MD. Tousif Zaman Fahim under my supervision. We further declare that no part of this thesis has been submitted elsewhere for the requirements of any degree, award or diploma or any other purposes except for publications. I further certify that the dissertation meets the requirements and standard for the degree in Computer Science and Engineering.

Supervisor

Mijanur Rahman

Assistant Professor

Department of Computer Science and Engineering

Bangladesh University of Business and Technology (BUBT)

Mirpur-2, Dhaka-1216, Bangladesh

Chairman

Md. Saifur Rahman

Assistant Professor

Department of Computer Science and Engineering

Bangladesh University of Business and Technology (BUBT)

Mirpur-2, Dhaka-1216, Bangladesh

ACKNOWLEDGMENT

We are deeply thankful to Bangladesh University of Business and Technology (BUBT) for providing us such a wonderful environment to peruse our research. We would like to express our sincere gratitude to Mijanur Rahman, Assistant Professor and project Supervisor, Department of CSE, BUBT. We have completed our research with his help. We found the research area, topic, and problem with his suggestions. He guided us with our study, and supplied us many research papers and academic resources in this area. He is patient and responsible. When we had questions and needed his help, he would always find time to meet and discuss with us no matter how busy he was. We would also like to acknowledge our team members for supporting each other and be grateful to our university for providing this opportunity for us. Lastly special thanks to Google scholar for collecting so many papers.

Contents

1	Introduction	1
1.1	Introduction	1
1.2	Background	2
1.3	Purpose of Statement	2
1.4	Problem Statement	3
1.5	Research Objectives	4
1.6	Research Contribution	4
1.7	Motivation	5
1.8	Flow of the Research	6
1.9	Significance of the Research	7
1.10	Thesis Organization	7
1.11	Summary	7
2	Literature Review	8
2.1	Introduction	8
2.2	Literature Review	8
2.3	Summary	10
3	Proposed Model	11
3.1	Introduction	11
3.2	Data Set	13
3.2.1	Collection Process	13
3.2.2	Dataset Statistics	13
3.2.3	Ethical Considerations	14
3.3	Data Analysis and Pre-processing	14
3.4	Model Development	14
3.4.1	Traditional OpenCV	14
3.4.2	Model Diagram	17
3.4.3	Flow Chart	18

4	Experimental Results	20
4.1	Code Implementation	20
4.1.1	views.py	22
4.1.2	req.txt	27
4.2	Detection Performing	28
4.3	Performance Analysis	33
4.3.1	Introduction	33
4.3.2	Analysis	33
4.3.3	Accuracy Graph	34
5	Standards, Constraints, Milestones	35
5.1	Introduction	35
5.2	Standards	35
5.2.1	Coding Standards	35
5.2.2	ASGI Standards	35
5.2.3	Django Framework	35
5.3	Constraints	35
5.3.1	Computational Resources	35
5.3.2	Real-time Processing	36
5.4	Milestones	36
5.4.1	Model Training Milestone	36
5.4.2	Integration with Django	36
5.4.3	Real-time Face Recognition	36
5.4.4	Privacy and Security Measures	36
6	Conclusion and Future Work	37
6.1	Conclusion	37
6.2	Future Work	38

List of Figures

1.1	Flow of the work	6
3.1	Outline of Detection of Human Entrance	12
3.2	Flow of our research	12
3.3	Installing OpenCV	16
3.4	Architecture	17
3.5	Flow Chart	19
4.1	File of Project	21
4.2	views.py	22
4.3	views.py	23
4.4	views.py	24
4.5	views.py	25
4.6	views.py	26
4.7	req.txt	27
4.8	Training Perform	29
4.9	Unknown Result	30
4.10	Result List	31
4.11	Result List	32
4.12	Accuracy Graph	34

Chapter 1

Introduction

1.1 Introduction

Security and surveillance play pivotal roles in safeguarding environments, prompting the need for innovative solutions to augment existing systems. This project introduces a comprehensive Detection of Detection of human entrance from video surveillance streaming image designed to enhance security measures at entry points. Leveraging advanced computer vision techniques, particularly deep learning, the system aims to accurately detect and track human presence in real-time. By focusing on entrance areas, the project addresses the vital task of fortifying access points, contributing to improved situational awareness and proactive security measures. This report outlines the development, implementation, and outcomes of the Detection of human entrance from video surveillance streaming image, showcasing its potential to elevate security standards across diverse settings. Eliminate false alarms and missed detections for a safer environment, Automate monitoring, freeing up resources and streamlining operations, Gain valuable insights on entrance activity for better security protocols and resource allocation, Works with existing infrastructure for a smooth and cost-effective implementation. This project will showcase the transformative power of human detection technology in securing building entrances, paving the way for smarter and safer environments. Ref.[1]

1.2 Background

Face recognition is crucial in daily life in order to identify family, friends or someone we are familiar with. We might not perceive that several steps have actually taken in order to identify human faces. Human intelligence allows us to receive information and interpret the information in the recognition process. We receive information through the image projected into our eyes, by specifically retina in the form of light. Light is a form of electromagnetic waves which are radiated from a source onto an object and projected to human vision. Robinson-Riegler, G., & Robinson-Riegler, B. (2008) mentioned that after visual processing done by the human visual system, we actually classify shape, size, contour and the texture of the object in order to analyze the information. The analyzed information will be compared to other representations of objects or face that exist in our memory to recognize. In fact, it is a hard challenge to build an automated system to have the same capability as a human to recognize faces. However, we need large memory to recognize different faces, for example, in the Universities, there are a lot of students with different race and gender, it is impossible to remember every face of the individual without making mistakes. In order to overcome human limitations, computers with almost limitless memory, high processing speed and power are used in face recognition systems. The human face is a unique representation of individual identity. Thus, face recognition is defined as a biometric method in which identification of an individual is performed by comparing real-time capture image with stored images in the database of that person (Margaret Rouse, 2012). 4 Nowadays, face recognition system is prevalent due to its simplicity and awesome performance. For instance, airport protection systems and FBI use face recognition for criminal investigations by tracking suspects, missing children and drug activities (Robert Silk, 2017). Apart from that, Facebook which is a popular social networking website implement face recognition to allow the users to tag their friends in the photo for entertainment purposes (Sidney Fussell, 2018). Furthermore, Intel Company allows the users to use face recognition to get access to their online account (Reichert, C., 2017). Apple allows the users to unlock their mobile phone, iPhone X by using face recognition (deAgonia, M., 2017). The work on face recognition began in 1960. Woody Bledsoe, Helen Chan Wolf and Charles Bisson had introduced a system which required the administrator to locate eyes, ears, nose and mouth from images. The distance and ratios between the located features and the common reference points are then calculated and compared. The studies are further enhanced by Goldstein, Harmon, and Lesk in 1970 by using other features such as hair colour and lip thickness to automate the recognition.

1.3 Purpose of Statement

The purpose of this project is to design, develop, and implement a Detection of human entrance from video surveillance streaming image that utilizes state-of-the-art computer vision and technology. The primary objective is to enhance security measures at entrances by accurately detecting and tracking human presence in real-time. By focusing on entry points, the system aims to contribute to the overall improvement of security infrastructure, providing a reliable and proactive solution to monitor and respond to potential security threats. This project seeks to demonstrate the effectiveness of the proposed system in fortifying access points, reducing response times, and ultimately creating a more secure environment in various applications such as public

spaces, institutions, and private facilities. It serves as a knowledge resource for stakeholders, researchers, and practitioners interested in leveraging advanced technologies to address security challenges effectively.

