**IOT BASED ANTI THEFT DETECTION AND ALERTING SYSTEM USING RASPBERRY PI**

**Abstract -** To secure and guard our house in our absence, we propose the IOT based Anti-theft detection and alert System using Raspberry Pi. One single step anywhere inside the room Is tracked and user is alarmed through mms and phone call over IOT..

***Key Words*: IOT (Internet of Things), Raspberry Pi, Call,**

**Message, Pi Camera.**

**1. INTRODUCTION**

Now-a-days, Security has become the most challenging task. Everyone wants safety but in present scenario, nothing is safe not even in their own houses. Home is a place where we keep our assets and our capital. But we can never be sure about the security of that asset behind us and the possibilities of intrusion are increasing day by day. We generally lock houses when going out of the house. But just locking the home is not enough, there must be a system which safety our home, belongings and income from theft is the necessary requirements for home security system and keep track of the activities and report to the owner accordingly and works according to the response of the owner.

**1.1 Proposed System**

**Anti-Theft Detection And Alert System Using Raspberry :**

Whenever the thief enters in the house, we use open cv library with python to detect thief’s entry and when thief is detected it sends a valid signal to the raspberry pi which turn alerts the owner through mms, phone call.

**2. SYSTEM BLOCK DIAGRAM**

**Fig. 1** Block diagram of Raspberry-pi based anti-theft detection system

Raspberry – pi zero

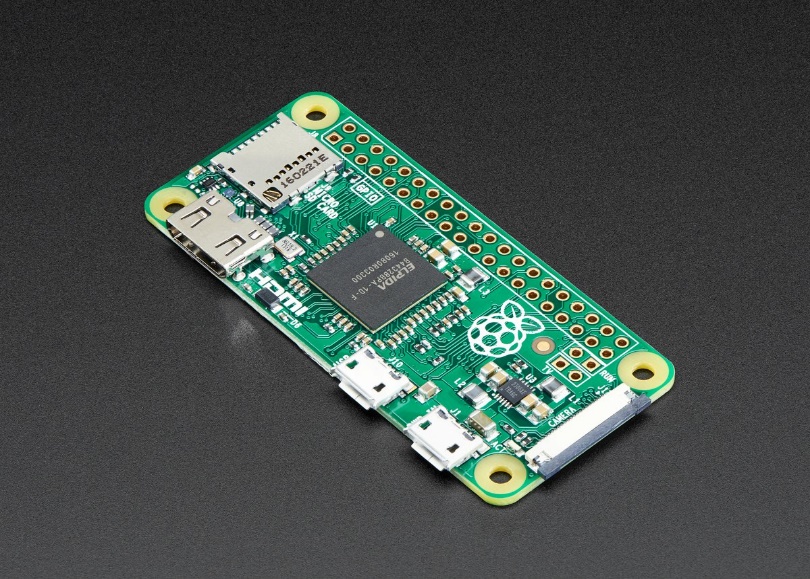
WH

Camera

**3. HARDWARE DESCRIPTION**

**1. Raspberry Pi**

The Raspberry Pi Zero W is size of only 65mm long by 30mm wide, with the addition of wireless LAN and Bluetooth. The Pi Zero W is designed to be as flexible and compact as possible with mini connectors and an unpopulated 40-pin GPIO. At the heart of the Raspberry Pi Zero W is a 1GHz BCM2835 single-core processor with 512MB RAM.



**2. Raspberry Pi Camera**

Camera module is Pi camera interfacing to the raspberry pi module. Its resolution is 5-megapixel and still picture resolution 2592 x 1944, Max image transfer rate 1080p: 30fps, this Pi camera module is used for captures an image and send captured image to the Raspberry pi module.



