A Project Report on

ML Enabled Surveillance System for Societies

Submitted in partial fulfillment of the requirements for the award
of the degree of

Bachelor of Engineering

in

Information Technology

by

Mr.Pranav Chauhan(17204004)
Mr.Sachin Gupta(17204001)
Mr.Rohit Arava(17204017)

Under the Guidance of

Dr. Sameer Nanivadekar
Mr. Vishal Badgujar

Department of Information Technology
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615
UNIVERSITY OF MUMBAI

Academic Year 2019-2020

Approval Sheet

This Project Report entitled ***"ML Enabled Surveillance System for Societies"*** Submitted by ***"Mr.Pranav Chauhan"(17204004), "Mr.Sachin Gupta"(17204001), "Mr.Rohit Arava"(17204017)***is approved for the partial fulfillment of the requirenment for the award of the degree of ***Bachelor of Engineering*** in ***Information Technology*** from ***University of Mumbai***.

Mr. Vishal Badgujar
Co-Guide

Dr. Sameer Nanivadekar
Guide

Mr. Kiran Deshpande
Head Department of Information Technology

Place:A.P.Shah Institute of Technology, Thane
Date:

CERTIFICATE

This is to certify that the project entitled ***"ML Enabled Surveillance System for Societies"*** submitted by ***"Mr.Pranav Chauhan" (17204004),"Mr.Sachin Gupta" (17204001),"Mr.Rohit Arava" (17204017)*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Information Technology*** to the University of Mumbai,is a bonafide work carried out during academic year 2019-2020.

Mr. Vishal Badgujar
Co-Guide

Dr. Sameer Nanivadekar
Guide

Mr. Kiran Deshpande
Head Department of Information Technology

Dr. Uttam D.Kolekar
Principal

External Examiner(s)

1.

2.

Place:A.P.Shah Institute of Technology, Thane
Date:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

_____

(Signature)

_____

(Mr.Pranav Chauhan 17204004)
(Mr.Sachin Gupta 17204001)
(Mr.Rohit Arava 17204017)

Date:

Abstract

Today every society faces the problem of intruders and illegal vehicle parking and many outsiders are entertained in the premises, which can make harm to the public and can damage their property, even the register book can be easily manipulated by the intruders. The purpose of this study is to successfully develop a system that will detect the vehicle number plate, recognize the human faces of the residential members and as well as for the outsiders in real-time using a deep neural network. The walls of our societies are not that safe, our model detects any intruder penetrating the society walls and as detected an alarm is raised to stop the intruder and make aware to the residential members. The application provides interesting features which improve the overall security of the residential society.

Contents

List of Figures

List of Tables

List of Abbreviations

ML:	Machine Learning
CNN:	Convolutional neural network
DL:	Deep Learning
CCA:	Connected Component Analysis
HMM:	Hidden Markov models
IA:	Image Acquisition
ANRP:	Automatic number-plate recognition
ALRP:	Automatic license-plate recognition
KNN:	k-Nearest Neighbors

Chapter 1

Introduction

Many residential societies in India face an impending problem of illegal vehicle parking inside their societies and theft of the vehicles. This issue is not bounded to just vehicles, but also adds to other security concerns inside the residential societies. Though there are solutions that exist in the market for monitoring through cameras and software system but are expensive and the affordability comes into the question. In this regard, we would like to have an affordable and innovative solution that caters to the Indian market. Scope of the System is notifying the vehicle owners about the check-in and check-out of their respective vehicle into their apartment's. The project will work as a security system for a residential society. Proper Details of the outsiders will be maintained in the form of an image which will be stored in the database. If any intruder comes from the boundary wall of the society can be easily detected and the alarm will on to restrict the intruder.

Our ML Enabled Surveillance System for Societies is based on three modules which are face detection, vehicle number plate detection and imaginary line/virtual line. The face detection module is used for detecting residential members and non-residential member's faces in real-time. If a person is detected non-residential, then the person's face will be stored in the database for future use. Another module is the vehicle number plate detection which is used for detecting residential and non-residential vehicles number plate in real-time from live footage. If a vehicle is detected non-residential vehicle, then the vehicle's number plate is stored in the database for future use. And the last module is the imaginary line/virtual line module which is placed at the boundary wall of the societies and if any intruder passed through the imaginary line/virtual line, it will detect the intruder and an alarm will ON to stop that intruder. If any object like a cat, dog, etc passed through the imaginary line/virtual line, it will detect the object but no alarm will ON for such objects.And the imaginary line/virtual line module is also used for counting the number of members which can be residential or non-residential in real-time.For the face detection module, we have used the OpenCV library and K-Nearest Neighbors (KNN) classifier.

OpenCV is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez. The library is cross-platform and free for use under the open-source BSD license. And K-Nearest Neighbors (KNN) is one of the simplest algorithms used in Machine Learning for regression and classification problems. KNN algorithms use data and classify new data points based on similarity measures (e.g. distance function).

Classification is done by a majority vote to its neighbors. The data is assigned to the class which has the nearest neighbors. As you increase the number of nearest neighbors, the value of k, accuracy might increase. For the vehicle number plate detection module, we have used the ANPR/ALPR is an open-source Automatic License Plate Recognition library to perform vehicle number plate recognition. The library analyses images and video streams to identify license plates. For creating the imaginary line/virtual line we have used python math function and we have used YOLOv3 model for object detection.YOLO is a clever convolutional neural network (CNN) for doing object detection in real-time. The algorithm applies a single neural network to the full image, and then divides the image into regions and predicts bounding boxes and probabilities for each region.

