Automatic Weapon Detection in CCTV systems Using Deep Learning

MSc Research Project
Data Analytics

Rohit Jadhav
Student ID: 20205350

School of Computing
National College of Ireland

Supervisor: Mr. Vladimir Milosavljevic

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Rohit Jadhav
Student ID:	20205350
Programme:	Data Analytics
Year:	2021-2022
Module:	MSc Research Project
Supervisor:	Mr. Vladimir Milosavljevic
Submission Due Date:	15/08/2022
Project Title:	Automatic Weapon Detection in CCTV systems Using Deep Learning
Word Count:	7629
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Rohit Jadhav
Date:	15th August 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Automatic Weapon Detection in CCTV systems Using Deep Learning

Rohit Jadhav
20205350

Abstract

Security surveillance is becoming highly crucial as the frequency of school shootings, armed robberies, and terrorist operations are increasing constantly. By catching these activities early on, there will be a lot less collateral harm, which will lower the number of offenses. This work introduces real-time weapon identification approach using surveillance systems based on computer-vision. For the purpose of finding the weapons and confusion objects in this study, different YOLO approaches are used. The study concentrates on Scaled-YOLOv4 and YOLOv4 to efficiently differentiate between confusion objects and real weaponry. With mean average accuracy (mAP@0.50) of 86.19%, Precision of 79%, and F1-score of 77%, YOLOv4 performs moderately better than Scaled YOLOv4. YOLOv4 also shows superior confidence score in pictures as well as video footage.

1 Introduction

The rate of crime has significantly increased over the past ten years. These occurrences often involve pistols or other weapons. Because of these instances, there have been several deaths reported from all over the world. This has immediate or long-term psychological repercussions for the victim's family, witnesses, or in certain situations, even the perpetrator's family. Studies show that the pistol is the weapon of choice in a wide range of crimes, including robberies, break-ins, rape, and theft. By identifying disruptive conduct early on and actively observing suspicious activities, these occurrences can be avoided and the required action can be taken by law enforcement.

1.1 Background & Importance

The humanity still present in people today has been called into doubt by a number of recent instances. First off with the shootings at the Christchurch mosque, which took place in New Zealand on March 15, 2019. The bulk of people had gathered for Jummah Salah because it was Friday (Muslim Friday prayer which is taken during midday). Nearly 44 unarmed worshipers were killed when a gunman stormed the 'Al Noor Mosque' at 13.40 with the intention of murdering as many people as possible. After 15 minutes, he murders 7 additional worshipers at Linwood Islamic Center. Brenton Harrison Tarrant, an Australian white nationalist, committed this crime Wikipedia contributors (2022a).

A large-scale shooting took place at Orlando's Pulse nightclub in 2016. This awful crime, which claimed the lives of 49 people and injured 53 more, was carried out by Omar Mateen Wikipedia contributors (2022b).

Additionally, there have been other school shootings in the USA in recent years. With easy access to gun license these days in United States of America, Even today number of shootings are rising in all states across USA. Sometimes youth steal these weapons from their own parents or from other extended family members as they have not kept them adequately hidden or secured in their safes. Once they get any such harmful objects in their hands, they sometimes carry out actions they otherwise wouldn't if they didn't have weapon in first place.

The events mentioned above are just a small sample of the innumerable ones that took place throughout the course of the past ten years worldwide. Almost everyday more than 10 kids die and more than 30 kids get injured from gun violence. Most of the times, at least one person other than malefactor knows about the plan of the execution and fails to report it on time in order to prevent it. Sometimes people who know about the deed early on do some necessary actions in order to prevent it. In 2019, Nicole Schubert's son was plotting Columbine school shooting anniversary by carrying out same thing in his school in Washington. Nicole found out about this through his journal. She also found out through his journal that he was planning to kill her before carrying out the massacre. After knowing about her own son's intentions she immediately called police on him and got him arrested. Her son is now 20 and still angry about the incidence, yet she is firm and happy about her decision which saved many lives including her own OLIVEIRA (2022). However, these types of decisions are hard to make. Therefore, we cannot expect this thing from every parent or other relatives. To avoid situations like these, we can make use of robust surveillance system that is already laid off.