**3. SOFTWARE DESCRIPTION**

The software components are used for the project has been mentioned below:

**3.1 Raspbian OS:**

Raspbian is an unofficial port of Debian Wheezy armhf with compilation settings adjusted to produce optimized "hard float" code that will run on the Raspberry Pi. This provides significantly faster performance for applications that make heavy use of floating point arithmetic operations. All other applications will also gain some performance through the use of advanced instructions of the ARMv6 CPU in Raspberry Pi. Although Raspbian is primarily the efforts of Mike Thompson (mpthompson) and Peter Green (plug wash), it has also benefited greatly from the enthusiastic support of Raspberry Pi community members who wish to get the maximum performance from their device.

**4.2 PYTHON:**

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python’s elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. The Python interpreter and the extensive standard library are freely available in source or binary form for all major platforms from the Python Web site, https://www.python.org/, and may be freely distributed. The same site also contains distributions of and pointers to many free third party Python modules, programs and tools, and additional documentation. The Python interpreter is easily extended with new functions and data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

**4.3 OPEN CV:**

****

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

**4.4 Firebase**

****

**Firebase** is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user **base**, and earn profit. It is built on Google's infrastructure. **Firebase** is categorized as a NoSQL database program, which stores data in JSON-like documents.

**4.5 Twilio**

****

**Twilio** is an American cloud communications platform as a service (CPaaS) company based in San Francisco, California. **Twilio** allows software developers to programmatically make and receive phone calls, send and receive text messages, and perform other communication functions using its web service APIs

**4.6 YOLO**

**YOLO**, a single CNN simultaneously predicts multiple bounding boxes and class probabilities for those boxes. **YOLO** trains on full images and directly optimizes detection performance. This model has a number of benefits over other object detection methods: **YOLO** is extremely fast.

**5. ARCHITECTURAL FLOW OF SYSTEM**

Following Figures shows the architectural flow of system installation process and working of the proposed system which will lead to prevention of Theft.

