REAL-TIME CCTV VIDEO ANALYSIS: DEEP LEARNING FOR WEAPON DETECTION

S. Vivek sagar,
Student,
Department of CSE,
St. Martin's engineering college,
Secunderabad, Telangana, India
singotamviveksagar@gmail.com

E. Soumya,
Assistant Professor,
Department of CSE,
St. Martin's Engineering College,
Secunderabad, Telangana, India
esoumyait@gmail.com

Abstract- CCTV cameras, you know those surveillance cameras you often see in public places, stores, and important buildings, play a crucial role in keeping us safe and secure. They constantly record video footage to monitor what's going on around and ensure our safety. However, as safety concerns grow, it's becoming increasingly important to improve these CCTV systems, making them capable of detecting potential threats, like weapons, in real-time.The current way CCTV surveillance works is that human operators must manually watch all those live video feeds from multiple cameras, which is error-prone, and honestly, a person can only effectively keep an eye on so many cameras at once. With the huge amount of video data these cameras generate, it's practically impossible for human operators to watch everything continuously, which means security threats, might get missed. The traditional approaches passive i.e., the security personnel or operators sit and watch the video feeds, hoping to spot anything suspicious, like someone carrying a weapon. But this ap0proach has its limitations too. Humans can make mistakes, and they might not react quickly enough in a real-time situation. In addition, as the number of cameras increases, it becomes tough to scale this method, and the costs can go up significantly. So, to overcome these challenges and enhance public safety, a more advanced solution is required. Therefore, this project develops a real-time CCTV video analysis with deep learning for weapon detection. By using deep learning models, a sophisticated system can be built that quickly analyzes the video streams from CCTV cameras in real-time. This means it can detect weapons and potential threats as they happen. It's super-fast and accurate, so it reduces the chances of false alarms or missing something important. In addition, it's scalable, cost- effective and helps security agencies respond quickly to potential threats and keeps us all protected in a better and more efficient way.

Keywords: CCTV cameras, Surveillance, Real-time, Weapon detection, Deep learning, Video analysis, Public safety, Threat detection, Scalability, Cost-effective, Security agencies, Response time, False alarms, Efficiency, Monitoring.

I. Introduction

The crime rate across the globe has increased mainly because of the frequent use of handheld weapons during violent activity. For a country to progress, the law-and-order situation must be in control. Whether we want to attract investors for investment or to generate revenue with the tourism industry, all these needs is a peaceful and safe environment [1]. The crime ratio because of guns is very critical in numerous parts of the world. It includes mainly those countries in which it is legal to keep a firearm. The world is a global village now and what we speak or write has an impact on the people. Even if the news they heard is crafted having no truth but as it gets viral in a few hours because of the media and especially social media, the damage will be done [2]. People now have more depression and have less control over their anger, and hate speeches can get those people to lose their minds. People can be brainwashed, and psychological studies show that if a person has a weapon in this situation, he may lose his senses and commit a violent activity [3].

High incidents were recorded in past few years with the use of harmful weapons in public areas. Starting with the past year's attacks on a couple of Mosques in New Zealand, on March 15, 2019 at 1:40 pm, the attacker attacks the Christchurch AL-Noor Mosque during a Friday prayer killing almost 44 innocent and unarmed worshippers. On the same day just after 15 minutes at 1:55 PM, another attack happened killing seven more civilians [4]. Active shooter incidents had also occurred in USA and then in Europe. The most significant cases were those at Columbine High School (USA, 37 victims), Andreas Broeivik's assault on Uotya Island (Norway, 179 victims) or the Charlie Hebdo newspaper attack killing 23. According to stats provided by the UNODC, among 0.1 million people of a country, the crimes involving guns are very high i-e. 1.6 In Belgium, United States having 4.7 and Mexico with a number of 21.5 [5].

CCTV cameras play an important role to overcome this problem and are considered to be one of the most important requirements for the security aspect. [3]. CCTVs are installed in every public place today and are mainly used for providing safety, crime investigation, and other security measures for detection. CCTV footage is the most important evidence in courts. After a crime is committed, law enforcement agencies arrive at the scene and take the recording of footage with them [6].