Chapter 2

Literature Review

[1]Real-Time vehicle detection and tracking using deep neural networks.
Authors: Xiao-Feng Gu, Zi-Wei Chen, Ting-Song Ma, Fan Li, Long Yan
This paper was published in the year 2016. they have labelled vehicles by themselves and Tested on ImageNet dataset. Their system was implemented on a build which consists of an 8-core 4GHz Intel I7 6700K, an NVIDIA GTX1080 GPU and 8GB of RAM and they train this network for approximately 5 days. Needs They need high processing power to run and get the output file. as well as they took a large amount of time just to train, detect and track vehicles.and they aren't getting the number plate information from those vehicles. and not even detecting pedestrians.
Advantages:
In this system, they have propose a network architecture for vehicle detection and tracking in real-time. In general, their network includes 9 convolutional layers, 4 inception modules, one SPP layer and 2 fully connect layers. It is 170 ideal for computer vision application. This proposed system can be applied in the fields of intelligent road routing, road traffic control, road planning and so on.
Disadvantages:
There are limitations in their network architecture. Their system struggles with small objects and nearby object in groups.Needs high processing power to run the system.

[2] A New Approach For Vehicle Number Plate Detection.
Authors: Sarthak Babbar, Saommya Kesarwani, Navroz Dewan, Kartik Shangle, Sanjeev Patel
This paper was published in the year 2018. There are several steps that they are performing to achieve the vehicle number plate. which are quite huge and less worthy as the output is considered. Those are 1. Gray Scale Conversion 2. Binarization 3. Number Plate Localization here there are furthermore steps: A) Connected Component Analysis(CCA) and B) Ratio Analysis then 4. Noise Reduction 5. Character Segmentation 6. Optical Character Recognition 7. Character Segmentation and Recognition. Their designed network is good for fetching information from an image of a single-vehicle but doesn't work well in the case of multiple vehicles. Here vehicle number plates are only detected not along with their associated vehicle, this can cause conflict while several vehicles are entering at the same time. even they aren't detecting pedestrians.
Advantages:
Their proposed technique proved to be successful for detecting number plates even in low

light conditions,extreme brightness which otherwise failed in previous machine learning systems.The system is also successful for detection number plates from skewed angles.
Disadvantages:
Suitable for detecting vehicles at fixed places only.It is also complex to be implemented.

[3] Forward Vehicle Detection Based on Incremental Learning and Fast R-CNN.
Authors: KaijingShi, Nan Ma, Hong Bao
This paper was published in the year 2017. They have used CNN and re-trained on ImageNet dataset. And able to recognized real-time traffic which including cars of all type. Their network model was divided into two phases, training and testing phase. At training phase, the convolution neural network of initial parameters after pre-training on ImageNet is re-trained to obtain the vehicle detection model. At test phase, the test samples are input into detection model to obtain the test result. They choose the KITTI public data set of cars derived from the KITTI Vision Benchmark Suite and BUU - T2Y data set. Use a trained model to test. Firstly, randomly selecting 20% KITTI data as a test set to test the model, adjust the parameters and get the best test results. Then, apply the trained model to the BUU-T2Y dataset to test.Finally, the BUU-T2Y data set and the KITTI data set are randomly combined with incremental testing.
Advantages:
In this system the experimental results showed that if the KITTI data set was used to train the model, and then the BUU-T2Y dataset was used to adjust to the optimization model. Finally, the BUU-T2Y data set was added to the KITTI dataset to form a new data set for incremental learning. The experimental result can reach 86.2%.
Disadvantages:
The system does not provide the end-to-end network structure. And the system does not execute in the real time environment.

[4] Video Based License Plate Recognition of Moving Vehicles Using Convolutional Neural Network.
Authors: Sanghyeop Lee, Keum-Young Son, Byung- Woo Yoon, Jangsik Park
This paper was published in the year 2018. In this paper, they have used AlexNet for training the ImageNet along with OpenALRP for detecting number plates. they need a lot of GPU performance to run and then to output varies on funky vehicle number plates, and even they aren't detecting pedestrians.They use DIGTIS to recognize numbers and letters first, then increase each recognized area by 40% to find objects around them. These discovered objects are identified by four consecutive numbers and two consecutive numbers, which are identified as the number plates.
Advantages:
In this paper, a built-in system was implemented with a GPU in order to recognize the license plate number without detection line. The deep-learning network to recognize the license plate number of the vehicle uses relatively simple AlexNet. The vehicle license plate can be effectively recognized using CNN.
Disadvantages:
The problem in this proposed system is that it is difficult to recognize the green coloured license plate.

[5] A Hidden Markov Model for Vehicle Detection and Counting.

Authors: Nicholas Miller, Mohan A. Thomas, Justin A. Eichel, Akshaya Mishra

This paper was published in the year 2015. Their goal is to measure real-world accuracy and reliability to assess suitability for deployment in an ITS system with realistic challenges, including changing lighting conditions, but using MatLab for suck task makes it less affordable and the biggest problem occurs when the driver drives on either half lane. this model raises a difficulty in handling this, and they are just detecting the vehicles in a particular area, recognization of number plate needs to be cared about. and even they aren't detecting pedestrians.

Advantages:

This method has been proposed to take advantage of constrained vehicle motion to detect and count vehicles using a hidden Markov model.The proposed method was also more reliable having fewer and less severe occurrences of 5- minute-bin errors throughout the testing set.

Disadvantages:

This hidden Markov models (HMM) approach fails to model a few aspects of the scene leading to some failures, particularly due to large vehicles out of the scale of the configured zone, and bumper to bumper vehicle traffic placing two vehicles in a zone at the same time.

Chapter 3

Problem Definition

In the current scenario, security is one of the major problem faced in societies. Boundaries of the society are also very less secure, an intruder can easily penetrate the boundary wall of the societies. Many outsider/salesman enter without permission into society. IN and OUT of the residential vehicle needs to be managed. Many societies also face the problem of illegal vehicle parking, to overcome such scenario, we are providing an extra layer of security in the residential societies by detecting human faces and vehicle number plate using deep learning algorithm such as CNN (convolutional neural network) having excellent performance compared to traditional algorithms.

