

# Chapter 1

## INTRODUCTION

### 1.1 Introduction

Suspicious activity detection (SAD) is a computer vision technology that automatically identifies and alerts security personnel to potentially dangerous or criminal activity in video surveillance footage. SAD systems are used in a variety of settings, including airports, train stations, schools, and other public places. SAD systems work by first identifying objects and people in the video footage. They then use a variety of algorithms to analyze the behavior of these objects and people. If the system detects any behavior that is considered to be suspicious, it will generate an alert for security personnel. The types of suspicious activity that SAD systems can detect vary depending on the specific system. However, some common examples include:

- people carrying weapons
- people fighting
- people vandalizing property
- car accident
- fire

SAD systems can be an effective tool for improving public safety. However, it is important to note that they are not perfect. False alarms can occur, and it is still up to security personnel to investigate and determine whether or not a real threat exists.

## 1.2 Motivation/ Relevance of Project

The motivation and relevance of a project involving a suspicious activity detector can be significant in various contexts. Here are some potential motivations and relevancies:

1. **Security Enhancement:** Suspicious activity detectors can play a crucial role in enhancing security measures.
2. **Crime Prevention:** Suspicious activity detectors can assist in proactive crime prevention. By analyzing patterns, anomalies, or deviations from normal behavior, these systems can alert law enforcement agencies or security personnel to potential criminal activities.
3. **Fraud Detection:** For financial institutions, suspicious activity detectors can be invaluable in identifying fraudulent transactions or activities.
4. **Public Safety:** Deploying suspicious activity detectors in public spaces, such as airports, train stations, or stadiums, can contribute to public safety. These systems can help identify behaviors or actions that deviate from normal conduct, potentially indicating threats or illegal activities.
5. **Proactive Monitoring:** Suspicious activity detectors enable continuous monitoring of various environments, both physical and digital.

## 1.3 Present Status

In the market, there are surveillance robo available, but they tend to be quite expensive. To make a more affordable option, a team created their own surveillance robo with additional features. Instead of requiring a specific app to operate like other surveillance robo car present in market, they made a object detection live streaming which is only accessible when connected to the surveillance robo with Remote Desktop. Additionally, they addressed a common issue with other surveillance robo cars by adding a rechargeable battery to solve the battery discharge problem. Overall, their surveillance robo is a more cost-effective and reliable option for those in the market for such a device.

## CHAPTER 2

### REVIEW OF LITERATURE

#### 2.1 Literature Survey:

[1] According to **V. Ratna Kumari, P.Siva Sanjay**

These days, we often get updated with the realities about lethal accidents because of harmful conditions, cavern investigation, mining and military surveillance and so forth. This has definitely been a zone of worry over an extensive stretch of time, as the valuable human life has been in question under the cost of investigative purpose.

[2] According to **V.Potluri , P.Siva Sanjay**

In today's world the age of apply autonomy field is developing exponentially and a part of the wellknown automated items are utilized to a great extent by the research networks, guard, scholastic and businesses. The idea and execution cost of a robot is less at that point procuring a human parental.

[3] According to **Narayan, Raja Naga Lochan ,Renish Kumar Kothadiya**

The need for surveillance in a society escalates with an increase in population. As more people reside in a specified area, monitoring their activity gets tougher, which gives a rise in security concerns. Not only in societies, need for surveillance, monitoring and detection has been a prime According requirement in the defence sector.

[4] According to **Mrs. A. Jansi Rani, Ms. A. Afna, Ms. A. Remsitha Banu, Ms. E. Sophiya** It presents the IOT Surveillance Robot Car project involves the use of an Raspberry Pi module as the main microcontroller with an inbuilt Wi-Fi module for connectivity between a desktop.

## Chapter 3

### SOFTWARE IMPLEMENTATION

#### 3.1 Implementation Details

##### a) Artificial Neural Network

An artificial neural network (ANN) is a set of layers of neurons (in this context they are called units or nodes). In the case of a fully connected ANN, each unit in a layer is connected to each unit in the next layer.