1.2 Approach towards the Solution

CCTV cameras play a crucial role in a surveillance system that works well. CCTV cameras are currently installed in every public space. After a crime has happened, evidence is gathered using CCTVs. In trials, CCTV footage is the most crucial piece of evidence. Security organisations respond to an incident and leave with video footage Ratcliffe (2006).

We currently have the greatest number of CCTV cameras deployed worldwide than ever before. In the UK alone, there are 4.5 million surveillance cameras in use. Around 50,000 cameras were installed in Sweden around the end of 2010. After installing just 450 cameras, the Polish government successfully put stop to drug street battles by 40%. Also, throughout the city of Poznan cases were reduced 60% Grega, Matiolański, Guzik and Leszczuk (2016).

Since such CCTV operator must occasionally sit in front of the monitors for ten hours straight, the management of the growing number of CCTV cameras is becoming more challenging. Previously, they could watch or spot suspicious activity more easily since there were fewer screens and cameras. However, it is currently challenging for them to keep track of more than 25 screens in front of them.

The answer to the aforementioned issue is to incorporate a camera that can automatically locate the weapons and notify them as soon as possible to the appropriate authorities. Object identification methods from domains of computer vision plus image processing are included into cameras to make this feasible. In any aggressive circumstance, portable weapons were often employed more than 90% of the time. Due of the relative ease with which handgun licences may be obtained in most nations, especially the USA, this research will concentrate more on handguns than any other sort of armament. Study will concentrate on identifying confusing things like Torch, Stapler, Remote control apart from an actual Handgun. Reason for concentrating on such common objects is if some of these objects are held in certain way, it can be misinterpreted as weapon. Though the objects looks similar to handgun, confusion can be avoided with precision in order to make less error and accurately predict the true positives where there is an actual revolver or pistol in the frame.

1.3 Research Question

Employment of YOLOv4 and Scaled-YOLOv4 to effectively discriminate between confusion objects and actual unconcealed firearms.

1.4 Research Objectives

The research study objectives listed below offer a clear, thorough transition plan for successfully completing the project on schedule and are expressed in connection to the research question addressed in this research effort.

1.4.1 Key Objective

Main focus of any implementation employed in this research will be to detect the handgun first.

1.4.2 Objective 1

Research will first focus on detecting and classifying an actual weapon apart from other confusion objects using YOLOv4.

1.4.3 Objective 2

Research will also focus on detecting and classifying an actual weapon apart from other common confusion objects using Scaled-YOLOv4.

1.4.4 Objective 3

Evaluation of better model amongst YOLOv4 and Scaled-YOLOv4 upon multiple evaluation metrics.

1.5 Assumptions

1. Confusion objects taken into consideration are considered based on their resemblance with an actual weapon if they are held in certain angle.

1.6 Limitations

1. Media files selected for this study are taken in broad daylight. Therefore, system can only function in daytime scenario.

2. Model works better with footage having good quality media. With the deterioration of quality, performance of overall model also decreases.

The sequence of the following structure will be covered in this thesis. In Section 2, we first go over relevant research. In Section 3, the research approach used in this study is covered. The design specification is covered in Section 4. The project's implementation is covered in Section 5. The assessment, experiments, and discussion from this research are included in Section 6. The research is concluded and future work is covered in Section 7.

2 Literature Review

Everywhere in the globe, the government is required to provide its citizens with a secure and livable environment. One approach to accomplish this is to continuously keep an eye out for dangerous or suspicious activity in the area to reduce the likelihood of life-threatening hostile circumstances for the local populace. Due to significant advancements in surveillance systems, including enhanced quality CCTV cameras and a wider variety of CCTV cameras, the issue of real-time object detection and categorization has arisen. Additionally, by utilizing dynamic deep learning & machine learning techniques, it is feasible to automate the process of identifying suspicious behaviours.