3.1 Objectives

Following are the major objectives of our system:

- To provide effective security to the residential society using deep learning algorithm.
- To detect residential and nonresidential vehicle through a vehicle number plate.
- Detect residential and non-residential member through face detection.
- Any intruder penetrating the boundary wall of the societies will be by the camera.
- The need for watchman in residential societies will be reduced.

3.2 Hardware and Software Requirements

3.2.1 Software Requirements:

Operating System	Ubuntu
IDE	Python3.7
Database	Mysql

Table 3.1: Software Requirements

3.2.2 Packages & Libraries used for Machine Learning:

Packages & Libraries
Pillow
Anaconda
TKinter
OpenCV3
ANPR/ALPR
face_recognization
Python math function

Table 3.2: Packages & Libraries used for Machine Learning

3.2.3 Hardware Requirements:

Laptop/PC	Minimum 4GB of RAM
Storage	Minimum 250GB Hard disk
Camera	Minimum 2mp with 720p

Table 3.3: Hardware Requirements

3.2.4 Costing of the project:

The software components which are used in this project are open source, therefore no significant amount of money was required to develop the software part of the project. And for the hardware part of the project, it will cost approximately 22,000 to 25,000 RS which is a one-time investment for the users or customers.

Chapter 4

Proposed System Architecture

A Proposed System Architecture is the model that defines the structure, behavior, and more views of a system. An Proposed System Architecture is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

4.1 Face Detection

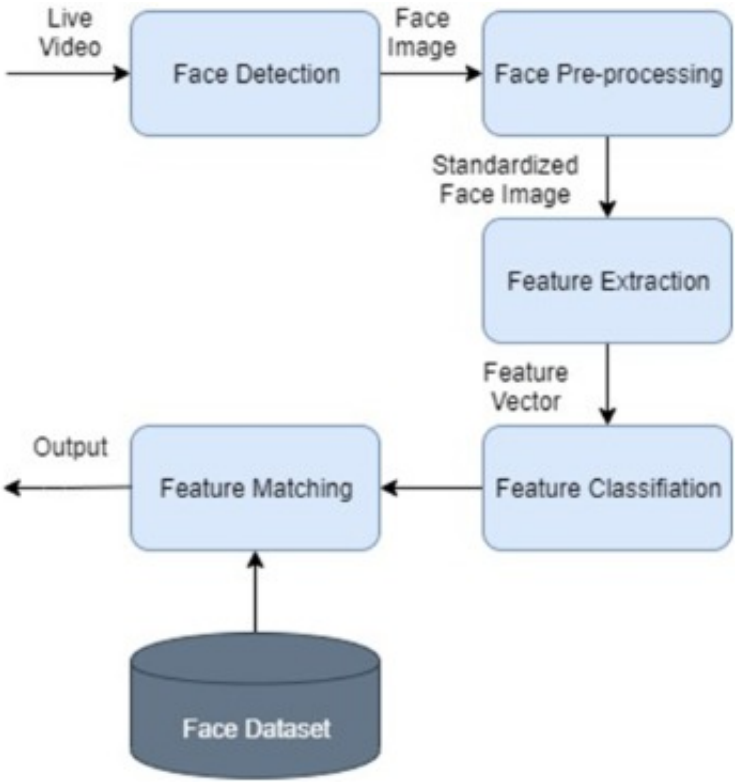

Figure 4.1: Face Detection Block Diagram

4.2 Vehicle Number Plate Detection

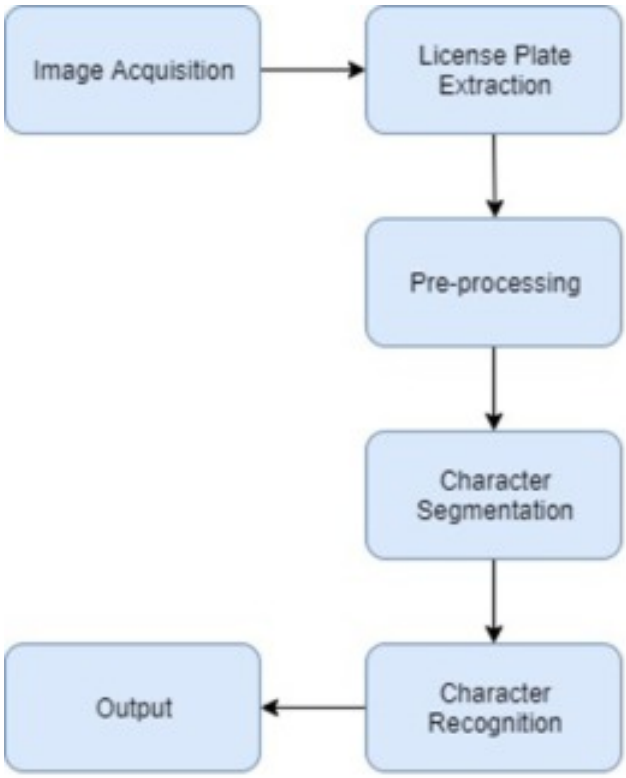

Figure 4.2: Vehicle Number Plate Detection Block Diagram

In the above Figure 4.1 and Figure 4.2, It is a proposed system architecture diagram for face detection and vehicle detection, here is a formal description and representation of the system. For face detection, live footage or video is taken as input and pre-processing is done on the captured face from image or video. Feature extraction and Feature classification are done on the pre-processed face. At last, Feature matching is done with face data set to get the final output.

And For Vehicle number plate detection, live footage or video is taken as input and license plate extraction is done on the captured image or video. Pre-processing and character segmentation are done on the extracted license plate. At last, Character recognition is applied to get the final output.

Chapter 5

Project Design

This Project Design module contains different UML diagrams for the representation of the systems.A UML diagram is a diagram based on the UML (Unified Modeling Language) with the purpose of visually representing a system along with its main actors, roles, actions, artifacts or classes, in order to better understand, alter, maintain, or document information about the system.