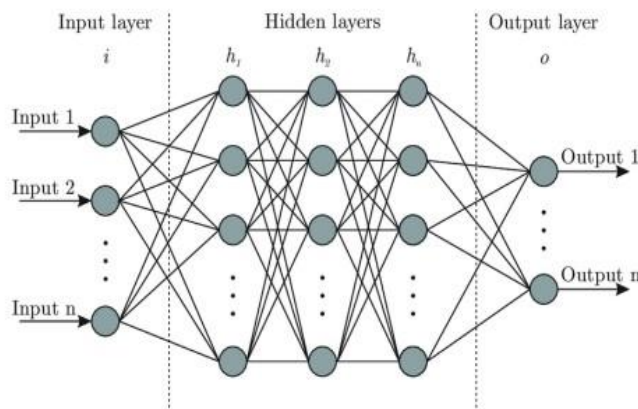


Fig.3.1.1 ANN Architecture

There is an input layer, where the network takes all the information needed, in this case the images to classify. Between the input layer and the output layer are hidden layers. Each hidden layer is used to detect a different set of features in an image, from less to more detailed. The output layer is where the network makes predictions. The predicted image categories are compared to the labels provided by humans. If they are incorrect, the network uses a technique called backpropagation to correct its learning, so it can make guesses more correctly in the next iteration. After enough learning, a network can make classifications automatically without human help.

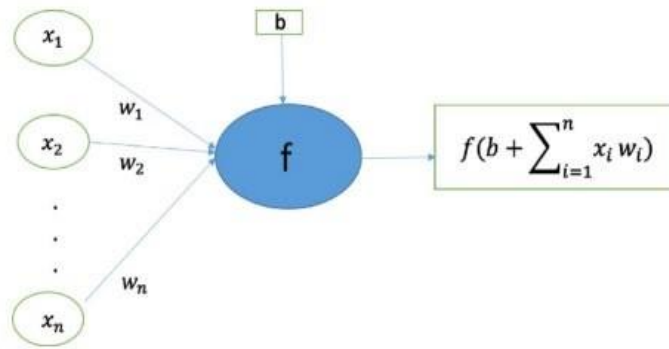
**c) Structure of Neuron, Weights and Biases**

Fig 3.1.2 Structure of Neuron

**3.2 Libraries used for implementation of CNN architecture.****a) TensorFlow:**

TensorFlow is an open source software library developed by Google that is widely used for machine learning tasks, particularly in the area of deep learning. TensorFlow is used for a wide variety of machine learning tasks, including:

- Natural language processing
- Computer vision
- Speech recognition
- Machine translation
- Recommender systems
- Reinforcement learning

**c) Matplotlib:**

Matplotlib is a popular open-source data visualization library in Python that is widely used for creating high-quality plots, charts, and figures. It provides a wide range of customizable 2D and 3D plotting tools and a flexible API that enables users to create a variety of visualization for different types of data. Matplotlib is particularly useful for visualizing scientific data, and is often used in fields such as a physics, engineering and data science. Some features of Matplotlib are:

**Easy Customization:** Matplotlib provides a wide range of customization options that allow users to create highly customized plots and charts. Users can adjust things like colours, labels and formatting, and can also add text, annotations, and other elements to their visualizations.

**Support:** It can support wide range of data formats, including NumPy arrays, Pandas data frames, and Python lists.

**Interactive visualization:** Matplotlib provides support for interactive visualization using the pyplot interface or the object-oriented API. This allows users to create interactive plots and charts that can be used to explore data and gain insights.

**d) Numpy:**

Numerical python is a popular open-source library for the python programming language that is widely used for numerical computing tasks. It provides powerful set of tools for working with arrays and matrices of numerical data, including a wide range of mathematical functions and operations. Some key features of NumPy are:

**f) Sklearn.metrics:**

Scikit-learn is a popular open source machine learning library for python. It provides a wide range of tools for various machine learning tasks, including classification, regression, clustering, and more.

### g) TensorFlow Lite:

TFLite is a lightweight version of TensorFlow that is designed for mobile and embedded devices. It provides tools for optimizing and deploying machine learning models on devices with limited resources, such as smartphones, tablets, and IoT devices. One of the key components of TensorFlow Lite is the TFLite convertor, which is used to convert TensorFlow models into a format that can be run on mobile and embedded devices.

### 3.3 Activation Function:

Choosing an activation function is crucial for any deep learning model, for YOLOv5 the authors went with SiLU and Sigmoid activation function. In YOLO v5 the Leaky ReLU activation function is used in middle/hidden layers and the sigmoid activation function is used in the final detection layer.

(a) SiLU function graph.

(b) Sigmoid function graph.