1.4 Problem Statement

The existing security and surveillance systems often face challenges in effectively monitoring and responding to human presence at entrance points, leading to potential security vulnerabilities. Traditional methods may lack accuracy and real-time capabilities, resulting in delayed or inadequate responses to security threats. The need for a robust solution becomes apparent as unauthorized access and security breaches continue to pose risks in various environments. This project aims to address these shortcomings by designing and implementing a Detection of Human Entrance System. The objective is to create a system that leverages advanced computer vision techniques, to accurately detect and track human presence in real-time at entrance areas. By doing so, the project seeks to overcome the limitations of existing systems, contributing to improved security measures and rapid response to potential security incidents. Ref.[2]

1.5 Research Objectives

Following analysis objectives achieved from this analysis area unit given below :

- Implement real-time video processing mechanisms to ensure the system can promptly analyze and respond to dynamic situations, providing timely updates on detected human activities.
- Focus on entrance zones and develop features that allow the system to monitor and assess human activities specifically at entry points, ensuring a heightened level of security for these critical areas.
- Implement advanced image processing techniques to enhance the quality of captured images and Face recognition and Identification.
- Develop a user-friendly interface that enables efficient detection and entrance for effective decision-making. Ref.[3]

1.6 Research Contribution

- By focusing on entrance zones, the project introduces specialized features that enhance security measures at critical access points. This targeted approach contributes to improved situational awareness and proactive security measures.
- The compilation of comprehensive documentation, including guidelines and best practices for deployment, maintenance, and future enhancements, contributes to knowledge transferability. This resource assists in facilitating the adoption and further development of the Detection of Human Entrance System.

1.7 Motivation

The project is motivated by the critical need to enhance security measures in various environments, addressing limitations in traditional surveillance systems. By focusing on human detection at entrance points, the project aims to leverage advanced computer vision technologies to provide real-time and accurate information, contributing to timely responses to security incidents. The goal is to create a system that integrates seamlessly with existing infrastructure, minimizes false alarms, and offers a user-friendly solution, ultimately advancing the effectiveness of security measures and situational awareness.

1.8 Flow of the Research

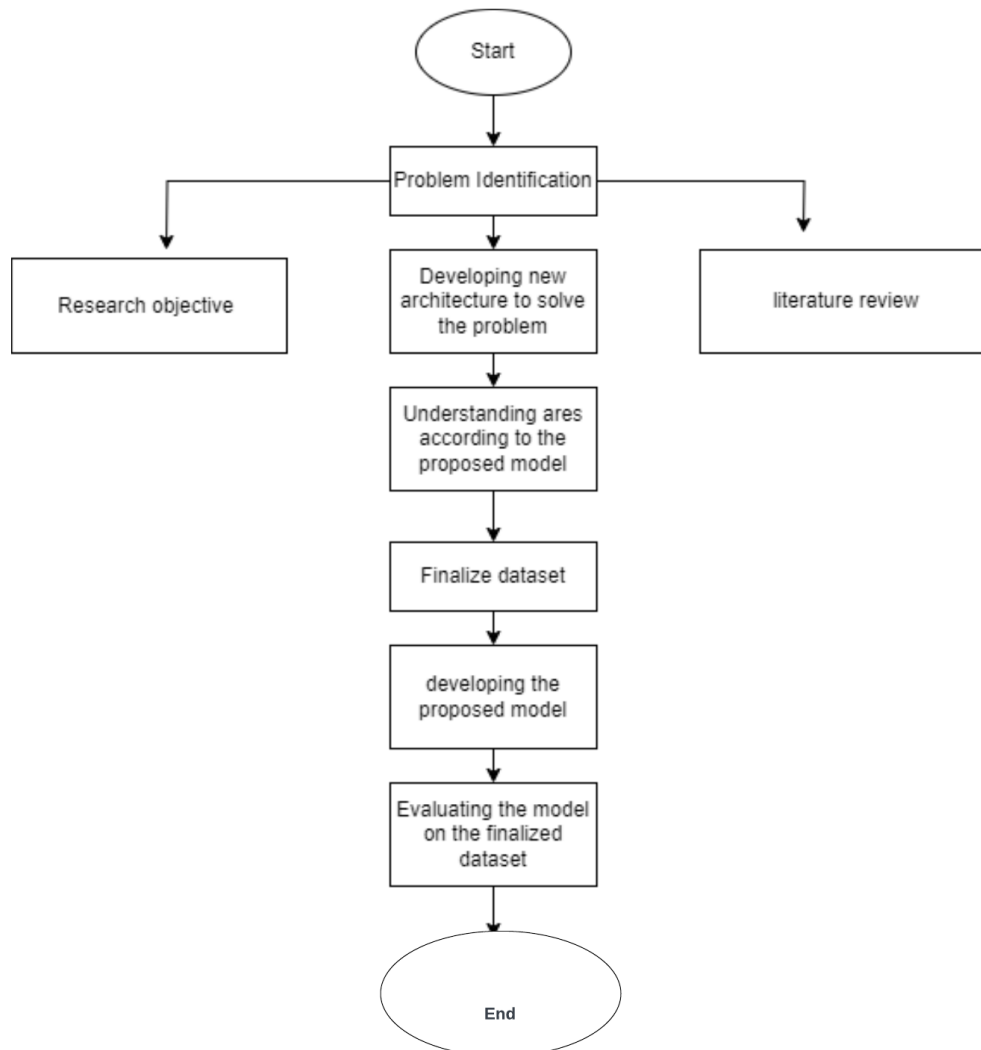


Figure 1.1: Flow of the work

1.9 Significance of the Research

The research holds significant value as it introduces "Detection of human entrance from video surveillance streaming image" that surpasses conventional methods. The system's real-time human detection, known and unknown facial recognition, and other features enhance security and safety measures. It offers actionable insights, proactive threat detection, and improved decision-making, contributing to safer environments in various domains.

1.10 Thesis Organization

This thesis work is organized as follows. Chapter 2 highlights the literature review on the field of the Real-time Face recognition and Identification with robust face detection algorithms using Dlib or deep learning models. Chapter 3 highlights the proposed model, data analysis and preprocessing, model development, and proposed Dlib or deep learning models. Chapter 4 contains the experimental results and performance evaluation metrics. Finally, Chapter 5 enclose our conclusion and future works in this field.

1.11 Summary

This chapter presented an outlook of our overall work. It introduces the research work, the problems leading to the choice of working on this as well as a brief overview of the actual proposed model.

Chapter 2

Literature Review

2.1 Introduction

The field of video surveillance and image processing has witnessed significant advancements in recent years, driven by the increasing demand for enhanced security and safety measures. With the ever-growing volume of visual data generated from surveillance cameras and other sources, there is a pressing need to develop advanced technologies capable of real-time analysis, intelligent insights, and proactive threat detection. This literature review provides an overview of the existing research and developments in the domain of advanced video surveillance and image processing. It aims to establish the context for the project and identify the gaps and opportunities for innovation. The review encompasses a wide range of topics, including computer vision algorithms, deep learning techniques, object detection, facial recognition.

2.2 Literature Review

Video surveillance has received a great attention as extremely active application-oriented research areas in computer vision, artificial intelligence, and image processing. The early use of monitoring system was the tube camera that deployed to broadcast and monitor the industrial processing in the 1930s and 1940s [1, 2]. The traditional video surveillance systems normally called Close-Circuit Television (CCTV) was defective and costly since they were deployed by security teams to observe events in the scenes via visual display. To this end, automated video surveillance systems utilize integration of real-time and more effective computer vision and intelligence techniques. Therefore, automated video surveillance systems succeed to assist security staffs by generating real-time alerts and forensics investigation due to support advanced video analysis techniques.

In recent decades, expansion in video surveillance systems lead to inspire evolution in various prominent domains of technology and science such as homeland security [3, 4], crime prevention through indoor and outdoor monitoring [2], elder care [5], accident detection [6, 7], traffic monitoring, controlling and traffic flow analysis [8-10], airborne traffic management [11], maritime traffic control [12, 13], counting moving object (pedestrians, vehicles) [14], human behavior understanding [15, 16], Motion detection, activity analysis, iden-

tification, tracking, and classification of (vehicles, peoples, and any object of interest). There is also a growing demand for applications to support monitoring indoor and outdoor environments like parking lots, shopping mall, airport, train station and so on due to the development, availability, and low price of processors and sensors [17-20]. Thus, research in video surveillance systems are multidisciplinary field associated to image analyzing and processing, pattern recognition, signal processing, embedded computing, and communication.