II. Related work

Bhatti, et al. [10] discussed a deep learning-based system for real-time weapon detection in CCTV videos. It likely presents a methodology and results related to weapon detection. However, this may not provide an extensive evaluation of the model's performance in various real-world scenarios.

Qi, et al. [11] introduced a dataset and system for real-time gun detection in surveillance videos using deep learning. It focused on the dataset's characteristics and system performance. But the dataset used may be limited in diversity and generalization, impacting the system's real-world applicability. In addition, the results presented are very limited.

In [12], authors discussed an algorithm for human pose estimation from CCTV images, likely focusing on the methodology used for posing estimation. However, the algorithm designed for specific types of CCTV images, limiting its generalizability. In addition, it has a lack of benchmarking.

Arya described an automatic and accurate weapons detection model using optimal neural network architecture [13]. But this work hasn't provided thorough validation on diverse real-world data, potentially limiting its practical utility. Moreover, the optimal architecture might require significant computational resources.

Akhila, and Ahmed [14] focused on firearm detection using deep learning, likely presenting a method and its results. However, this work lacksthe detailed information and the system's performance in real-world surveillance scenarios hasn't thoroughly discussed.

In [15] authors discussed a computer vision-based framework for gun detection using the Harris interest point detector. But the use of the Harris interest point detector has a limitation that the model's performance compared to modern techniques. In addition, availability of suitable training data also limited.

Research gap Many of the references focus on the development of deep learning models or algorithms for various types of object detection (e.g., weapons, firearms, human poses) in CCTV or surveillance videos. While these papers contribute to the advancement of object detection techniques, there is a significant research gap in terms of comprehensive real- world evaluation and practical deployment of these models. This gap includes the following aspects:

• Dataset Diversity: Most references may lack diverse and representative datasets for training and evaluation. Real-world surveillance scenarios vary widely, and the absence of such diversity can limit the generalization of these models.

• Performance in Challenging Conditions: Surveillance videos can involve various challenging conditions such as low light, occlusions, and camera angles. There is often limited discussion in the references about how well the models perform under such conditions.

• Real-Time Application: While some references mention real-time detection, the actual real-time performance in practical surveillance systems is often not explored. Real-time object detection has stringent computational requirements that need to be addressed.

• Deployment Challenges: There is limited information on the challenges and considerations for deploying these models in real-world surveillance systems. Deployment may involve hardware requirements, integration with existing systems, and ethical considerations.

• Long-Term Reliability: The references may not address the long-term reliability and stability of these models when deployed in continuous surveillance operations.

• Ethical and Privacy Implications: Ethical and privacy considerations are crucial when deploying surveillance systems. These references may not adequately discuss the ethical implications and potential privacy concerns associated with object detection in public spaces.

ML falls under the larger canvas of Artificial Intelligence. ML seeks to build intelligent systems or machines that can automatically learn and train themselves through experience, without being explicitly programmed or requiring any human intervention.In this sense, ML is a continuously evolving activity. It aims to understand the data structure of the dataset at hand and accommodate the data into ML models that can be used by companies and organizations. Following are the benefits of ML. Enhanced decision-making: ML uses advanced algorithms to improve the decision- making process capacity. It facilitates innovative models and business services simultaneously. It provides a deep understanding of the variations and types of data patterns. You can determine which step to take next based on the variations and data patterns.

• Increases business productivity: It improves the business process and productivity, contributing to business growth. It helps you to adapt to the changing situations at workplaces quickly. The data continue to be updated daily. So, the work environment, too, keeps on changing quickly. ML reduces the chances of error occurrence by half. Hence, it boosts business productivity. This aspect is important to consider when carrying out deep learning vs neural network.

• Removes manual data entry: One of the most common concerns in many organizations is the usage of duplicate records. ML algorithms use predictive models that significantly avoid any errors caused by manual data entry. The corresponding programs use the discovered data to enhance these processes. Hence, the employees can save time to focus on other important business tasks.

• Guarantees customer satisfaction: The ML algorithms are uniquely designed to continue attaining experience with time. They are accurate and efficient. These algorithms improve the machines' decision-making skills. ML can anyhow find a way to make accurate decisions or predictions, although the data is overwhelming and ever-increasing. It benefits businesses with the latest market opportunities related to revenue. As a result, it can satisfy the customers' expectations and boost your business' sales in less time. Moreover, it can quickly recognize threats in the market. You can compare deep learning vs neural networks based on this aspect to have a clear judgment.

