WIMAGE

A PROJECT REPORT

Submitted to

Visvesvaraya Technological University

BELAGAVI - 590 018

by

ANANTHA KRISHNA K S	**USN: 4SU16CS012**
BHARATH KISHORE	**USN: 4SU16CS022**
BHARATH S JAIN	**USN: 4SU16CS023**
IMRAN SM	**USN: 4SU15CS027**

Under the guidance of

Ms. Sneha Shetty R

Assistant Professor

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

Bachelor of Engineering

Department of Computer Science & Engineering

SDM INSTITUTE OF TECHNOLOGY

UJIRE - 574 240

2019-2020

SDM INSTITUTE OF TECHNOLOGY

(Affiliated to Visvesvaraya Technological University, Belagavi)

UJIRE – 574 240

Department of Computer Science & Engineering

CERTIFICATE

Certified that the Project Work titled '**Wimage**' is carried out by **Mr. ANANTHA KRISHNA K S**, USN: **4SU16CS012, BHARATH KISHORE,** USN: **4SU16CS022 BHARATH S JAIN,** USN: **4SU16CS023, IMRAN SM,** USN: **4SU15CS027** bona-fide students of SDM Institute of Technology, Ujire, in partial fulfillment for the award of the degree of **Bachelor of Engineering** in Electronics and Communication Engineering of Visvesvaraya Technological University, Belagavi during the year 2018-2019. It is certified that all the corrections/ suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree.

Ms. Sneha Shetty R	**Dr. Thyagaraju G S**	**Dr. Ashok Kumar T**
Asst. Professor and Guide	Professor and Head	Principal

Signature with date and seal:

External Viva

Name of the Examiners: **Signature with Date**

1.

2.

Acknowledgement

We express our deepest gratitude to our guide Ms. Sneha Shetty R, Asst. Professor, Department of Computer Science & Engineering, for his valuable guidance and encouragement while doing this project work.

We are indebted to Dr. Thyagaraju G S, Head of the Department, and Dr. Ashok Kumar T, Principal, for their advice and suggestions at various stages of the work. We also extend our heartfelt gratitude to the management of SDM Institute of Technology, Ujire, for providing us with a good learning environment, library and laboratory facilities.

Lastly, we take this opportunity to offer our regards to all of those who have supported us directly or indirectly in the successful completion of this project work.

ANANTHA KRISHNA

BHARATH KISHORE

BHARATH S JAIN

IMRAN S M

Abstract

This project helps to detect various weapons via object detection algorithms which can have various real time applications such as detection of suspicious objects or people.It can be integrated into CCTV cameras which can be installed in Banks, Airport, Private and Commercial places, etc.The main motive of this project is to safe guard people from the threats by warning them if something suspicious is detected.We can do this by providing a notification/message to the user whenever an object of interest is detected.It is a small system that helps in reducing the crime rate by providing proper surveillance and notifying the authorities resulting in reduced response time and due to which will gradually reduce the crime rate.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Project Introduction

To provide a safer place to people using AI technology, we envision a world where I, you and we all live must be safe, no mass shootings, security products are more intelligent, more…human. We believe that AI will unlock another era of public safety, and that the R&D and deployment of computer vision and deep learning technologies will redefine human life as we know it and better protect us from gun violence at large. Not only will this transform this industry as a whole, but it will push our safety forward as well. Our strategy is to build top-in-class AI platforms and safety services for mankind. This project is built to detect weapons and to rise alerts when person carrying gun or knife I detected.

1.2 Problem Description

Weapons are now a days are easily available to criminals, which is a huge danger for public around the world. To safeguard the security and safety for the people from criminals ,such as children's holding the gun without their knowledge and this may lead to an accidental death or serious injury. Safety for the common man who travels all around such as , railway stations, hotels , public places and public transport. Many a times a person is not safe around his house and he may notice suspicious activities which may lead to murder. It can also be private and commercial places such as hotels, shopping malls, parks, religious places and historical places among few are tourist attraction. Can be implemented in forest areas to avoid animal hunting and other illegal activities.

