Research & Reviews: Journal of
Embedded System & Applications
https://journals.stmjournals.com/rrjoesa

ISSN: 2395-6712 (Online)
ISSN: 2321-8533 (Print)
Volume 13, Issue 1, 2025
January–April
DOI (Journal): 10.37591/RRJoESA

Review RRJoESA

Advance Surveillance System Integrated with Weapon Detection and Accident Detection

Amisha Vartak[1,*], Himanshu Ghode[1], Aniket Kashid[1], Prajakta Jadhav[2]

Abstract

Security concerns have become paramount as there is rise in crime rates in crowded events and isolated areas. Abnormal event detection and monitoring system, utilizing computer vision, are crucial for tackling these challenges. In parallel, reducing mortality rates from accidents by ensuring timely emergency response in essential. This study presents the implementation of automatic weapon detection and accident detection. In weapon detection, YOLO v4, Convolutional Neural Networks (CNN), and Faster RCNN algorithms were used. Results indicate both algorithms provide good accuracy, though their real-world application depends on speed and precision. In Accident detection, Keras model, CNN, and RCNN were used. This study also explores the development of Weapon Detection and Accident Detection and Alert System using software to detect weapons and vehicle accidents. In a camera prototype, GPS modules provide the accident location, and GSM sends email notifications. This system ensures instant alerts to emergency services in case of an accident, delivering real-time updates on the vehicle's location. By reducing response time, it enhances the chances of timely medical intervention and increases survival rates.

Keywords: Convolutional neural network, single shot detection, YOLOv4, region convolutional neural networks (R-CNN), object detection

INTRODUCTION

In today's world, public safety is the most important topic to concern, with the rise in global security threats and the increasing number of road accidents. The ability to detect potential threats, such as weapons or respond quickly to accidents, is essential to minimize harm and to ensure timely emergency responses. Traditionally surveillance and monitoring system were manual oversight, which can result in delay detection, human errors and limited scalability [1]. This led to shift toward intelligent systems powered by computer vision and machine learning, which can automatically detect and send alert to relevant emergency services.

***Author for Correspondence**
Amisha Vartak
E-mail: amishavartak1980@gmail.com

[1]Student, Department of Computer Engineering Vishwaniketan's Institute of Management Entrepreneurship and Engineering Technology (VIMEET), Khalapur, Maharashtra, India
[2]Assistant Professor, Department of Computer Engineering Vishwaniketan's Institute of Management Entrepreneurship and Engineering Technology (VIMEET), Khalapur, Maharashtra, India

Received Date: February 02, 2025
Accepted Date: February 17, 2025
Published Date: March 18, 2025

Citation: Amisha Vartak, Himanshu Ghode, Aniket Kashid, Prajakta Jadhav. Advance Surveillance System Integrated with Weapon Detection and Accident Detection. Research & Reviews: Journal of Embedded System & Applications. 2025; 13(1): 47–54p.

Weapon detection in public spaces is of particular concern due to the growing number of incidents involving armed violence (Figure 1). In response to this, this study investigates the implementation of automatic weapon detection using Convolutional Neural Network (CNN) and advance object detection algorithms, such as Single Shot Detection (SSD) and Faster Region Convolutional Neural Networks (Faster R-CNN) [2]. This model has high accuracy in real-time firearm detection, providing alert. This study aims to balance the speed and accuracy which ensure optimal performance for real-world applications. Road accident can cause death and harm in worldwide, not due to lack of medical care or inadequate emergency services, but often because

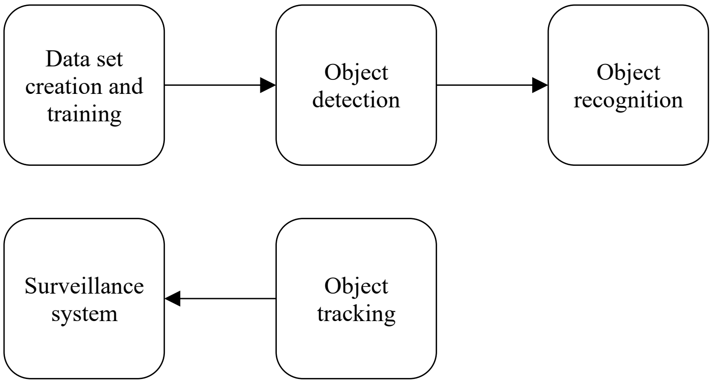


Figure 1. Detection and tracking.

of delayed communication between accident scenes and emergency responders [3]. The ever-growing number of vehicles on the road has led to increase in traffic jam and cause frequent road accidents. In urban areas, accidents primarily result from reckless driving, while in rural areas, intoxicated driving can cause accidents. Other factors, such as driving without seat belts, further elevate the risk of fatalities. Thousands of lives are lost every year because families and emergency personnel were not notified on time. To address this problem many systems were made to detect accident, vehicle tracking and notification, but they still possess certain limitations that require improvement. This system calls for improved emergency response systems, which is the main motive of this research [4].