5.1 Usecase diagram for face detection

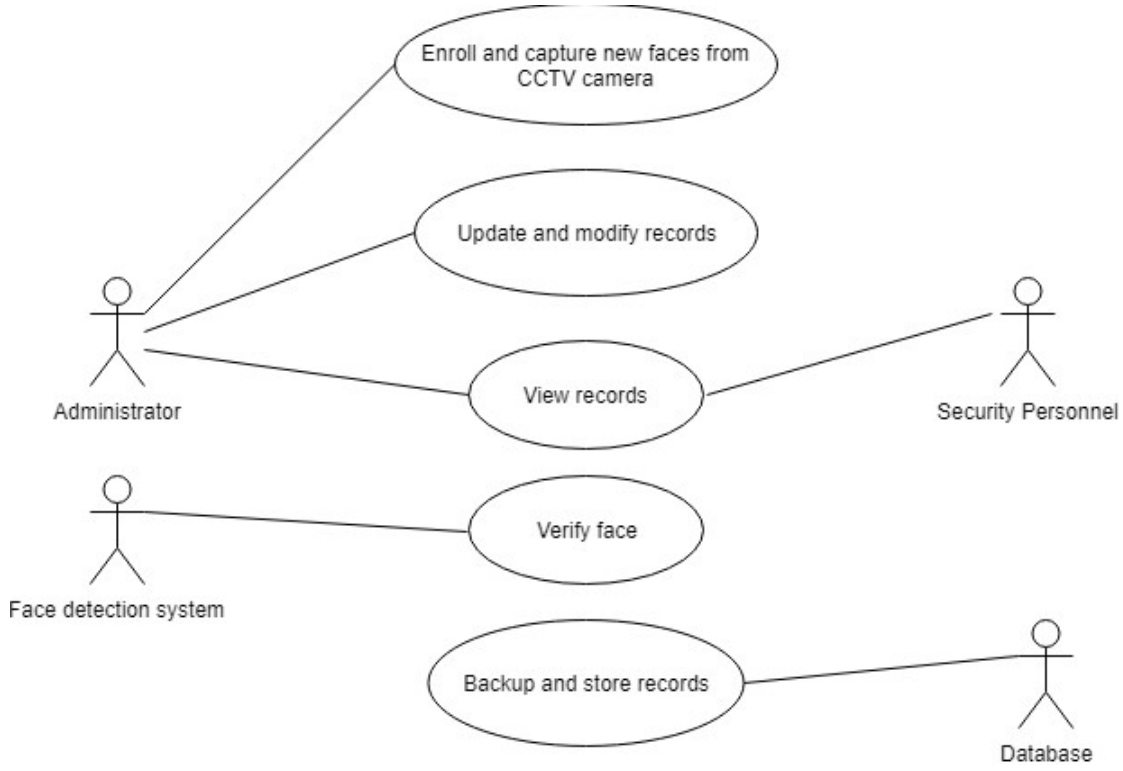

Figure 5.1: Usecase diagram for face detection

The above figure 5.1 represents the use-case diagram for face detection. The diagram is

used to model the system/subsystem of an application. The above use-case diagram consists of following actors like Administrator, Security personnel, Face detection system and database. The diagram also consists of use-cases like Enroll and capture new faces from CCTV cameras, Update and modify records, view records, verify face, backup and store records.

