Smart Video Surveillance System

PROJECT SUPERVISOR
DR. ZULFIQAR ALI MEMON

PROJECT CO SUPERVISOR
SIR BEHRAJ KHAN

Submitted By

Syed Rayyan Adil	17K-3904
Hassan Saeed	17K-3693
Zakir Yousuf	17K-3748

SUBMISSION DATE
May 25, 2021

Submitted in partial fullment of the requirements for the degree of Bachelor of Science

DEPARTMENT OF COMPUTER SCIENCE
NATIONAL UNIVERSITY OF COMPUTER AND EMERGING SCIENCES
KARACHI CAMPUS

ACKNOWLEDGEMENT

We are Thankful to Allah and His blessings.
We would like to express our utmost gratitude to our supervisor Dr.
Zulfiqar Ali Memon, for his enthusiasm, patience, insightful comments, helpful information, practical advice, and unceasing ideas that have helped us tremendously at all times especially during research and development of this project. His immense knowledge, profound experience, and professional expertise in Computer Vision and Embedded Systems has enabled us to complete this project successfully. Without his support and guidance, this project would not have been possible.

We are also thankful to our FYP Jury and FYP Committee for their guidance at each evaluation.

CONTENTS

1. ABSTRACT

Theft detection (specifically GUN detection) is one of the most problematic crime that citizens face in their everyday life such as at homes, railways, airports or other public places. Since traditional approaches have long since failed to address this issue there is a dire need of a smart surveillance system that is able to distinguish among actions recorded in real time and checks if there is any such activity occurring, would generate alarming condition and sending the notification to the registered number. The proposed system does not waste its memory by recording the activity unnecessarily as it detects on real time, hence saves lot of wastage of memory of hard disk.

2. INTRODUCTION

In today's world theft, crime and burglaries have gone to much extent, which causes damage in several aspects such as in banking, streets crime, shopping malls, educational institutes, housing societies and some others. Video Surveillance System provides an ultimate solution for such activities (specifically GUN detection), used for "Remote Video Monitoring" for the protection mainly against theft, burglaries, also "Loss prevention" to protect the assets, Public Safety: routinely used for streets, malls, communities, and neighborhoods to help deter crime and enhance public safety. Through surveillance cameras, crimes can be prevented from such happenings and cases could be solved with material evidence. For meeting the security objectives, cameras must be visible and must be able to record, store and transmit footage. In recent years, a great influence by deep learning in the field of machine learning especially in the object detection for classification, and CNN achieved great results in this respect.

In this project, we aim to detect the gun held by a person in the monitoring areas through cameras using Convolutional Neural Network and YOLO, and recognizing the face with the age classification of the person using some machine learning and deep learning algorithms such as FaceNet and AgeNet-Caffe Model, which produces great results in the real-time.

3. PROJECT OBJECTIVE

The main concern of the project is the detection of guns and the person holding the in the premises of the monitoring area, also recognizing as known or unknown to the organization and it will be achieved by applying algorithms like You Only Look Once (YOLO) and Convolutional Neural Network (CNN). After successful detection of the gun held by a person, an alarm will be generated in that area and notification is sent to the relevant authority, which indicates an anomalous activity with respect to the gun detection happened in the monitoring area of the premises.

4. MOTIVATION

The need of these systems is increasing now a days as theft and burglary cases are growing rapidly. For this purpose, CCTV systems are used in different banks, malls, streets etc. In earlier stages, the cameras are capable of only monitoring the premises where they get installed and act passively. It needs an operator who can monitor the system manually after the activity got reported. Now this system works more effectively and efficiently which detects the activity in which weaponry (i.e. gun) is used at run time and generate the alarm for notification purpose. The vision of this project is to provide security by detecting gun held by a person, through system without the help of any monitoring person.

5. LITERATURE REVIEW

5.1 DATASET

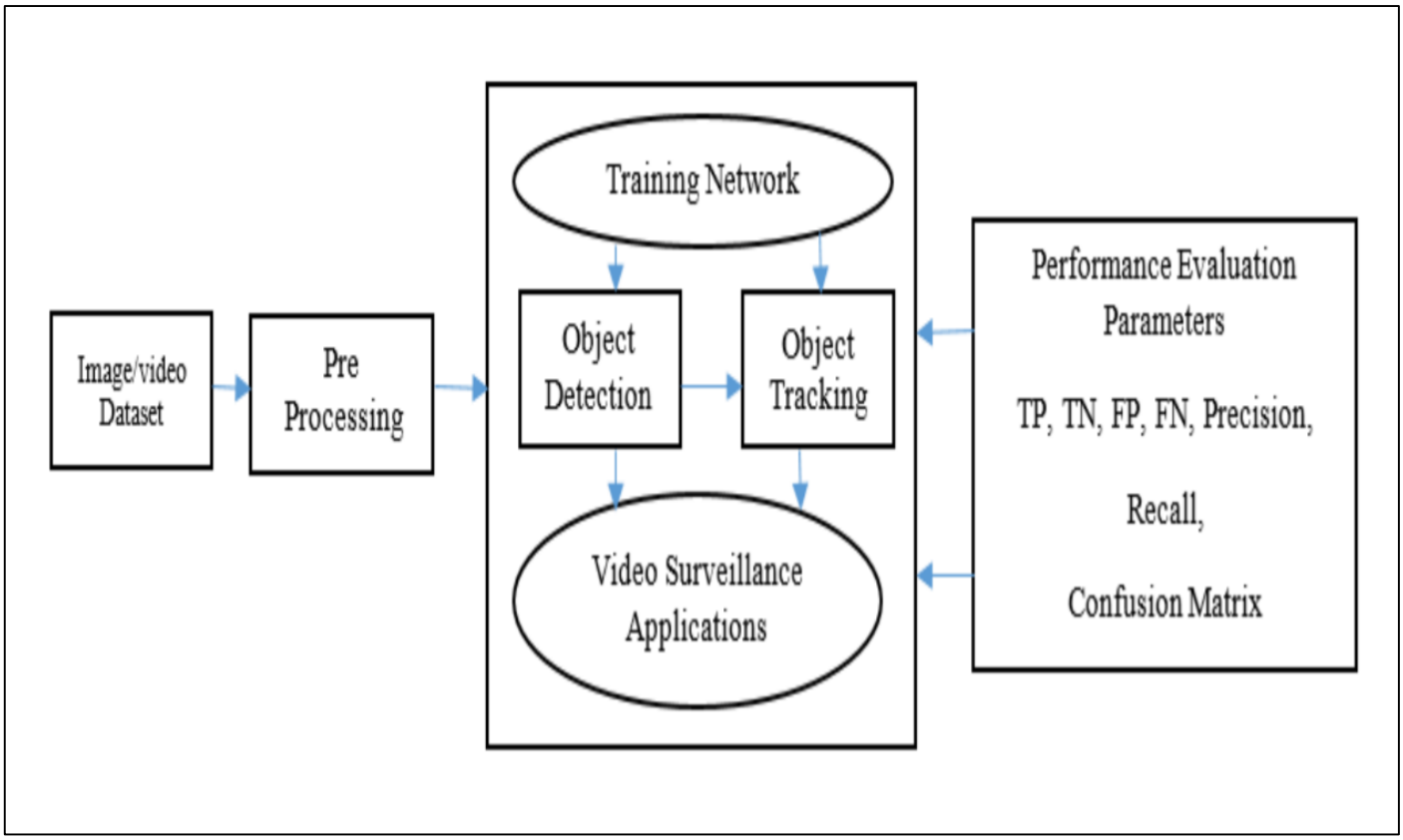
The dataset is being collected from Kaggle and other online resources. In the initial phase of the project, we worked on small datasets of good quality images to test our system's performance. After that, we collected data in the form of CCTV footage videos for training the system on low quality images and identify the results in real form scenarios.

5.2 RESEARCH WORK

Video surveillance system as a whole is about detecting all the burglary and theft activities, means to automate the process and provides a better result in the field of security. Several algorithms, approaches, methods are used to restrict the theft and enhance the system with regards to the security. In our project, a system is developed to detect gun in the respective premises where the camera is installed, and for that we use Convolutional Neural Network and YOLO (You Only Look Once) architecture. Different approaches have been used and implemented in this frame of work and all of them are fine enough in one way or other. Models such as RCNN, FAST R-CNN, CNN, SSD, YOLO, HOG, etc. All the frameworks have their own significance such as object detection, noise removal, quality of frames and etc. Among then we used CNN and YOLO for gun detection held by a person unknown to the organization, recognizing the age, generating alarm in the premises and

notifying the registered number by using some machine learning and deep learning algorithms.

The further details of the above mentioned algorithms are described in the Proposed Work section.



6. OVERVIEW

6.1 SIGNIFICANCE OF THE PROJECT

Video surveillance is one of the trading technologies that is used detect threat and criminal activities like street snatching, burglary in banking sectors, shopping malls etc. This field is basically based on remote technology. In today's modern world, the use of this technology is widely used all across the globe. Mainly computer vision and some methods of deep learning are used in the project.