• Provides product recommendation: Unsupervised research assists in the development of suggestion systems depending on goods. Currently, most e- commerce platforms use ML to provide product recommendations. ML algorithms use the consumers' purchasing experience to balance it with the assets' huge inventory. This helps in detecting secret trends and connects identical products. Finally, these goods are recommended to the consumers.

• Detects spam: ML is widely used for spam detection. It uses spam filters to identify spam and phishing communications.

• Improves network security: ML improves an organization's security. It helps organizations to develop new systems capable of quickly and efficiently recognizing unknown threats. It can track abnormalities present in network activity and automatically execute relevant actions. When the ML algorithm is used for self-training, it removes manual research and analysis. So, it enhances the organization's network security. Many deep learning neural networks are also used for this purpose.

• Simplifies business analysis: ML is used in business analysis that involves huge volumes of precise and quantitative historical data. It is widely used for algorithmic trading, portfolio management, fraud detection, and lending in finance. The future ML applications for finance will entail Chatbots and a few other interfaces for improving customer service, security, and sentiment analysis. Many neural networks and deep learning algorithms are also used to streamline finance analysis.

III. PROPOSED WORK

The project aims to develop a system that can automatically and rapidly detect weapons in real-time surveillance video streams. This project leverages the YOLO (You Only Look Once) model, a popular deep learning framework for object detection, to achieve its objectives. Below is an overview of the key components and goals of this project:

1. Objectives: The primary objectives of this project include:

• Real-Time Weapon Detection: Developing a system capable of identifying and localizing weapons such as firearms, knives, or other dangerous objects in real- time from live CCTV video feeds.

• Accuracy and Speed: Leveraging the YOLO model to strike a balance between high detection accuracy and real-time processing speed, making it suitable for live surveillance applications.

2. YOLO Model Implementation: The YOLO model, known for its efficiency in object detection tasks, is a crucial component of the project. YOLO divides the image into a grid and predicts bounding boxes and class probabilities for each grid cell. This architecture enables the model to detect multiple objects in a single pass through the network.

3. Dataset and Training: To train the YOLO model for weapon detection, a substantial and diverse dataset of CCTV footage containing instances of weapons is required. The project includes data collection, annotation, and preprocessing to create a suitable training dataset. The YOLO model is then trained on this dataset to learn the features and patterns associated with different types of weapons.

4. Real-Time Video Analysis: The core functionality of the system is real-time video analysis. It involves continuously processing live video feeds from CCTV cameras and applying the trained YOLO model to detect weapons as they appear in the footage. This analysis is performed frame by frame, ensuring that the system can respond immediately to any potential threats.

5. User Interface: For ease of use and monitoring, the project may include a user interface that provides a visual representation of the CCTV video feeds and highlights detected weapons in real-time. This interface can also display alerts and allow security personnel to take action.

6. Evaluation and Testing: The system's performance is rigorously evaluated and tested using various real-world scenarios and conditions. Testing includes assessing detection accuracy, false positive rates, and system responsiveness.

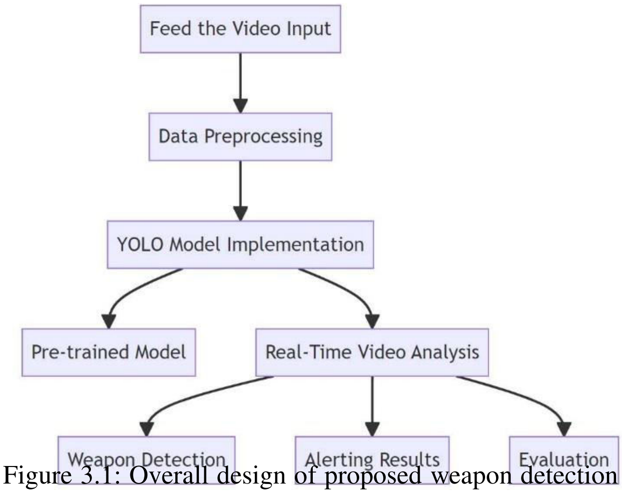


Figure 3.1: Overall design of proposed weapon detection model.

