## VIDEO BASED SURVEILLANCE SYSTEM AND

**PATH PREDICTION**

**A PROJECT REPORT**

***Submitted by***

### P.POOJA [REGISTER NO:211414104187] S.OMEZHILE [REGISTER NO:211414104175]

***in partial fulfillment for the award of the degree of***

### BACHELOR OF ENGINEERING

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.**

### ANNA UNIVERSITY: CHENNAI 600 025

**APRIL 2021**

### BONAFIDE CERTIFICATE

Certified that this project report **“VIDEO BASED SURVEILLANCE SYSTEM AND PATH PREDICTION ”** is the bonafide work of “P.POOJA (211417104187) and S.OMEZHILE (211417104175)” who carried out the project work under my supervision.

**SIGNATURE SIGNATURE**

**Dr.S.MURUGAVALLI,M.E.,Ph.D., MS.V.SATHIYA HEAD OF THE DEPARTMENT SUPERVISOR**

**ASSISTANT PROFESSOR**

DEPARTMENT OF CSE, DEPARTMENT OF CSE,

PANIMALAR ENGINEERING COLLEGE, PANIMALAR ENGINEERING COLLEGE, NAZARATHPETTAI, NAZARATHPETTAI,

POONAMALLEE, POONAMALLEE,

CHENNAI-600 123. CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voce Examination held on...........................

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like express our deep sense of gratitude and sincere thanks to our Directors **Tmt.C.VIJAYA RAJESWARI, Dr.C.SAKTHIKUMAR, M.E.,**

**Ph.D.,** and **Tmt. SARANYASREE SAKTHIKUMAR B.E.,M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. S.MURUGAVALLI , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my **Project Guide Ms.V.SATHIYA** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

P.POOJA S.OMEHILE

### TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT** | iii |
|  | **LIST OF SYMBOLS, ABBREVIATIONS** | ix |
| **1.** | **INTRODUCTION** | 8 |
|  | 1.1 Overview | 8 |
|  | 1.2 Problem Definition | 8 |
| **2.** | **LITERATURE SURVEY** | 10 |
| **3.** | **SYSTEM ANALYSIS** | 14 |
|  | 2.1 Existing System | 14 |
|  | 2.2 Proposed system | 14 |
|  | 2.3 Requirement Analysis and Specification | 15 |
|  | 2.3.1 Input Requirements | 15 |
|  | 2.3.2 Output Requirements | 15 |
|  | 2.3.3 Functional Requirements | 15 |
|  | 2.4 Technology Stack | 16 |

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
| **4.** | **SYSTEM DESIGN** | 18 |
|  | 4.1. ER diagram | 18 |
|  | 4.2 Data dictionary | 19 |
|  | 4.3 UML Diagrams | 20 |
| **5.** | **SYSTEM ARCHITECTURE** | 22 |
|  | 5.1 Architecture Overview | 23 |
|  | 5.2 Program Design Language | 24 |
| **6.** | **SYSTEM IMPLEMENTATION** | 28 |
|  | 6.1 Client-side coding | 28 |
|  | 6.2 Server-side coding | 30 |
| **7.** | **SYSTEM TESTING** | 40 |
|  | 7.1 Unit Testing | 40 |
|  | 7.2 Integration Testing | 41 |
| **8.** | **CONCLUSION** | 43 |
|  | 8.1 Conclusion and Future Enhancements | 43 |
|  | **APPENDICES** | 44 |
|  | A.1 Sample Screens | 46 |
|  | A.2 Publications | 49 |

**ABSTRACT**

As an important branch of Security and surveillance systems, face recognition technology has the characteristics of convenient acquisition and high reliability that is widely used in the fields of information security, national security, traffic monitoring, security cameras and organisations like school, college.

This project is to Detect Unknown or specified given person(criminal) in Surveillance camera at Traffic signals and predicting his probability path further then reporting to their nearby controlled rooms

# CHAPTER 1

**1 INTRODUCTION**

**1.1 OVERVIEW**

In this project , we first define visual tracking of a particular person, say criminal. Subsequently we present another background algorithm for face recognition.

Once the criminal is detected in the camera an alert message is passed to all the nearby control rooms. And even if the criminal tries to escape he cannot, because his path is also predicted in our project .

* 1. **PROBLEM DEFINITION**

Anti-theft Security Systems need the use of software parts . Firstly, the software part will be used as Telegram.

System will start surveillance function and capture photos sent to administrator to notify him with Telegram when a stranger tries to enter.And later it predicts the direction of the thief and send another alert to the administrator

**CHAPTER 2**

# LITERATURE SURVEY

### LITERATURE SURVEY



**Title :** Human Action Recognition and Prediction: A Survey

**Authors :** Yu Kong, Yun Fu

**Abstract :** Derived from rapid advances in computer vision and machine learning, video analysis tasks have been moving from inferring the present state to predicting the future state. Vision-based action recognition and prediction from videos are such tasks, where action recognition is to infer human actions (present state) based upon complete action executions, and action prediction to predict human actions (future state) based upon incomplete action executions. These two tasks have become particularly prevalent topics recently because of their explosively emerging real-world applications, such as visual surveillance, autonomous driving vehicle, entertainment, and video retrieval, etc.