Nowadays, many researchers have attempted to provide effective and appropriate video surveillance services to users by proposing and implementing network infrastructure, user interface, and middleware components. However, various published journal and conference articles have shown researchers interest in video surveillance systems; hence, it can be a time for an overview analysis. Due to diversity of articles in various journals it seems quite difficult to have direct comparison; therefore, the primary objectives of this review are:

- To summarize and classify research related to video surveillance systems.
- To prepare and offer a conceptual framework to integrate and classify articles correspondingly.
- To provide some suggestions for video surveillance systems researches according to this review.

Section I presents a methodology to perform the systematic literature. The overall features of the articles are shown in Section III. The proposed classification framework is shown in Section IV. Each characteristic of the proposed classification is described in Section IV, whereas discussions and suggestions are reviewed in Section V. Finally, the paper is concluded with the concise conclusion in Section VI.

The area of this review was limited to academic research on video surveillance systems and its application. Furthermore, the scope of the study covered only the literature published in the time interval 2000- 2011. The authors classify the criteria for selecting and searching articles, and devise a procedure to organize chosen articles. The research was performed using the keyword "video surveillance" through six online databases. Although, video surveillance is a broad area of research from computer science to computer graphic, computer vision, etc., nevertheless, the authors limit the area for exploring through the online databases to the subjects which are summarized in Table 1, The following inclusion and exclusion criteria have been used for extracting of the articles. The authors included journal articles due to these reasons: first, the highest levels of research publish in journal, and second, both academics and professionals alike mostly use journal articles to publicize their novel finding and obtain new information. It depicts the selected journal and distribution of the selected articles accordingly. Ref.[4]

The abstract architecture of video surveillance systems according to the explored articles, and it provides appropriate classification criteria for organizing the articles. The abstract layer architecture is prepared by fully reviewing, Visual Inspection and Evaluation of Widearea Scenes (VIEW S), Video Surveillance and Monitoring (VSAM), IBM Smart Surveillance System (IBM S3), Hierarchical and Modular Surveillance System (HMSS), Defense Advanced Research Projects Agency (DARPA), Proactive Integrated Systems for Security Management by Technological and Communication Assistance (PRISMATICA), Cooperative Surveillance MultiAgent System (CS-MAS), Gaussian Mixture Model Adaptive Mean Shift (GMM -SAMT), Reading People Tracker (RPT), simultaneous localization and mapping (SLAM), Sensor Placement Algorithms, Facial Expression Recognition, Maximal Lifetime Sensor Target Surveillance (MLSTS), Surveillance

Security (SurvSec). Therefore, the abstract layer architecture of video surveillance systems is extracted in accordance with reviewing the aforementioned articles. Ref.[5]

The abstract architecture of video surveillance systems are used to develop the classification framework with the purpose of classifying extracted articles in this review. The proposed classification consists of these six layers: user interaction layer, application layer, communication layer, processing layer, network infrastructure layer, and concept and foundation layer. The abstract architecture and classification framework vary on concept and foundation layer; each classification layer is composed of its own categories in order to classify the articles appropriately. Ref [6]

This review was conducted based on the proposed framework with the intention of presenting a comprehensive review on video surveillance systems. The review initiated the video surveillance concept and foundation, network infrastructure, processing, communication, application, user interaction and listed all research related to each layer of the proposed framework on video surveillance consequently. The review examined all the research of the concept and applications, and reviewed them according to current and ongoing problem of video surveillance systems. Ref[7]

The review conducted by exploring the video surveillance systems literature from 2000 to 2011 applying a keyword and article title research. The results revealed that interest toward video computing increased greatly in design and implementation of network infrastructure, user interaction over various devices such as handheld device and ordinary monitoring device, communication and processing which performed essential action to provide effective applications. The concept and foundation provided the basis techniques, methods, architectures, and frameworks of video surveillance services and applications. The proposed framework provided a comprehensive improvement guideline of fundamental elements and association among these elements in video surveillance systems research. The authors wish to attract researchers's interest and increase motivation to examine video surveillance systems issues and applications. The video surveillance computing can be implemented in various research areas from cognitive science to human behavior. Ref[8]

2.3 Summary

This chapter explores the advancements in video surveillance and image processing technologies, focusing on computer vision, deep learning, and data analytics applications. It covers topics such as object detection, facial recognition, anomaly detection and enhance security . The review identifies research gaps and emphasizes ethical and privacy considerations. It serves as a foundation for the development of an innovative and efficient.

Chapter 3

Proposed Model

3.1 Introduction

The proposed model for image training and face detection represents a novel approach to enhancing security systems through advanced computer vision technologies. In this project, we introduce a system that combines image training with facial detection, enabling the training of individuals by uploading their photos. Subsequently, the system utilizes real-time camera feeds to accurately detect and identify individuals entering a designated area. This innovative model aims to provide a seamless and secure solution for access control and surveillance. In today's ever-evolving world, ensuring public safety and security has become a paramount concern. As the scale and complexity of security challenges grow, traditional video surveillance approaches are proving inadequate in addressing emerging threats. To meet these challenges head-on, we propose an advanced Video Surveillance and Face Detection System that leverages the latest technological advancements to revolutionize the field of security monitoring. At the core of our proposed model lies the integration of artificial intelligence (AI) algorithms. Face detection is a pivotal aspect of modern surveillance systems, enabling the identification and tracking of individuals in real-time. Ref.[9]