3.1 Data Preprocessing

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this, we use data pre- processing task. A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

• Getting the dataset

• Importing libraries

• Importing datasets

• Finding Missing Data

• Encoding Categorical Data

• Splitting dataset into training and test set

Importing Libraries: To perform data preprocessing using Python, we need to import some predefined Python libraries. These libraries are used to perform some specific jobs. There are three specific libraries that we will use for data preprocessing, which are:

Numpy: Numpy Python library is used for including any type of mathematical operation in the code. It is the fundamental package for scientific calculation in Python. It also supports to add large, multidimensional arrays and matrices. So, in Python, we can import it as:

Import Numpy as nm.

Here we have used nm, which is a short name for Numpy, and it will be used in the whole program.

Matplotlib: The second library is matplotlib, which is a Python 2D plotting library, and with this library, we need to import a sub-library pyplot. This library is used to plot any type of charts in Python for the code. It will be imported as below:

Import matplotlib.pyplot as mpt

Here we have used mpt as a short name for this library.

Pandas: The last library is the Pandas library, which is one of the most famous Python libraries and used for importing and managing the datasets. It is an open-source data manipulation and analysis library. Here, we have used Pd as a short name for this library. Consider the below image:

Handling Missing data: The next step of data preprocessing is to handle missing data in the datasets. If our dataset contains some missing data, then it may create a huge problem for our machine learning model. Hence it is necessary to handle missing values present in the dataset.There are mainly two ways to handle missing data, which are:

• By deleting the particular row: The first way is used to commonly deal with null values. In this way, we just delete the specific row or column which consists of null values. But this way is not so efficient and removing data may lead to loss of information which will not give the accurate output.

• By calculating the mean: In this way, we will calculate the mean of that column or row which contains any missing value and will put it on the place of missing value. This strategy is useful for the features which have numeric data such as age, salary, year, etc.

Encoding Categorical data: Categorical data is data which has some categories such as, in our dataset; there are two categorical variables, Country, and Purchased. Since machine learning model completely works on mathematics and numbers, but if our dataset would have a categorical variable, then it may create trouble while building the model. So, it is necessary to encode these categorical variables into numbers.

Feature Scaling: Feature scaling is the final step of data preprocessing in machine learning. It is a technique to standardize the independent variables of the dataset in a specific range. In feature scaling, we put our variables in the same range and in the same scale so that no variable dominates the other variable. A machine learning model is based on Euclidean distance, and if we do not scale the variable, then it will cause some issue in our machine learning model. Euclidean distance is given as:

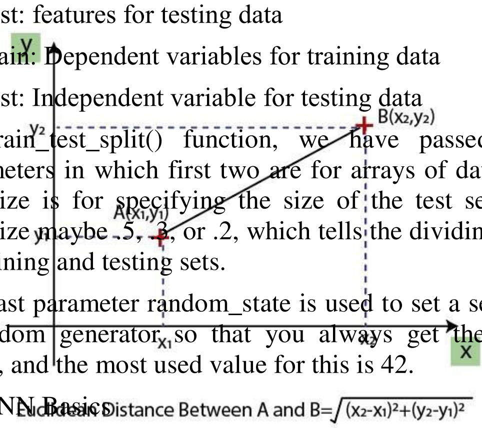


Figure 3.2: Feature scaling.

If we compute any two values from age and salary, then salary values will dominate the age values, and it will produce an incorrect result. So, to remove this issue, we need to perform feature scaling for machine learning.

3.2 Splitting the Dataset

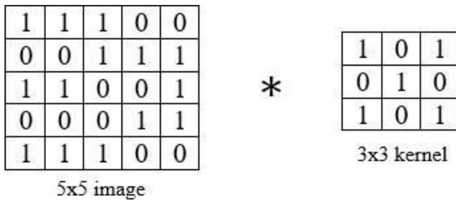
In machine learning data preprocessing, we divide our dataset into a training set and test set. This is one of the crucial steps of data preprocessing as by doing this, we can enhance the performance of our machine learning model. Suppose if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models. If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as:

Figure 3.3: Splitting the dataset.