This study reviews various accident detection, vehicle tracking and notification systems which depend on various technologies and controllers. By sensing various parameters such as nearness of vehicles, collision impact, etc., this system can provide timely notification on system where location can be seen. This system is evolving to provide more efficient and reliable solutions.

Objective of The Project

The main aim of this project is to make an integrated system of weapons detection and accident detection by developing an intelligent real-time detection system. This research offers a comprehensive solution for enhancing public safety. The findings contribute not only to the field of security and road safety but also to the applications to the computer version [5].

Existing System

At the start of modern age of mobile phones, the idea of utilizing GPS sensors for security applications was first proposed. Initially, these solutions required additional hardware which led to the disadvantage of high cost. However, with the massive development in the mobile technology in the past decade and integrate new sensors, the extra hardware can be avoided. Also, there were no integrated system based on weapon detection and accident detection. Today, such applications are available separately in few countries, but by extending their reach, an enable communication with emergency services and families, their effectiveness can be enhanced [6].

Problem Statement

By literature review we get to know that the existing systems associate with image processing, machine learning technology, sensors, Raspberry Pi, MEMS sensors, GPS, GMS module, flame sensors, Arduino which provides average 31–70% of Generalizability, 0–30% of Controllability and 31–70% of Efficiency for created Safe Alert Automated Emergency Detection and Response System.

LITERATURE REVIEW

The system is created to reduce the time between accident occurrence and emergency response, thereby decreasing mortality rates and improving survival chances. It provides detailed accident information and alerts authorities promptly, potentially saving lives. MEMS sensors lack inbuilt analog to digital converters in Raspberry Pi, requiring additional components like ADC for operation. Use Raspberry Pi, MEMS sensors, vibration sensors, GPS, GSM module, flame sensor, pulse sensor, and other components to detect accidents, collect data, and send alerts with location and vital details to emergency services. The study proposes a weapon detection system utilizing the YOLOv5 deep learning model, achieving an F1-score of 95.43%, a detection accuracy of 90.66%, and a Mean Intersection over Union (mIoU) of 88.74%. The methodology involves using YOLOv5 for weapon detection and Mask R-CNN for instance-level segmentation [7].

In modern surveillance and safety system, the integration of accident detection and weapon detection have the significant role which shares the features in terms of objective, technologies used and system architecture. Both systems aim to enhance the public safety through real-time detection and alert system. The integration of both systems can lead to comprehensive solution in smart city, high risk environment and transportation sector [8].

Weapon detection and accident detection both rely on various technologies such as machine learning models, datasets, real-time communication models. These technologies enable efficient detection and identification of weapons and vehicle collision.

Applied deep learning for weapon detection using Faster-RCNN focusing on real-time surveillance and similarly for accident detection, CNN model is trained. Both weapon detection and accident detection depend on deep learning models for expected detection. In weapon detection, models like YOLOv4 have been used for real-time detection for weapon detection in crowded areas [9]. This model detects objects and classifies them with high accuracy and ensures identification of threats. Similarly in accident detection, machine learning models use convolutional neural networks (CNNs) and real time data processing from common ground in both domains. In integrated system, these models could detect both accident and weapons, as shown in Figure 2.

Real-time monitoring and alert system are integral for both weapon and accident detection solutions. In weapon detection, real-time surveillance system uses images fed, processed by YOLOv4 to detect weapons and alert security. Similarly, accident detection system uses CNN model to detect collision of vehicles and alerts nearby police station and medical facilities. While both weapon and accident detection systems share many technological similarities, integrating them can cause real challenge in terms of data processing, real-time efficiency and system coordination [10].

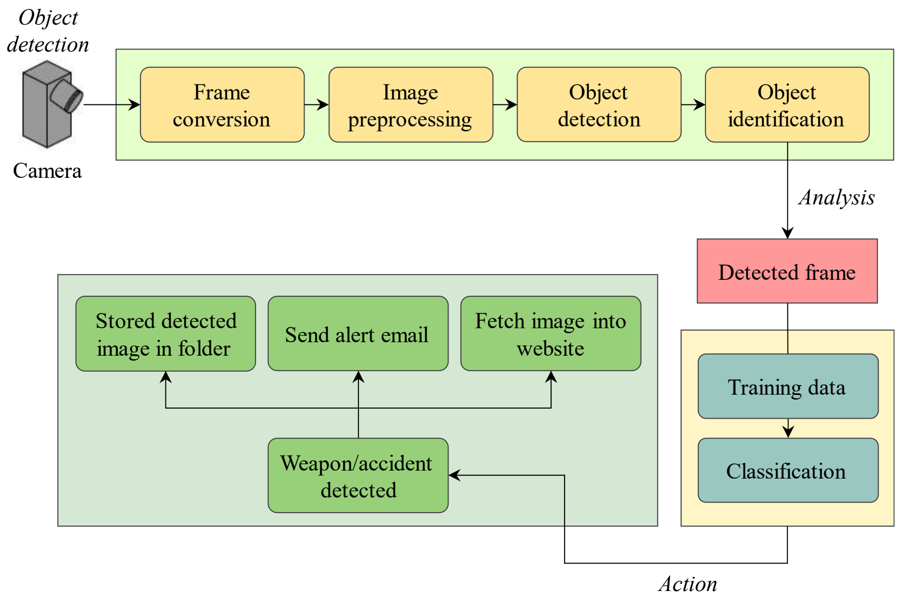


Figure 2. System architecture.