**Title :** Video-Based Motion Trajectory Forecasting Method for Proactive Construction Safety Monitoring Systems

**Authors :** [Shuai Tang](https://ascelibrary.org/author/Tang%2C%2BShuai), [Mani Golparvar-Fard](https://ascelibrary.org/author/Golparvar-Fard%2C%2BMani), A.M.ASCE, [Milind Naphade](https://ascelibrary.org/author/Naphade%2C%2BMilind) and [Murali M. Gopalakrishna](https://ascelibrary.org/author/Gopalakrishna%2C%2BMurali%2BM)

**Abstract** : Falls, struck-bys, and caught-in/betweens are among the most common types of fatal accidents on construction sites. Despite their significance, the majority of today’s accident prevention programs react passively to situations in which workers or equipment enter predefined unsafe zones. To support systems that proactively prevent these accidents, this paper presents a path prediction model for workers and equipment. The model leverages the extracted video frames to predict upcoming worker and equipment motion trajectories on construction sites. Specifically, the model takes

two-dimensional (2D) tracks of workers and equipment from visual data—based on computer vision methods for detection and tracking—and uses a long short-term memory (LSTM) encoder-decoder followed by a mixture density network (MDN) to predict their locations. A multihead prediction module is introduced to predict locations at different future times. The method is validated on an existing dataset, TrajNet, and a new dataset of 105

high-definition videos recorded over 30 days from a real-world construction site.

**Title :** Video-Based Surveillance Systems

**Authors :** Paolo Remagnino, Graeme A. Jones, Nikos Paragios

**Abstract:** Monitoring of public and private sites has increasingly become a very sensitive issue resulting in a patchwork of privacy laws varying from country to country -though all aimed at protecting the privacy of the citizen. It is important to remember, however, that monitoring and visual surveillance capabilities can also be employed to aid the citizen. The focus of current development is primarily aimed at public and corporate safety applications including the monitoring of railway stations, airports, and inaccessible or dangerous environments.

**CHAPTER 3**

# SYSTEM ANALYSIS

### EXISTING SYSTEM

Based on the research found, there are some project works related to face recognition on security systems. Through a research paper “Web-based online embedded door access control and home security system based on face recognition” written by Sahani, M., Nanda, C., Sahu, A.K. and Pattnaik, B..

The strength and weakness of their product can be identified after analysis. The strength of their product is they used wireless network technique ZigBee based. The ZigBee module combine with electromagnetic door lock module to operate the door accessibility

### PROPOSED SYSTEM

The proposed system is designed with wireless access control so the lock module can be added easily if needed. Email and SMS are used to notify the house owner when a stranger faces. It helps to reduce the need of the server so the user can directly login and control the embedded system in real time. Users can control the system with SMS, email and website.

However, it still can be improve from it weaknesses

. The face recognition can be bypassed with a photo of the owner's face. The system can be improved with add on password authentication, sound recognition or fingerprint authentication. The product cost can be lower with reducing the SMS module and using the WIFI module as replacement. Since our phone always connected to internet and the latency should be lower if compare with GSM network

### REQUIREMENT ANALYSIS AND SPECIFICATION

* + 1. **INPUT REQUIREMENTS**
* System with 8GB RAM ,above i5(9th gen),with Graphic processor-NVIDIA.
* Visual studio C (to code)
* A dataset to encode and train

### OUTPUT REQUIREMENTS

* + Camera with 16MP

### FUNCTIONAL REQUIREMENTS

Step 1: Enroll and Encode the faces(common faces and specified person's face)in a Database

Step 2: Train the Given set of faces

Step 3: Then to detect that specified person among common people

Step 4: If it matches with the specified person ,It send alert notification to the master control room

Step 5: It send 5 sec of the captured video and along with the predicted map in Telegram

### TECHNOLOGY STACK

**HARDWARE:**

SURVEILLANCE CAMERA-To capture and detect the specified person MONITORING SCREEN-To monitor the movement of capturing video