5.2 Usecase diagram for vehicle number plate detection

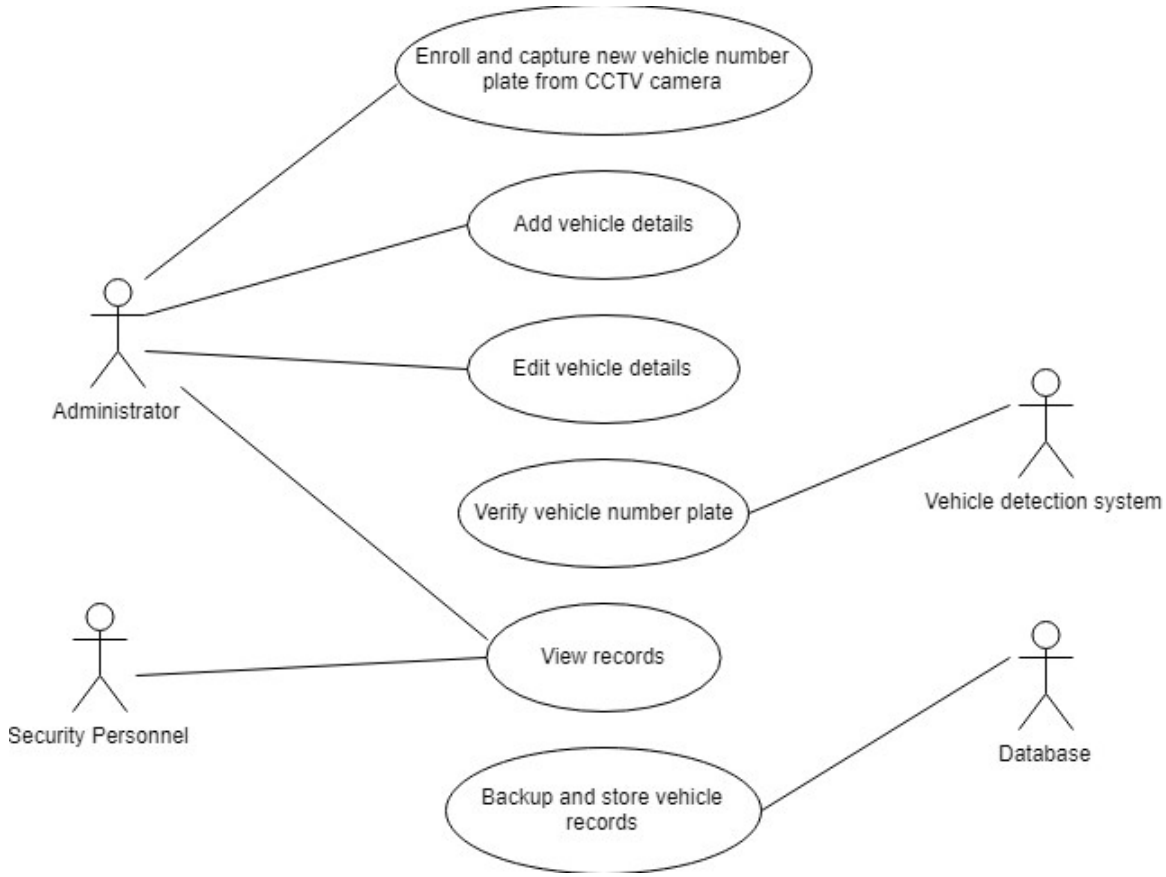

Figure 5.2: Usecase diagram for vehicle number plate detection

The above figure 5.2 represents the use-case diagram for vehicle number plate detection. The diagram is used to model the system/subsystem of an application. The above use-case diagram consists of following actors like Administrator, Security personnel, Vehicle detection system and database. The diagram also consists of use-cases like Enroll and capture new vehicle number plate from CCTV cameras, Add vehicle details, Edit vehicle details, Verify vehicle number plate, View records, Backup and store vehicle details or records.

5.3 Activity diagram for face detection

Activity diagrams are graphical representations of workflows of step-wise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes, as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores.

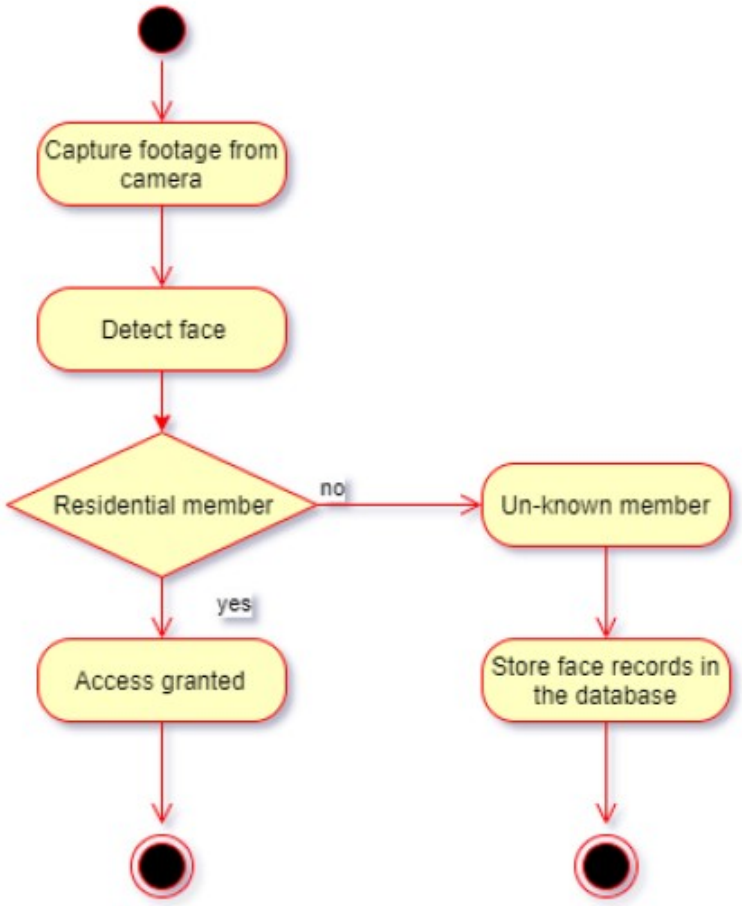

Figure 5.3: Activity diagram for face detection

The above figure 5.3 represents the activity diagram for face detection. The activity diagram is another important diagram in UML to describe the dynamic aspects of the system. The activity diagram is a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all types of flow control by using different elements such as fork, join, etc. The above activity diagram consists of following activities like Capture

footage from the camera, Detect face, Un-known member, Store face records in the database, and Access granted. The diagram also consists of one condition which is Residential member.

5.4 Activity diagram for vehicle number plate detection

The above figure 5.4 represents the activity diagram for Vehicle number plate detection.The above activity diagram consists of the following activities like Capture footage from the camera, Detect vehicle, Detect vehicle number plate, Un-known vehicle detected, Store vehicle records in the database, and Access granted.

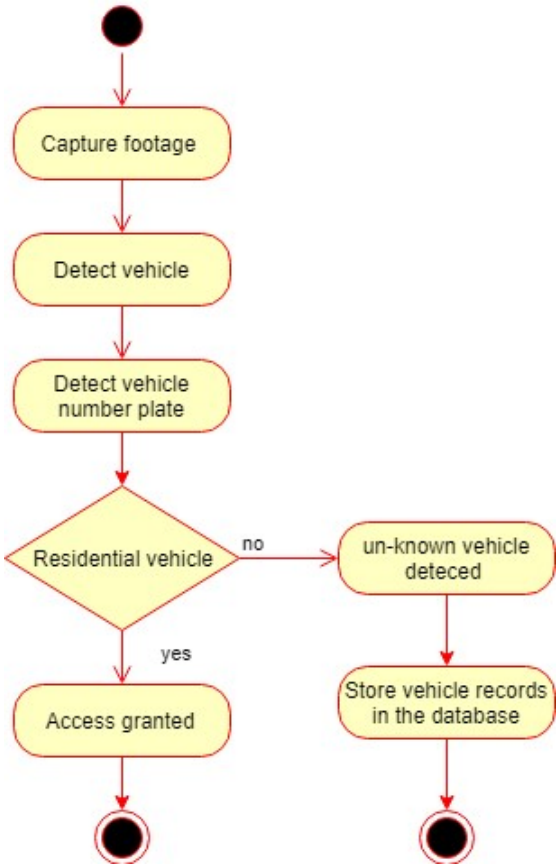

Figure 5.4: Activity diagram for vehicle number plate detection

The diagram also consists of one condition which is Residential vehicle.The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

5.5 Activity diagram for imaginary line

The above figure 5.5 represents the activity diagram for the imaginary line. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part. It does not show any message flow from one activity to another.