PROPOSED METHODOLOGY

We split our dataset into three sets: 70% training set, 20% validation set and 10% for testing set as shown in Figures 3 and 4.

Precision, Recall and F1-Score

These are used to evaluate the performance of detection models.

- *Precision = True positives/(True positives + False positives)*
- *Recall = True positives/(True positives + False negatives)*
- *F1-Score =2×{(Precision × Recall)/(Precision + Recall)}*

Loss Function

There are three components for the total loss: classification loss, objectness loss, and bounding box loss.

Total loss:

$$L_{total}=L_{class}+L_{object}+L_{box}$$

Where, L_{class} is the classification loss.
L_{object} is the objectness loss.
L_{box} is the bounding box loss.

Region Proposal Network

In Mask R-CNN architecture, the Region Proposal Network (RPN) generates region proposal. The RPN consist of classification loss and regression loss.

$$L_{RPN}=L_{cls}+L_{reg}$$

Where, L_{cls} is the binary cross entropy loss for classifying objects.
L_{reg} is the smooth L1 loss for refining the bounding boxes.

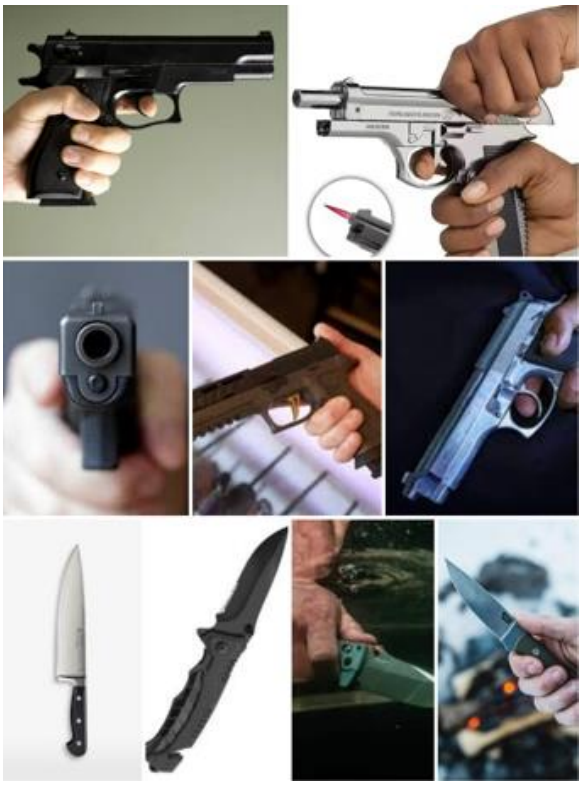


Figure 3. Samples from the weapon detection dataset.

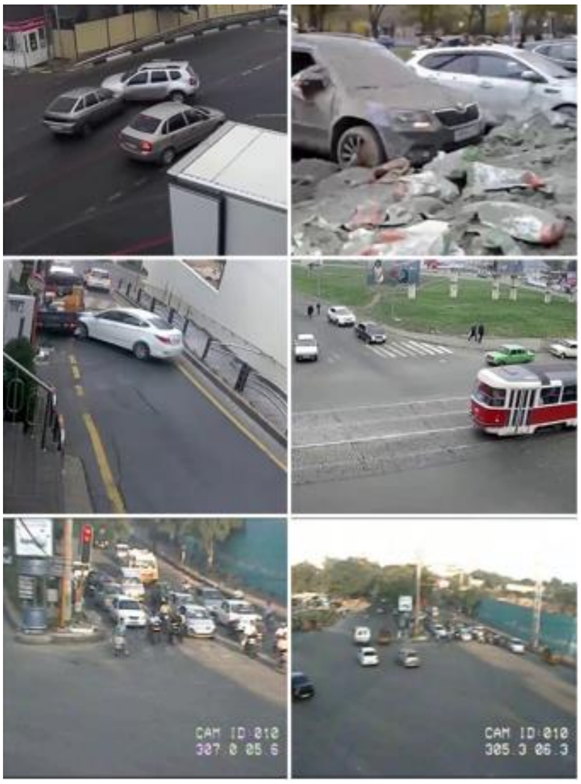


Figure 4. Samples from the weapon detection dataset in vehicle.