Chapter 2

Literature Review

2.1 General Introduction

Literature Survey is an important activity, which we have to do while gathering information about a particular topic. It will help us to get required information or ideas to do work. The following paragraphs discuss the related work and issues in the area of object detection in a Computer Laboratory using machine learning algorithm.

2.2 Literature Survey

Paper 1:

Kan-ichi Koyama,Yasuhachi Hamamoto, Hiroshi Kano, Haiyuan Wu , "K-means Tracker-based Object Tracking Method for Digital Camera" [1]. This paper proposes a new modified object tracking algorithm based on K-means tracker and implemented it on a prototype digital camera. The tracking performance was evaluated by using six computer-generated movie sequences simulating critical conditions of real scenes. Consequently, it is confirmed that the performance of the modified method was high enough for the practical use.

Paper 2:

Rumin Zhang,Yifeng Yang,Wenyi Wang,Liaoyuan Zeng"An Algorithm for Obstacle Detection based on YOLO and Light Filed Camera" [2]. In this paper we a new method to detect obstacles in the indoor environment is proposed that introduces deep learning technology and the light field camera to recognize the obstacle and perceive its information. According to the recognitionresult and depth map, the object filter is applied to remove the unconcern obstacle. To demonstrate the performance of our method, different types of scene, including pedestrian, chair,book and so on, are demonstrated. The experimental results prove the effectiveness of our obstacle detection algorithm. Inthe future, we will further investigate how to detection the obstacles in real time base on the light filed camera and the deep learning technology.

Paper 3:

Tianmei Guo, Jiwen Dong ,Henjian Li,Yunxing Gao, “Simple Convolutional Neural Network on Image Classification” [3]. This paper proposed a simple Convolutional neural network on image classification. This simple convolutional neural network imposes less computational cost. On the basis of the convolutional neural network , we also analyzed different methods of learning rate set and different optimization algorithm of solving the optimal parameters of the influence on image classification. We also verify that the shallow network also has a relatively good recognition effect.

Paper 4:

Bojian Ding, Lize Gu,Xiaoning Zhu,” Real-Time Detection for Camera Sensing System Adaptive Cascade Single-Shot Detector” [4].This paper introduces ACSD as a real-time end-to-end convolutional network for object detection. Which is efficient and high-quality for camera sensing. We use spectral clustering to change classification datasets to detection which enrich the variety of detection samples. Comprehensively, the introduce of cascade detector perfectly solves the unbalance of positives and negatives which cause over-fitting. a multiresolution network model for object detection is constructed and all experimental results are carefully analyzed. Compared to different start-of-art in both accuracy and speed performance.

Paper 5:

Seshadri Sastry Kunapuli, Praveen Chakravarthy BH and Upasana Singh. Caytiles, “Designing and Implementing a Real time weapons detection system on ODROID-XU4” [5]. ODROID is a thunder packed on-chip single board computer with 2 GB Ram and 8 core processor. The main difference between the original CPU and the ODROID-XU4 box is the built in hardware kernels. With respect to the size, we can take it to everywhere by carrying it in your pocket. It will also be helpful in designing/implementing a real time model which is explained in this paper by building a weapon detection system with an accuracy of nearly 100%. One can use it in making the drones/robots for the real time purpose, which will be our future scope, detecting weapons through drones/mini-robots made up with Odroid box that will be helpful in army borders and highly delicate areas, where person can’t go. But the main drawback with this Odroid box is that it is quite costly when comparing to the raspberry pi kits but we can use this box in IFTTT also with the help of android operating system.

2.3 Summary

After going through the IEEE papers, it was found that there are various algorithms available for training the machine to identify to identify objects using a custom or a pre built labeled data set provided by coco. Each algorithm had its own strength and weaknesses in terms of no. of objects detected, frame rate and precision. Algorithms for object detection based on yolo are faster when compared with r-cnn, fast r-cnn which used a method called selective search which require less number of bounding boxes that the algorithm tests, thus giving a better object tracking results. Another algorithm SSD for object detection forwards the image once through a deep learning network, but YOLOv3 is much faster than the SSD while achieving very comparable accuracy. It was also found that while yolo performed better than others at detecting multiple objects it faces difficulty detecting small sized objects.