6.2 DESCRIPTION OF THE PROJECT

The System is programmed in a way that it identifies the gun at real time in the monitoring area along with the person who is holding it and labelling them as known or unknown to the organization. Additionally, the system will recognize the age of a person who is holding the gun and if the age of person is above threshold, the system will generate the alarm and notify the relevant authority, that indicates the anomalous activity in the area.

6.3 PROJECT SCOPE

Basically, the project will cover the boundaries of security. The system will provide safety in terms of detecting the use of illegal weapons i.e. gun. The system will be applicable where camera systems are installed for security and monitoring purposes. The project will detect the human holding weapon (gun) in the premises by applying algorithms of machine learning, deep learning along with Convolutional Neural Network (CNN) and You Only Look Once (YOLO), generating alarm and sending notification to the related organizations.

6.4 NOT IN SCOPE

The project will not include the features of working on facial expressions and anomalous activities. The burglary and robbery activities without the use of guns will not be entertained.

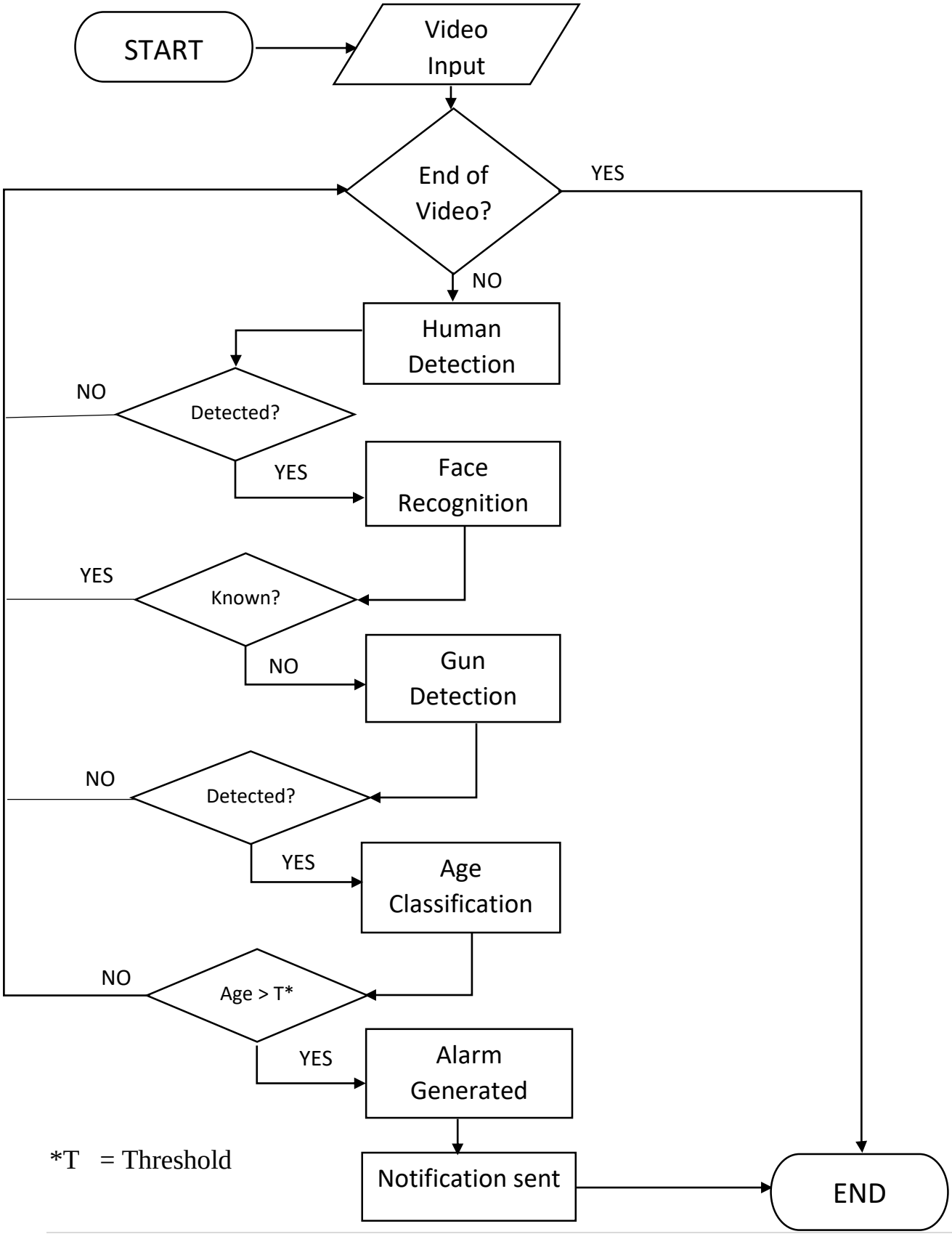
6.5 STAKEHOLDERS

The IT staff of the company where system is installed might be in connection with the system and to handle the issues if occur on sight. The ones who will work on this systems maintenance and to ensure that the quality assurance team will also be the stakeholders. All the areas where system will be placed are the stakeholders of this project.

6.6 OPERATING ENVIRONMENT

The system will operate in particular areas such as companies, malls, banks and streets etc. where monitoring of the premises will take place and where security measure is a main concern like in all banking sectors, malls and most of the streets are now monitored by CCTV camera systems. This project is a new innovation in the field of CCTV systems.

7. PROJECT WORKFLOW



*T = Threshold

8. REQUIREMENTS

8.1 FUNCTIONAL HIERARCHY

- **Camera Monitoring:**

  Camera/System constantly monitoring the on-going activities, and checking whether any gun is detected in the frame or not.

- **Extraction of frames:**

  System extracts the frames from the video and send it to the trained models to carry out the process of gun detection held by a person unknown to the organization.

- **Human Detection:**

  System detects the suspect in the frame, whose coordinates are retrieved to detect the gun held by that person.

- **Burglary action (Gun detection):**

  The camera/system then detects the person holding gun in the coordinates specified for the person by using trained model like YOLO. Afterwards, it will recognize the person as known or unknown to the related organization.

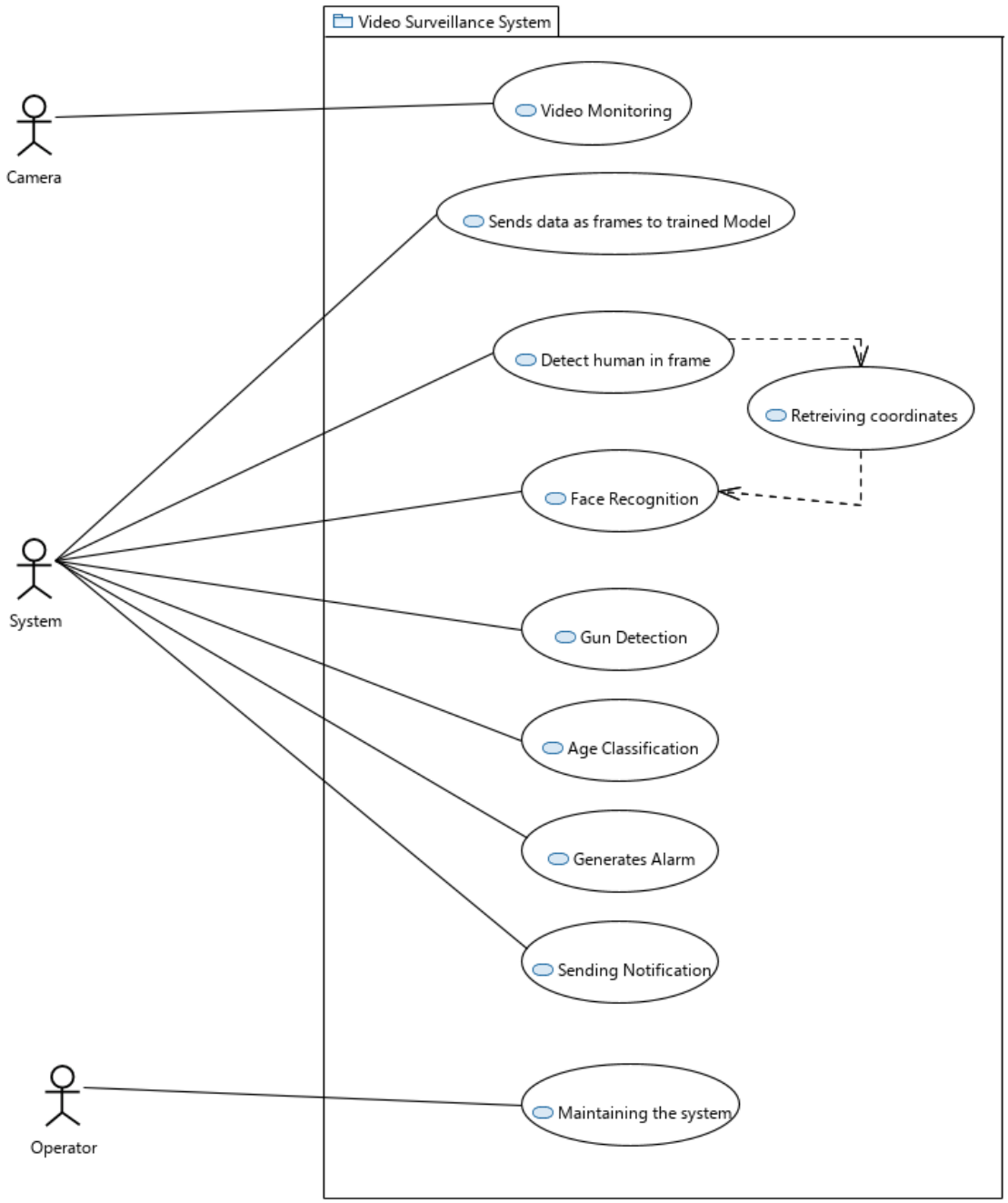
- **Face and Age Classification:**

  Once the face is detected with the label of Unknown, the system then classifies the age of the person who is holding the gun, and label the person with the age.