Data Preparation

Before proceeding with detection, model setup and training is necessary. Gathering and labeling a dataset, obj.data and obj.names file need to be created. Once we have our data ready, we need to zip both the folders.

Setup

To setup Virtual Machine, we need to do:

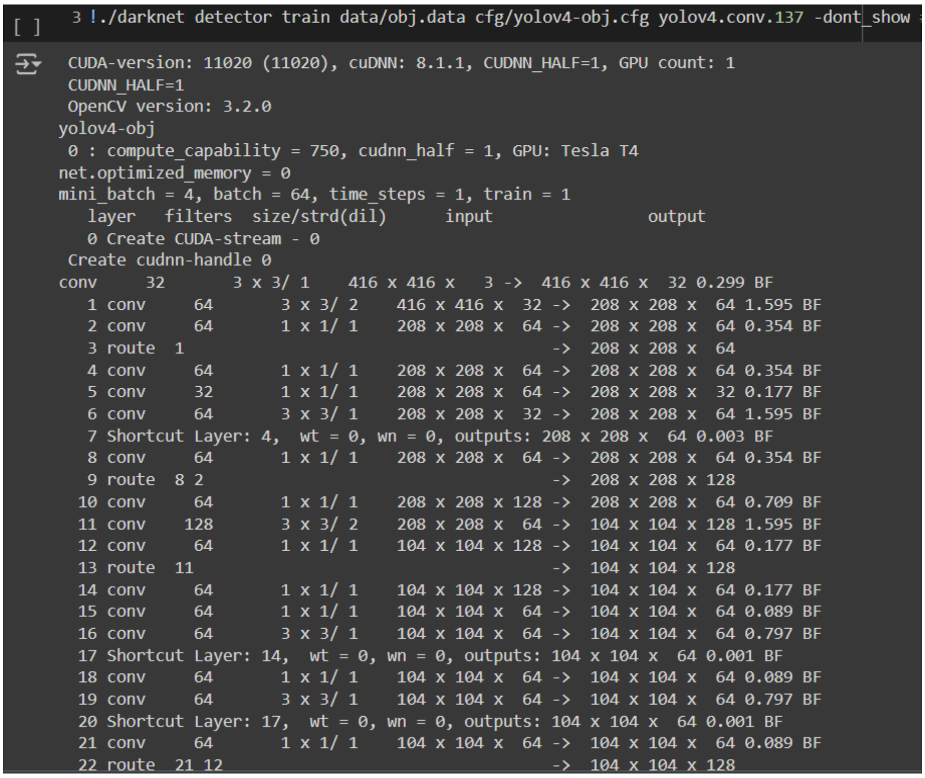
- Enable GPU within the notebook.
- Connect to Google Drive.
- Clone and build Darknet.
- Add data.
- Download .cfg file.
- Generate train.txt and test,txt files.
- Download pre-trained weights.

Data Augmentation

Data augmentation is used to enhance the dataset. Image fusion, geometric alteration, and affine transformations were the techniques used to generate new image. This method is designed to create different versions of original image that replicates weapon and accident appear based on camera angle, light and weather condition, as shown in Figures 5 and 6.