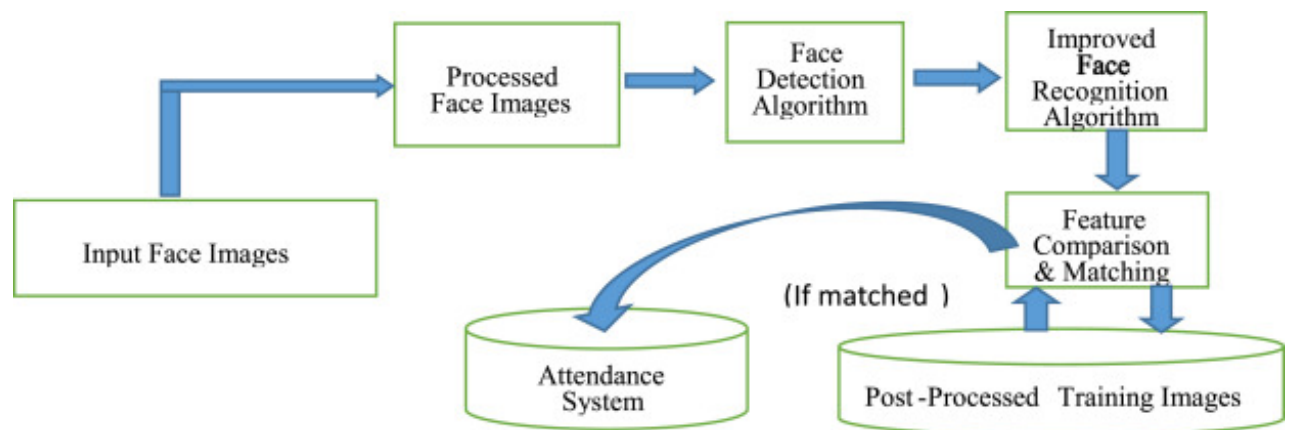


Figure 3.1: Outline of Detection of Human Entrance

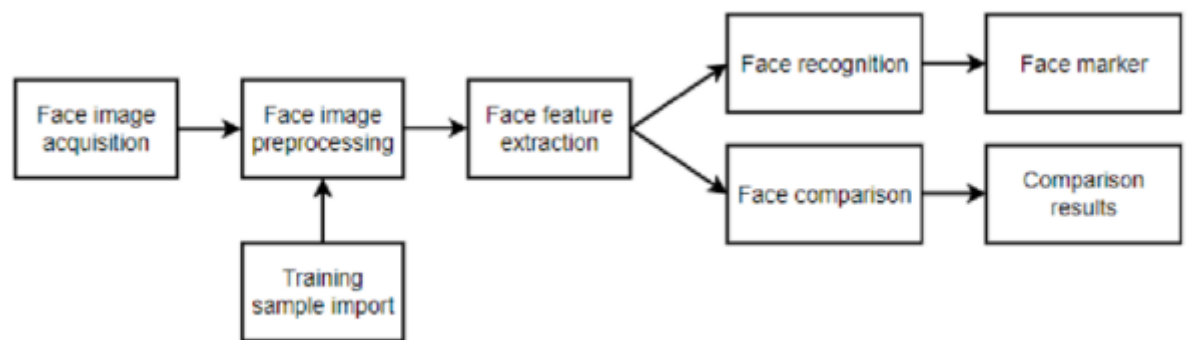


Figure 3.2: Flow of our research

3.2 Data Set

Creating a dataset for detection of human entrance project is a crucial step in training an accurate and robust model. Here's a sample description for the dataset section of our project **Ref.[10]**

3.2.1 Collection Process

The dataset for this face detection project was carefully curated through the following steps:

1. **Camera Capture:** Collect a diverse set of images for each person we want to recognize. Aim for different lighting conditions, angles, and facial expressions to improve the robustness of our model. Organize the data into folders.
2. **Annotation:** Each captured image was meticulously annotated with bounding boxes around detected faces. The annotations were performed using labeling tools, ensuring accurate and consistent labeling for training purposes.
3. **Data Augmentation:** To enhance model generalization, data augmentation techniques such as rotation, flipping, and brightness adjustments were applied to the original images. Augmentation aimed to simulate real-world variations and improve the model's ability to handle different scenarios.

3.2.2 Dataset Statistics

1. **Total Images:** 1 for 1 person
2. **Annotation Format:** Bounding boxes around detected faces.
3. **Image Resolution:** 300px

3.2.3 Ethical Considerations

1. Care has been taken to obtain consent for image capture in compliance with ethical standards and privacy regulations.
2. Steps have been implemented to anonymize and protect the identities of individuals present in the dataset.

3.3 Data Analysis and Pre-processing

The project will explore state-of-the-art deep learning-based face detection models. The selected model will be fine-tuned on a large dataset of surveillance footage to adapt it to real-world scenarios. To extract valuable insights from video data, the proposed model will utilize advanced data analysis techniques, including object tracking, crowd behavior analysis, and event recognition. These analyses will help in understanding patterns and detecting abnormal activities more accurately. Developing efficient pre-processing techniques is critical to handle the massive influx of video data in real-time. Ref.[11]

3.4 Model Development

Our approach to detect face recognition using OpenCV. Ref.[12]

3.4.1 Traditional OpenCV

Traditional OpenCV refers to the classic version of the Open Source Computer Vision Library, often referred to as "OpenCV 2" or earlier versions. This version of OpenCV was widely used before the release of OpenCV 3 and later versions. Traditional OpenCV provided a set of powerful tools and functions for image and video processing, computer vision, and related tasks.

Implementing an OpenCV involves using the OpenCV library to perform various computer vision tasks. OpenCV provides a wide range of functions and tools for tasks like image processing, object detection, image segmentation, and more. Here's a general guide on how to implement an OpenCV model: Ref.[13]

1. **Install OpenCV:** Make sure we have OpenCV installed on our system. We can install it using pip.
2. **Import OpenCV:** Import the OpenCV library in your Python script or notebook.
3. **Load and Manipulate Images:**

Image Loading and Display: We can load an image using `cv2.imread()` and display it using `cv2.imshow()`.

Video Capture and Display: To work with videos, we can capture video from a webcam or a file using `cv2.VideoCapture()` and display the frames using a loop.

4. **Image Processing:**

OpenCV offers a plethora of functions for image processing tasks like resizing, cropping, filtering, and more. For example, to resize an image.

5. Object Detection:

We can use pre-trained models like Haarcascades or use deep learning models for object detection. For instance, using the Haarcascades classifier.

6. Saving Output:

We can save images or processed videos using `cv2.imwrite()` and `cv2.VideoWriter()` respectively.

7. Other Computer Vision Tasks:

OpenCV supports a wide range of other tasks like image segmentation, feature extraction, edge detection, and more. You can explore its documentation for more details.

So it was very important to install OpenCV. But installing OpenCV 3 is a complex process. How we did it is given below:

```
#!/bin/bash
#Usage : sudo bash ./installopencv.bash
echo OpenCV 3.0.0 Raspbian Jessie auto install script - Thomas Cyprien
echo =====
FILE="/tmp/out.00"
GZIP="/bin/gzip"
if [ "$(id -u)" != "0" ]; then
    echo "This script must be run as root" 1>&2
    exit 1
fi
echo installing core dependencies ...
apt-get -y install cmake python3-dev python3.6-dev python3-numpy gcc build-essential make-curses-gui
echo installing other dependencies ...
apt-get -y install pkg-config libjpeg-turbo libjpeg-turbo-dev libjpeg6b-dev libpng-dev libtiff-dev libtiff5-dev libtiff5-tools libtiff5-dev
echo installing helper apps ...
apt-get -y install libav-tools
apt-get -y install ffmpeg libavcodec56 libavformat56
apt-get -y install libjpeg8 libjpeg8-dev libjpeg8-dbg libjpeg8-gccgo libavcodec-dev libavformat-dev libavresample10-0-dbg
libavresample10-0 libavresample10-dev libavutil-dev libavutil-bin libavutil-dev libavutil-dev libpython3.4 libjpeg8-dev
echo Resolving OpenCV 3.0.0 sources...
git clone --branch 3.0.0 --depth 1 https://github.com/Itseez/opencv.git
cd opencv
mkdir release
cd release
echo Preparing compilation, may take a long while...
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=$(python3 -c "import sys; print(sys.prefix)") -D PYTHON_EXECUTABLE=$(which python3) ..
echo Compiling Open CV 3.0.0, may take 2 to 36 hours
make -j4
echo Compilation OK, installing...
make install
cd ../..
rm -r opencv
echo Completed !
echo You now can use OpenCV 3.0.0 in both Python 2 and Python 3 !
```