- **Alerting the relevant authorities:**

  If the labelled age of a person is above the threshold, it will generate the alarm by alerting the system/organization about the theft occurred in that area and by sending notification to the related organization or relevant authorities.

8.1.1 USE CASES



Video Monitoring		
Use case Id:	01	
Actors:	CAMERA	
Feature:	To Detect the GUN held by a person.	
Pre-condition:	Camera should be in operating state. System should be installed. System should be functional/Working properly. Model should be trained.	
Scenarios		
Step#	***Action***	***Software Reaction***
1.	Camera monitoring the particular area.	Camera is sending frames to the system.
Alternate Scenarios:		
1a: Camera and system isn't working properly and fails to detect the gun.		
Post Conditions		
Step#	***Description***	
	Gun is detected in the frame held by a person. System generates the alarm and sends the notification. The frames then analyzed by the system.	
Use Case Cross referenced	Send data as frames to trained model	

Send data as frames to trained model		
Use case Id:	02	
Actors:	SYSTEM	
Feature:	System send the frames for extraction of images.	
Pre-condition:	System should be installed. System should be functional/Working properly. Model should be trained.	
Scenarios		
Step#	***Action***	***Software Reaction***
1.	System sends the frames to the model.	System then detects the images with help of trained model (like CNN/YOLO)
2.	System use these frames for detection.	System detects gun held by a person using model YOLO/CNN.
Alternate Scenarios:		
1a: System fails to send the video as frames. *2a: Gun is not detected.* *3a. Alarm doesn't activate.*		
Post Conditions		
Step#	***Description***	
	Gun is detected in the frame held by a person. System recognizes the human (known/unknown). System generated the alarm and sent the notification.	
Use Case Cross referenced	Detect Human in frame.	

Detect Human in frame		
Use case Id:	03	
Actors:	SYSTEM	
Feature: Detecting human holding gun in the frame and retrieving the coordinates of the detected person.		
Pre-condition:	Camera should be in operating state. System should be installed. System should be functional/Working properly. Model should be trained. System should have sent the frames extracted from video.	
Scenarios		
Step#	***Action***	***Software Reaction***
1.	System detects human holding the gun.	System is labelling the person holding the gun in the frame.
2.	System retrieved the coordinates of the labelled image.	After detecting human in the frame, coordinates are retrieved to detect the gun in it.
Alternate Scenarios:		
1a: System isn't working properly. 2a: Human not detected. 3a. System failed to retrieve the coordinates.		
Post Conditions		
Step#	***Description***	
	Human is detected in the frame. Gun is detected held by a person. Generates alarm and notification is sent.	
Use Case Cross referenced	Face Recognition. Gun Detection	

Retrieving Coordinates		
Use case Id:	04	
Actors:	SYSTEM	
Feature: After successful detection of human, retrieve the coordinates of human detected.		
Pre-condition:	Camera should be in operating state. System should be installed. System should be functional/Working properly. Model should be trained. System should have sent the frames extracted from video. System should have detected the human in the frame.	
Scenarios		
Step#	***Action***	***Software Reaction***
1.	System retrieves the coordinates.	System, after detecting human in the frame, retrieved the coordinates of a person for gun detection.
Alternate Scenarios:		
1a: System isn't working properly. 2a: Human not detected. 3a. System failed to retrieve the coordinates.		
Post Conditions		
Step#	***Description***	
	Human is detected in the frame. Gun is detected held by a person. Coordinates of the detected human is retrieved,	
Use Case Cross referenced	Face Recognition.	

Face Recognition		
Use case Id:	05	
Actors:	SYSTEM	
Feature: Recognizing face of the person holding gun is known or unknown to the related organization.		
Pre-condition:	Camera should be in operating state. System should be installed. System should be functional/Working properly. Model should be trained. System should have sent the frames extracted from video. Human is detected and coordinates of human is retrieved.	
Scenarios		
Step#	***Action***	***Software Reaction***
1.	System recognizes the face of person holding the gun.	System detects the human and then recognizes if a person is known or unknown to the organization.
2.	Labelling the person.	System labels the person with the name if known person else labelled as unknown person.
Alternate Scenarios:		
1a: System isn't working properly. 2a: Human not detected. 3a: System failed to retrieve the coordinates. 4a: System didn't label the person.		
Post Conditions		
Step#	***Description***	
	Human is detected in the frame. Recognition of human is done. Gun is detected held by a person. Generates alarm and notification is sent.	
Use Case Cross referenced	Gun Detection	

Gun Detection		
Use case Id:	06	
Actors:	SYSTEM	
Feature: Detecting gun in the frame where human is recognized as unknown to the organization.		
Pre-condition:	Camera should be in operating state. System should be installed. System should be functional/Working properly. Model should be trained. System should have sent the frames extracted from video. Human is detected, coordinates of human is retrieved, and recognized as unknown.	
Scenarios		
Step#	***Action***	***Software Reaction***
1.	System detects the gun in the frame.	System detects the gun held by a person detected earlier.
Alternate Scenarios:		
1a: System isn't working properly. 2a: Gun not detected.		
Post Conditions		
Step#	***Description***	
	Gun is detected held by a person. Generates alarm and notification is sent to the relevant authority.	
Use Case Cross referenced	Age Classification	

Age Classification		
Use case Id:	07	
Actors:	SYSTEM	
Feature:	Age of person holding gun is classified as the range.	
Pre-condition:	System should be functional/Working properly. Model should be trained. Human is detected holding the gun, coordinates of human is retrieved, and recognized as unknown.	
Scenarios		
Step#	***Action***	***Software Reaction***
1.	System classifies the age of a person in the frame.	After successful detection of gun and human in the frame, it classifies the age of a person.
Alternate Scenarios:		
1a: System isn't working properly. 2a: Gun not detected. 3a: Face not recognized properly and can't classify the image of person.		
Post Conditions		
Step#	***Description***	
	Gun is detected held by a person. Generates alarm and notification is sent to the relevant authority, if the age is classified beyond the standard age limit.	
Use Case Cross referenced	Generate Alarm. Sending notification	

Generates Alarm		
Use case Id:	08	
Actors:	SYSTEM	
Feature:	System generates the alarm, once the human holding gun and age above the threshold is detected.	
Pre-condition:	System should be installed. System should be functional/Working properly. Model should be trained. System should have the human holding the gun is detected.	
Scenarios		
Step#	***Action***	***Software Reaction***
1.	System generates the alarm.	System is generating the alarm on the basis of gun detection.
Alternate Scenarios:		
1a. System doesn't generate alarm.		
Post Conditions		
Step#	***Description***	
	System generated the alarm. Operator then deactivated the alarm. Notification is sent to the authority.	
Use Case Cross referenced	-	

Sending Notification		
Use case Id:	09	
Actors:	SYSTEM	
Feature:	Notification is sent to the relevant authority.	
Pre-condition:	System should be functional/Working properly. Model should be trained. Human is detected holding the gun, coordinates of human is retrieved, and recognized as unknown. Alarm is generated.	
Scenarios		
Step#	***Action***	***Software Reaction***
1.	System sends the notification to the relevant authority.	After successful detection of gun and human in the frame, classified the age of a person, generated the alarm and then sends the notification to the registered number of the related organization.
Alternate Scenarios:		
1a: System isn't working properly. 2a: Gun not detected. 3a: Face not recognized properly and can't classify the image of person, alarm not generated.		
Post Conditions		
Step#	***Description***	
	Generated alarm and notification is sent to the relevant authority, if the age is classified beyond the standard age limit.	
Use Case Cross referenced	-	

Maintaining the System		
Use case Id:	10	
Actors:	OPERATOR	
Feature:	Operator maintains the system	
Pre-condition:	System should be installed. Camera should be in operating state. System should be functional/Working properly. Model should be trained.	
Scenarios		
Step#	***Action***	***Software Reaction***
1.	Operator is maintaining the system regularly.	Camera is being checked, either working properly or not or updating its specification
2.	Management of data	Data is being updated after certain interval or filtered for database management.
Alternate Scenarios:		
1a. System isn't working properly. 2a. Camera stopped working. 3a. Mismanagement of data.		
Post Conditions		
Step#	***Description***	
	Camera worked properly and perfectly monitoring. Data is updated or filtered on certain interval of time.	
Use Case Cross referenced	-	

8.2 NON-FUNCTIONAL REQUIREMENTS

8.2.1 PERFORMANCE REQUIREMENTS

Firstly, the priority was to develop the software covering all the basic features and functionalities then we proceeded to improve its accuracy, execution time, precession etc.