start

On camera with program

Checking for movement in each frame

Movement detected

NO

YES

Take image and send to cloud message and call the owner

Checking for movement in each 100th frame

Movement detected

NO

YES

Take image and send to cloud, message the owner

**5.Code**

**sender\_camera.py**

**import cv2, datetime, os, time, numpy as np, subprocess**

**# Camera ON**

**video = cv2.VideoCapture(0)**

**image\_frame = 0**

**# dl datasheets**

**net = cv2.dnn.readNet("yolov3.weights", "yolov3.cfg")**

**with open("coco.names", "r") as f:**

**classes = [line.strip() for line in f.readlines()]**

**layer\_names = net.getLayerNames()**

**output\_layers = [layer\_names[i[0] - 1] for i in net.getUnconnectedOutLayers()]**

**# Image processing**

**def dl():**

**img = cv2.resize(frame1, None, fx=2.0, fy=2.0)**

**height, width, channels = img.shape**

**blob = cv2.dnn.blobFromImage(img, 1 / 255.0, (416, 416), swapRB=True, crop=False)**

**net.setInput(blob)**

**outs = net.forward(output\_layers)**

**class\_ids = []**

**confidences = []**

**boxes = []**

**for out in outs:**

**for detection in out:**

**scores = detection[5:]**

**class\_id = np.argmax(scores)**

**confidence = scores[class\_id]**

**if confidence > 0.5:**

**center\_x = int(detection[0] \* width)**

**center\_y = int(detection[1] \* height)**

**w = int(detection[2] \* width)**

**h = int(detection[3] \* height)**

**x = int(center\_x - w / 2)**

**y = int(center\_y - h / 2)**

**boxes.append([x, y, w, h])**

**confidences.append(float(confidence))**

**class\_ids.append(class\_id)**

**indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)**

**colors = np.random.uniform(0, 255, size=(len(classes), 3))**

**for i in range(len(boxes)):**

**if i in indexes:**

**x, y, w, h = boxes[i]**

**label = str(classes[class\_ids[i]])**

**color = colors[class\_ids[i]]**

**cv2.rectangle(img, (x, y), (x + w, y + h), color, 2)**

**cv2.putText(img, label, (x, y - 5), cv2.FONT\_HERSHEY\_SIMPLEX, 2/2, color, 2)**

**cv2.imwrite("/home/linuxlite/Desktop/frames/snap.jpg", img)**

**cv2.imwrite(f"/home/linuxlite/Desktop/Dl\_processed/{frame\_uid}", img)**

**time.sleep(2)**

**os.remove("/home/linuxlite/Desktop/frames/snap.jpg")**

**# http connection for transferring files(like socket, ftp...)**

**# process = subprocess.Popen("xfce4-terminal -x python3 -m http.server 22111", shell=True)**

**# Above line triggers the terminal for http connectivity but it isn't working!!**

**# Movement detecting**

**while video.isOpened:**

**# difference**

**check1, frame1 = video.read()**

**check2, frame2 = video.read()**

**diff = cv2.absdiff(frame1, frame2)**

**# frame conversions**

**gray = cv2.cvtColor(diff, cv2.COLOR\_BGR2GRAY)**

**blur = cv2.GaussianBlur(gray, (5, 5), 0)**

**\_, thresh = cv2.threshold(blur, 20, 255, cv2.THRESH\_BINARY)**

**dilated = cv2.dilate(thresh, None)**

**contours, \_ = cv2.findContours(dilated, cv2.RETR\_TREE, cv2.CHAIN\_APPROX\_SIMPLE)**

**# if motion...**

**for contour in contours:**

**(x, y, w, h) = cv2.boundingRect(contour)**

**if cv2.contourArea(contour) >= 2500:**

**if (image\_frame % 50) == 0:**

**print("Movement Detected")**

**frame\_uid = str(datetime.datetime.now()) + ".jpg"**

**dl()**

**original\_img = cv2.resize(frame1, None, fx=2.0, fy=2.0)**

**cv2.imwrite(f"/home/linuxlite/Desktop/Movement\_images/{frame\_uid}", original\_img)**

**image\_frame += 1**

**# Showing frames**

**cv2.imshow('CCTV', frame1)**

**# Quitting program**

**if cv2.waitKey(1) == ord('q'):**

**break**

**video.release()**

**cv2.destroyAllWindows()**

**receiver\_system.py**

**import requests, os, datetime, time, pyrebase, numpy as np, cv2**

**from twilio.rest import Client**

**from gpiozero import LED**

**led1 = LED(23)**

**led2 = LED(24)**

**# To avoid multiple calls**

**called = 0**

**# Firebase Storage Connectivity**

**config = {**

**"apiKey": "AIzaSyB0QxPtqRbAvbb0bZshtdPh\_5PYuvMIkeE",**

**"authDomain": "theft-detector-iot.firebaseapp.com",**

**"databaseURL": "https://theft-detector-iot.firebaseio.com",**

**"projectId": "theft-detector-iot",**

**"storageBucket": "theft-detector-iot.appspot.com",**

**"messagingSenderId": "1060700333231",**

**"appId": "1:1060700333231:web:f4be0b491e6451c074e05a",**

**"measurementId": "G-Q1GW9RH818"}**

**firebase = pyrebase.initialize\_app(config)**

**storage = firebase.storage()**

**# Twilio call & message connectivity**

**TWILIO\_PHONE\_NUMBER = "+12243026503"**

**DIAL\_NUMBERS = ["+918608550403"]**

**TWIML\_INSTRUCTIONS\_URL = \**

**"http://static.fullstackpython.com/phone-calls-python.xml"**

**client = Client("ACbf7167e386e0c2cc7966460a82dcaac2", "0bd6964c5bc9ee6613b9f4a553ca8ad7") # Check Auth token, it may change in times!**