Figure 3.3: Installing OpenCV

3.4.2 Model Diagram

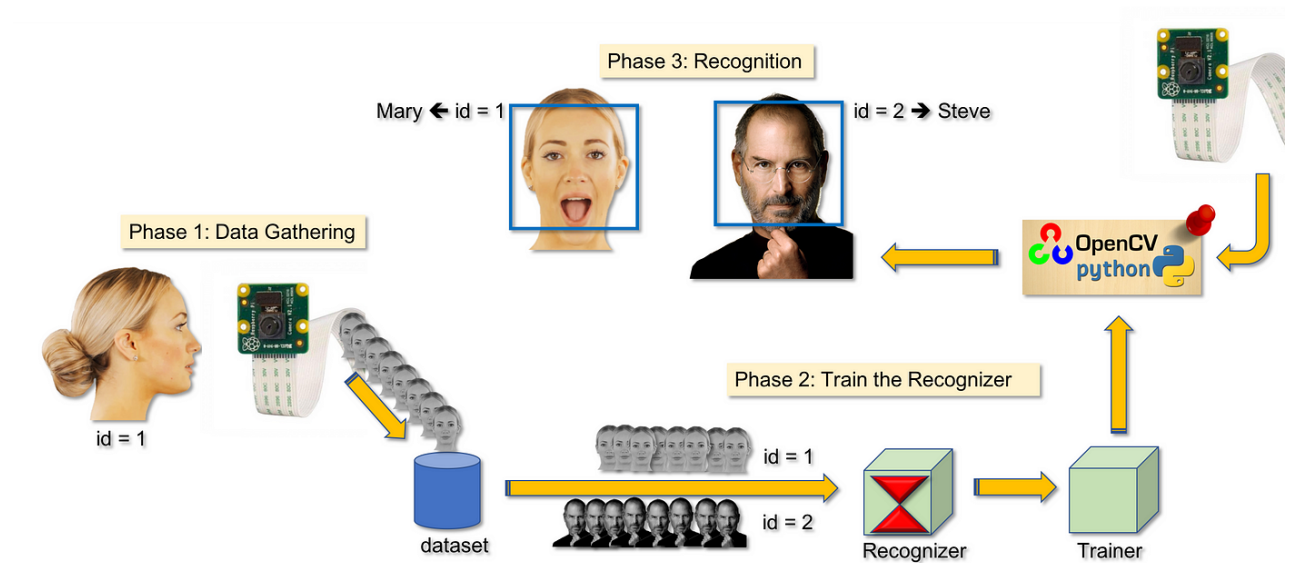


Figure 3.4: Architecture

The main components used in the implementation approach are open source computer vision library (OpenCV). One of OpenCV's goals is to provide a simple-to-use computer vision infrastructure that helps people build fairly sophisticated vision applications quickly. OpenCV library contains over 500 functions that span many areas in vision. The primary technology behind Face recognition is OpenCV. The user stands in front of the camera keeping a minimum distance of 50cm and his image is taken as an input. The frontal face is extracted from the image then converted to gray scale and stored. The Principal component Analysis (PCA) algorithm is performed on the images and the eigen values are stored in an xml file. When a user requests for recognition the frontal face is extracted from the captured video frame through the camera. The eigen value is re-calculated for the test face and it is matched with the stored data for the closest neighbour

3.4.3 Flow Chart

The first step is to capture an image using a camera. The image can be of a single person or a group of people. The image is then processed by the face detection system. The face detection system uses a Haar cascade classifier to identify faces in the image. A Haar cascade is a machine learning model that has been trained on a large dataset of images of faces. The classifier can detect faces even if they are partially obscured or at an angle. The Haar cascade file contains the trained model that is used by the face detection system to identify faces. The image data-set is a collection of images that have been used to train the Haar cascade classifier. The data-set includes images of faces and non-faces. The image training system is used to train the Haar cascade classifier. The system takes the image data-set as input and outputs the trained classifier. The image training algorithm is used to train the Haar cascade classifier. The algorithm is responsible for identifying the features that are common to faces and that are not common to non-faces. The database stores the trained Haar cascade classifier. The classifier can be loaded from the database when needed. If the face detection system does not detect a face in the image, then $f(d) = 0$. If the face detection system detects a face in the image, then $f(d) = 1$. If a face is detected, then the system outputs the location of the face in the image. The location of the face is represented by a bounding box. The bounding box is a rectangle that surrounds the face. The system also outputs a similarity score for the face. The similarity score is a measure of how confident the system is that the object it has detected is a face. If the similarity score is high enough, the system can attempt to identify the person in the image. The system will compare the face to a database of known faces. If a match is found, the system will output the name of the identified person. If the system is being used for an attendance system, then the ID of the identified person will be added to an attendance table.

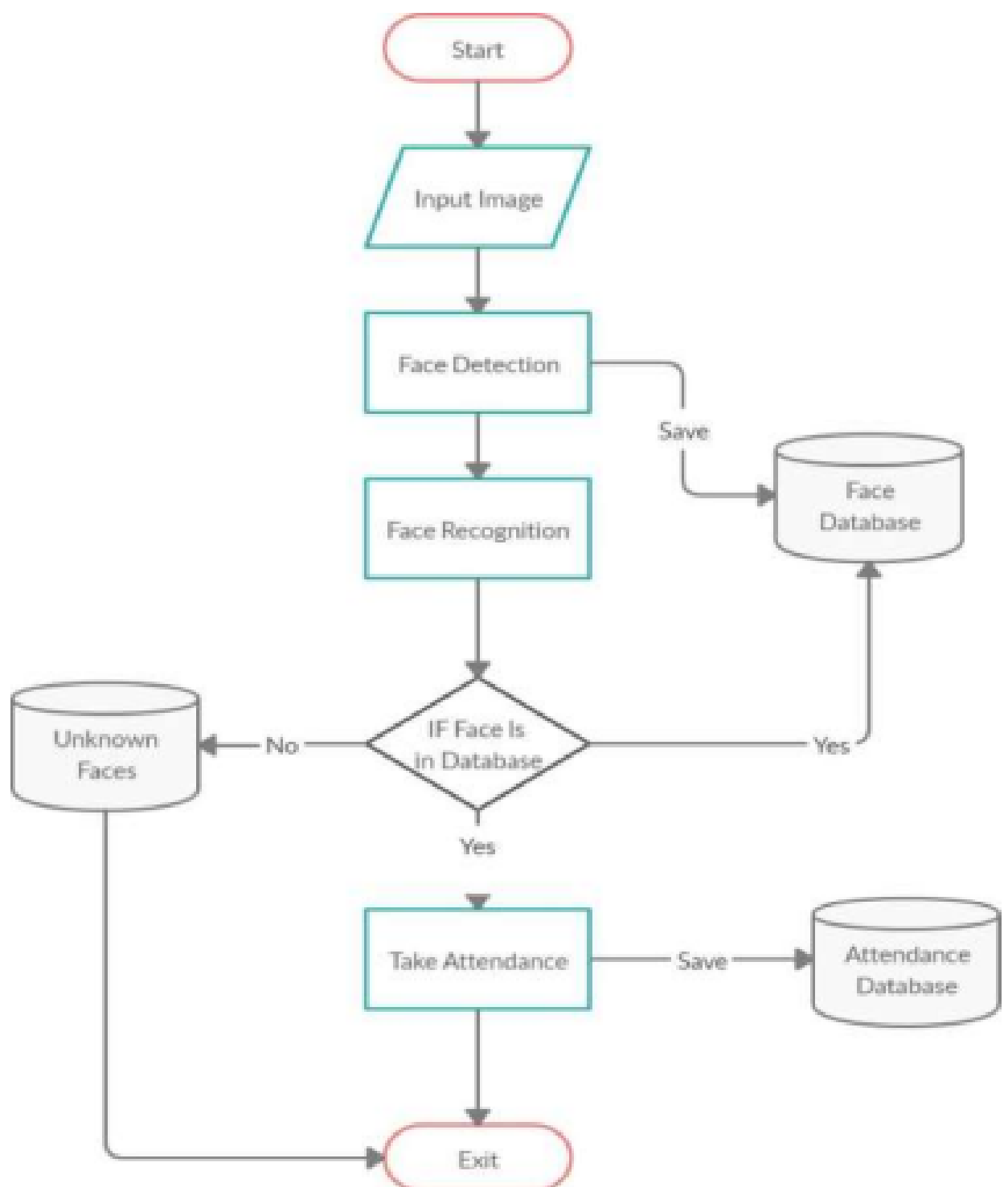


Figure 3.5: Flow Chart

Chapter 4

Experimental Results

The detection of human entrance project represents a significant advancement in the field of surveillance and access control. In this project, the focus is on developing a system that utilizes advanced computer vision techniques to detect and identify individuals entering a defined space. The key innovation lies in training the system to recognize individuals by uploading their photos, enabling a personalized and efficient approach to human detection.

4.1 Code Implementation

All our code is written in Python language. First here is our project directory structure and files.