All the above mentioned points have been considered while developing the project and these requirements will be check by the client and quality assurance team when the project will get evaluated.

8.2.2 SAFETY REQUIREMENTS

- **Camera/System Maintenance:**

  Monitoring system should be working properly and maintained or to replace it with new after a certain interval of time, as it is monitoring (constantly) (24/7). And after detecting the burglary (gun detection) action an alarm will be generated, because if the camera isn't working, burglary (gun detection) would not be captured.

- **Storage Management:**

  A proper disk storage management is needed in order to gather more data, so need to filter the frames/images stored in it.

- **Alarm System:**

  Alarm is the main source of alerting the organization about the theft, so it should be checked on regular basis.

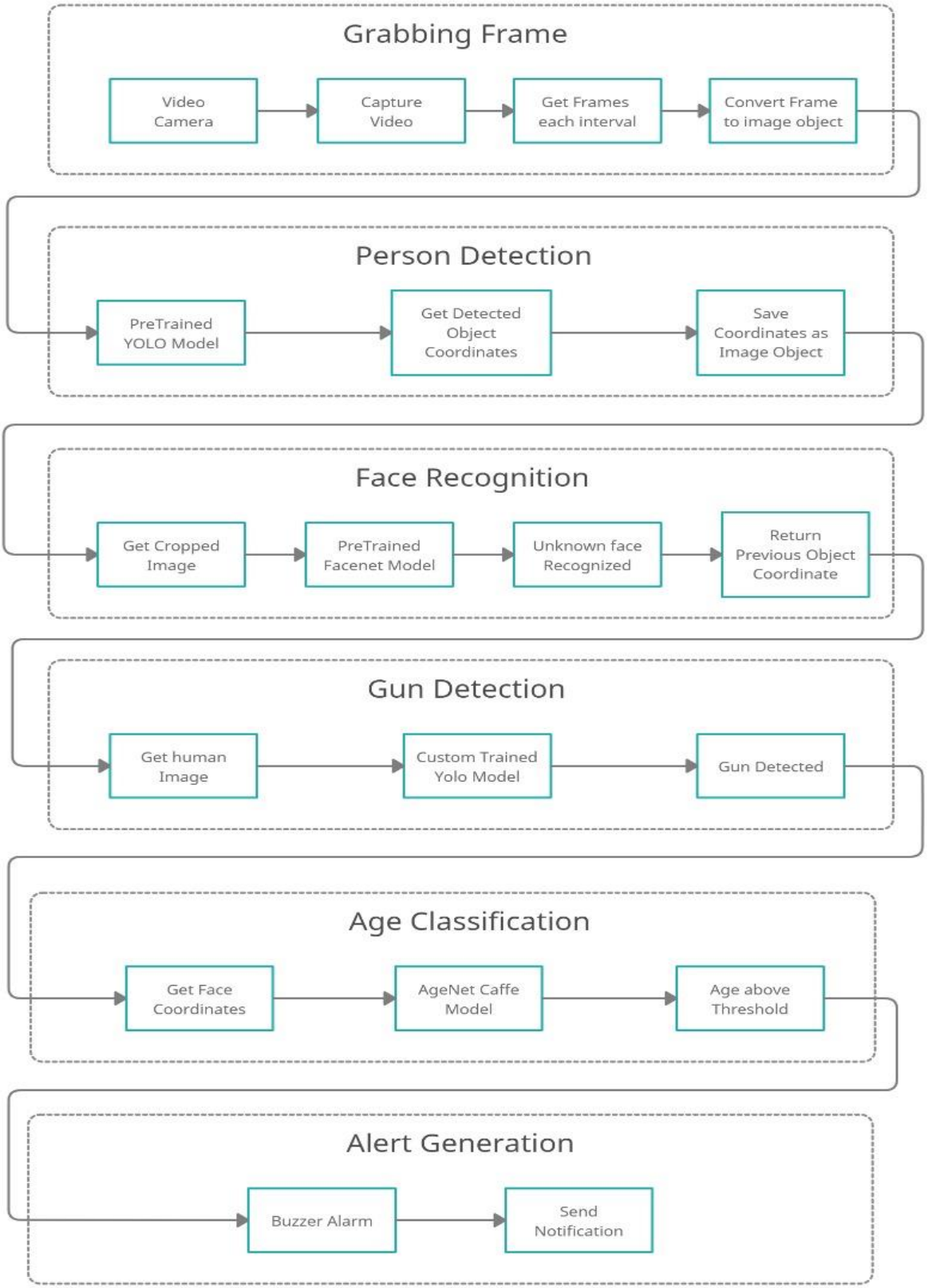
- **Data collection/Training model:**

  Data is collected from different online resources such as KAGGLE, etc, using this dataset we trained our model using YOLO and CNN in order to achieve the better results.

- ***Alarm deactivation***

  After the successful detection of gun held by a person, alarm is generated which can be deactivated by the operator only.

9. SOFTWARE ARCHITECTURE



10. DESIGN STRATEGY

10.1 FUTURE SYSTEM EXTENSION OR ENHANCEMENT

In the initial launch of this system, we cover the most basic points like detection of gun, alarm generation and sending notification to the registered number of the related organization. Furthermore, in near future we set an objective to track suspect by sequence of cameras connected, whereas alerting the relevant authorities, that includes all the branches of the related company, bank, office and mall, police etc.

10.2 USER INTERFACE PARADIGMS

Camera is being operated to monitor the scenario and detecting the guns in the frame of that particular area.

The system is operating on windows 10, using Python on Visual Studio code for processing videos, extracting images and detecting frames as gun or not gun, recognizing the age of that person.

Alarm is used to generate noise and sending notification to the related authority that indicates about the theft in the area.

10.3 DATA MANAGEMENT (STORAGE, DISTRIBUTION, PERSISTENCE)

The system is tested on the files that are stored in a folder for checking the accuracy and results of the system.

10.4 CONCURRENCY AND SYNCHRONIZATION

The projects achieve synchronization as the data of the system is shared effectively among the sub-components, like it extracts the frames from video and detects gun in each frame. Similarly, the system is achieving concurrency while providing synchronization as multiple tasks are processing simultaneously like monitoring of video, after detection of gun, system will generate the alarm, recognize the age of the person holding gun and notifying the registered number of the related organization.