Chapter 3

Problem Formulation

3.1 General

Before attempting to solve a problem, we need to first formulate or define the problem. It is important to precisely define the problem you intend to solve. Problem formulation is the act of a problem, determining the cause of the problem and, identifying the solution.

3.2 Problem Statement

Nowadays, safety is a biggest measure and due to increase in illegal and criminal activities safety is becoming just a myth. In many countries' guns are given to citizens of country for their safety and many times this safety measures leads to a dangerous scene of misusing the guns. In shopping malls, Schools, community halls, Religious places safety should be main concern and many times we see crimes taking place at these areas where someone is shot by gun. In forest areas many Criminals do smuggling of weapons and kill innocent animals by shooting them for their benefits. Few times you may never know who is standing outside your house with a gun to shot you down. To maintain safety and secured surrounding around you. Hence, we want to build a peaceful and safety environment around everyone.

3.3 Objectives of the Present Study

The objectives of the proposed project are as follows:

1. To develop a weapon detection system.
2. Training the machine using datasets to identify weapons.
3. Improving the currently used cctv cameras by using one's which can process important information as well.
4. Raising alerts by calling the authorities when weapon is detected.

3.4 Summary

The proposed project on Artificial Intelligence based weapon detection system based on object detection. Detection of weapons can be a big help to the authorities as it reduces their response time and help bringing the suspect to justice while saving lots of lives in the process.

Chapter 4

Requirements and Methodology

4.1 Requirements

The proposed project consists of following modules:

4.1.1 Hardware requirements

4.1.2 Software requirements

4.1.1 Hardware Requirements

The hardware requirements for the proposed project are depicted in the table below:

Table 4.1: Hardware requirements

Sl.No	Hardware/Equipment	Specification (min)
1.	Processor	Intel i3
2.	Disk Space	40 GB
3.	RAM	4 GB
4.	Graphics	NVIDIA/ AMD/ INTEGRATED
5.	Camera	Minimum 2MP

4.1.2 Software Requirements

The software requirements for the proposed project are depicted in the table below:

Table 4.2: Software requirements

Sl.No	Software	Specification
1.	Anaconda 5.3	Jupyter Notebook
2.	Browser	Google Chrome
3.	Libraries	jupyter, pillow, cython, OpenCV, matplotlib, Keras, Cmake, Twillio
4.	Programming Language	Python

4.2 Methodology Used

Methodology is the systematic, theoretical analysis of the methods applied to a field of study. It comprises the theoretical analysis of the body of methods and principles associated with a branch of knowledge. Typically, it encompasses concepts such as paradigm, theoretical model, phases and quantitative or qualitative techniques.

Steps followed for implementing the Proposed Project

1. Exploring the most relevant IEEE papers related to the project.
2. Preparing the Literature survey document.
3. Preparing the report including details like Problem Statement, Brief introduction, Applications, Business Scope and Research scope.
4. Setting up the list of objectives for the proposed project ideas.
5. Framing the methodology for the execution of the project.
6. Identification of the Hardware and Software Requirements for the project and budget estimation.
7. Identifying algorithms to be used - YOLOv3
8. Preparing the data sets for training using annotation software.
9. Dividing the data set in a ratio of 80 - 20 for training and testing respectively.
10. The training produces weights used to prepared a model that does the object detection.
11. Development of project code for accessing the camera module and performing detection.
12. Integrating an api in the code for communicating the user upon detection of the weapon.
13. Performing alterations in the code to contact the user only when the detection confidence value is more than 8.0 to avoid false alerts.

Chapter 5

System Design

5.1 System Design

System design is a one important phase in software or system development. System design can be defined as method of defining different modules required for software or system to fulfil all requirements.