RESULT

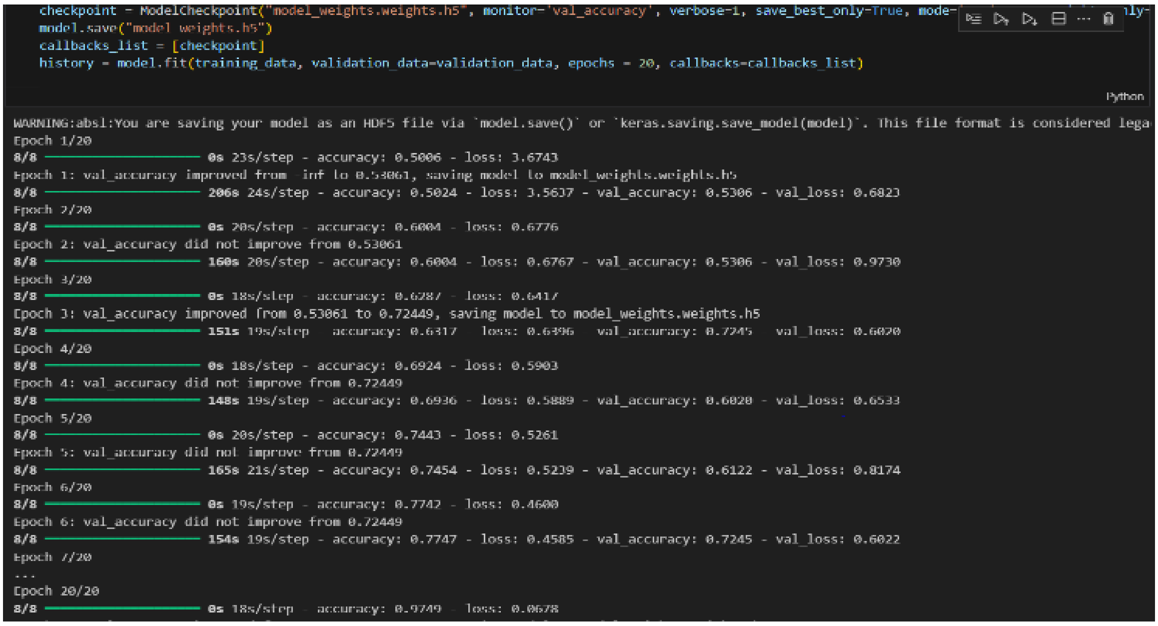
The overall result of this project is that whenever weapon or accident is been detected it sends the alert message to police station and nearby hospitals on website and on email too. This website also contains the analysis report of the year 2022–2023 which includes Cases Under Arms Act, Traffic Fatalities. Website also contains chatbot to have interaction and to solve query of users, as shown in Figures 7–13.



```
[ ]  3 !./darknet detector train data/obj.data cfg/yolov4-obj.cfg yolov4.conv.137 -dont_show

 CUDA-version: 11020 (11020), cuDNN: 8.1.1, CUDNN_HALF=1, GPU count: 1
 CUDNN_HALF=1
 OpenCV version: 3.2.0
yolov4-obj
 0 : compute_capability = 750, cudnn_half = 1, GPU: Tesla T4
net.optimized_memory = 0
mini_batch = 4, batch = 64, time_steps = 1, train = 1
   layer   filters  size/strd(dil)      input                output
   0 Create CUDA-stream - 0
 Create cudnn-handle 0
conv     32       3 x 3/ 1    416 x 416 x   3 ->  416 x 416 x  32 0.299 BF
   1 conv     64       3 x 3/ 2    416 x 416 x  32 ->  208 x 208 x  64 1.595 BF
   2 conv     64       1 x 1/ 1    208 x 208 x  64 ->  208 x 208 x  64 0.354 BF
   3 route  1                                    ->  208 x 208 x  64
   4 conv     64       1 x 1/ 1    208 x 208 x  64 ->  208 x 208 x  64 0.354 BF
   5 conv     32       1 x 1/ 1    208 x 208 x  64 ->  208 x 208 x  32 0.177 BF
   6 conv     64       3 x 3/ 1    208 x 208 x  32 ->  208 x 208 x  64 1.595 BF
   7 Shortcut Layer: 4,  wt = 0, wn = 0, outputs: 208 x 208 x  64 0.003 BF
   8 conv     64       1 x 1/ 1    208 x 208 x  64 ->  208 x 208 x  64 0.354 BF
   9 route  8 2                                  ->  208 x 208 x 128
  10 conv     64       1 x 1/ 1    208 x 208 x 128 ->  208 x 208 x  64 0.709 BF
  11 conv    128       3 x 3/ 2    208 x 208 x  64 ->  104 x 104 x 128 1.595 BF
  12 conv     64       1 x 1/ 1    104 x 104 x 128 ->  104 x 104 x  64 0.177 BF
  13 route  11                                   ->  104 x 104 x 128
  14 conv     64       1 x 1/ 1    104 x 104 x 128 ->  104 x 104 x  64 0.177 BF
  15 conv     64       1 x 1/ 1    104 x 104 x  64 ->  104 x 104 x  64 0.089 BF
  16 conv     64       3 x 3/ 1    104 x 104 x  64 ->  104 x 104 x  64 0.797 BF
  17 Shortcut Layer: 14,  wt = 0, wn = 0, outputs: 104 x 104 x  64 0.001 BF
  18 conv     64       1 x 1/ 1    104 x 104 x  64 ->  104 x 104 x  64 0.089 BF
  19 conv     64       3 x 3/ 1    104 x 104 x  64 ->  104 x 104 x  64 0.797 BF
  20 Shortcut Layer: 17,  wt = 0, wn = 0, outputs: 104 x 104 x  64 0.001 BF
  21 conv     64       1 x 1/ 1    104 x 104 x  64 ->  104 x 104 x  64 0.089 BF
  22 route  21 12                                ->  104 x 104 x 128
```