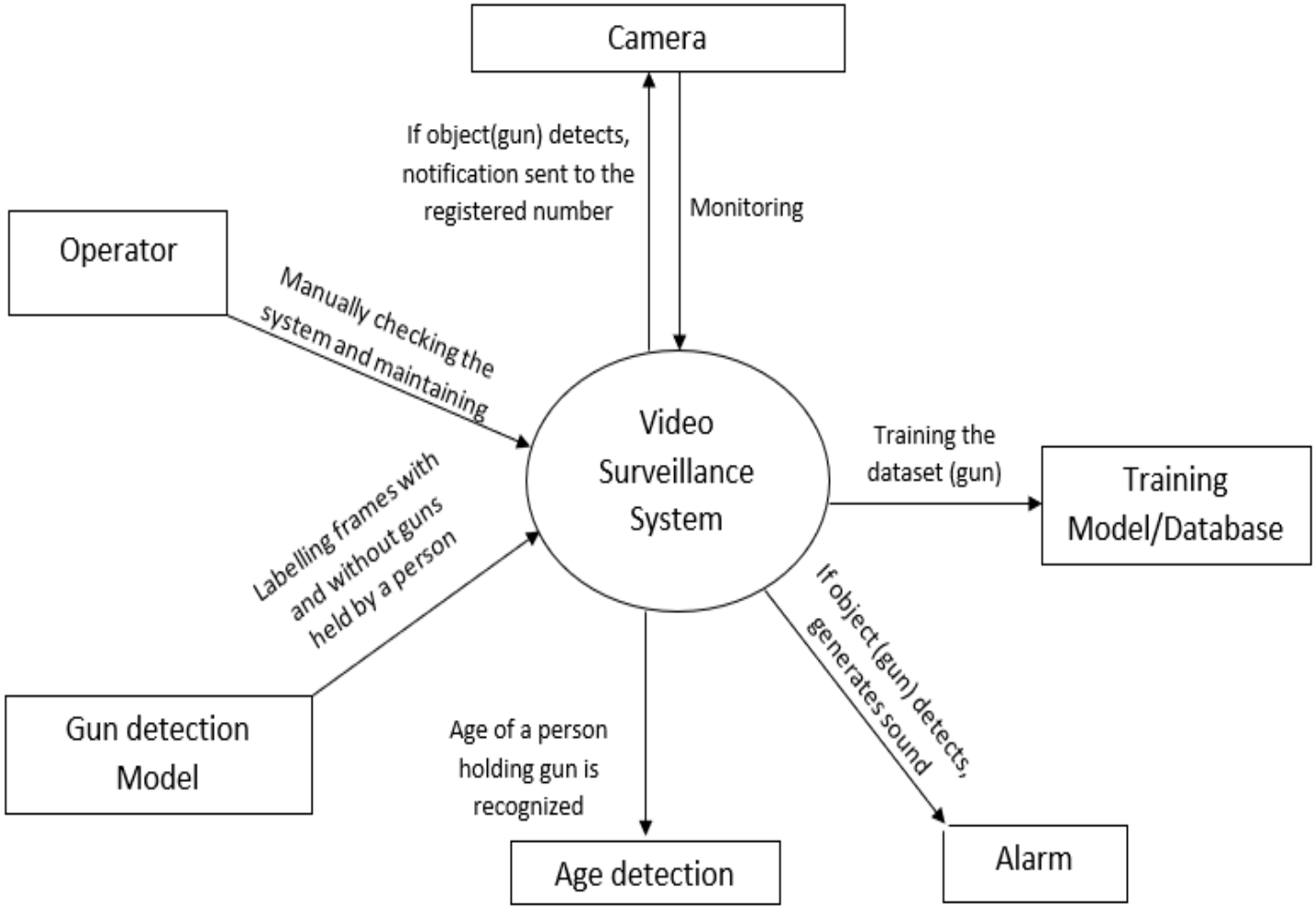


Figure 1: Process Diagram

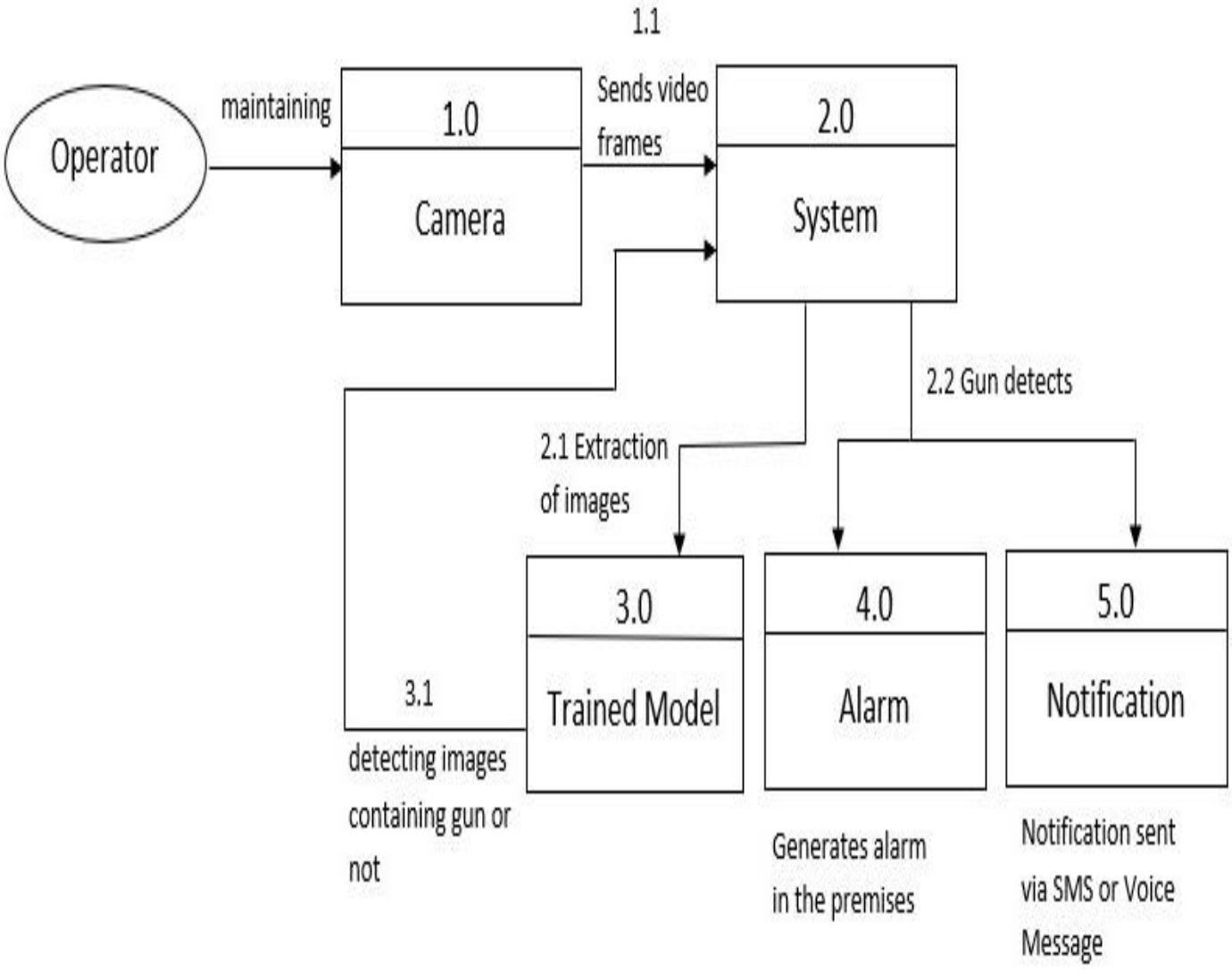


Figure 2: Data Flow Diagram

11. IMPLEMENTATION

PROPOSED WORK:

Here the video is captured by the web camera or the image is processed, the extracted frames are then processed to detect as containing gun or not held by a person.

At the present moment, many researches have been made in detecting the gun in images and video frames. We also did research for image classification classes through which we are able to detect the concealed weapon (gun) in the frame.

If the gun is detected in the monitoring area, then the surveillance system will be able to generate the alarm also notify the registered number by SMS or Voice Message. The algorithms that we choose for our project perspective are widely used in Video Surveillance system, Convolutional Neural Network (CNN) and YOLO along with deep learning and machine learning algorithms for recognizing the age of a person also to notify the registered number about the detection of gun in the particular area.

i. CONVOLUTIONAL NEURAL NETWORK:

The Convolutional Neural Network (CNN) is a deep learning algorithm which takes image or video as input. The pre-processing of the images is done using Keras. CNN based upon some filters which provide a better Spatial and Temporal Dependencies. Hence, it will train the images better. Filters are used in ConvNet to detect spatial patterns i.e., edges and lines of particular given image by detecting the fluctuations in intensity value of the images. The ConvNet split the images into much smaller parts which provides a better predictability for the image without losing any distinct feature.

The image or video that is chosen to detect gun held by a person is tested on the

trained CNN model, recognizing the age of a person and sending notification to the related person of the domain.

ii. YOLO:

Yolo came on Computer Vision scene with the seminal 2015 paper by Joseph Redmon et al. “You Only Look Once: Unified, Real-Time Object Detection,” Yolo “You Only Look Once” is an object detection algorithm. Yolo is basically based on regression principle. It processes 45 frames per second. It does not only consider the interest part of the image although it predicts the classes and bound box of the whole image. This approach is effective and efficient then classification-based algorithm because in that algorithm firstly, it looks for the interest part of the image and then classify these interest regions of the images. The processing is done in two steps and it is slow then Yolo which takes one run time to predict the image.

The bounding box is consist of four terms:

I. Center of bounding box (bxby).

II. Width (bw).

III. Height (bh).

IV. Value C correlate with object of the class.

The algorithm uses the images in both training and testing time so it will be able to completely encode contextual information about the classes. Here, the frame is processed then it is cropped and labelled only that person who is holding the gun, also the notification to the registered number is sent when the person detected holding gun is UNKNOWN, else the system ignores if the person is KNOWN to the organization.

12. TESTING AND EVALUATION:

For testing, dataset is collected from CCTV footages and other online resources to check the accuracy and predicting the results of the project.

12.1 CODES

12.1.1 YOLO

```
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()]
height, width, channels = img.shape
blob = cv2.dnn.blobFromImage(img, 0.00392, (416, 416), (0, 0, 0), True, crop=False)
net.setInput(blob)
outs = net.forward(output_layers)
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*1.13)
            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)

for i in range(len(boxes)):
    if i in indexes:
        if (classes[class_ids[i]] == 'person'):
            x, y, w, h = boxes[i]
            label = str(classes[class_ids[i]])
            label2 = label+str(count)
```

Figure 3: Person Detection

```
net = cv2.dnn.readNetFromDarknet("yolov3_custom.cfg", r"yolov3_custom_4000.weights")
classes = ['notGun', 'gun']
img = cv2.imread(FilePath)
hight, width, _ = img.shape
blob = cv2.dnn.blobFromImage(img, 1 / 255, (416, 416), (0, 0, 0), swapRB=True, crop=False)
net.setInput(blob)
output_layers_name = net.getUnconnectedOutLayersNames()
layerOutputs = net.forward(output_layers_name)
boxes = []
confidences = []
class_ids = []
for output in layerOutputs:
    for detection in output:
        score = detection[5:]
        class_id = np.argmax(score)
        confidence = score[class_id]
        if confidence > 0.2:
            center_x = int(detection[0] * width)
            center_y = int(detection[1] * hight)
            w = int(detection[2] * width)
            h = int(detection[3] * hight)
            x = int(center_x - w / 2)
            y = int(center_y - h / 2)
            boxes.append([x, y, w, h])
            confidences.append((float(confidence*100)))
            class_ids.append(class_id)
indexes = cv2.dnn.NMSBoxes(boxes, confidences, .5, .4)

if len(indexes) > 0:
    for i in indexes.flatten():
        x, y, w, h = boxes[i]
        label = str(classes[class_ids[i]])
        confidence = str(round(confidences[i], 2))
        color = [0,0,255]
```