5.1.1 Architecture of proposed system

System design shows the overall design of the proposed human emotion recognition system, which is depicted in figure 5.1 shown below.

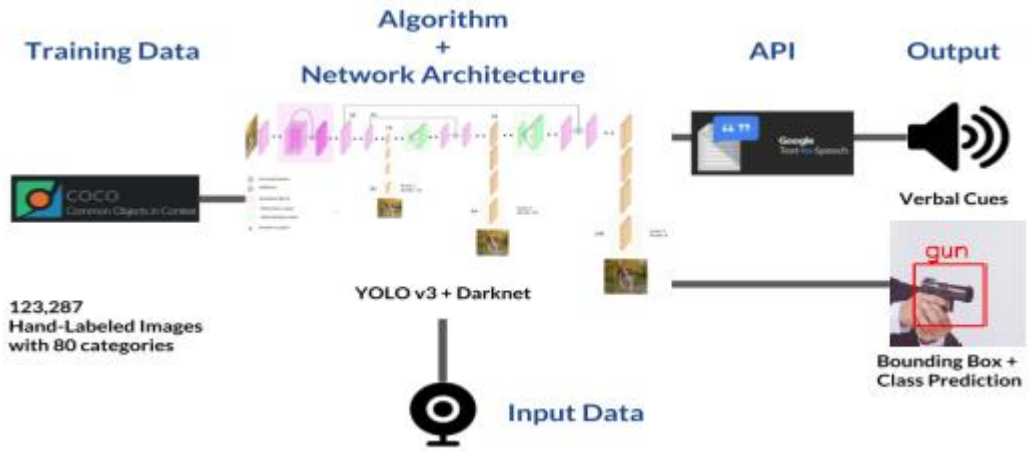

Figure 5.1: Design of the proposed weapon detection system

Description:

1. The system model is trained using hand labeled images of the target object for detection.

2. The system consists of a camera gives the live feed of images as input for detection.

3. The YOLOv3 is the algorithm used for detection using the weights obtained by training.

4. The detected object is marked by a bounding box and the lable name i.e the class name is previewed.

5. Alert is sent to the user using an api known as twillio.

5.1.1.1 Phases in weapon detection System

The object detection system is trained using supervised learning approach in which it takes live feed from a camera as input. The system includes the training and testing phase followed by object classification and detection. Weapon detection and classification are carried training weights which aid the mode in recognizing similar objects in various unseen circumstances like low lighting or partial visibility. When weapon is detected depending upon the confidence value, the user is contacted. Phases for implementing the Proposed Project are:

1) Live Video feed as input

2) Detection of firearm or knife

3) Alerting via call

5.1.1.2 Sequence Diagram

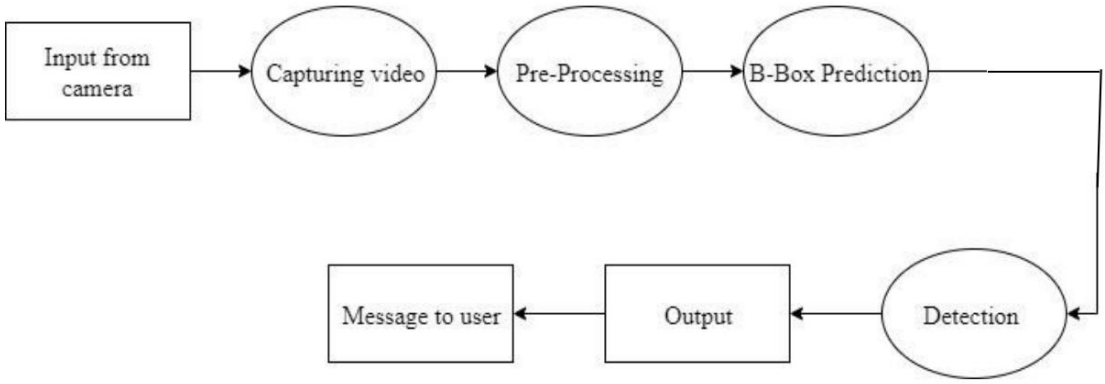

Figure 5.2 Sequence Diagram for the system

Discription:

1. Camera is used to capture the video and each frame is processed.

2. A single neural network is applied to the full image.This network divides the image into regions and predicts bounding boxes and probabilities for each region.