### SOFTWARE:

★ Opencv to train images

★ Window 10-operating system

★ Python language

★ API to send messages

★ Algorithm-CNN

★ Telegram

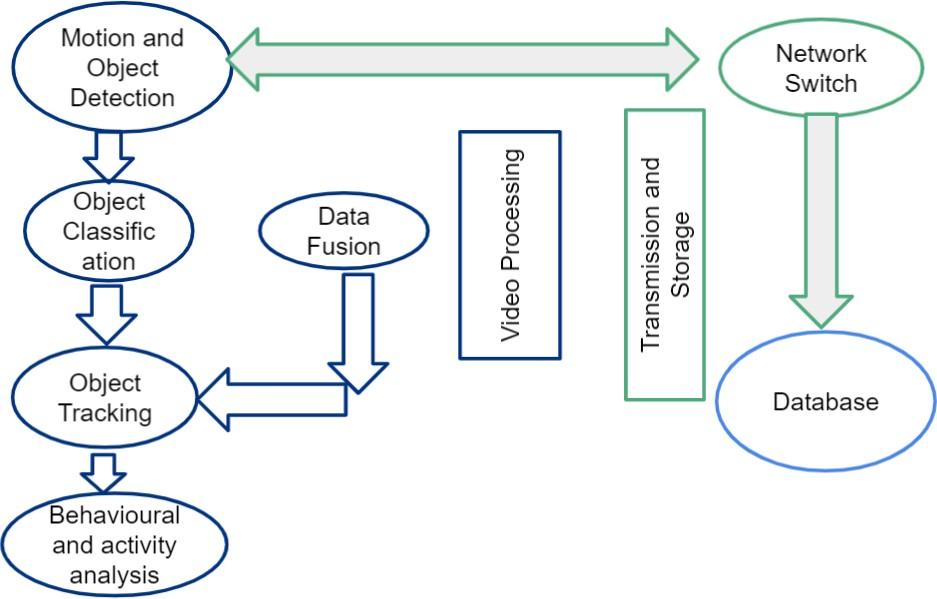
**CHAPTER 4**

**4.SYSTEM DESIGN**

### UML DIAGRAM

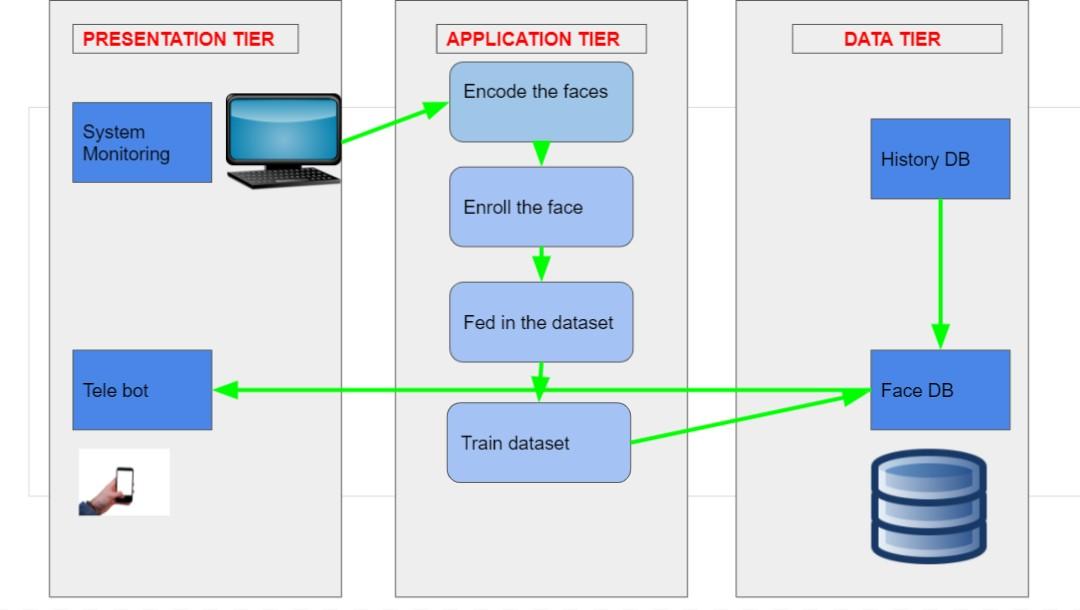


**4.2 WORKFLOW DIAGRAM**



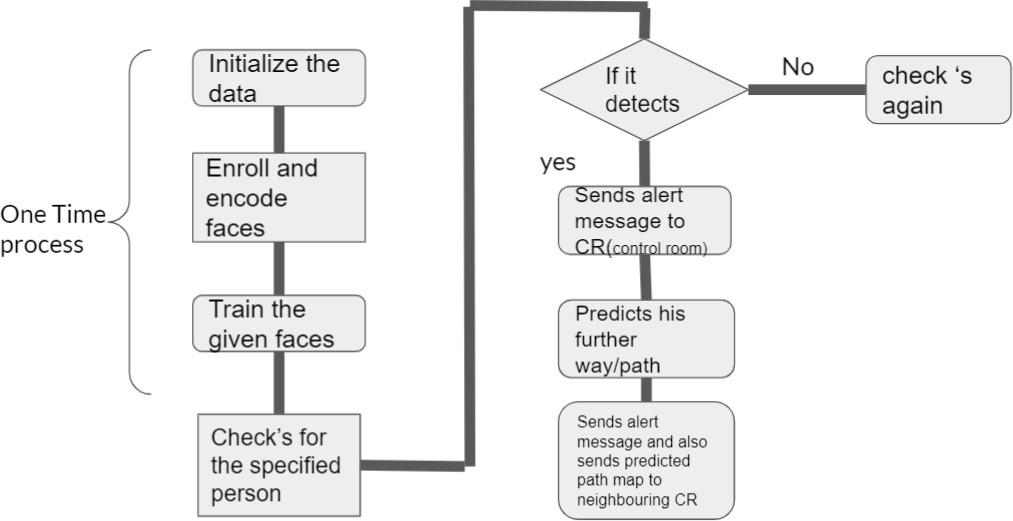
**CHAPTER 5**

1. **SYSTEM ARCHITECTURE**



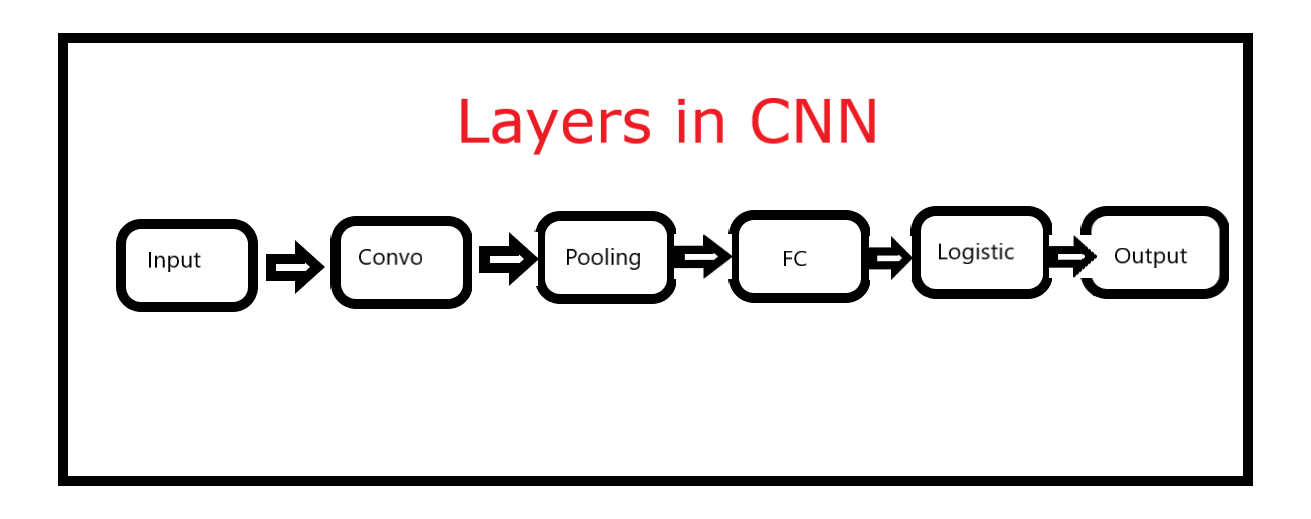


* 1. **ARCHITECTURE OVERVIEW**



### 5.3 PROGRAM DESIGN LANGUAGE

**A.TYPES OF LAYERS**

****

* 1. **Input layer**

Image data should be included in the CNN input layer. As we saw earlier, image data is represented by a three-dimensional matrix. You must resize it to fit into a single column. If you have a 28 x 28

= 784 image, you must convert it to 784 x 1 before feeding it into the input. If you have "m" training instances, the input dimension would be "m" (784, m).

* 1. **Convo Layer**

The Convo layer is also called the Feature Extractor Layer because it extracts features from the image. To begin, a portion of the image is connected to the Convo layer, which performs the convolution operation we saw earlier as well as calculating the dot product between the receptive field (a local region of the input image the same size as the filter) and the filter.The operation yields a single integer representing the output volume. Then we use a Stride to slide the filter over the next receptive field of the same input image and repeat the procedure. We'll keep repeating the process until we've gone through the entire image. The output would be the next layer's input.

ReLU activation is also present in the Convo layer, which reduces all negative values to zero.

* 1. **Pooling Layer**

A pooling layer is used to reduce the spatial volume of the input signal. It's used in the middle of two convolution layers. It would be computationally costly to apply FC after the Convo layer without using pooling or max pooling, which we do not want. As a result, the only way to reduce the spatial volume of the input image is to use maximum pooling.

You can observe the 4 x 4 dimension input is reduced to 2 x 2 dimension.

The pooling layer has no parameters, but it does have two hyperparameters: Filter(F) and Stride (S). In general, if we have W1 x H1 x D1 as input dimensions, then

W2 = (W1−F)/S+1 H2 = (H1−F)/S+1 D2 = D1

W2, H2, and D2 are the output width, height, and depth, respectively.