Training Set: A subset of dataset to train the machine learning model, and we already know the output.

Test set: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

For splitting the dataset, we will use the below lines of code: from sklearn.model_selection import train_test_split



x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.2, random_state=0)

Explanation

• In the above code, the first line is used for splitting arrays of the dataset into random train and test subsets.

• In the second line, we have used four variables for our output that are

• x_train: features for the training data

• x_test: features for testing data

• y_train: Dependent variables for training data

• y_test: Independent variable for testing data

In train_test_split() function, we have passed four parameters in which first two are for arrays of data, and test_size is for specifying the size of the test set. The test_size maybe .5, .3, or .2, which tells the dividing ratio of training and testing sets.

The last parameter random_state is used to set a seed for a random generator so that you always get the same result, and the most used value for this is 42.

3.3 CNN Basics

According to the facts, training and testing of proposed model involves in allowing every source image via a succession of convolution layers by a kernel or filter, rectified linear unit (ReLU), max pooling, fully

4

connected layer and utilize SoftMax layer with classification layer to categorize the objects with probabilistic values ranging from [0,1]. Convolution layer as is the primary layer to extract the features from a source image and maintains the relationship between pixels by learning the features of image by employing tiny blocks of source data. It's a mathematical function which considers two inputs like source image $I(x, y, d)$ where x and y denotes the spatial coordinates i.e., number of rows and columns. d is denoted as dimension of an image (here d = 3, since the source image is RGB) and a filter or kernel with similar size of input image and can be denoted as $F(kx, ky, d)$.

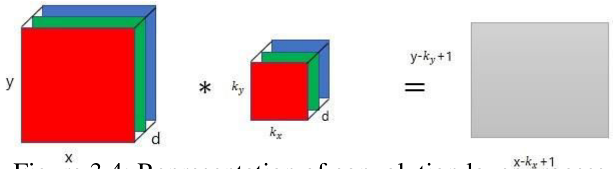
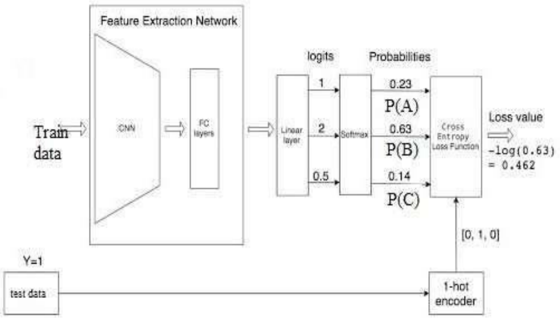


Figure 3.4: Representation of convolution layer process.

The output obtained from convolution process of input image and filter has a size of

red as
size of
eature
input



(a) an
kernel

Re LU layer: Networks those utilizes the rectifier operation for the hidden layers are cited as rectified linear unit (ReLU). This ReLU function $\mathcal{G}(\cdot)$ is a simple computation that returns the value given as input directly if the value of input is greater than zero else returns zero. This can be represented as mathematically using the function $max(\cdot)$ over the set of 0 and the input x as follows: $\mathcal{G}(x) = \max\{0, x\}$

Max pooling layer :This layer mitigates the number of parameters when there are larger size images. This can be called as sub sampling or down sampling that mitigates the dimensionality of every feature map by preserving the important information. Max pooling considers the maximum element form the rectified feature map.

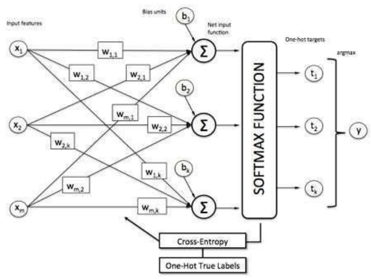


Figure 3.6: Vehicle prediction using Softmax classifier.