3. These bounding boxes are weighted by the predicted probabilities.

4. It looks at the whole image at test time so its predictions are informed by global context in the image, thus model has several advantages over classifier-based systems.

5. As the object is detected an alert message is sent to the user.

5.1.1.3 System Flowchart

The flowchart of the proposed system is depicted in figure 5.2 shown below:

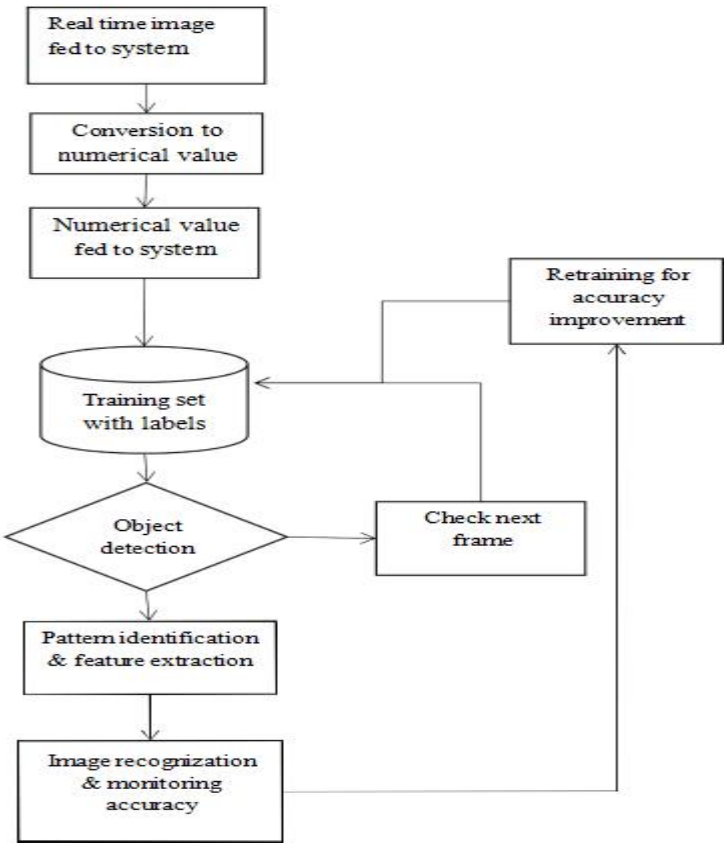

Figure 5.3: Flowchart of the proposed system

Description:

1. The real time video fed to the system which is converted to numerical values.
2. The values are then matched with the values used while training for object recognition.
3. If the object is detected by matching the values, then the object in the environment is localized by a bound box around it, along with the class name it belongs to.
4. Training the system with a larger data set would provided better results.

Chapter 6

Implementation

6.1 Pseudocode

Pseudocode is a step-by-step written outline of the code that can gradually transcribed into the programming language.

//Pseudocode for motion and object detection

1: Calculate pythagorean distance between two frame.
2:Use standard deviation to calculate where the motion is significant enough to trigger an alarm. If standard deviation > Threshold value , start object detection
3:Detect objects using the training weights.
4:If object detected with confidence >0.5 show bound box
5:If object detected with confidence >0.9 call make_call()
6:make_call() uses twilio api for contacting the user on the registered number.