* 1. **Fully Connected Layer**

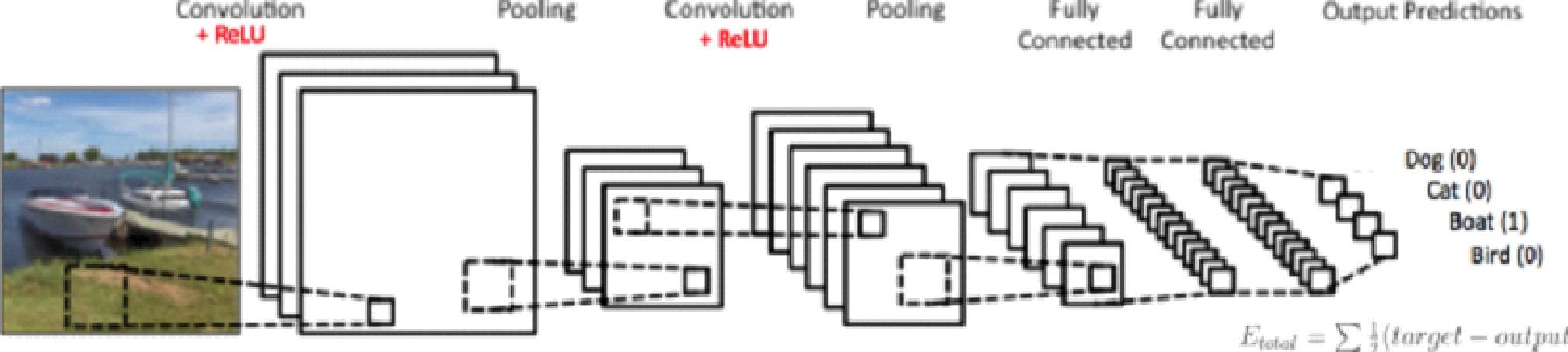
Weights, biases, and nerves are all part of the completely connected layer. It binds neurons from one layer to those from another. It is used to train people to identify images into various categories.

* 1. **Logistic Layer**

The last layer of CNN is the logistic layer. It is located at the bottom of the FC layer. Softmax is used for multi-classification and logistic is used for binary classification.

* 1. **Output Layer**

Output layer contains the label which is in the form of one-hot encoded.Now you have a good understanding of CNN. Let’s implement a CNN in Keras.



Feature Extraction from Image Classification

**CHAPTER 6**

**6 SYSTEM IMPLEMENTATION**

### 6.1SERVER SIDE CODING

**INITIALIZE THE DATA**

import cv2

cam = cv2.VideoCapture(0) cam.set(3, 640) # set video width cam.set(4, 480) # set video height

face\_detector = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml') # For each person, enter one numeric face id

face\_id = input('\n enter user id end press <return> ==> ')

print("\n [INFO] Initializing...")

# Initialize individual sampling face count count = 0

while(True):

ret, img = cam.read()

img = cv2.flip(img, 1) # flip video image vertically gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY) faces = face\_detector.detectMultiScale(gray, 1.3, 5)

for (x,y,w,h) in faces:

cv2.rectangle(img, (x,y), (x+w,y+h), (255,0,0), 2)

count += 1

# Save the captured image into the datasets folder cv2.imwrite("dataset/Sampled Image." + str(face\_id) + '.' + str(count) +

".jpg", gray[y:y+h,x:x+w])

cv2.imshow('image', img)

k = cv2.waitKey(1) & 0xff # Press 'ESC' for exiting video if k == 27:

break

elif count >= 30: # Take 30 face sample and stop video break

# Do a bit of cleanup

print("\n [INFO] Image Captured") cam.release() cv2.destroyAllWindows()

### FACE TRAINING

''''

Training Multiple Faces stored on a DataBase:

==> Each face should have a unique numeric integer ID as 1, 2,,

==> LBPH computed model will be saved on trainer/ directory. (if it does not exist, pls create one)

'''

import cv2

import numpy as np from PIL import Image import os

# Path for face image database path = 'dataset'

recognizer = cv2.face.LBPHFaceRecognizer\_create()

detector = cv2.CascadeClassifier("haarcascade\_frontalface\_default.xml");

# function to get the images and label data def getImagesAndLabels(path):

imagePaths = [os.path.join(path,f) for f in os.listdir(path)] faceSamples=[]

ids = []

for imagePath in imagePaths:

PIL\_img = Image.open(imagePath).convert('L') # convert it to grayscale img\_numpy = np.array(PIL\_img,'uint8')

id = int(os.path.split(imagePath)[-1].split(".")[1]) faces = detector.detectMultiScale(img\_numpy)

for (x,y,w,h) in faces: faceSamples.append(img\_numpy[y:y+h,x:x+w]) ids.append(id)

return faceSamples,ids

print ("\n [INFO] Training faces. It will take a few seconds. Wait ...") faces,ids = getImagesAndLabels(path)

recognizer.train(faces, np.array(ids))

# Save the model into trainer/trainer.yml

recognizer.write('trainer/trainer.yml') # recognizer.save() worked on Mac, but not on Pi

# Print the number of faces trained and end program print("\n [INFO] {0} faces trained. Exiting Program".format(len(np.unique(ids))))

### FACE RECOGNITION

''''

Real Time Face Recognition

==> Each face stored on dataset/ dir, should have a unique numeric integer ID as 1, 2,,

==> LBPH computed model (trained faces) should be on trainer/ dir

'''

import cv2

recognizer = cv2.face.LBPHFaceRecognizer\_create() recognizer.read('trainer/trainer.yml')

cascadePath = "haarcascade\_frontalface\_default.xml" faceCascade = cv2.CascadeClassifier(cascadePath);

font = cv2.FONT\_HERSHEY\_SIMPLEX

#initiate id counter id = 0

# names related to ids: example ==> Marcelo: id=1, etc names = ['None', '1', '2']

# Initialize and start realtime video capture cam = cv2.VideoCapture(0)

cam.set(3, 640) # set video width cam.set(4, 480) # set video height

# Define min window size to be recognized as a face minW = 0.1\*cam.get(3)

minH = 0.1\*cam.get(4) while True:

ret, img =cam.read()

img = cv2.flip(img, 1) # Flip vertically

gray = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY) faces = faceCascade.detectMultiScale(

gray,

scaleFactor = 1.2,

minNeighbors = 5,

minSize = (int(minW), int(minH)),

)

for(x,y,w,h) in faces:

cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)

id, confidence = recognizer.predict(gray[y:y+h,x:x+w])