Figure 4: Gun Detection

```
while True:
    ret, frame = video_capture.read()
    bounding_boxes, _ = detect_face.detect_face(frame, minsize, pnet, rnet, onet, threshold, factor)
    faceNum = bounding_boxes.shape[0]
    if faceNum > 0:
        det = bounding_boxes[:, 0:4]
        img_size = np.asarray(frame.shape)[0:2]
        cropped = []
        scaled = []
        scaled_reshape = []
        for i in range(faceNum):
            emb_array = np.zeros((1, embedding_size))
            xmin = int(det[i][0])
            ymin = int(det[i][1])
            xmax = int(det[i][2])
            ymax = int(det[i][3])
            try:
                if xmin <= 0 or ymin <= 0 or xmax >= len(frame[0]) or ymax >= len(frame):
                    print('Face is very close!')
                    continue
                cropped.append(frame[ymin:ymax, xmin:xmax,:])
                cropped[i] = facenet.flip(cropped[i], False)
                scaled.append(np.array(Image.fromarray(cropped[i]).resize((image_size, image_size))))
                scaled[i] = cv2.resize(scaled[i], (input_image_size,input_image_size),
                                       interpolation=cv2.INTER_CUBIC)
                scaled[i] = facenet.prewhiten(scaled[i])
                scaled_reshape.append(scaled[i].reshape(-1,input_image_size,input_image_size,3))
                feed_dict = {images_placeholder: scaled_reshape[i], phase_train_placeholder: False}
                emb_array[0, :] = sess.run(embeddings, feed_dict=feed_dict)
                predictions = model.predict_proba(emb_array)
                best_class_indices = np.argmax(predictions, axis=1)
                best_class_probabilities = predictions[np.arange(len(best_class_indices)), best_class_indices]
                if best_class_probabilities>0.7:
                    cv2.rectangle(frame, (xmin, ymin), (xmax, ymax), (0, 255, 0), 2)
                    for H_i in HumanNames:
                        if HumanNames[best_class_indices[0]] == H_i:
                            result_names = HumanNames[best_class_indices[0]]
                            cv2.rectangle(frame, (xmin, ymin-20), (xmax, ymin-2), (0, 255,255), -1)
                            cv2.putText(frame, result_names, (xmin,ymin-5), cv2.FONT_HERSHEY_COMPLEX_SMALL,
                                        1, (0, 0, 0), thickness=1, lineType=1)

                else :
                    cv2.rectangle(frame, (xmin, ymin), (xmax, ymax), (0, 255, 0), 2)
                    cv2.rectangle(frame, (xmin, ymin-20), (xmax, ymin-2), (0, 255,255), -1)
                    cv2.putText(frame, "Unknown", (xmin,ymin-5), cv2.FONT_HERSHEY_COMPLEX_SMALL,
                                        1, (0, 0, 0), thickness=1, lineType=1)
            except:

                print("error")

    fps = 1/(endtimer-timer)
    cv2.rectangle(frame,(15,30),(135,60),(0,255,255),-1)
    cv2.putText(frame, "fps: {:.2f}".format(fps), (20, 50),cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 0), 2)
    cv2.imshow('Face Recognition', frame)
```

Figure 5:Face Recognition

12.1.2 CNN

```
data = ImageDataGenerator(rescale = 1./255, zoom_range = 0.3,horizontal_flip=True,
rotation_range= 15).flow_from_directory(path,target_size= (224,224),
color_mode= "rgb",classes= ["gun", "no_gun"],batch_size=64)

model= Sequential()

model.add(Convolution2D(16,(3,3),input_shape=(224,224,3),padding = "Same",activation="relu"))
model.add(MaxPooling2D(pool_size=(3,3)))
model.add(Dropout(0.2))
model.add(Convolution2D(32,(3,3),padding = "Same",activation="relu"))
model.add(MaxPooling2D(pool_size=(3,3)))
model.add(Dropout(0.2))
model.add(Convolution2D(64,(3,3),padding = "Same",activation="relu"))
model.add(MaxPooling2D(pool_size=(3,3)))
model.add(Dropout(0.2))
model.add(Flatten())
model.add(Dense(64,activation="relu"))
# model.add(Dropout(0.3))
model.add(Dense(2,activation="sigmoid"))

clbk= keras.callbacks.EarlyStopping(monitor='accuracy',mode='max')
model.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])

history_1 = model.fit(data,steps_per_epoch=10,epochs=5,shuffle=False,callbacks=[clbk])
model.save("Mymodel_1.h5")
```

Figure 6: Training Model

```
def extract_images(fln):
    cam = cv2.VideoCapture(fln)
    try:
        if not os.path.exists('\data'):
            os.makedirs('\data')
    except OSError:
        print ('Error: Creating directory of data')

    currentframe = 0
    dir_path = "/data/"
    while(True):
        ret,frame = cam.read()

        if ret:
            name = './data/frame' + str(currentframe) + '.jpg'
            print ('Creating...' + name)
            # plt.imshow(name)
            cv2.imwrite(name, frame)
            currentframe += 1
        else:
            break
```

Figure 7: Extraction of Images from Videos

```
def gun_detect(fln):
    img = Image.open(fln)
    img.thumbnail((350,350))
    img = ImageTk.PhotoImage(img)
    # Load the model
    filepath = 'Mymodel_1.h5'
    model = load_model(filepath, compile = True)

    new_image = plt.imread(fln)
    # img = plt.imshow(new_image)
    # plt.show()

    classification = ['gun', 'not_gun']
    from skimage.transform import resize
    resized_image = resize(new_image, (224,224,3))
    img = plt.imshow(resized_image)

    predictions = model.predict(np.array( [resized_image] ))
```

Figure 8: Testing Gun Detection

```
def age_detector(frame, list_index):
    parser = argparse.ArgumentParser()
    parser.add_argument('--input')
    args = parser.parse_args()

    faceProto = r"opencv_face_detector.pbtxt"
    faceModel = r"opencv_face_detector_uint8.pb"

    ageProto = r"age_deploy.prototxt"
    ageModel = r"age_net.caffemodel"

    MODEL_MEAN_VALUES = (78.4263377603, 87.7689143744, 114.895847746)
    ageList = ['(0-2)', '(4-6)', '(8-12)', '(15-20)', '(25-32)', '(38-43)',
               '(48-53)', '(60-100)']
    # Load network
    ageNet = cv.dnn.readNet(ageModel, ageProto)
    faceNet = cv.dnn.readNet(faceModel, faceProto)
    padding = 20
    t = time.time()
    frameFace, bboxes = getFaceBox(faceNet, frame)
    if not bboxes:
        print("No Face Detected!")
    for bbox in bboxes:
        face = frame[max(0,bbox[1]-padding):min(bbox[3]+padding,frame.shape[0]-1),
        max(0,bbox[0]-padding):min(bbox[2]+padding, frame.shape[1]-1)]
        blob = cv.dnn.blobFromImage(face, 1.0, (227, 227), MODEL_MEAN_VALUES, swapRB=False)
        ageNet.setInput(blob)
        agePreds = ageNet.forward()
        age = ageList[agePreds[0].argmax()]
```

Figure 9: Age Classification

```
def notify():
    customer_id = "****************************************"
    api_key = "****************************************"

    print("notification send")
    phone_number = "92*********"
    message = "Gun Detected."
    message_type = "ARN"

    messaging = MessagingClient(customer_id, api_key)
    response = messaging.message(phone_number, message, message_type)

    voice = VoiceClient(customer_id, api_key)
    response1 = voice.call(phone_number, message, message_type)
```

Figure 10: Notification via Number

```
def email_alert(subject, body, to):
    msg = EmailMessage()
    msg.set_content(body)
    msg['subject'] = subject
    msg['to'] = to

    user = "abc123@gmail.com"
    msg['from'] = user
    password = "******************"

    server = smtplib.SMTP("smtp.gmail.com", 587)
    server.starttls()
    server.login(user, password)
    server.send_message(msg)
    server.quit()
```