6.2 Implementation Code

// Code for motion detection (motion.py)

```
import numpy as np

import cv2

import os

#import main as mainfile

# Set global variables and functions

# We need variables for setting the motion level threshold and display font.

# "sdThresh" is used for motion level threshold.

# "font" is used for setting for for text display on video.

sdThresh = 10

font = cv2.FONT_HERSHEY_COMPLEX

def distMap(frame1, frame2):

    """outputs pythagorean distance between two frames"""
```

```
    frame1_32 = np.float32(frame1)
    frame2_32 = np.float32(frame2)
    diff32 = frame1_32 - frame2_32  # Difference between two frames
    norm32 = np.sqrt(diff32[:,:,0]**2 + diff32[:,:,1]**2 + diff32[:,:,2]**2)/np.sqrt(255**2 + 255**2 + 255**2)
    dist = np.uint8(norm32*255)
    return dist
cv2.namedWindow('frame')
cv2.namedWindow('dist')
# capture video stream from camera source. 0 refers to first camera, 1 referes to 2nd and so on.
cap = cv2.VideoCapture(0)
_, frame1 = cap.read()
_, frame2 = cap.read()
# Begin the main video loop
while True :
    _, frame3 = cap.read()
    # Get frame3's cols and rows matrix.
    rows, cols, _ = np.shape(frame3)
    # take the difference between two frames.
    dist = distMap(frame1, frame3)
    # Now We can shift our frames.
    frame1 = frame2
    frame2 = frame3
    # Apply Gaussian smoothing to even out our distance mapping.
```

```
    mod = cv2.GaussianBlur(dist, (9,9), 0)

    # Threshold this result to retrieve a binary mapping of where motion is taking place.

    _, thresh = cv2.threshold(mod, 100, 255, 0)

    # At this point, we have a binary array that indicates where motion has occurred and where it has not. Now,

    # ..we will use standard deviation to calculate where the motion is significant enough to trigger an alarm.

    _, stDev = cv2.meanStdDev(mod)   # Calculate the standard deviation.

    # Lets show what we found after standard deviation and display that value on the video.

    cv2.imshow('dist', mod)

    cv2.putText(frame2, "Standard Deviation - {}".format(round(stDev[0][0],0)), (70, 70), font, 1, (255, 0, 255), 1, cv2.LINE_AA)

    # If standard deviation is more than our threshold, then print a message.

    if stDev > sdThresh:

        print("Motion detected.. Do something!!!")

        cap.release()

        cv2.destroyAllWindows()

        import main as mainfile

        mainfile

    cv2.imshow('frame', frame2)

    if cv2.waitKey(1) & 0xFF == 27:

        break

cap.release()

cv2.destroyAllWindows()
```