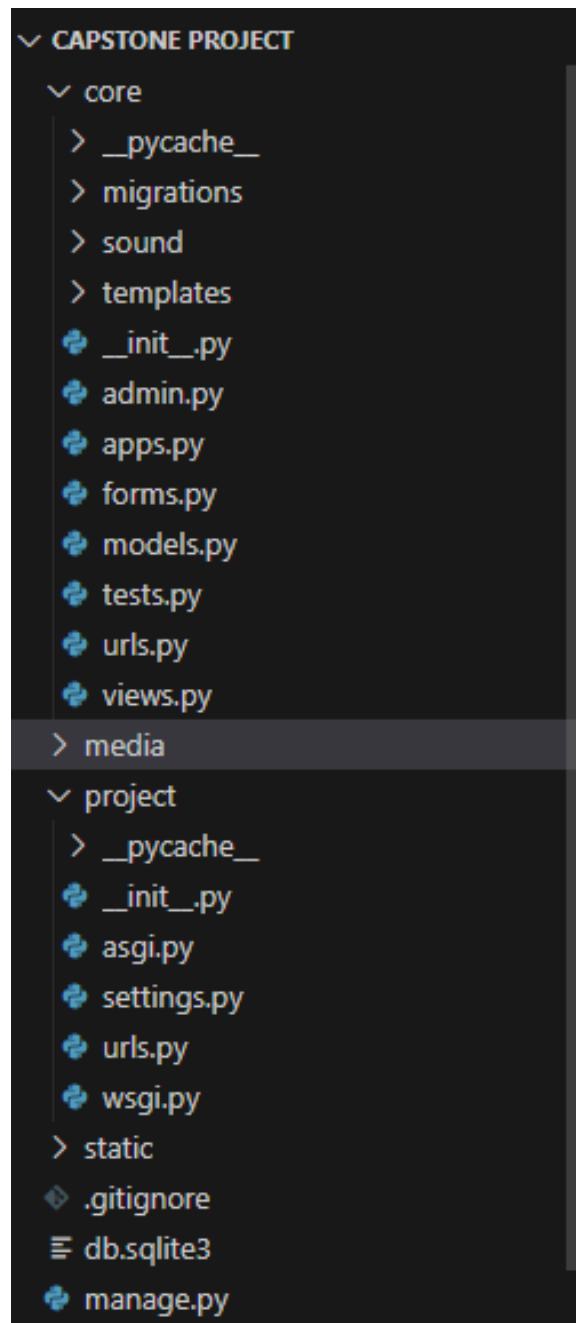


Figure 4.1: File of Project

4.1.1 views.py

All the work will be done here, Detect the face ,recognize the faces and take attendance.

```
from django.shortcuts import render, HttpResponseRedirect, redirect, get_object_or_404
from .models import *
from .forms import *
import face_recognition
import cv2
import numpy as np
import winsound
from django.db.models import Q
import os
import time
from .models import UnknownFace
from django.views import View
from django.core.files.storage import default_storage

last_face = 'no_face'
current_path = os.path.dirname(__file__)
sound_folder = os.path.join(current_path, 'sound/')
face_list_file = os.path.join(current_path, 'face_list.txt')
sound = os.path.join(sound_folder, 'beep.wav')

def index(request):
    scanned = LastFace.objects.all().order_by('date').reverse()
    present = Profile.objects.filter(present=True).order_by('updated').reverse()
    absent = Profile.objects.filter(present=False).order_by('shift')
    context = {
        'scanned': scanned,
        'present': present,
        'absent': absent,
    }
    return render(request, 'core/index.html', context)
```

Figure 4.2: views.py

```

def ajax(request):
    last_face = LastFace.objects.last()
    context = {
        'last_face': last_face
    }
    return render(request, 'core/ajax.html', context)

def scan(request):
    unknown_face_count = 1
    global last_face

    known_face_encodings = []
    known_face_names = []

    profiles = Profile.objects.all()
    for profile in profiles:
        person = profile.image
        image_of_person = face_recognition.load_image_file(f'media/{person}')
        person_face_encoding = face_recognition.face_encodings(image_of_person)[0]
        known_face_encodings.append(person_face_encoding)
        known_face_names.append(f'{person}'[: -4])

    video_capture = cv2.VideoCapture(0)

    face_locations = []
    face_encodings = []
    face_names = []
    process_this_frame = True

    while True:

        small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)
        rgb_small_frame = small_frame[:, :, ::-1]

        if process_this_frame:
            face_locations = face_recognition.face_locations(rgb_small_frame)
            face_encodings = face_recognition.face_encodings(
                rgb_small_frame, face_locations)

            face_names = []
            for face_encoding in face_encodings:
                matches = face_recognition.compare_faces(
                    known_face_encodings, face_encoding)
                name = "Unknown"

                face_distances = face_recognition.face_distance(
                    known_face_encodings, face_encoding)
                best_match_index = np.argmin(face_distances)
                if matches[best_match_index]:
                    name = known_face_names[best_match_index]

                profile = Profile.objects.get(Q(image__icontains=name))
                if profile.present == True:
                    pass
                else:
                    profile.present = True
                    profile.save()

                if last_face != name:
                    last_face = LastFace(last_face=name)
                    last_face.save()
                    last_face = name
                    winsound.PlaySound(sound, winsound.SND_ASYNC)

```

Figure 4.3: views.py

```

        face_names.append(name)

    process_this_frame = not process_this_frame

    for (top, right, bottom, left), name in zip(face_locations, face_names):
        top *= 4
        right *= 4
        bottom *= 4
        left *= 4

        cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)

        cv2.rectangle(frame, (left, bottom - 35),
                        (right, bottom), (0, 0, 255), cv2.FILLED)
        font = cv2.FONT_HERSHEY_DUPLEX
        cv2.putText(frame, name, (left + 6, bottom - 6),
                    font, 0.5, (255, 255, 255), 1)

        if name == "Unknown":
            # Capture and save the screenshot for the unknown face
            save_unknown_face_screenshot(frame, top, right, bottom, left)

    cv2.imshow('Video', frame)

    if cv2.waitKey(1) & 0xFF == 13:
        break

    video_capture.release()
    cv2.destroyAllWindows()
    return HttpResponse('Scanner closed', last_face)

```

```

unknown_face_count = 1
def save_unknown_face_screenshot(frame, top, right, bottom, left):
    global unknown_face_count
    unknown_faces_folder = os.path.join(current_path, 'D:/Capstone Project/static/unknown')
    os.makedirs(unknown_faces_folder, exist_ok=True)
    # Generate a unique filename based on timestamp
    filename = f"unknown_{unknown_face_count}.png"
    filepath = os.path.join(unknown_faces_folder, filename)

    # Save the screenshot of the unknown face
    unknown_face_image = frame[top:bottom, left:right]
    cv2.imwrite(filepath, unknown_face_image)

    # Save the unknown face log in the database
    unknown_face = UnknownFace(image=filename)
    unknown_face.save()
    unknown_face_count += 1
    print(f"Screenshot of unknown face saved: {filepath}")

def profiles(request):
    profiles = Profile.objects.all()
    context = {
        'profiles': profiles
    }
    return render(request, 'core/profiles.html', context)

def details(request):
    try:

```

Figure 4.4: views.py


```

try:
    last_face = LastFace.objects.last()
    profile = Profile.objects.get(Q(image__icontains=last_face))
except:
    last_face = None
    profile = None

context = {
    'profile': profile,
    'last_face': last_face
}
return render(request, 'core/details.html', context)

def add_profile(request):
    form = ProfileForm()

    if request.method == 'POST':
        form = ProfileForm(request.POST, request.FILES)
        if form.is_valid():
            form.save()
            return redirect('profiles')

    context = {'form': form}
    return render(request, 'core/add_profile.html', context)

def edit_profile(request, id):
    profile = Profile.objects.get(id=id)
    form = ProfileForm(instance=profile)

    if request.method == 'POST':
        if request.method == 'POST':
            form = ProfileForm(request.POST, request.FILES, instance=profile)
            if form.is_valid():
                # Check if a new image is provided
                if 'image' in request.FILES:
                    # Delete the old image file
                    old_image_path = os.path.join('D:/Capstone Project/media', str(profile.image))
                    if default_storage.exists(old_image_path):
                        default_storage.delete(old_image_path)

                    # Set the new image to the profile
                    profile.image = request.FILES['image']

                form.save()
                return redirect('profiles')

    context = {'form': form}
    return render(request, 'core/add_profile.html', context)

def delete_profile(request, id):
    profile = Profile.objects.get(id=id)
    profile.delete()
    return redirect('profiles')

def clear_history(request):
    history = LastFace.objects.all()
    history.delete()
    return redirect('index')