Figure 5. Training the model for weapon detection.



```
checkpoint = ModelCheckpoint("model_weights.weights.h5", monitor='val_accuracy', verbose=1, save_best_only=True, mode=
model.save("model_weights.h5")
callbacks_list = [checkpoint]
history = model.fit(training_data, validation_data=validation_data, epochs = 20, callbacks=callbacks_list)

WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered lega
Epoch 1/20
8/8 ──────────── 0s 23s/step - accuracy: 0.5006 - loss: 3.6743
Epoch 1: val_accuracy improved from -inf to 0.53061, saving model to model_weights.weights.h5
8/8 ──────────── 206s 24s/step - accuracy: 0.5024 - loss: 3.5637 - val_accuracy: 0.5306 - val_loss: 0.6823
Epoch 2/20
8/8 ──────────── 0s 20s/step - accuracy: 0.6004 - loss: 0.6776
Epoch 2: val_accuracy did not improve from 0.53061
8/8 ──────────── 160s 20s/step - accuracy: 0.6004 - loss: 0.6767 - val_accuracy: 0.5306 - val_loss: 0.9730
Epoch 3/20
8/8 ──────────── 0s 18s/step - accuracy: 0.6287 - loss: 0.6417
Epoch 3: val_accuracy improved from 0.53061 to 0.72449, saving model to model_weights.weights.h5
8/8 ──────────── 151s 19s/step - accuracy: 0.6317 - loss: 0.6396 - val_accuracy: 0.7245 - val_loss: 0.6020
Epoch 4/20
8/8 ──────────── 0s 18s/step - accuracy: 0.6924 - loss: 0.5903
Epoch 4: val_accuracy did not improve from 0.72449
8/8 ──────────── 148s 19s/step - accuracy: 0.6936 - loss: 0.5889 - val_accuracy: 0.6020 - val_loss: 0.6533
Epoch 5/20
8/8 ──────────── 0s 20s/step - accuracy: 0.7443 - loss: 0.5261
Epoch 5: val_accuracy did not improve from 0.72449
8/8 ──────────── 165s 21s/step - accuracy: 0.7454 - loss: 0.5239 - val_accuracy: 0.6122 - val_loss: 0.8174
Epoch 6/20
8/8 ──────────── 0s 19s/step - accuracy: 0.7742 - loss: 0.4600
Epoch 6: val_accuracy did not improve from 0.72449
8/8 ──────────── 154s 19s/step - accuracy: 0.7747 - loss: 0.4585 - val_accuracy: 0.7245 - val_loss: 0.6022
Epoch 7/20
...
Epoch 20/20
8/8 ──────────── 0s 18s/step - accuracy: 0.9749 - loss: 0.0678
```

Figure 6. Training the model for accident-detection.