Figure 3.7: Example of Softmax classifier.

the model processes the pictures, send it to the hidden layers and then finally send to SoftMax for classifying the picture. The SoftMax uses a One-Hot encoding Technique to calculate the cross-entropy loss and get the max.One-Hot Encoding is the technique that is used to categorize the data. In the previous example, if SoftMax predicts that the object is class a then the One-Hot Encoding for:

Class A will be [1 0 0]

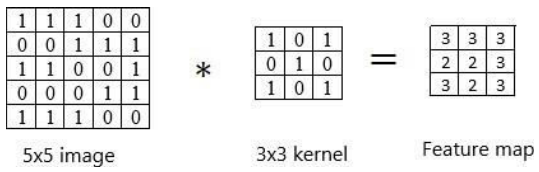
Class B will be [0 1 0]

Class C will be [0 0 1]

From the diagram, we see that the predictions are occurred. we don't know the predictions. But the machine must choose the correct predicted object. So, for machine to identify an object correctly, it uses a function called cross-entropy function.

Figure 3.8: Example of Softmax classifier with test data.

In the above example we see that 0.462 is the loss of the



function for class specific classifier. In the same way, we find loss for remaining classifiers. The mathematical representation for loss function can be represented as: $LOSS = np$

IV. EXPERIMENTAL ANALYSIS

This project uses the OpenCV (cv2) library and a pre-trained YOLOv3 model to perform object detection on either a video feed from a webcam or a video file. Specifically, it is designed to detect objects related to guns and knives, and it draws bounding boxes around those objects when detected, it shows a still frame or a snapshot from a video captured by a closed-circuit television (CCTV) camera. The key characteristic of this figure is that it does not contain any visible weapons. This is another example of a scenario where the system does not detect any weapons.

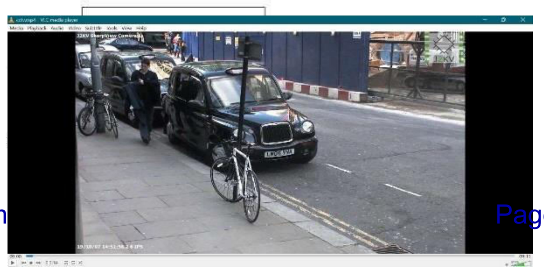


Figure 4.1: Sample video 1 collected from CCTV footage without any weapons.

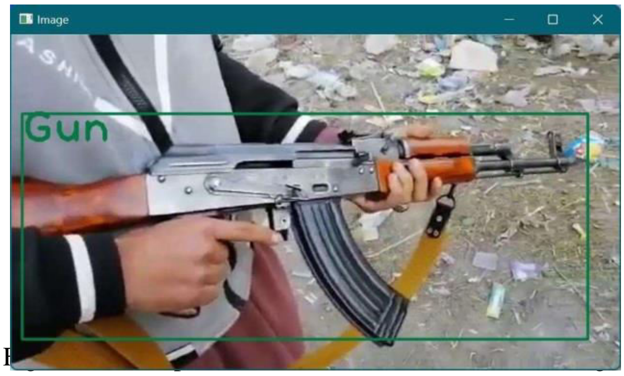
Figure 4.2: Sample video 2 collected from CCTV footage suspecting weapon.

Figure 4.3: Sample video 1 collected from CCTV footage suspecting weapon.

Figure 4.4: Sample video 4 collected from CCTV footage suspect clear.

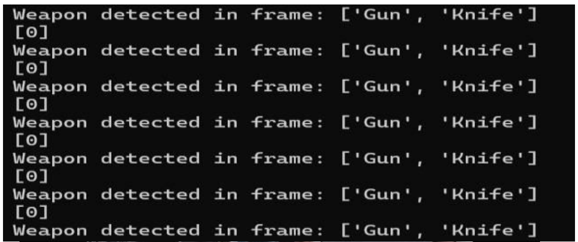
Figure 4.5: Sample video 5 collected from CCTV footage weapon move detected.

Figure 4.5 is a snapshot from a video captured by a CCTV camera. The significant aspect of this figure is that it contains a visible weapon. It is used as an example to demonstrate how the system detects and highlights the presence of a weapon in the video feed, shows some sample frames from CCTV footage. The key feature of figure is the presence of a weapon within the scene .It provides an additional example of a scenario where the system successfully identifies and highlights the presence of a weapon.