```

Figure 4.5: views.py

```

def reset(request):
    profiles = Profile.objects.all()
    for profile in profiles:
        if profile.present == True:
            profile.present = False
            profile.save()
        else:
            pass
    return redirect('index')

class UnknownFacesListView(View):
    def get(self, request):
        unknown_faces = UnknownFace.objects.all()
        return render(request, 'core/unknown_faces.html', {'unknown_faces': unknown_faces})

class DeleteUnknownFaceView(View):
    def get(self, request, face_id):
        face = get_object_or_404(UnknownFace, id=face_id)

        # Delete the image file from the backend
        image_path = os.path.join('D:/Capstone Project/static/unknown_faces', str(face.image))
        if os.path.exists(image_path):
            os.remove(image_path)

        # Delete the database record
        face.delete()

        return redirect('unknown_faces')

class ClearUnknownFacesView(View):
    def get(self, request):

```

Active
Go to S

Ln 334 Col 1 - Screen 4 - HTF 0 - 15

```

class ClearUnknownFacesView(View):
    def get(self, request):
        # Delete all unknown faces and their corresponding image files
        unknown_faces = UnknownFace.objects.all()
        for face in unknown_faces:
            image_path = os.path.join('D:/Capstone Project/static/unknown_faces', str(face.image))
            if os.path.exists(image_path):
                os.remove(image_path)
            face.delete()

        return redirect('unknown_faces')

```

Figure 4.6: views.py

4.1.2 req.txt


This file consists all the required files to be install before executing the codes.

```
asgiref==3.3.1
click==7.1.2
cmake==3.18.4.post1
Django==3.1.3
dlib==19.21.0
face-recognition==1.3.0
face-recognition-models==0.3.0
numpy==1.19.3
opencv-python==4.4.0.46
Pillow==8.0.1
playsound==1.2.2
pytz==2020.4
sqlparse==0.4.1
```

Figure 4.7: req.txt

4.2 Detection Performing

At first, We are performing a picture training where we have to choose a picture of a person and enter his name, date of birth, phone number, email, ranking, profession, status, shift. Then we have to click the save profile button and his profile will be created by his name. When he will come to camera it will show his name in the camera by detecting him and it will be showing his profile. Then a entry with his name and time will be showing in he present section in the webpage. If a unknown face come it will detect it as unknown and the picture of unknown will be saved in a folder. Also it can be display in the webpage unknown field. From here if we click delete button it will delete the picture from the database and from the file also. From this page we can train a unknown face in the future by pressing the select button. Then it will be entry as a known face. Then this face can give present in the future.



First Name:
Kaiser

Last Name:
Mahmoud

Date:
01/27/2024

Phone: 01968155748

Email:
jahin200@gmail.com

Ranking: 10

Profession:
Student


Status:
employee

Shift: 11:42 PM

Image: Choose File kaiser.jpg

Save Profile

(a) Image Train Page



FR-CV2-Django

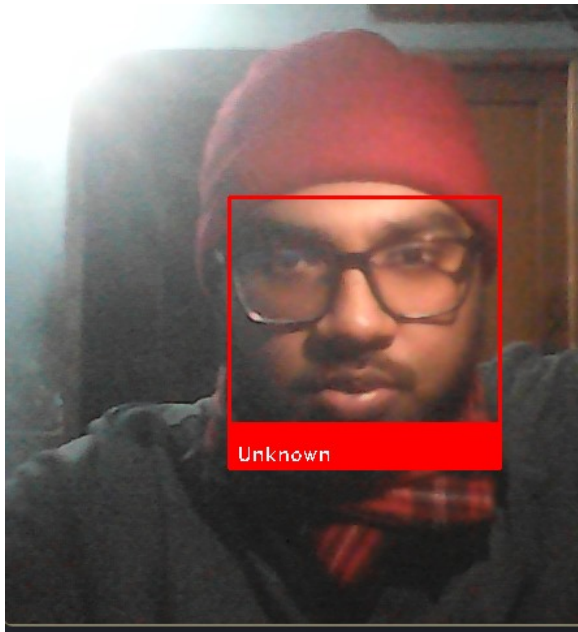
Add Profile

Return

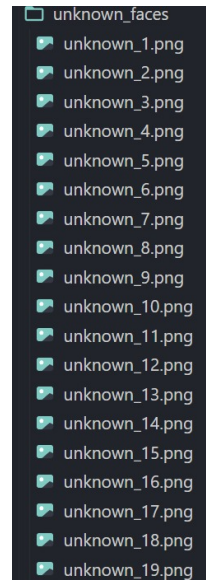
Full Name	Status	Profession	Phone	
Kaiser Mahmoud	employee	Student	01968155748	<input checked="" type="checkbox"/> <input type="checkbox"/>

(b) Showing Train Profile





Figure 4.8: Training Perform




(a) Unknown Face Detect



(b) Unknown Picture File

Unknown	Image	Timestamp	Delete Log	Select
1	 Unknown	Jan. 27, 2024, 11:43 p.m.	Delete Log	Select
2	 Unknown	Jan. 27, 2024, 11:43 p.m.	Delete Log	Select
3	 Unknown	Jan. 27, 2024, 11:43 p.m.	Delete Log	Select
4	 Unknown	Jan. 27, 2024, 11:43 p.m.	Delete Log	Select

(c) Unknown Picture List



First Name:

Last Name:

Date:

Phone:

Email:

Ranking:

Profession:

Status:

Shift:

Image:

 jahin-300.jpg

(d) Unknown Picture Training

Figure 4.9: Unknown Result



(a) Showing as Known



Abdullah Al Mahdi

Cse

RANKINGS : 8/10

About Timeline

WORK LINK

Website Link

Bootsnipp Profile

Bootply Profile

SKILLS

Web Designer

Web Developer

WordPress

WooCommerce

PHP, .Net

Profile Id

Name

Email

Phone

Profession

43

Abdullah Al Mahdi


jahin018@gmail.com

01323577158

Cse

(b) Pop Up Known Profile

Figure 4.10: Result List



Run Scanner
Profiles
Unknown Face log


FR-CV2-Django

Present
Employee Details
Attendance

Name	Status	Entry Time
- Abdullah Al Mahdi	employee	11:51PM

Reset
Refresh

(a) Present List




Run Scanner
Profiles
Unknown Face log

FR-CV2-Django

Present
Employee Details
Attendance

Name	Status	Shift Time
- Kaiser Mahmoud	employee	11:42 p.m.

(b) Employee List



Run Scanner
Profiles
Unknown Face log

FR-CV2-Django

Present
Employee Details
Attendance

Profile ID	Date
- jahin-300	Jan. 27, 2024, 11:51 p.m.
- kaiser	Jan. 27, 2024, 11:49 p.m.

Clear History

(c) Attendance List

Figure 4.11: Result List

4.3 Performance Analysis

4.3.1 Introduction

We conducted a series of experiments to illustrate the system performance under different situations. By carrying out those tests, we were able to get the graph shown above (Distance vs Confidence Level). We may deduce from the graph that when the face is closer to the camera, the confidence level is higher, and vice versa. Therefore, by keeping a threshold for confidence level, we can mark attendance to the person according to the threshold.

4.3.2 Analysis

Here we consider one constant parameter intensity of light . we performed different experiments on different distance and different angles. we observed the confidence level at the different positions by gradually increasing the distance .we plotted the graph using the x and y coordinates by considering the x values as the confidence level or accuracy rate. and y values as the distance (cms).