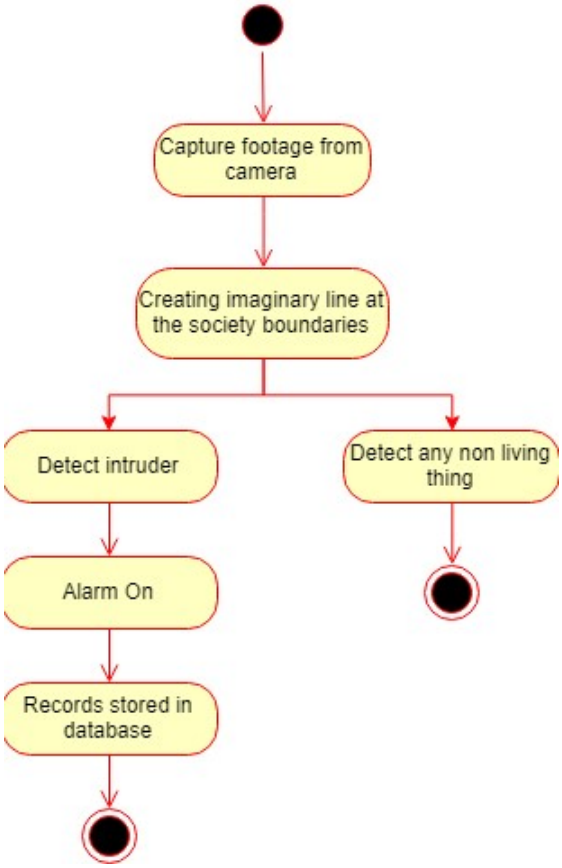

Figure 5.5: Activity diagram for imaginary line

The activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single. The above activity diagram consists of the following activities like Capture footage from the camera, Creating an imaginary line at the society boundaries, Detect intruder, Detect any non-living things, Alarm ON, and records store in the database.

5.6 Class diagram

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of object oriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as a structural diagram.

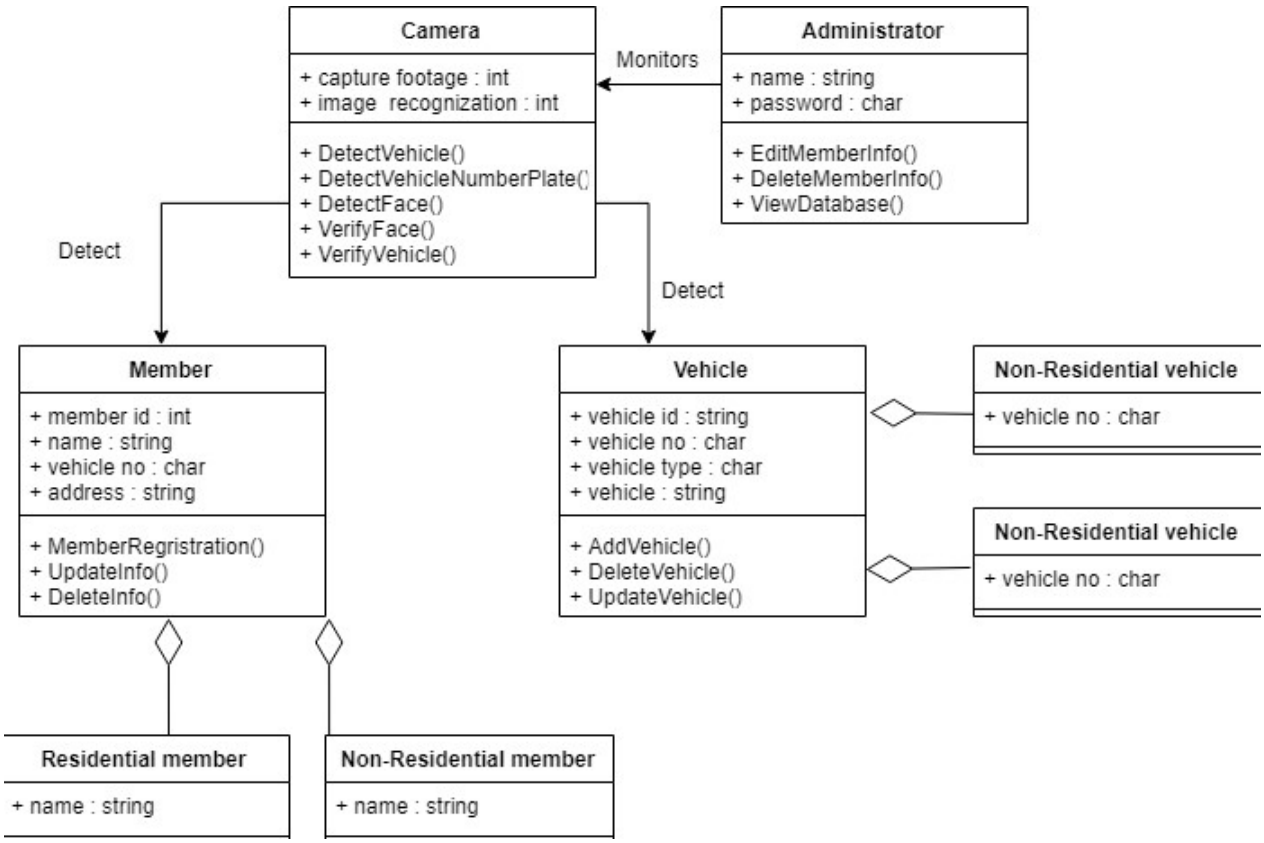

Figure 5.6: Class diagram

The above figure 6.1 represents the class diagram, A class diagram in our system helps in understanding various different modules that are functioning such as camera, Administrator, Member, Vehicle. Class diagram not only provide visual overview of project instead it helps in understanding various operations such as DetectVehicle(), DetectVehicleNumberPlate(), DetectFace(), etc. that a particular module performs. It also helps in understanding various attributes(such as Camera footage, image recognition value etc) involved in the project.