// code for object detection (main.py)

```
import cv2

import numpy as np

import time

# Download the helper library from https://www.twilio.com/docs/python/install

from twilio.rest import Client

from datetime import datetime

## Load YOLO

net = cv2.dnn.readNet("yolov3_custom_train_final.weights","yolov3_custom_train.cfg")

classes = []

with open("yolo.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()]

colors = [(255, 0, 0), (0, 255, 0), (0, 0, 255), (255, 0, 0), (0, 255, 0), (0, 0, 255), (255, 0, 0), (0, 255, 0), (0, 0, 255), (255, 0, 0), (0, 255, 0), (0, 0, 255), (255, 0, 0), (0, 255, 0), (0, 0, 255), (255, 0, 0), (0, 255, 0), (0, 0, 255)]

# Loading images

cap = cv2.VideoCapture(0)

time_start = time.time()

frame_id = 0

font = cv2.FONT_HERSHEY_COMPLEX

def name():

        # This function is used to give a unique name to saved output file based on time
```

```
        now = datetime.now()
        current_time = now.strftime("%H:%M:%S")
        return current_time
        #return ".join(map(str, current_time))
# Make call function
def make_call():
        # Your Account Sid and Auth Token from twilio.com/console
        # DANGER! This is insecure. See http://twil.io/secure
        account_sid = 'AC3d88bcb14fededd0b9e9cc29ee2bb49b'
        auth_token = 'a253f05eda3ce414749bd93d7fdedb5c'
        client = Client(account_sid, auth_token)
        call = client.calls.create(
                        url='http://demo.twilio.com/docs/voice.xml',
                        to='+91XXXXXXXXXX',
                        from_='+14012982112'
                    )
        print(call.sid)
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out = cv2.VideoWriter('output.avi',fourcc, 20.0, (640,480), True)
while True:
        _,frame = cap.read()
        frame_id+= 1
        #  Save Video: FourCC is a 4-byte code used to specify the video codec.
```

```
#  The list of available codes can be found in fourcc.org. It is platform dependent.


# * In Fedora: DIVX, XVID, MJPG, X264, WMV1, WMV2. (XVID is more preferable. MJPG results in high size video. X264 gives very small size video)

# * In Windows: DIVX (More to be tested and added)

#  FourCC code is passed as cv2.VideoWriter_fourcc('M','J','P','G') OR

#  cv2.VideoWriter_fourcc(*'MJPG) for MJPG.

out.write(frame)

print("saving!!")

height, width, channels = frame.shape

# Detecting Objects

blob = cv2.dnn.blobFromImage(frame, 0.00392, (320,320), (0,0,0), True,crop=False)

# Feeding to nets

net.setInput(blob)

outs = net.forward(output_layers)

# Showing informations on the screen

class_ids = []

confidences = []

boxes = []

# Showing informations on screen

for out in outs:

        for detection in out:

                scores = detection[5:]

                class_id = np.argmax(scores)
```

```
                confidence = scores[class_id]
                if confidence > 0.5:
                        # Object Detected
                        center_x = int(detection[0] * width)
                        center_y = int(detection[1] * height)
                        w = int(detection[2] * width)
                        h = int(detection[3] * height)
                        # Rectangle Co-Ordinate
                        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)
                if confidence > 0.9:
                        # Make call Function
                        make_call()
indexes = cv2.dnn.NMSBoxes(boxes, confidences, 0.5, 0.4)
print(indexes)
font = cv2.FONT_HERSHEY_COMPLEX
for i in range(len(boxes)):
        if i in range(len(boxes)):
                        x, y, w, h =boxes[i]
                        label = str(classes[class_ids[i]])
```

```
                    color = colors[i]
                    label = label+"; confidence:  "+repr(confidences[i])
                    cv2.rectangle(frame, (x,y), (x + w, y + h), color, 2)
                    cv2.putText(frame, label, (x, y+30),font, 0.5, color, 2)
        elapsed_time = time.time() - time_start
        fps = frame_id / elapsed_time
        cv2.putText(frame, "FPS:  "+str(fps), (10,30), font, 0.5, (0,0,0), 1)
        cv2.imshow("Video",frame)
        key = cv2.waitKey(1)
        if key == 27:
            break;
cap.release()  # Commenting this line will keep the camera alive all the time
out.release()  # release "out" variable of videowriter
cv2.destroyAllWindows()
```

Chapter 7

System Testing, Results and Discussion

7.1 System Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner.

Software testing is the process of checking whether the developed system is working according to the original objectives and requirements. Software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the project is not aid to be complete.

7.2 Result Analysis

The aim of weapon detection project is to develop an object detection system that has an addition feature to start detection only after detecting some motion and send the user an alert when the object is detected with a high confidence rate. The system was trained using hand labeled data set of different weapons in various positions and different lighting conditions. It is supervised learning as a lot of training examples are used. In each datasets the data were partitioned into two parts for training and testing. All the training examples well labeled using LableImg software and the annotations were saved in yolo format. The test example were not labeled to check the performance after the training. The performance improves with the increase in the number of training examples.

7.3 Summary

The Weapon detection system starts detection as it detects any motion, if a weapon is detected, it captures image of the suspect and calls the users mobile device with an alert message.