# Check if confidence is less than 100 ==> "0" is perfect match conf=round(100 - confidence)

if (conf > 50):

id = names[id]

confidence = " {0}%".format(conf) else:

id = "unknown"

confidence = " {0}%".format(conf)

cv2.putText(img, str(id), (x+5,y-5), font, 1, (255,255,255), 2)

cv2.putText(img, str(confidence), (x+5,y+h-5), font, 1, (255,255,0), 1) cv2.imshow('camera',img)

k = cv2.waitKey(1) & 0xff # Press 'ESC' for exiting video if k == 27:

break

# Do a bit of cleanup

print("\n [INFO] Exiting Program and cleanup stuff") cam.release()

cv2.destroyAllWindows()

### TO START THE CAMERA

import cv2

cap = cv2.VideoCapture(0)

cap.set(3, 640) # set video width cap.set(4, 480) # set video height

while(1):

#read the frame

ret, frame = cap.read() if ret==True:

frame = cv2.flip(frame,0) #show the frame cv2.imshow('frame',frame)

k = cv2.waitKey(1) & 0xff # Press 'ESC' for exiting video if k == 27:

break

else:

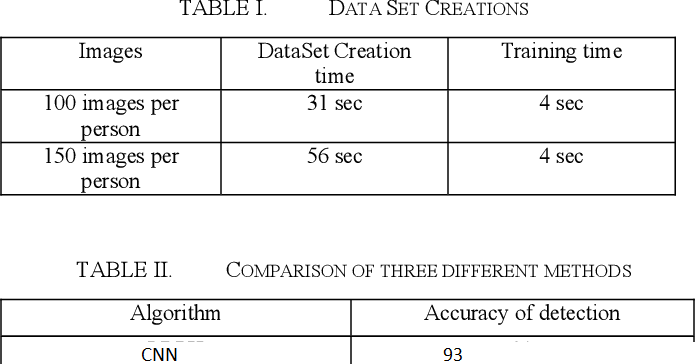
break

# Release everything if job is finished cap.release() cv2.destroyAllWindows()

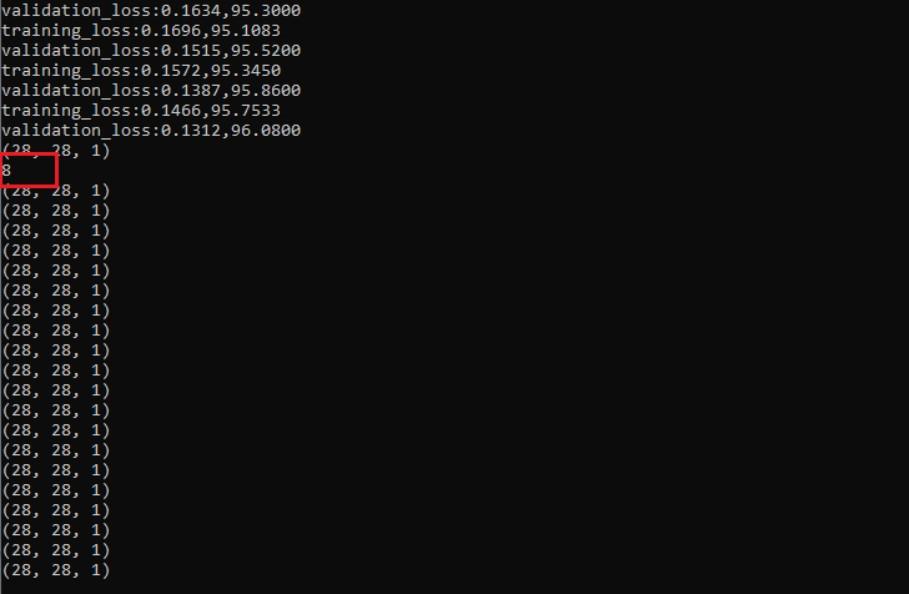
**CHAPTER 7**

1. **SYSTEM TESTING**

### 7.1 UNIT TESTING



* 1. **INTEGRATION TESTING**



**CHAPTER 8**

## CONCLUSION

### 8.1 CONCLUSION AND FUTURE ENHANCEMENTS CONCLUSION

This project is used to implement face recognition. Face recognition nowadays has been widely used in many areas especially on security. The house security can be improved with the implementation of this product. It is designed with low cost and efficient material. The improvement of technology has made the internet of things no longer an expensive stuff and it can be modified and customized depending on our needs.