Figure 11: Notification via Email

12.2 RESULTS

12.2.1 YOLO

Figure 12: Gun Detected held by person

Figure 13:Face Recognition

12.2.2 CNN

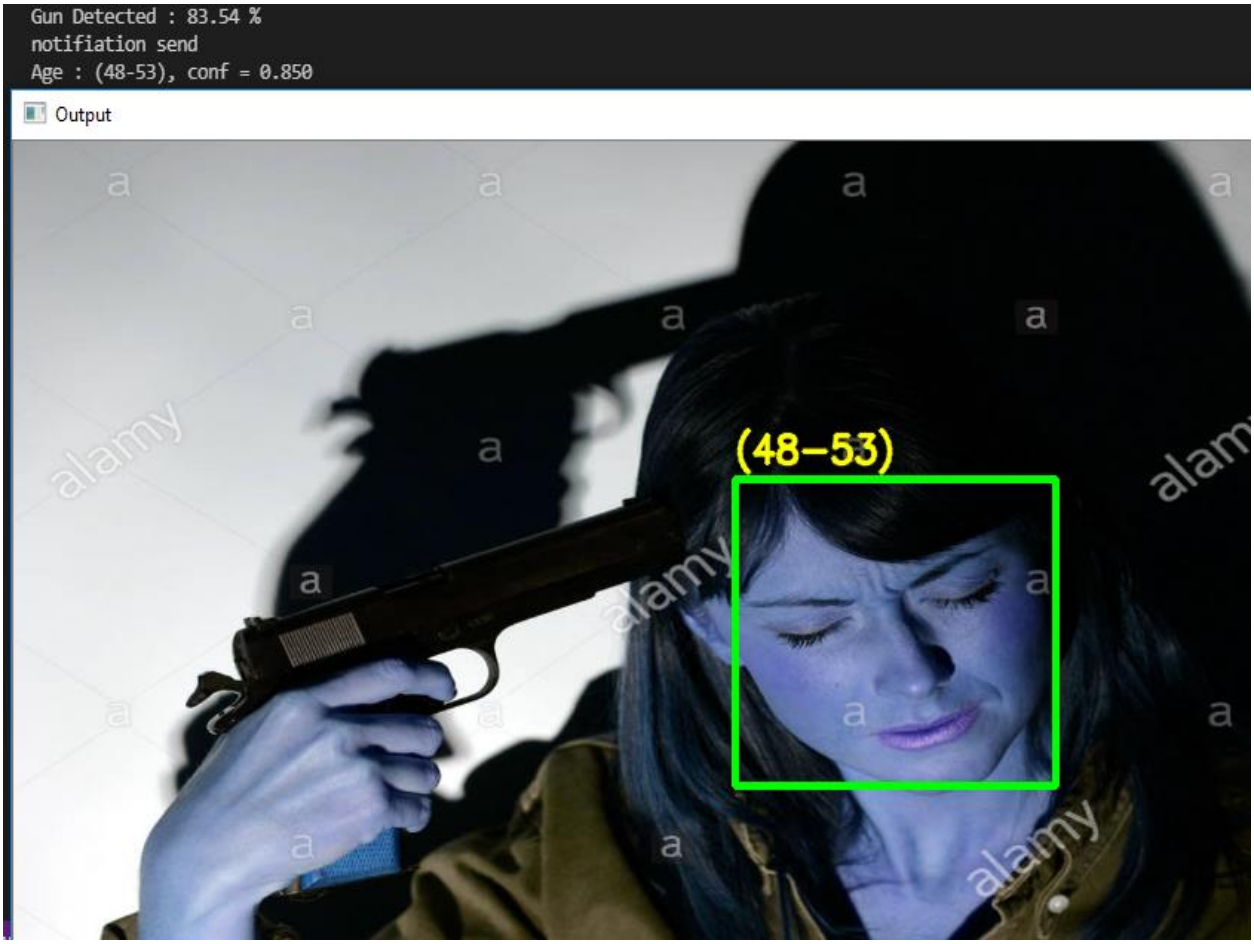


Figure 14: Detected

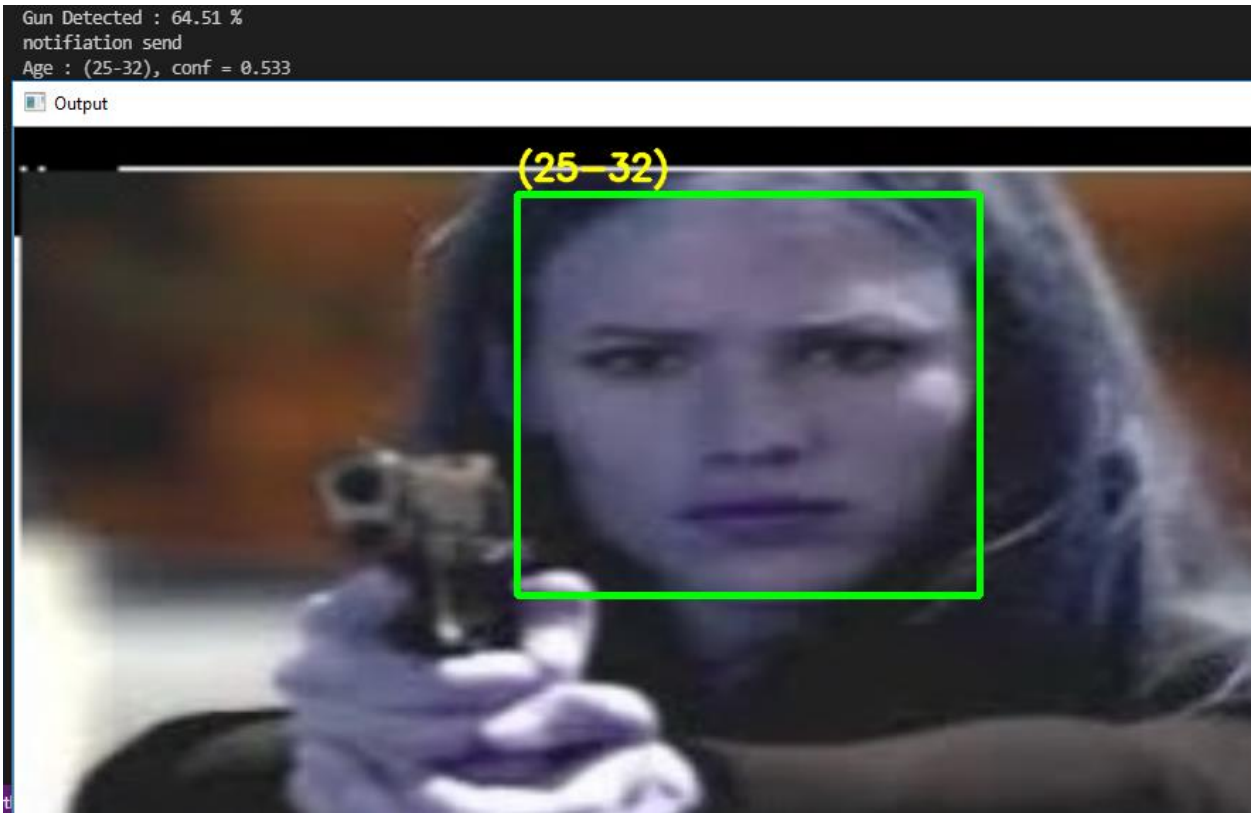


Figure 15:Detected

Figure 16: Not Detected

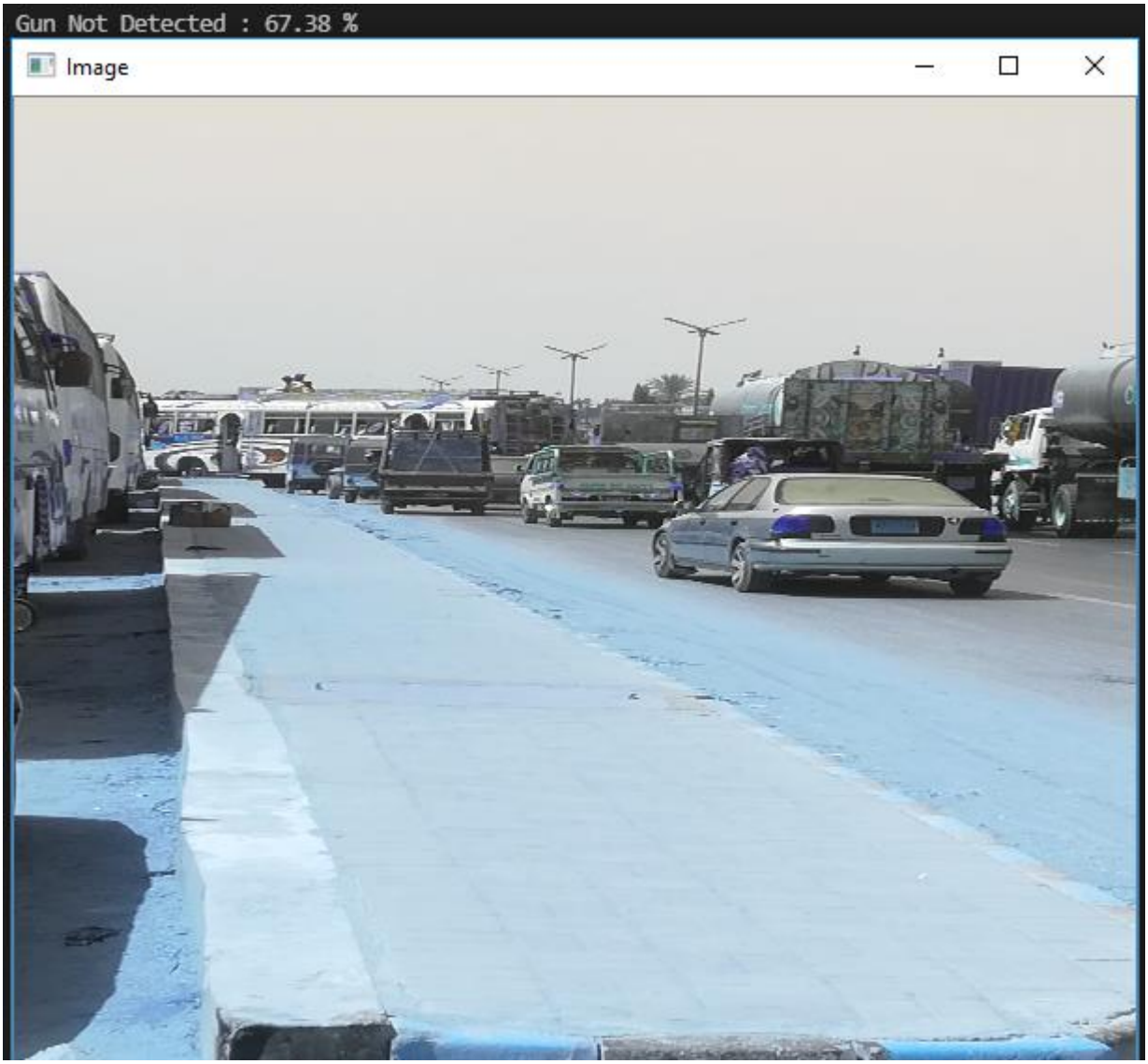


Figure 17: Not Detected

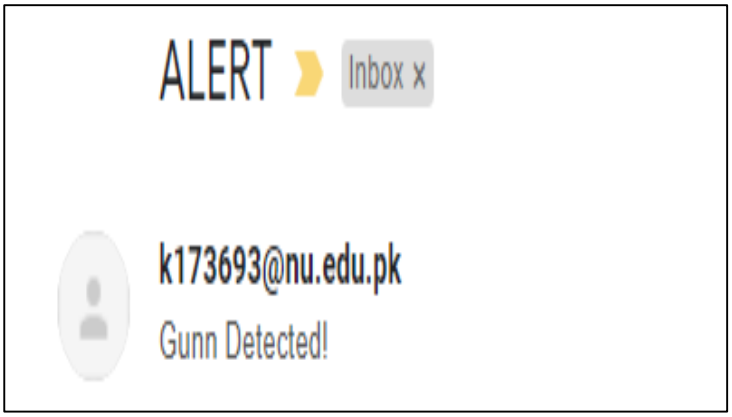


Figure 18: Notification via Email

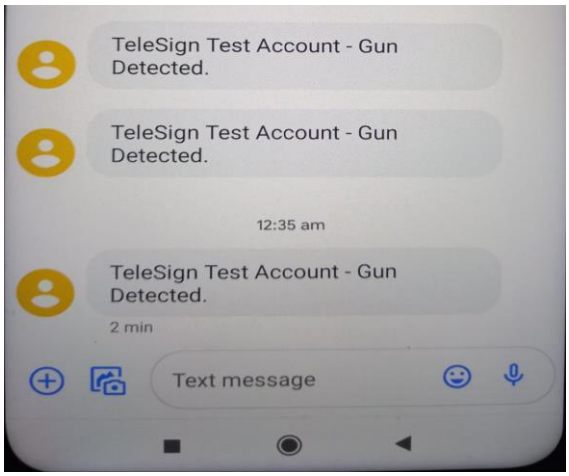


Figure 19: Notification via Message

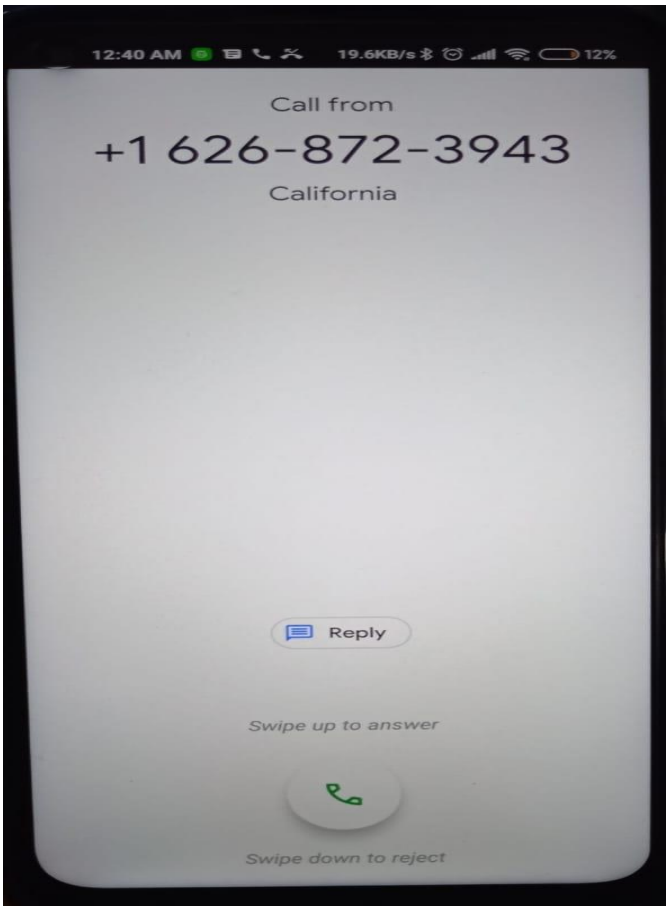


Figure 20: Notification via Voice Message

13. CONCLUSION

In order to provide security around a specific region, many researches and algorithms are implemented to make the environment secure.

In this project, analysis of Video Surveillance System based on specific problem for the detection of gun is carried out, which further proceeded to detect the suspect holding the gun by classifying *it* as unknown person to the organization and age above the threshold, then generating the alarm and notifying the relevant authorities to alert them about the theft.

To do the analysis, we have used the models YOLO and CNN for the detection of gun and human in the premises and recognizing face using FaceNet, whereas classifying age using AgeNET-Caffe Model. This implementation is done on very small scale and the accuracy may vary depending upon the quality of image.

Object tracking still remains a challenging issue which is difficult to analyze on the small scale, but this issue requires more effort and research on large scale, so listed as the future work.

14. REFERENCES

[1] Video Surveillance Systems – A Survey, C. Lakshmi Devasena 1 , R. Revathí 2 , M. Hemalatha 3, IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 4, No 1, July 2011

[2]. AUTOMATIC THEFT SECURITY SYSTEM (SMART SURVEILLANCE CAMERA)
Veena G.S 1, Chandrika Prasad2 and Khaleel K3

[3]. Lun Zhang Li, S.Z. Xiaotong Yuan Shiming Xiang, Beijing, "Real-time Object Classification in Video Surveillance Based on Appearance Learning", Computer Vision and Pattern Recognition, 2007, IEEE Conference Issue Date: 17-22 June 2007, Print ISBN: 1-4244-1180-7, pp.1-8