Chapter 6

Project Implementation

The project is divided into four modules face detection, vehicle number plate detection, object detection and imaginary line/virtual line. The Object Detection Module can detect common objects like a chair, mug, person, etc with an accuracy of 82% after being trained on a set of only 60 images of each object. The face recognition module can detect faces by using embedding's of faces such as nose, eyes, ears, and eyebrows with an accuracy of 80% after being trained on a single image. The vehicle number plate detection module can recognize number plate characters with an accuracy of 75% to 80% after being trained on a single image.

6.1 Face detection module output

Figure 6.1: Output for the Face detection module

In the above figure 6.1, we have used OpenCV3 and KNN classifier to perform face recognition. To build our face recognition module, we have first performed face detection, extract face embedding's from each face using deep learning, train a face recognition model on the embedding's, and then finally recognize faces in both images and video streams with OpenCV3.

6.2 Vehicle number plate detection module output

6.2.1 Original Image

Figure 6.2: Original Image

6.2.2 The output for the vehicle detection module

Figure 6.3: Final output of vehicle number plate recognition

Figure 6.2, was the original image taken as input. we have used ANPR/ALPR is an open-source Automatic License Plate Recognition library to perform vehicle number plate recognition. The library analyses images and video streams to identify license plates. The output is the text representation of any license plate characters recognized in the processed image which was shown in figure 6.3.

6.3 Object detection module output

In the above figure 6.4, we have used YOLOv3 model for object detection and human detection. The object detection module is a part of imaginary line/virtual line module. Object

detection is a computer vision task that involves both localizing one or more objects within an image and classifies each object in the image. Our model can detects common objects like cup,chair,phone,etc.

Figure 6.4: Output for the Object and Human detection module

6.4 Imaginary line/Virtual line module output

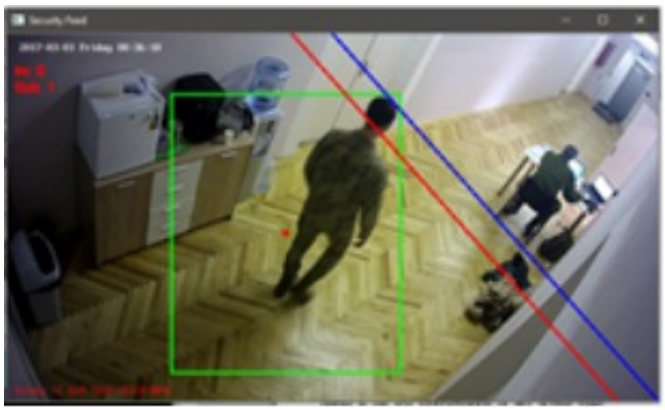

Figure 6.5: Output for the Imaginary line/Virtual line module

In the above figure 6.5, we have used opencv2 to detect the person and python math function to create the blue and red line which was represented as the imaginary line/virtual line. The system can also detect and count how many people cross the imaginary line/virtual line.

Chapter 7

Testing

Testing is the process of evaluating a system or its components with the intent to find whether it satisfies the specified requirements or not. In simple words, testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

Black-box testing is a method of software testing that examines the functionality of an application based on the specifications. It is also known as Specifications based testing. The technique of testing without having any knowledge of the interior workings of the application is called black-box testing. The tester is oblivious to the system architecture and does not have access to the source code. Typically, while performing a black-box test, a tester will interact with the system's user interface by providing inputs and examining outputs without knowing how and where the inputs are worked upon.Independent Testing Team usually performs this type of testing during the software testing life cycle.This method of test can be applied to each and every level of software testing such as unit, integration, system and acceptance testing.

The following represents some of the techniques which could be used to perform blackbox testing on our system:

- Functional Testing
  This is a type of black-box testing that is based on the specifications of the software that is to be tested. The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for.

- Non-Functional Testing
  This section is based upon testing an application from its non-functional attributes. Non-functional testing involves testing a software from the requirements which are nonfunctional in nature but important such as performance, security, user interface, etc.

- Regression Testing
  Whenever a change in a software application is made, it is quite possible that other areas within the application have been affected by this change. Regression testing is performed to verify that a fixed bug hasn't resulted in another functionality or business rule violation. The intent of regression testing is to ensure that a change, such as a bug fix should not result in another fault being uncovered in the application.

Functional, Non-Functional and Regression Testing are the testing methods because of which Blackbox Testing becomes best suited for this project, Since performance testing along with spotting a bug and fixing it, is the primary requirement of this project.

In our system, higher testing preference is given to end user experience rather than internal code implementation. In this project, testing at the end user side is more important i.e. how system functions and show its effectiveness after being implanted on residential societies. Thus Blackbox testing is the best suited testing method for managing our system. blackbox testing checks the functionality of system without checking internal code structure, implementation detail or internal paths of the software. This testing is purely based only on software requirement and specifications. It only check whether correct input (i.e. residential members face details/residential member vehicle details) is provided to the system and correspondingly the system provide correct output (i.e. detect the person is residential or not/detect the vehicle is residential or not).