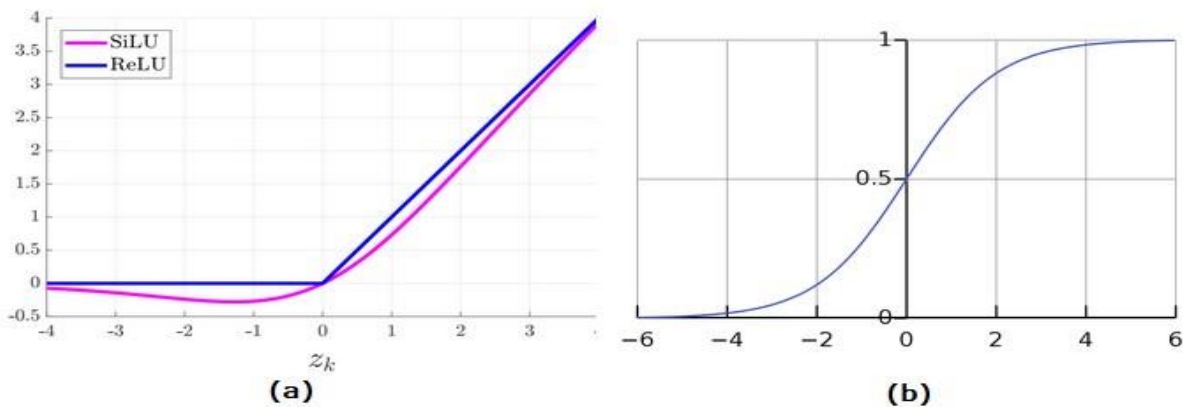


Fig.3.3.1. Sigmoid and Silu function

SiLU stands for Sigmoid Linear Unit and it is also called the swish activation function. It has been used with the convolution operations in the hidden layers. While the Sigmoid activation function has been used with the convolution operations in the output layer.

### 3.4 Loss Function

YOLOv5 returns three outputs: the classes of the detected objects, their bounding boxes and the objectness scores. Thus, it uses BCE (Binary Cross Entropy) to compute the classes loss and the objectness loss. While CIoU (Complete Intersection over Union) loss to compute the location loss. The formula for the final loss is given by the following equation YOLOv5 was released with five different sizes:

$$Loss = \lambda_1 L_{cls} + \lambda_2 L_{obj} + \lambda_3 L_{loc} \quad (1)$$

### 3.5 Collecting Our Training Images:

In order to get your object detector off the ground, you need to first collect training images. You want to think carefully about the task you are trying to achieve and think ahead of time about the aspects of the task your model may find difficult. I recommend narrowing the domain that your model must handle as much as possible to improve your final model's accuracy.

**3.6 Download Dataset:** This is the full 2017 object detection dataset (train and valid), which is a subset of the most recent 2020 object detection dataset. COCO is a large-scale object detection, segmentation, and captioning dataset of many object types easily recognizable by a 4-year-old. The data is initially collected and published by Microsoft.

### 3.7 Labelling to data using labelling tools:

There's a limited amount of annotated data currently available for object detection. That can be a hurdle. If you're not able to find an annotated dataset from the above mentioned resources then you need to create one by yourself. A few of the more popular are: • Label Img • Labelme • DarkLabe 4.4.6 Split The Data Into Training And Validation: Once the labelling is done, we will split our data into training and validation sets. The ratio of split is noted in the train\_test\_split.py file and you can change it according to your specific needs.