[4]. Theft detection using computer vision, Patil Siddesh et al.; International Journal of Advance Research, Ideas and Innovations in Technology, ISSN: 2454-132X

[5]. Theft Detection System using Convolutional Neural Network and Object Tracking, Pallav Doshi1, Shubhankar Punktambekar2 , Niraj Kini3 , Simarjeet Singh Dhami 4, Vol-5 Issue-3 2019 IJARIIE-ISSN(O)-2395-4396

[6]. M. Naphade, et al. "On the Detection of Semantic Concepts at TRECVID", ACM International Conference on Multimedia, 2004.

[7]. High Definition Surveillance System Using Motion Detection Method based on FPGA DE-II 70 Board, V. B. Jagdale, R. J. Vaidya, International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-2, Issue-2, December2012

[8]. A Survey on Visual Surveillance of Object Motion and Behaviors, Weiming Hu, Tieniu Tan, Fellow, IEEE, Liang Wang, and Steve Maybank, IEEE TRANSACTIONS ON
SYSTEMS, MAN, AND CYBERNETICS—PART C: APPLICATIONS AND REVIEWS, VOL. 34, NO. 3, AUGUST 2004

[9]. Moving Object Detection Using Scanning Camera on a High-Precision Intelligent Holder, Shuoyang Chen,1 Tingfa Xu,1,2,* Daqun Li,1 Jizhou Zhang,1 and Shenwang Jiang1

[10]. An Improved Motion Detection Method for Real-Time Surveillance, Nan Lu, Jihong Wang, Q.H. Wu and Li Yang, IAENG International Journal of Computer Science, 35:1, IJCS_35_1_16

[11]. Automatic Theft Detection System under Video Surveillance Mr.L. Ashokkumar 1, Mrs.D.Rajeswari2, J.Jovin3, S.Karthikeyan4, Newbin Wilfred5

[12]. Survey Paper on Smart Surveillance System Shivprasad Tavagad1, Shivani Bhosale2, Ajit Prakash Singh3, Deepak Kumar4

[13]. Camera based Smart Surveillance System-Literature Survey Ishan Kokadwar, Anurag Kulkarni, Sayali Khare, Vaibhav Limbhore, Swati Chandurkar

[14].VIDEO SURVEILLANCE SYSTEMS Ľuboš OVSENÍK, Anna KAŽIMÍROVÁ KOLESÁROVÁ, Ján TURÁN

[15]. BOJKOVIČ, Z. – SAMČOVIČ, A. – TURÁN, T.: "Object Detection and Tracking in Video Surveillance Systems", COST 276 Workshop, Trondheim, Norvegia, pp. 113–116, May 25–26, 2005