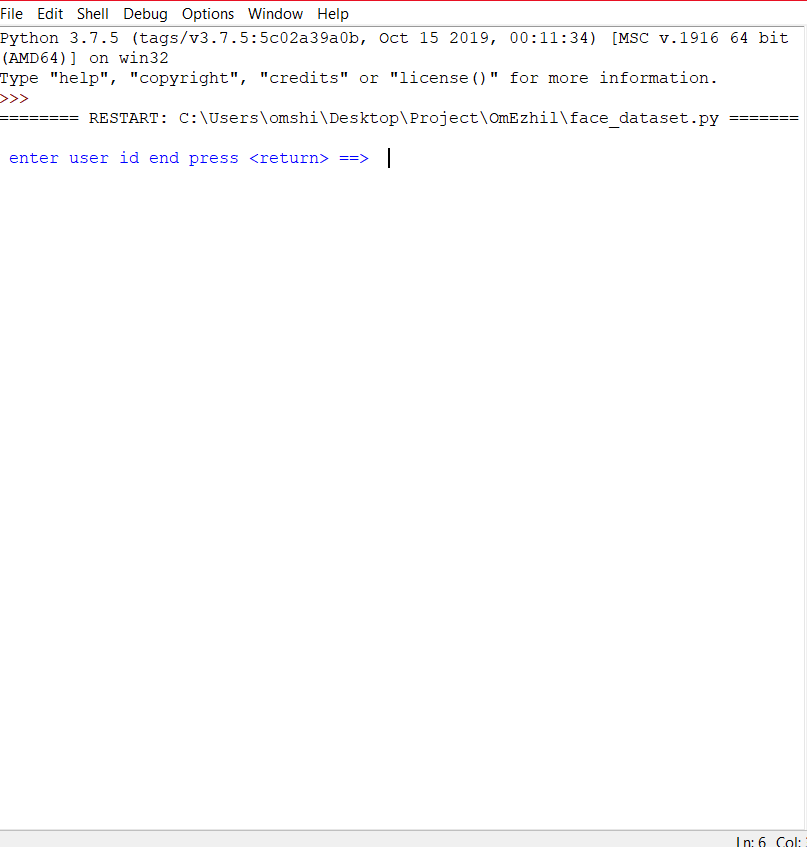
### FUTURE ENHANCEMENTS

The future enhancement of this project is to develop the hardware module to be used in the camera which provides extra efficiency and precision scale to detect the thief with less effort and time.

And also to use a tribrid algorithm for better extraction and high efficiency which works only on the deep scale extraction and layer conversion.

**APPENDICES**

### SAMPLE SCREENS





35

3%

37

36

39

40

41

42

43

44

45

4%

47

46

49

52

53

54

57

%2

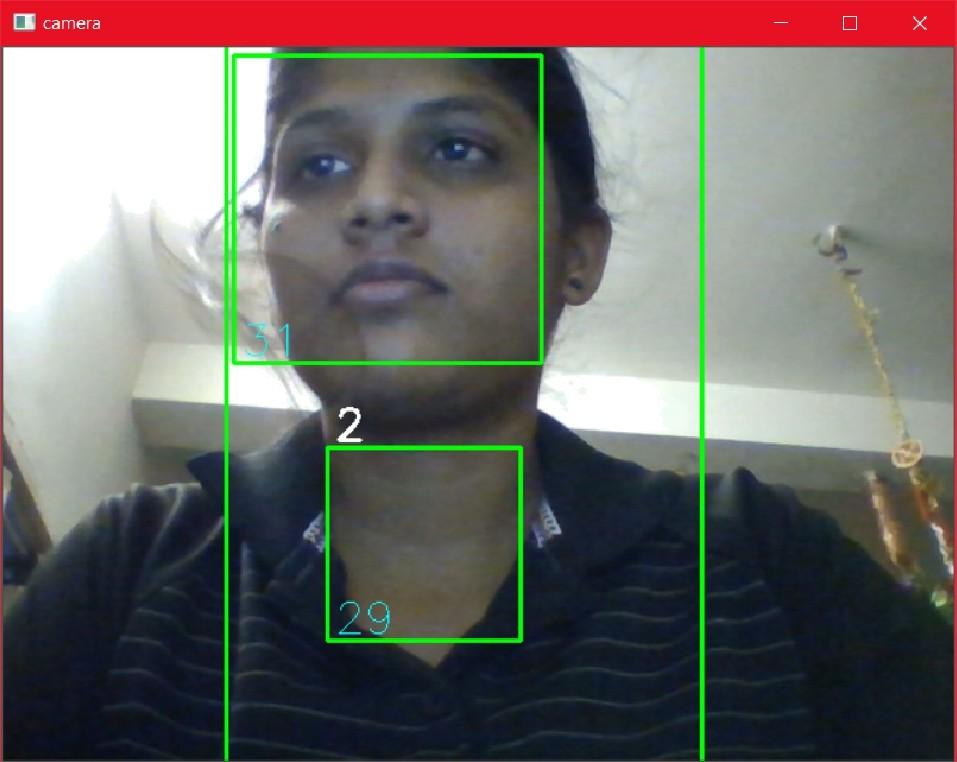
%3

%4

67

[INFO] Exiting Program and cleanup stuff





INr04 and cleanup stuff





surveillance system,is Active Now

Surveillance system Tracked Something!!!