### 3.8 Google Teachable

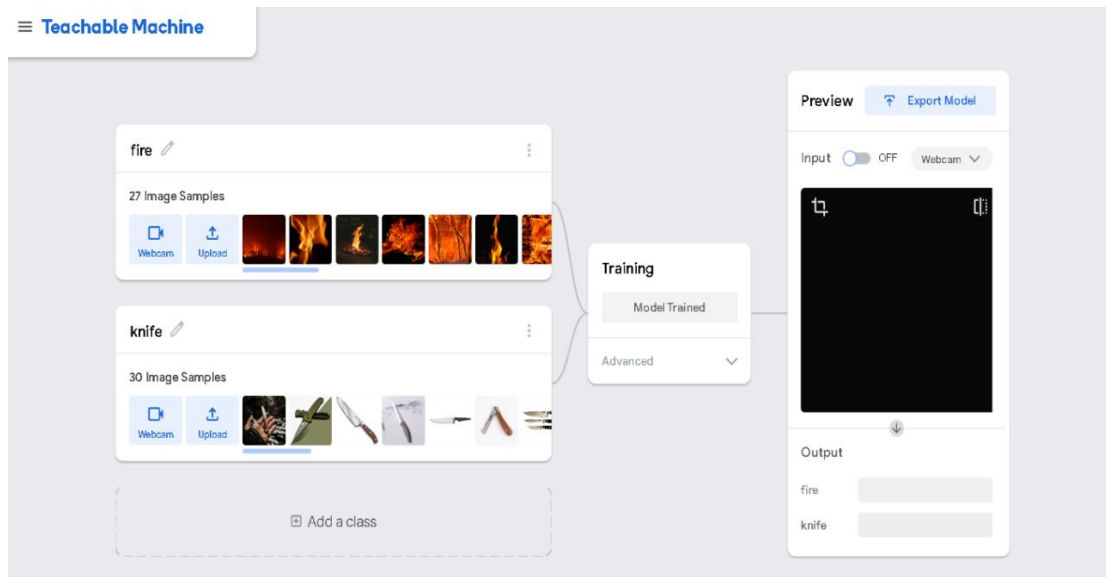


Fig. 3.8.1 Implemented Model Training.

Teachable Machine is a web-based tool that makes it fast and easy to create machine learning models for your projects, no coding required. Train a computer to recognize your images, sounds, & poses, then export your model for your sites, apps, and more.

#### Features

- Create machine learning models to recognize images, sounds, and poses.
- Export your models to HTML, JavaScript, and Python.
- Train your models with just a few clicks.
- Use Teachable Machine on any device with a web browser.

#### Benefits

- Easy to use: Teachable Machine is designed to be easy to use, even for beginners.
- Free: Teachable Machine is free to use.
- Online: Teachable Machine is an online tool, so you can use it from anywhere.

Accuracy per class

CLASS	ACCURACY	# SAMPLES
Class 1	0.60	5
Class 2	1.00	5
Class 3	1.00	1
Class 4	1.00	4
Class 5	1.00	2

### 3.8.2 Accuracy per class

Underfit: a model is underfit when it classifies poorly because the model hasn't captured the complexity of the training samples.

Overfit: a model is overfit when it learns to classify the training samples so closely that it fails to make correct classifications on the test samples

Confusion Matrix

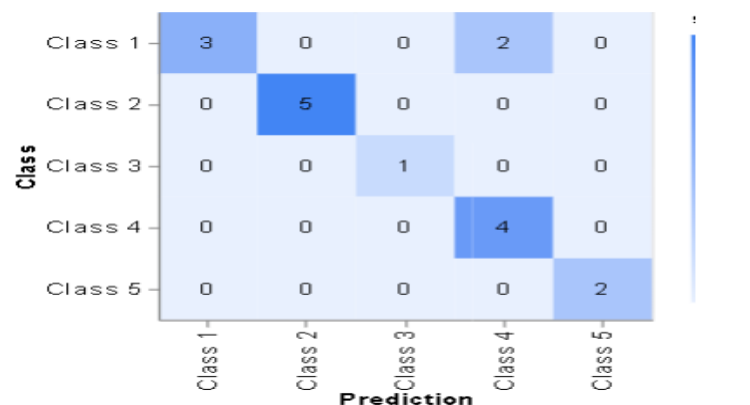


Fig.3.8.3 Confusion Matrix

Epochs: One epoch means that every training sample has been fed through the model at least once. If your epochs are set to 50, for example, it means that the model you are training will work through the entire training dataset 50 times.

**3.9 Visualization the Model:** Now that we are done training our model, let's look at the results. Once the custom weights have been added to the yolov5 directory on your local computer, then you can run the model in your webcam after changing to the yolov5 directory in your terminal.