Chapter 8

Conclusion and Future Scope

In our work, we propose a system for vehicle detection and tracking in real-time. Our system is also used to detect the human face in real-time. This proposed system can be applied in the fields of the society security system and also in the smart vehicle parking system. However, there are limitations in our system. Our system struggles to detect human faces covered with any object such as cloth or any type of mask. Future works will contain the improvement of that.

Bibliography

[1] Gu, Xiao-Feng, et al. "Real-Time vehicle detection and tracking using deep neural networks." 2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP). IEEE, 2016.

[2] Babbar, Sarthak, et al. "A New Approach for Vehicle Number Plate Detection." 2018 Eleventh International Conference on Contemporary Computing (IC3). IEEE, 2018.

[3] Shi, Kaijing, Hong Bao, and Nan Ma. "Forward vehicle detection based on incremental learning and fast r-cnn." 2017 13th International Conference on Computational Intelligence and Security (CIS). IEEE, 2017.

[4] Lee, Sanghyeop, et al. "Video Based License Plate Recognition of Moving Vehicles Using Convolutional Neural Network." 2018 18th International Conferenceon Control, Automation and Systems (ICCAS). IEEE, 2018.

[5] Miller, Nicholas, Mohan A. Thomas, Justin A. Eichel, and Akshaya Mishra. "A hidden markov model for vehicle detection and counting." In 2015 12th Conference on Computer and Robot Vision, pp. 269-276. IEEE, 2015.

Appendices

Appendix-A: Download and Installation of the ML Components

To run "ML Enabled Surveillance System for Societies project", one needs to have the following software components like packages/libraries of machine learning in their local system. The following components are Python, Pillow, Anaconda, TKinter, OpenCV2, ANPR, Mysql, etc.

1.Download and Installation of Python
Download python for ubuntu:
https://www.python.org/downloads/release/python-382/

Install python :
$ sudo apt-get update
$ sudo apt-get install python

2.Install pillow
$ sudo apt install python3-pil

3.Download and Installation of Anaconda
Download Anaconda for ubuntu:
https://repo.anaconda.com/archive/Anaconda3-2020.02-Linux-x86$_6$4.*sh*

Install Anaconda :
$ sha256sum /path/filename
$*bash* /*Downloads*/*Anaconda*3 − 2020.02 − *Linux* − *x*86$_6$4.*sh*

4.Install TKinter
$ sudo apt-get install python3-tk

5.Download OpenCV and OpenCV contrib
Download opencv from Github
git clone *https* : //*github.com*/*opencv*/*opencv.git*
cd opencv
git checkout 3.3.1
cd ..

5.1 Compile and install OpenCV with contrib modules
Create a build directory:
cd opencv
mkdir build
cd build

5.2 Run CMake

5.3 Compile and Install:
nproc
make -j4
sudo make install
$sudosh - c'echo"/usr/local/lib" >> /etc/ld.so.conf.d/opencv.conf'$
sudo ldconfig
5.4 Create symlink in virtual environment
$find/usr/local/lib/ - typef - name"cv2 * .so"$

6.Install MySQL
$ sudo apt install mysql-server
To check whether the MySQL server is running:
$ sudo systemctl status mysql

Appendix-B: Sample Code

Sample code for the administrator login page. Where the administrator can log in and have the right to add, delete or modify the residential member's details and add, delete or modify residential member's vehicle details.

```
from tkinter import *
import tkinter.messagebox as tm
import sys
import os
import tkinter

class LoginFrame(Frame):
        def _init_(self, master):
            super()._init_(master)
            self.label_username = Label(self, text="Username")
            self.label_password = Label(self, text="Password")
            self.entry_username = Entry(self)
            self.entry_password = Entry(self, show="*")

            self.label_username.grid(row=0, sticky=E)
            self.label_password.grid(row=1, sticky=E)
            self.entry_username.grid(row=0, column=1)
            self.entry_password.grid(row=1, column=1)
            self.logbtn = Button(self, text="Login", command=self._login_btn_clicked)
            self.logbtn.grid(columnspan=2)

            self.pack()

        def _login_btn_clicked(self):
            # print("Clicked")
            username = self.entry_username.get()
            password = self.entry_password.get()

            # print(username, password)

            if username == "admin" and password == "admin":
                top=tkinter.Tk()
                top.geometry("300x300")

                def helloCallBack_1():
                    os.system('python face.py')
                def helloCallBack_2():
                    os.system('python Main.py')
                def helloCallBack_3():
                    os.system('python face.py')
```

```
                B=tkinter.Button(top,text="face Detection ",command= helloCallBack_1)
                B.pack()
                B=tkinter.Button(top,text="Car Number Plate Detection ",command= hel-
loCallBack_2)
                B=tkinter.Button(top,text="Boundaries ",command= helloCallBack_3)
                B.pack()
                top.mainloop()
        else:
                tm.showerror("Login error", "Incorrect username")

root = Tk()
root.geometry("300x300")
lf = LoginFrame(root)
root.mainloop()
```

Acknowledgement

We have great pleasure in presenting the report on **"ML Enabled Surveillance System for Societies".** We take this opportunity to express our sincere thanks towards our guide **Dr. Sameer Nanivadekar** & Co-Guide **Mr. Vishal S. Badgujar** Department of IT, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.
We thank **Mr. Kiran B. Deshpande** Head of Department,IT, APSIT for his encouragement during progress meeting and providing guidelines to write this report.
We thank **Mr.Vishal S. Badgujar** BE project co-ordinator, Department of IT, APSIT for being encouraging throughout the course and for guidance.
We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Pranav Chauhan
17204004

Sachin Gupta
17204001

Rohit Arava
17204017

Publication

Paper entitled **"ML Enabled Surveillance System for Societies"** is presented at **"2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)"** by **"Mr. Pranav Chauhan","Mr. Sachin Gupta","Mr. Rohit Arava","Dr. Sameer Nanivadekar","Mr. Vishal Badgujar".**
https://ieeexplore.ieee.org/document/9077746