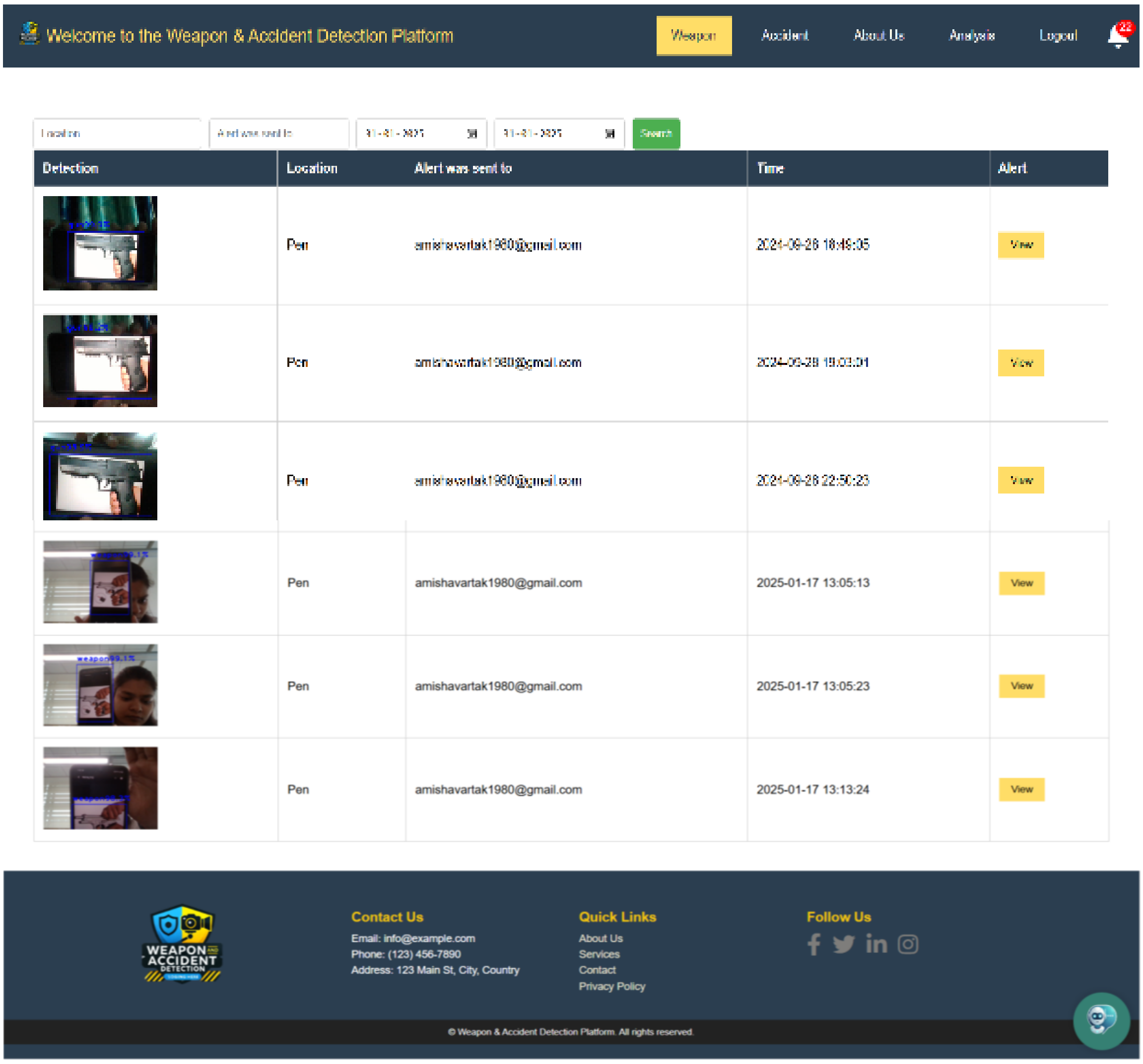


Figure 7. Website image.

Figure 8. (a) and (b) Weapon detection with image.

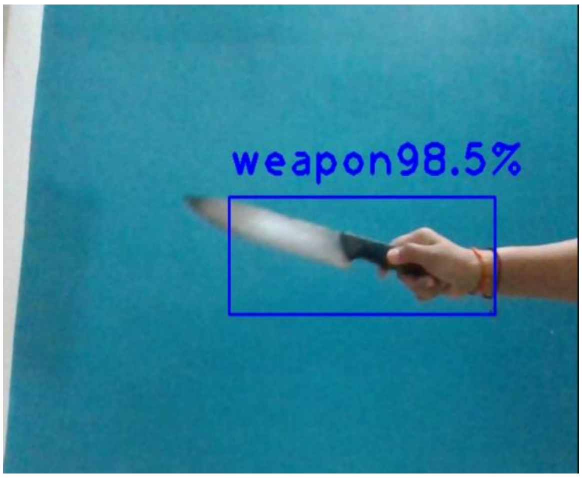


Figure 9. Weapon detection with real weapon.

Figure 10. Accident detection.

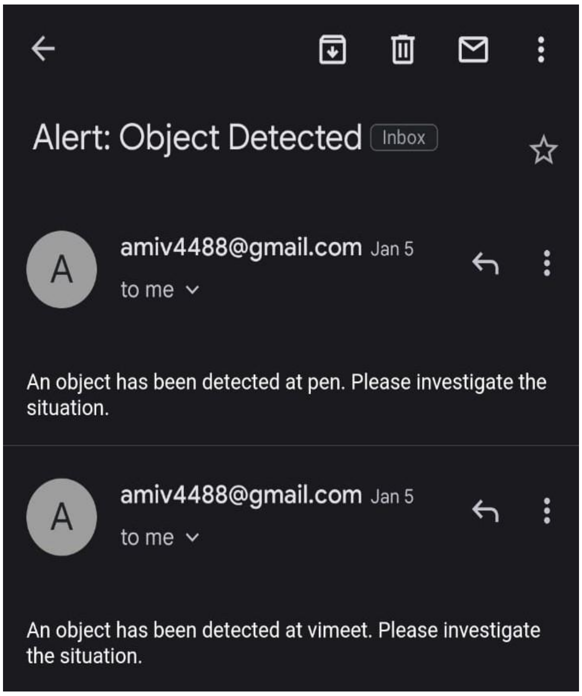


Figure 11. Alert email.

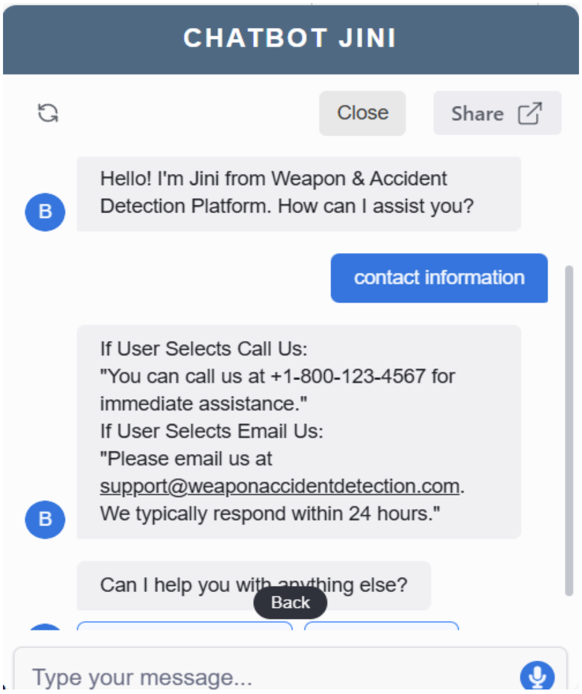


Figure 12. ChatGPT.