### 3.10 Flow Chart

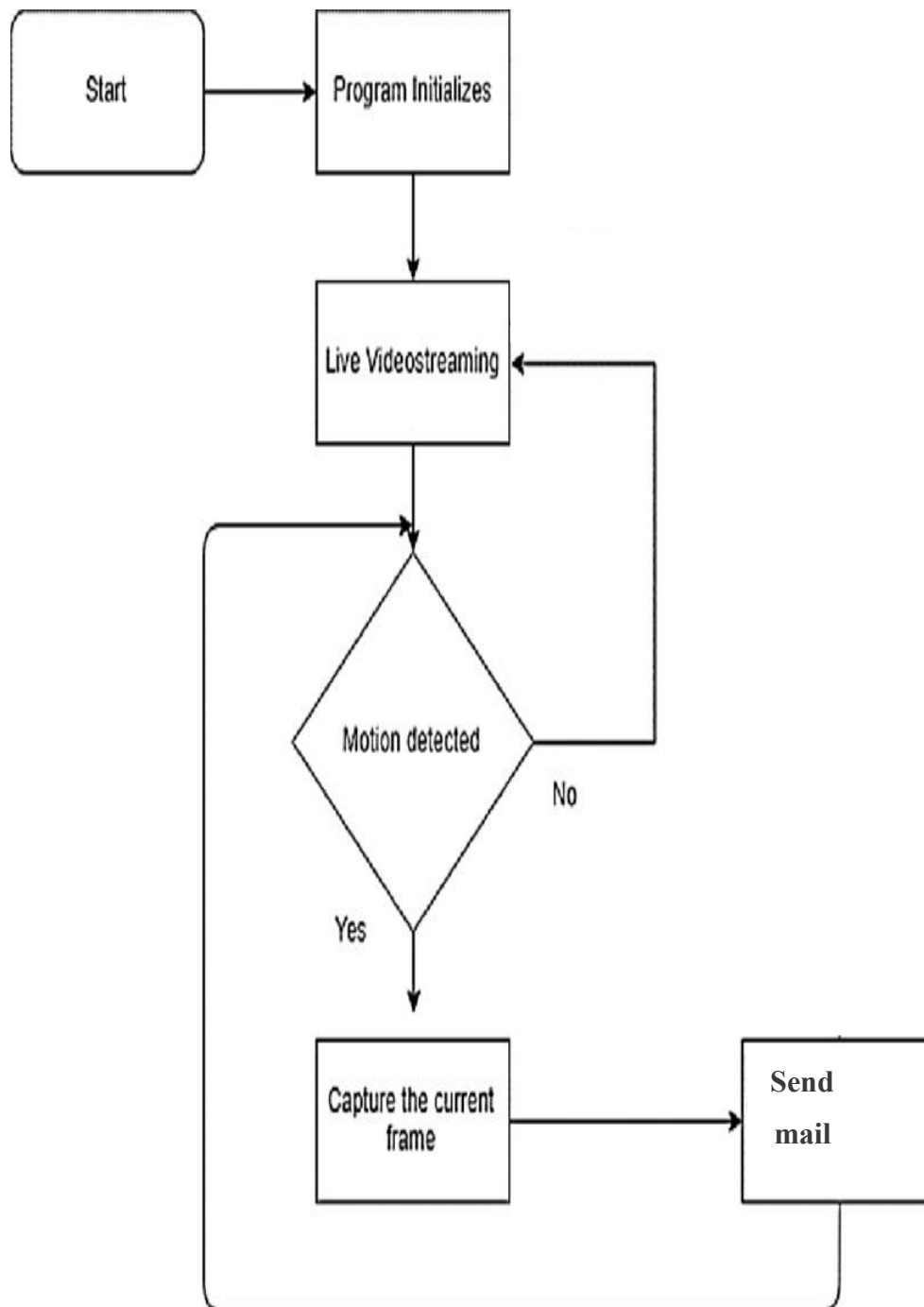


Fig. 3.10.1 flow chart

Input: Video or sensor data

Preprocessing:

- Convert data to a suitable format
- Normalize or enhance data if necessary

Feature Extraction:

- Extract relevant features from the data
- Examples: Motion, shape, appearance, audio characteristics

Anomaly Detection:

- Apply anomaly detection algorithm or model
- Compare extracted features with normal behavior patterns
- Identify deviations or anomalies

Thresholding:

- Set a threshold to classify activities as suspicious or normal
- If anomaly score exceeds the threshold, mark as suspicious

Alert Generation:

- If suspicious activity is detected:
- Generate an alert/notification
- Notify appropriate personnel or authorities

Postprocessing:

- Optional filtering or refinement of detected activities
- Merge or split activities as needed
- Remove false positives or noise

Visualization:

- Visualize detected activities or anomalies
- Generate reports or visual summaries

## Chapter 4

# Training the Model

### 4.1 Introduction to training the model

Set up your camera and environment

Neural networks can learn to detect and generalize small gestural changes that only affect a few dozen pixels on the screen, but it takes much more data and training time than we want to spend.

