

# Thief Detection System

Nukala Naga Sri Ganesh, 11715703, K17JT, Computer Science and Engineering Department, Lovely Professional University, sri,ganesh@aiesec.net

## Objective:

The objective of this report is to explain each and every function of the code used to develop the Project. The Project Aim is to develop a Thief detection portable appliance which can be attached to security prone devices. The project is crudely based on face detection technology. This project was developed on Python 3.6.

## Modules imported:

1. 'face\_recognition'
2. 'cv2'
3. 'numpy'

'face\_recognition' is a face recognition module developed by Adam Geitgey. This module is used to process an input image and identify any variable image as a known or unknown face with the help of inputs. Inputs provided to 'face\_recognition' module are through 'Image' module in Python

'cv2' module refers to opencv-python module which is Computer Vision module available in python. Computer vision is used in the Project to access live video and detect images from live video.

'numpy' is used to pre process available input variables into arrays which are then provided to 'face\_recognition' module for further processing and producing result.

## Code Snippet:

```
import face_recognition # Face Recognition Module by Adam Geitgey
import cv2               #OpenCV
import numpy as np
# PLEASE NOTE: This requires OpenCV (the `cv2` library) to be installed to read from your webcam.
# Get a reference to webcam
video_capture = cv2.VideoCapture(0)
# Loading pictures of the specific people who have access to enter
# The image files specified in load_image_file() function refers to image location on the pc where program
was developed, need to change for each pc
accessed_member1_image =
face_recognition.load_image_file("/home/laikhan13/Downloads/my_picture.jpg")
accessed_member1_face_encoding = face_recognition.face_encodings(accessed_member1_image)[0] # Face
encoding returns a list of length 1
accessed_member2_image = face_recognition.load_image_file("/home/laikhan13/Downloads/ganesh2.JPG")
accessed_member2_face_encoding = face_recognition.face_encodings(accessed_member2_image)[0]
# Array of face encodings of members having access to enter
known_face_encodings = [
accessed_member1_face_encoding,
accessed_member2_face_encoding,
]
known_face_names = [
"Ganesh Nukala",
"Ganesh Nukala"
]
sms_limit=0
while True:
# Grabs the video frame
ret, frame = video_capture.read()
```

```

# Converting the image from BGR color (which OpenCV uses) to RGB color (which face_recognition uses)
rgb_frame = frame[:, :, ::-1]
# Finding all the faces and face encodings in the frame of video to identify suspicious people
face_locations = face_recognition.face_locations(rgb_frame)
face_encodings = face_recognition.face_encodings(rgb_frame, face_locations)
# Loop through each face in this frame of video
for (top, right, bottom, left), face_encoding in zip(face_locations, face_encodings):
# See if the face is a match for the authorized face(s)
matches = face_recognition.compare_faces(known_face_encodings, face_encoding)
name = "Unauthorized Suspicious Person"
# if True in matches: Does Nothing as the Person is authorized
# first_match_index = matches.index(True)
# name = known_face_names[first_match_index]
# Or instead, use the known face with the smallest distance to the new face
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]
if(name=='Unauthorized Suspicious Person'):
print('Suspicious Person near the Vault')
sms_limit+=1
if sms_limit%5==0 and name=='Unauthorized Suspicious Person':
print("Thief detected")
# Draw a box around the face
cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)
# Draw a label with a name below the face
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, 1.0, (255, 255, 255), 1)
# Display the resulting image
cv2.imshow('Video', frame)
# Hit 'q' on the keyboard to quit!
if cv2.waitKey(1) & 0xFF == ord('q'):
break
# Release handle to the webcam
video_capture.release()
cv2.destroyAllWindows()
-----End of Code-----

```

### Steps Involved:

- Initially input is provided in form of Images to face\_recognition module
- Open CV launched the System Camera for face detection
- Face is detected from know faces array created using Numpy and face\_encodings
- If a detected face is authorized to be near the vault or the security prone region, the persons name will appear in a rectangle on the camera launched using Computer Vision
- If the face is not known, the face will be detected as ‘Unauthorized Suspicious Person’ on Camera and the same will be detected on the Python Script
- If the alert message pops on the Screen in a multiple of 5 another alert messages will be shown saying that a possible theft is detected near the vault
- On pressing q, the camera module will be closed and the Program will End

### GitHub Link:

[https://github.com/N-Ganesh/Thief-Detection-System/blob/master/Thief\\_Detection\\_System.py](https://github.com/N-Ganesh/Thief-Detection-System/blob/master/Thief_Detection_System.py)