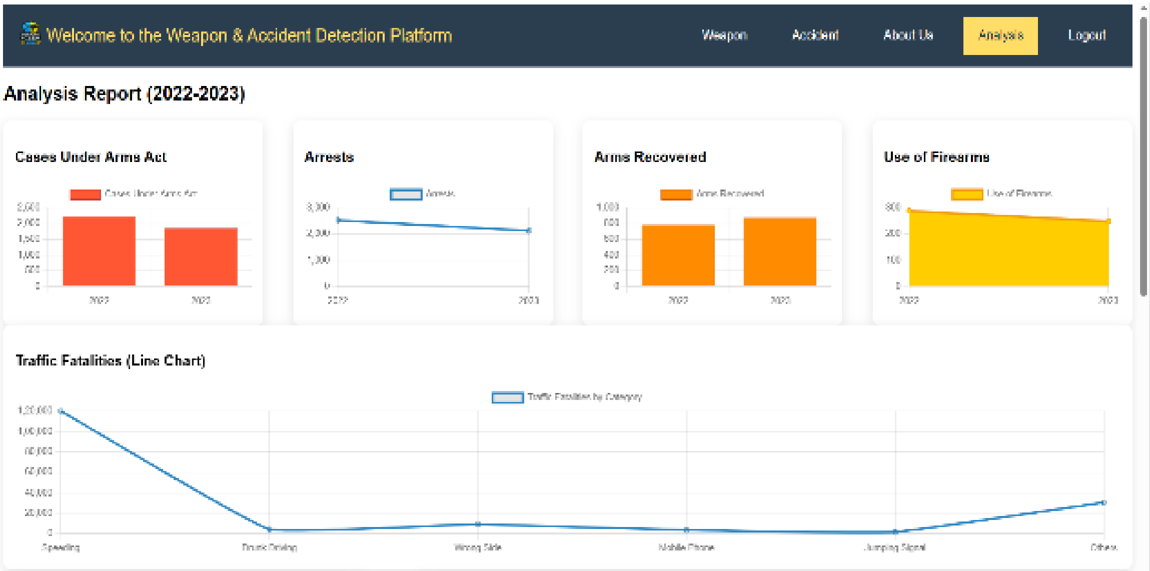


Figure 13. Analysis report.

CONCLUSION

The developed weapon and accident detection system displays the effectiveness of advance deep learning techniques which enhance public safety and security. By using YOLO V4 architecture for rapid object detection and Mask R-CNN for precise instance-level segmentation, the system demonstrates high performance in identifying object and localizing potential hazards in real-time. The deployment of this system can enhance the awareness for authorities and contribute towards effective safety measures. Future work could focus on incorporating additional data sources and sensor inputs to enhance accuracy and response time.

REFERENCES

1. Jain H, Vikram A, Kashyap A, Jain A. Weapon detection using artificial intelligence and deep learning for security applications. In 2020 IEEE International Conference on Electronics and Sustainable Communication Systems (ICESC). 2020 Jul 2; 193–198.
2. Aditya DB, Naresh N, Kumar KV, Raju BG. Accident Detection and Alert System. J Eng Sci. 2023; 14(06): 312–319.
3. Gomathy CK, Rohan K, Reddy BM, Geetha V. Accident detection and alert system. Journal of Engineering, Computing & Architecture. 2022 Mar; 12(3): 32–43.
4. Parameswaran S, Anusuya P, Dhivya M, Banu AH, Kumar DN. Automatic vehicle accident detection and messaging system. Int J Eng Res Technol. 2016; 4(11): 1–3.
5. Singh H, Hand EM, Alexis K. Anomalous motion detection on highway using deep learning. In 2020 IEEE international conference on image processing (ICIP). 2020 Oct 25; 1901–1905.
6. Singh B, Singh D, Singh G, Sharma N, Sibbal V. Motion detection for video surveillance. In 2014 IEEE International Conference on Signal Propagation and Computer Technology (ICSPCT 2014). 2014 Jul 12; 578–584.
7. Parsa AB, Taghipour H, Derrible S, Mohammadian AK. Real-time accident detection: Coping with imbalanced data. Accid Anal Prev. 2019 Aug 1; 129: 202–10.
8. Ghosh S, Sunny SJ, Roney R. Accident detection using convolutional neural networks. In 2019 IEEE international conference on data science and communication (IconDSC). 2019 Mar 1; 1–6.
9. Bari AS, Falalu MA, Umar MA, Sulaiman YY, Gamble AM, Baballe MA. Accident Detection and Alerting Systems: A Review. Glob J Res Eng Comput Sci. 2022 Jul; 2(4): 24–29.
10. Kattukkaran N, George A, Haridas TM. Intelligent accident detection and alert system for emergency medical assistance. In 2017 IEEE international conference on computer communication and informatics (ICCCI). 2017 Jan 5; 1–6.