To make it easier for the neural network to learn from limited data, place the camera so that:

- 1) Your gestures are roughly in the center of the camera view.
- 2) You are close enough to the camera that your gesture affects a significant number of pixels.

A good rule is to place the camera in front of you so that your hands can reach the edges of the frame when you stretch your hands out in front of you. To preview what the record and run scripts will see, use the `preview.py` script with this command:

It does not matter if the camera is the correct way up as long as its position remains consistent. For best results, ensure that:

- 1) There is not too much activity happening in the background.
- 2) There is good lighting and the camera has a good view of the subject.
- 3) Your arms are in the picture even when they are extended.

When you are happy with the positioning of the camera and the lighting, either close the preview window or press `ctrl-c` at the console to exit the preview app.

## 4.2 Steps to Training the model

1. First create an ML model having datasets of normal activities such as walking, talking, reading, sitting etc. Then feed the datasets of Suspicious Activity such as fighting, boxing, pointing guns or any other violent movement deemed suspicious into the ML model.
2. Also, perform such activities in front of the smart camera so that various movements are captured. This will be useful for training the ML model and deploying it on Raspberry Pi to make a smart AI camera.
3. To create and train an ML model, there are several flexible options such as TensorFlow, Google Teachable, Edge Impulse, Lobe etc. You can choose any and work on the above idea. Here, I am demonstrating with the help of Google Teachable.
4. In Google Teachable, select the PoseNet option for tracking the various body movements and actions. Earlier, we performed different actions like walking, talking, eating, standing etc. By correctly labelling them, feed these datasets into the ML model. Similarly, feed the datasets of activities like pointing guns, firing guns, fighting, beating etc.

## Chapter 5

# USER INTERFACE SPECIFICATION

### 5.1 How the user interface behaves

#### 5.1.1 Loading a video

For demonstration purposes the option to load videos manually was used. The user can also use this feature to play old recordings so that the system helps them find suspicious behaviour. If no camera is detected, the system does not stop. It still allows you to select a pre-recorded video.

#### 5.1.2 Display the alert

Mplayer is used to show the alert signal and produce the sound alarm too. Each screen will have its own alert video so that the user knows immediately where the suspicious activity is occurring.

#### 5.1.3 Alert Notifications

Implement a notification system within the UI to alert users promptly about any critical or high-priority suspicious activities. This could include pop-up alerts, sound notifications, or visual indicators that draw attention to urgent events. Users should have the ability to configure the types of notifications they want to receive and how they prefer to be notified.