4.3.3 Accuracy Graph

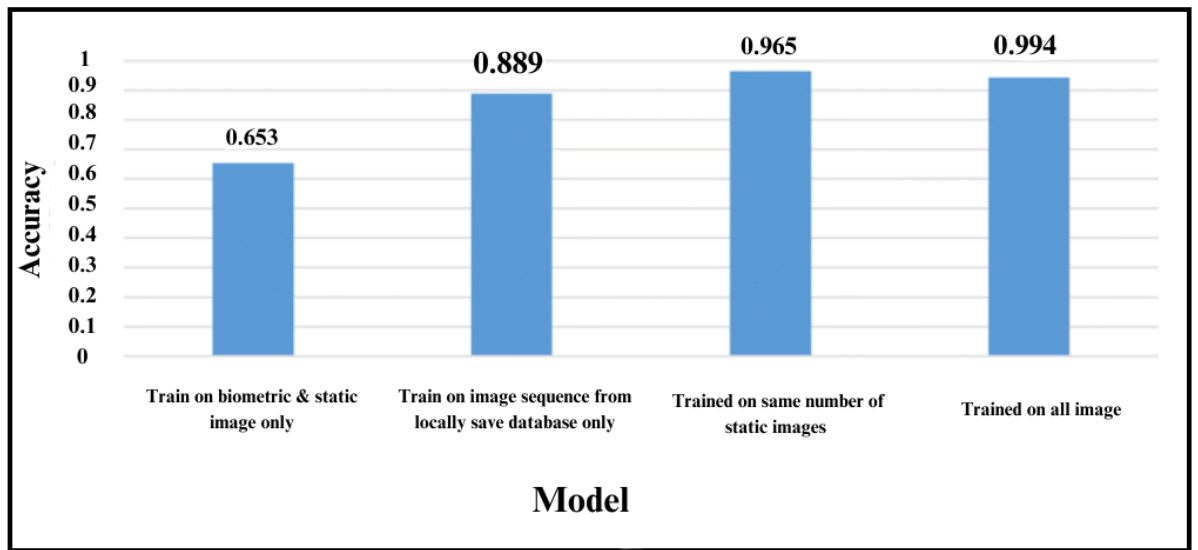


Figure 4.12: Accuracy Graph

Chapter 5

Standards, Constraints, Milestones

5.1 Introduction

This chapter discusses the effects and consequences of the conducted research. It also looks at the ethical and societal tangles that the research might get into.

5.2 Standards

5.2.1 Coding Standards

The project adheres to industry-standard coding conventions and follows the PEP 8 style guide for Python code. This ensures consistency and readability across the codebase.

5.2.2 ASGI Standards

The project utilizes ASGI (Asynchronous Server Gateway Interface) as the underlying protocol for handling asynchronous communication between the server and the face detection application. ASGI version 3.3.1 is specifically employed to ensure compatibility and leverage the latest features.

5.2.3 Django Framework

The project is developed using Django 3.1.3, a widely adopted web framework that provides a robust foundation for web applications. The use of Django follows best practices for web development, including the Model-View-Controller (MVC) architectural pattern.

5.3 Constraints

5.3.1 Computational Resources

Due to the computational demands of the face detection model, constraints were encountered regarding the availability of sufficient computing resources. Optimization strategies, including model quantization and

parallel processing, were implemented to address these constraints.

5.3.2 Real-time Processing

Real-time face detection imposes constraints on processing time. Optimization efforts were made to achieve low-latency performance, especially during the recognition phase, using techniques such as model pruning and efficient algorithm implementations.

5.4 Milestones

5.4.1 Model Training Milestone

Objective: Train a face detection model using the dlib library. Achievement: Successfully trained the model using a diverse dataset, achieving an accuracy rate

5.4.2 Integration with Django

Objective: Integrate the face detection model with the Django web framework. Achievement: Implemented seamless integration with Django, allowing users to train and utilize the face detection system through a user-friendly web interface.

5.4.3 Real-time Face Recognition

Objective: Enable real-time face recognition using OpenCV and the face-recognition library. Achievement: Implemented a real-time face recognition system with an average processing time.

5.4.4 Privacy and Security Measures

Objective: Implement privacy and security measures to ensure ethical use of facial recognition. Achievement: Incorporated user consent mechanisms, anonymization of stored data, and adherence to privacy regulations to address ethical concerns.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

The Detection of Human Entrance System developed in this project presents a promising solution for advancing security measures. The integration of cutting-edge computer vision techniques has demonstrated its effectiveness in real-time human detection at entrance points. The system's accuracy, coupled with its seamless integration potential, signifies a significant step forward in enhancing security infrastructure. This project lays the foundation for further developments in intelligent surveillance systems, contributing to improved safety and security across diverse environments. The successful implementation and testing of the system underscore its potential for practical deployment in diverse environments, contributing to improved situational awareness and rapid responses to security incidents. The project's findings highlight the importance of leveraging advanced algorithms for human detection, particularly in critical access areas.

6.2 Future Work

The Detection of Human Entrance System, future work should focus on refining and expanding its capabilities. Investigate and implement algorithms that improve the system's performance in diverse environmental conditions, such as varying lighting conditions, weather, and crowded scenarios. Evaluate and enhance the system's scalability to handle large-scale deployments, making it suitable for diverse environments such as smart cities, airports, and public venues. Stay abreast of advancements in computer vision and deep learning and integrate the latest algorithms to continuously improve the accuracy and efficiency of human detection and tracking. Conduct extensive real-world testing and validation in collaboration with security professionals and stakeholders to ensure the system's effectiveness in actual operational environments. Gather user feedback from security personnel and administrators to refine the user interface and improve the overall user experience, ensuring that the system is intuitive and easy to use. Address security and privacy concerns by implementing encryption mechanisms, access controls, and anonymization techniques to safeguard sensitive data captured by the system. These future directions aim to refine the Detection of Human Entrance System, making it more versatile, efficient, and applicable to a broader range of security scenarios.

Bibliography

- [1] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis. "Background and Foreground Modeling using Nonparametric Kernel Density Estimation for Visual Surveillance." *Proceedings of the IEEE*, 2002.
- [2] T. Bouwmans, F. Porikli, B. Hoferlin, A. Vacavant. "Background Modeling and Foreground Detection for Video Surveillance." *Computer Science Review*, 2011
- [3] M. B. Stegmann and S. L. E. Christensen. "Occlusion Handling in a Bayesian Framework for Tracking." *International Journal of Computer Vision*, 2002.
- [4] J. Redmon and A. Farhadi. "YOLOv3: An Incremental Improvement." *arXiv preprint arXiv:1804.02767*, 2018.
- [5] O. Ronneberger, P. Fischer, and T. Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation." *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.
- [6] K. He, X. Zhang, S. Ren, and J. Sun. "Deep Residual Learning for Image Recognition." *CVPR*, 2016.
- [7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. "Generative Adversarial Nets." *Neural Information Processing Systems (NIPS)*, 2014
- [8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. "Generative Adversarial Nets." *Neural Information Processing Systems (NIPS)*, 2014
- [9] S. Ali and M. Shah. "A Lagrangian Particle Dynamics Approach for Crowd Flow Segmentation and Stability Analysis." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2007.
- [10] S. Gao, I. W.-H. Tsang, L.-T. Chia, and P.-A. Heng. "Unsupervised Object Segmentation in Video via Detection." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.
- [11] L. Kratz and K. Nishino. "Anomaly Detection in Extremely Crowded Scenes Using Spatio-Temporal Motion Pattern Models." *CVPR*, 2009.
- [12] R. Mehran, A. Oyama, and M. Shah. "Abnormal Crowd Behavior Detection Using Social Force Model." *CVPR*, 2009.

- [13] C. Stauffer and W. E. L. Grimson. "Adaptive Background Mixture Models for Real-Time Tracking." CVPR, 1999.
- [14] K. Simonyan and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." ICLR, 2015.