**# Function to make call**

**def dial\_numbers():**

**led1.on()**

**for number in DIAL\_NUMBERS:**

**print("Dialing " + number)**

**client.calls.create(to=number, from\_=TWILIO\_PHONE\_NUMBER, url=TWIML\_INSTRUCTIONS\_URL, method="GET")**

**led1.off()**

**# Function for Image processing**

**def cloud():**

**led2.on()**

**storage.child(f"images/{frame\_uid}").put(f"/home/pi/Desktop/dl\_processed/{frame\_uid}")**

**view\_img = storage.child(f"images/{frame\_uid}").get\_url(None)**

**# For sending messages**

**for number in DIAL\_NUMBERS:**

**print("messaging" + number)**

**client.messages.create(body=f'Movement detected @ {frame\_uid[:-4]} - Click below url to view the spotted image {view\_img}', from\_=TWILIO\_PHONE\_NUMBER, to=DIAL\_NUMBERS)**

**led2.off()**

**while True:**

**#driver.refresh()**

**time.sleep(2)**

**url = 'http://192.168.43.41:22111/Desktop/frames/snap.jpg'**

**r = requests.get(url)**

**open('/home/pi/Desktop/frames/snap.jpg', 'wb').write(r.content)**

**size = os.path.getsize('/home/pi/Desktop/frames/snap.jpg')**

**if size > 500:**

**print("image got")**

**frame\_uid = str(datetime.datetime.now()) + '.jpg'**

**open(f'/home/pi/Desktop/dl\_processed/{frame\_uid}', 'wb').write(r.content)**

**if called == 0:**

**dial\_numbers()**

**called = 1**

**cloud()**

**os.remove('/home/pi/Desktop/frames/snap.jpg')**

**6. WORKING AND RESULT**

In this project raspberry Pi zero (model) has been used as heart of system. This proposed system is an intelligent system and it eliminates the need of continuous by human resource. Thus, any human extra work is ruled out. This system continuously checks the status of place by sensors that is anyone entering in the home or not. And sends the alert message to the owner with live images by using camera with different angles. The main aim of this project is to make an automated security system for Home, Banks or jewelry shops. The project consists of Raspberry Pi with sensor and camera. The whole system is placed in that place. If system detect someone in Home/Bank/shop it sets the capture the live images and sent it on phone through mms.

**7. ADVANTAGES**

1. The device was capable in distinguishing between human and object using AI.

2. It was using an alarm system which uses to alert the owner by making phone call, mms.

3. It was convenient in use, relatively free from false alarms and does not require frequent user action to arm and disarm the system.

**8. DIS-ADVANTAGES**

1. The sound was made by device will not be recognized by the owner, if he/she is not present there.

**9. APPLICATIONS**

 Jewellery Shops

 Army Surveillance

 Bank Locker Room

 Museum Security

 Home Security

**10. CONCLUSIONS**

The research work that will be carried out in this thesis would be mainly focused to design and develop efficient and convenient motion detection surveillance i.e. an Anti-Theft device to solve security problems which will help to reduce/stop theft. This system is suitable for small personal area surveillance. I.e. personal office cabin, bank locker room, parking entrance. Whenever the motion is detected through. The main Advantage of the project is Easy to implement, Low cost with High quality.

**REFERENCES**

answers.opencv.org, stackoverflow.com, Stackexchange.com

**BIOGRAPHIES**

**A.AKASH(180701017)**

A UG Third Year student seeking his degree in Computer Science Engineering at Rajalakshmi Engineering College, Thandalam, Chennai.

**V.AKASH(180701018)**

A UG Third Year student seeking his degree in Computer Science Engineering at Rajalakshmi Engineering College, Thandalam, Chennai

**M.ARUN KUMAR(180701035)**

A UG Third Year student seeking his degree in Computer Science Engineering at Rajalakshmi Engineering College, Thandalam, Chennai

**V. BALAJI(180701047)**

A UG Third Year student seeking his degree in Computer Science Engineering at Rajalakshmi Engineering College, Thandalam, Chennai

**B. CHRIS GAVIN(180701056)**

A UG Third Year student seeking his degree in Computer Science Engineering at Rajalakshmi Engineering College, Thandalam, Chennai