#### 5.1.4 Activity Details

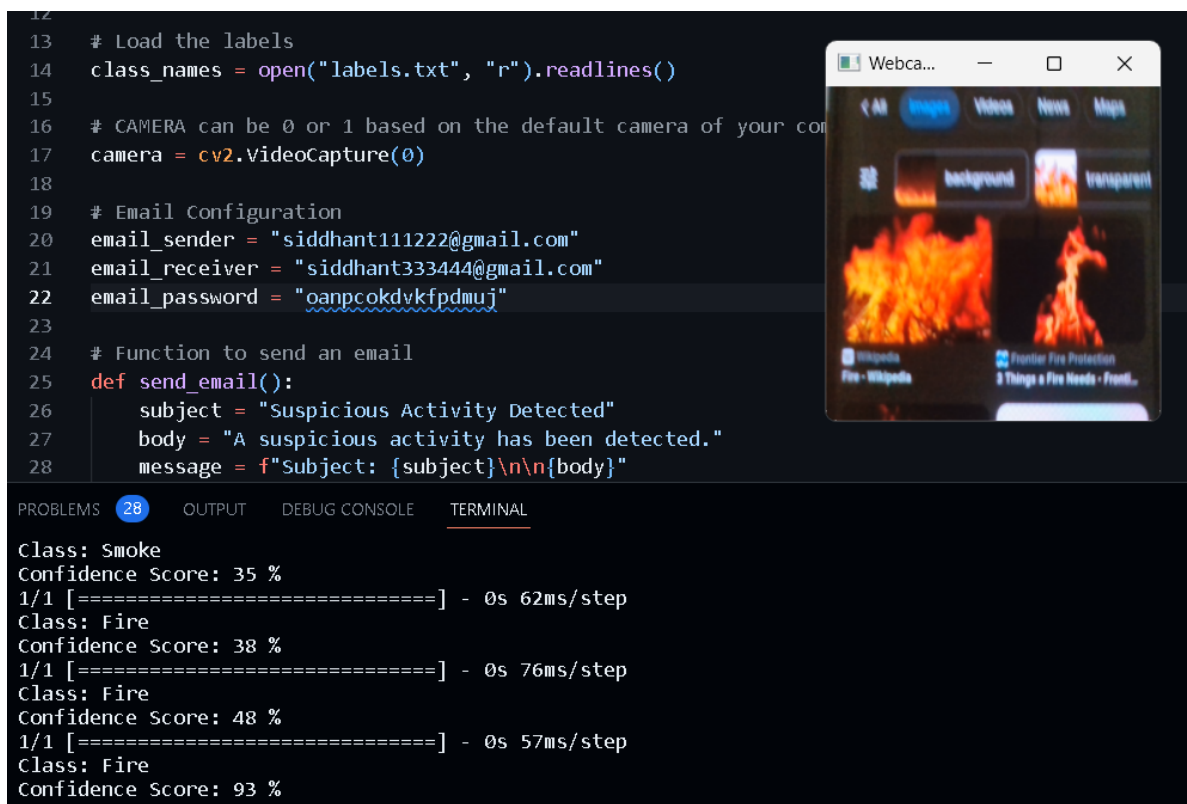
Provide a detailed view of each suspicious activity when selected or clicked. This view should present comprehensive information about the detected activity, such as timestamps, location, description, and any associated entities or users. It should also display any related evidence, such as images, videos, or transaction logs, to help users assess the nature and severity of the activity.

## Chapter 6

### Result

This code appears to be a Python script that uses a pre-trained Keras model to perform image classification on webcam images. It detects suspicious activities by predicting the class of the captured image and taking specific actions if the confidence score of the predicted class is above a certain threshold.

- The webcam feed window displaying the captured image.
- The predicted class name for each image frame, representing the detected activity.
- The confidence score for each predicted class, indicating the model's confidence in its prediction.
- If the confidence score is above 0.80, an email will be sent and an alert sound will be played.
- The script will continue running until the 'Esc' key is pressed.



```

12
13 # Load the labels
14 class_names = open("labels.txt", "r").readlines()
15
16 # CAMERA can be 0 or 1 based on the default camera of your com
17 camera = cv2.VideoCapture(0)
18
19 # Email Configuration
20 email_sender = "siddhant111222@gmail.com"
21 email_receiver = "siddhant333444@gmail.com"
22 email_password = "oanpcokdvkfpdmuj"
23
24 # Function to send an email
25 def send_email():
26     subject = "Suspicious Activity Detected"
27     body = "A suspicious activity has been detected."
28     message = f"Subject: {subject}\n\n{body}"

```

PROBLEMS 28 OUTPUT DEBUG CONSOLE TERMINAL

```

Class: Smoke
Confidence Score: 35 %
1/1 [=====] - 0s 62ms/step
Class: Fire
Confidence Score: 38 %
1/1 [=====] - 0s 76ms/step
Class: Fire
Confidence Score: 48 %
1/1 [=====] - 0s 57ms/step
Class: Fire
Confidence Score: 93 %

```

Fig. 4.1.1 Result



### A) Confidence score

As an AI language model, I don't have a built-in confidence scoring system for individual responses. However, I strive to provide accurate and reliable information based on the training data I've been provided. It's important to note that while I can generate responses and information, it's always a good idea to verify any critical information or seek additional sources when needed. Confidence score, in the context of AI models like represents the level of certainty or reliability associated with a generated response. It indicates the model's confidence in the accuracy or relevance of its answer. The confidence score is typically expressed as a value between 0 and 1, with 1 indicating high confidence and 0 indicating low confidence.