Hold on please for 10 sec Recoding Completed

Uploading video

Predicting path please wait Predicted Path person Moving Left

direction

### PUBLICATIONS

Video based surveillance system and path prediction Published in : Journal of emerging technologies and innovative research

[video based surveillance system and path prediction - Journal of …](https://www.jetir.org/view?paper=JETIR2105831)

**REFERENCE FOR PUBLICATION**





A. 4. F u II y Co nnec Ccd L ay er

A. 3. Pos lin g La ye r

A. S. Log ‹sï ic La ye r

A. 6.0 u ï pu t La yer





1. TELEGRAM 4 BOT 

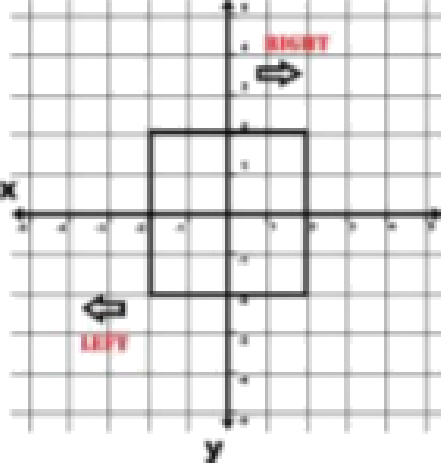


\*j TcMgram If<a APJ: Ttuz 7cfegzaoz Ben .4Pt is a

Tckgraos £I4as izsiz g jxograzzauog Maguag<s like 3as x, fi L”- —.. TTx• Tctcgzwn B<K dve

A•h tbc rules Inn ” uW ottB•z

J ] Telegram bat- A ToJzgram bol is a <orzyIuter

1. PATH PREDICTION 4.5ystem Architecture





S.CONCLUSlOLt



tic hnolu«ts hJ> mzJc the interns-i ol” thtnu> n-u Iun¿•cr use c xpczt>iY”c Ihtne: azuJ ii



Jcpcnr1iJiq us uuz nccsJ».

”I”hc .r.i i.ri. iI rcpun ,a iJ fi "L ol” the IheJI a lñund u Uh the help ul” L”t"J”\” JJxJ



”I”h i» Pn›|>tr+cJ J dcc uscJ tu Jezc<t rinJ imyilcrncnl i i\v hi › yco|c<t w il I xuzcls help

£tG cfi£ ñI1Lt \” iA Eh1c\ cabz § .

6.BE FEnENC ES







* ..s iaLri . › an.l I :l li , [ '” i” i»d ink i\*u‹l J«

\'’ i\*lg.• ’I’’’’’’’' ',



Intcmzzuurua L'unt'c-rcrx:c on 'L”omptzrer





n.IJ wtyu ei ru Jttnj e 1” zn. Xiu L . uruJ





J 1 Jou+r•N o use’g’’g e<r•usi‹zg+ss +«eo u•sovaU•e oven NJ \ ••• i

# REFERENCES

Erdem, H., Uner and A. (2009) ‘A multi-channel remote controller for home and office appliances’, IEEE Xplore Digital Library, vol. 55, no. 4, pp. 2184-2189.

Yuksekkaya, B., Kayalar, A.A., Tosun, M.B., Ozcan, M.K., Alkar and A.Z., (2006) ‘A GSM, internet and speech controlled wireless interactive home automation system’, IEEE Xplore Digital Library, vol. 52, no. 3, pp. 837-843.

Vernon, S., Joshi and S.S., (2011) ‘Brain—Muscle—Computer Interface: MobilePhone Prototype Development and Testing’, IEEE Xplore Digital Library, vol. 15, no. 4, pp. 531-538.

Faundez-Zanuy and M. (2005) ‘Privacy issues on biometric systems’ , IEEE Xplore Digital Library, vol. 20, no. 2, pp. 13-15.

P. B. Saurabh and D.S. Chaudhari (2012) ‘Principal Component Analysis for Face Recognition’, International Journal of Engineering and Advanced Technology, , IEEE Xplore Digital Library, vol. 1, pp. 91-94.

Sahani, M., Nanda, C., Sahu, A.K.,Pattnaik, B. (2015) 'Web-based online embedded door access control and home security system based on face recognition', IEEE Xplore Digital Library[online], 19-20 March 2015, pp.1-6. Available from: <http://ieeexplore.ieee.org.libezp.utar.edu.my/xpls/icp.jsp?arnumber=7159473&ta> g =1[Accessed 1 August 2017]

Thabet, A. and Amor, N. (2015) ‘Enhanced smart doorbell system based on face recognition’, IEEE Xplore Digital Library[online], 21-23 Dec. 2015,

pp.373-377. Available from: <http://ieeexplore.ieee.org.libezp.utar.edu.my/document/7505106/>[Accessed 1

August 2017]

Neelam Majgaonkar, Ruhina Hodekar & Priyanka Bandagale (2016) Automatic Door Locking System Vol.4 Issue-1, 2016 p.495 – 499.