F
handling weapon

Figure 4.6 represents the result of a proposed YOLOv3-based model for analysing CCTV video footage. This figure shows a visual representation of the weapon detection results obtained from the video analysis process, provides annotated frames from the video, bounding boxes around detected weapons, labels indicating the types of weapon detected (e.g., "Gun" or "Knife")



```
Weapon detected in frame: ['Gun', 'Knife']
[0]
Weapon detected in frame: ['Gun', 'Knife']
[0]
Weapon detected in frame: ['Gun', 'Knife']
[0]
Weapon detected in frame: ['Gun', 'Knife']
[0]
Weapon detected in frame: ['Gun', 'Knife']
[0]
Weapon detected in frame: ['Gun', 'Knife']
[0]
Weapon detected in frame: ['Gun', 'Knife']
```

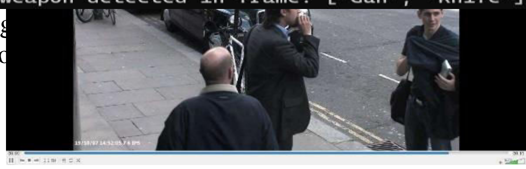


Fig olov3
mo

This project implemented a weapon detection system utilizing a pre-trained YOLO model. Its current functionality allows for the real-time detection of guns and knives in video streams, offering potential applications in security and safety monitoring. However, the system's accuracy hinges on the quality of the pre-trained model and the dataset used for training. To enhance detection accuracy and minimize false positives and negatives, further refinement through fine-tuning or dataset expansion is advisable. It's worth noting that, like any object detection system, it may not achieve perfect accuracy and could produce occasional inaccuracies. The system's potential for growth lies in enhancing the object detection model. This can be achieved by training it on more comprehensive and diverse datasets, ultimately leading to heightened accuracy and reliability. Tailoring the model to specific scenarios, such as indoor or outdoor surveillance, is another avenue to explore.

Real-time Alerts: Future iterations of the system could feature alert mechanisms, such as notifications or alarms, triggered when a weapon is detected. This real-time response capability is invaluable in security applications.

Multi-Camera Support: Expanding the system to support multiple cameras concurrently would broaden its applicability. This capability is especially useful for monitoring expansive areas or complex environments.

Privacy and Ethics: Consideration of the ethical and legal aspects of deploying such a system is crucial, particularly in public spaces. Integrating privacy features to safeguard individuals' privacy should be a priority.

REFERENCES

1. K. Akhila, and K. R. Ahmed. Firearm Detection Using Deep Learning. Intelligent Systems and Applications: (2022).
2. Ahmed, S.; Bhatti, M.T.; Khan, M.G.; Lövström, B.; Shahid, M. Development and Optimization of Deep Learning Models for Weapon Detection in Surveillance Videos. Appl. Sci. 2022.
3. Ahmed S, Bhatti MT, Khan MG, Lövström B, Shahid M. Development and Optimization of Deep Learning Models for Weapon Detection in Surveillance Videos. 2022.
4. Ahmed, Soban, Muhammad Tahir Bhatti, Muhammad Gufran Khan, Benny Lövström, and Muhammad Shahid. 2022. "Development and Optimization of Deep Learning Models for Weapon Detection in Surveillance Videos.
5. Ratcliffe, J. Video Surveillance of Public Places. US Department of Justice, Office of Community Oriented Policing Services: Washington, DC, USA, 2006.
6. Cohen, N.; Gattuso, J.; MacLennan-Brown, K. CCTV Operational Requirements Manual 2009. Home Office Scientific Development Branch: St. Albans, UK, 2009.
7. Murthy, C.B.; Hashmi, M.F.; Bokde, N.D.; Geem, Z.W. Investigations of Object Detection in Images/Videos Using Various Deep Learning Techniques and Embedded Platforms—A Comprehensive Review. Appl. Sci. 2020.
8. Bhatti, M.T.; Khan, M.G.; Aslam, M.; Fiaz, M.J. Weapon Detection in Real-Time CCTV Videos Using Deep Learning. IEEE Access 2021.
9. M.T. Bhatti, M.G. Khan, M. Aslam, and M.J. Fiaz. Weapon detection in real- time cctv videos using deep learning. IEEE Access 9: 34366-34382 (2021).
10. M.T. Bhatti, M.G. Khan, M. Aslam, and M.J. Fiaz. Weapon detection in real- time cctv videos using deep learning. IEEE Access 9: 34366-34382 (2021).