Chapter 8

Conclusion and Scope for Future Work

8.1 Conclusion

In this project, we evaluated the performance of the YOLOv3 based detector on real time images and videos. The objective was to minimize the false positive using YOLOv3 algorithm. YOLOv3 based model has been trained with a dataset containing ImageNet and our own customized dataset. It is clear from the results that YOLOv3 has a good detection performance even in low quality videos as than faster RCNN. The advantage of YOLOv3 over Faster RCNN is its speed. And we have developed this system for suspicious activity detection to improving the functionality of the current surveillance cameras to convert them to an intelligent device capable of detecting suspicious activities. The system is also capable of notifying the appropriate authorities about the suspicious activities occurring. The most important aspect about the proposed system is that it works real-time. It takes in live camera streams from the multiple cameras, performs activity detection, and calculates whether it is suspicious or not. And this occurs in real time, thus providing live and real-time updates of suspicious or criminal activities occurring in the specific area.

8.2 Scope for Future Work

As present and future work, we are evaluating reducing the number of false positives, of Faster detection, by preprocessing the videos, i.e.,increasing their contrast and luminosity, and also by enriching the training set with pistols in motion. We will also evaluate different CNNs-based classifier and consider a higher number of classes. Our goal is to improve the system by improving our detection and notifying system by adding various other features such as expanding the dataset, experimenting in different locations. We aim to make our system scalable and improve efficiency in crowded places. Also, generalization for various stores is utmost importance. There is room to increase the accuracy of the system to reduce the false positives and false negatives and to investigate the category of activity and level of impact.

References

[1] https://ieeexplore.ieee.org/document/5418814 (K-means tracker-based object tracking method for digital camera)

[2] https://ieeexplore.ieee.org/document/8603600 (An Algorithm for Obstacle Detection based on YOLO and Light Filed Camera)

[3] https://ieeexplore.ieee.org/document/8078730 (Simple convolutional neural network on image classification)

[4] https://ieeexplore.ieee.org/document/8834260 (Real-Time Detection for Camera Sensing System: Adaptive Cascade Single-Shot Detector)

[5] https://ieeexplore.ieee.org/document/8783187 (Designing and Implementing a Real Time Weapons Detection System on ODROID-XU4)

https://towardsdatascience.com/object-detection-simplified-e07aa3830954

https://towardsdatascience.com/object-detection-using-yolov3-and opencv19ee0792a420

https://machinelearningmastery.com/how-to-perform-object-detection-with-yolov3-in-keras/

https://www.twilio.com/

http://www.robindavid.fr/opencv-tutorial/motion-detection-with-opencv.html

https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088

https://stackoverflow.com/

Personal Profile

Miss Sneha Shetty R,

Asst. Prof.

Project Guide

Prof. Sneha Shetty R received the B.E. degree in Computer Science and Engineering from SDM Institute of Technology College in the year 2015, and M.E. in Computer Science and Engineering form Sahyadri College of Enginnering and Management in 20017.

Her subjects of interest include IOT, Computation in Graphics Processing Unit.

Prof. Sneha Shetty R is a life member of Indian Society of ISTE and member of IEEE.

Name: Anantha Krishna K S

USN: 4SU16CS012

Address: No 8 Anugraha , near mount senoria school road , laggere, bangalore-560058

E-mail ID: ananthaskurthukote@gmail.com

Contact Phone No: 9945615045

Name: Bharath Kishore

USN: 4SU16CS022

Address: House no. 18, Lane no 6, Vaski Vihar, Barnai Road, Bantalab, Jammu, Jammu & Kashmir

E-mail ID:bkishore770@gmail.com

Contact Phone No: 9419292924

Name: Imran S M

USN: 4SU15CS027

Address: #2-5(4) Ashwatha Nagar House, Reshme road, Odilnala post and village, Belthangady taluk, Dakshina Kannada-574217

E-mail ID: imran.malik01@icloud.com

Contact Phone No: 8971727696

Name: Bharath S Jain

USN: 4SU16CS023

Address: #48/49,Valmikimahakavi road, 11th cross, 3rd main Bassapalayout, Hanumanth nagar, Bangalore-560019

E-mail ID: bharathjain@gmail.com

Contact Phone No: 8095659993