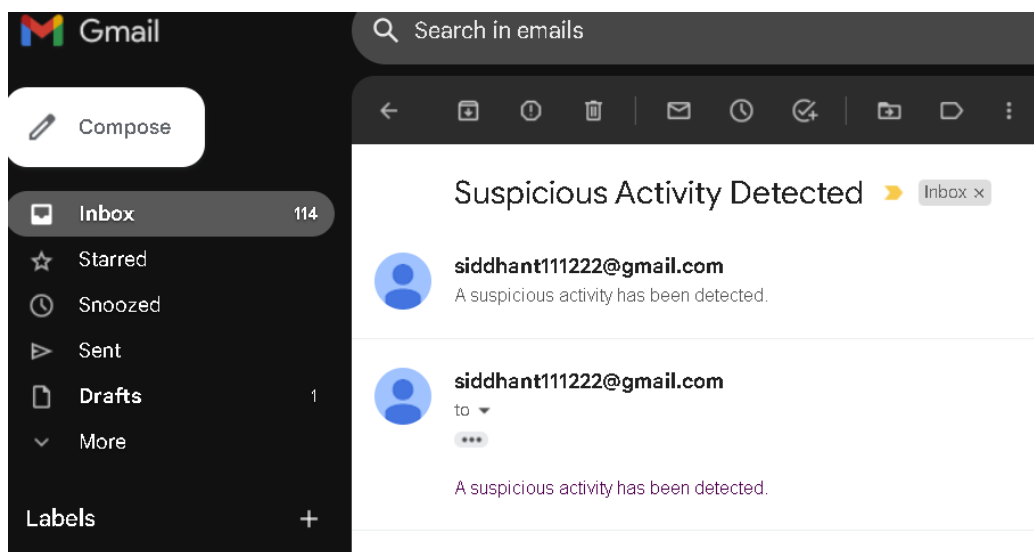


Fig 4.2.2 Mail

## Chapter 7

# SUMMARY AND CONCLUSION

### 7.2 Conclusions

In conclusion, a suspicious activity detector plays a crucial role in enhancing security, preventing crimes, and detecting fraudulent or malicious activities. By analyzing patterns, anomalies, or deviations from normal behavior, these systems can provide early warnings and alerts to help mitigate risks and protect individuals and assets. Whether deployed in public spaces, financial institutions, or online platforms, suspicious activity detectors contribute to proactive monitoring, real-time detection, and effective response to potential threats. The user interface of such a system should prioritize ease of use, clear information presentation, and efficient interaction to enable users to effectively monitor and manage detected activities. However, it's important to note that the confidence of the system's responses should be considered along with critical evaluation to ensure accurate and reliable results.

### 7.3 Future Scope

The future scope of a suspicious activity detector is promising, with several potential advancements and areas of development. Here are some future directions and possibilities for such systems:

1. **Improved Detection Algorithms:** Future developments may involve the use of advanced machine learning techniques, such as deep learning or reinforcement learning, to improve the detection capabilities of the system.
2. **Integration of Multi-Modal Data:** Future systems may leverage the power of data fusion and multi-modal analysis to detect and correlate activities across different data sources, leading to more robust and reliable detection.
3. **Real-time Analytics and Predictive Capabilities:** This proactive approach can aid in preventive measures and early intervention.

## Chapter 8

### REFERENCES

- 1) [1] “Computer Vision Zone” <https://www.computervision.zone/>
- 2) M Manoj krishna1\*, M Neelima2 , M Harshali3 , M Venu Gopala Rao4. “Image classification using Deep learning”. International Journal of Engineering & Technology , 7 (2.7) (2018) 614-617. [18] <https://www.quora.com/How-is-CNN-better-for-an-image-classification-problem-over-ANN>
- 3) [3]<https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6> : Activation Function.
- 4) [4]<https://medium.com/analytics-vidhya/a-complete-guide-to-adam-and-rmsprop-optimizer75f4502d83be> : RMSprop
- 5) <https://www.ibm.com/topics/convolutional-neural-networks> : CNN information.
- 6) <https://www.mathworks.com/help/deeplearning/ref/alexnet.html>
- 7) [https://www.tensorflow.org/lite/android/lite\\_build](https://www.tensorflow.org/lite/android/lite_build)
- 8) [https://d2l.ai/chapter\\_convolutional-neural-networks/lenet.html](https://d2l.ai/chapter_convolutional-neural-networks/lenet.html)
- 9) <https://www.mathworks.com/help/deeplearning/ref/vgg16.html>
- 10) <https://keras.io/api/applications/mobilenet/>
- 11) <https://cloud.google.com/tpu/docs/inception-v3-advanced>