2.1 History of CCTV

The first country to use CCTV was Germany. In 1946, German scientists created the technology to see the launch of the V2 Rocket. This same kind of camera monitoring was used later, when atomic bombs were tested in the US. The first video tape recorder (VTR) was created in 1951. The VTR recorded live pictures from a television camera on a magnetic storage strip. Later, this technique was combined with CCTV cameras. Towards the 1960s, CCTV use had become increasingly widespread, and by the end of the decade, it was also included into home security systems. In order to monitor the interactions, CCTVs were placed in ATM systems by the 1990s. At the same time, individuals began employing a camera inside their homes, known as a "Nanny Cam," to keep an eye on their children who were being watched by babysitters. People started concentrating their efforts on creating a more stringent monitoring system that was supported by a facial recognition system and installing additional security cameras after the disastrous second attack on the World Trade Center.(Delgado; 2013).

2.2 Object Detection Algorithms

The hardest challenge for computer vision is identifying objects or patterns in static or moving pictures. In the realm of computer vision, a variety of object recognition techniques have been put out to improve surveillance systems. In reality, a smart surveillance architecture is constructed at a number of different levels, including data extraction from the bare minimum, object tracking, feature engineering, the detection of anomalous or abnormal events, the identification of odd human actions, patterns, or behaviours, and many others.

Zhang, Wang, Chan, Wei and Ho (2014) talked about a variety of fresh approaches to object tracking and detection in CCTV systems. For better-occluded object segmentation and backdrop modelling, color-based object segmentation using mean shift was suggested. Then BKF-SGM-IMS followed the Segmented Objects. Finally, SP-EMD distortion assessment unit was employed to propose a non-training based object recognition approach for the identification of analogous objects captured by nearby cameras in order to achieve network-based tracking. The performance of SP-EMD-based and ICR-based algorithms increased by 13.63% and 1.53% correspondingly, with the use of object masks. Even though there are numerous object tracking algorithms, it is challenging to discover one that can track a firearm.

Buckchash and Raman (2017) came up with a creative way to automatically find knives in video footage. Changes in posture, scaling, or rotation have no impact on the procedure. Knives come in a variety of shapes, sizes, and materials, making it difficult to identify them. Foreground segmentation, important feature discovery utilising Features from the Accelerated Segment Test (FAST) for image classification and localization, and target confirmation by employing Multi-Resolution Analysis are the three steps of the approach (MRA). The framework makes use of multithreading and socket-based parallel processing to boost KDF's overall performance. The author found that interest points were a more effective method than AAMs and HOG features. FAST-FREAK has outperformed other available interest point detectors.

Darker, Gale, Ward and Blechko (2007) were the pioneers to consider utilising CCTV cameras to find weapons. They said even increasing the detection of gun-related crime will reduce the likelihood that it will occur. If carrying a gun has a substantial risk of being discovered and punished, carrying one is likely to be discouraged. By moving beyond accurate weapons detection, MEDUSA may be able to significantly lower gun crime if the right steps are taken.

The monitoring network was created by the same folks a year later to find unmasked weaponry. However, it was reliant on test-run CCTV footage assessment techniques that worked suit well for a real-life situation (Darker, Gale and Blechko; 2008).

Tiwari and Verma (2015b) proposed using the K-mean clustering approach with a color-based segmentation strategy to remove irrelevant elements from the image. The firearm was located in the segmented images using the speeded-up robust features (SURF) along with interest point detector. Color-based segmentation is used to remove irrelevant colours or objects. The resemblance of each segmented item to the gun description is then calculated using SURF features. It is categorised as a gun if the object's SURF features match more than half of the gun descriptor's features.

Tiwari and Verma (2015a) utilised a mix of Fast Retina Keypoint (FREAK) descriptors and Harris interest point detectors after discovering a technique to identify interest regions and extract attributes to match with descriptions of guns. Once more, uninteresting colours or objects are eliminated using color-based segmentation. The next step is to evaluate whether each segmented item and the gun description are the same using the FREAK features of each Harris interest point. If more than good portion of the gun descriptors for the item fit its freak traits, the item is classified as a gun.

2.3 Object Detection with Deep Learning

The use of deep learning to address problems with artificial intelligence is growing in popularity. Deep learning is a technique for training a artificial intelligence system to behave like a person in particular situations.

By building a crucial training data set based on a CNN's findings and comparing the two techniques of region proposal and sliding window, Olmos, Tabik and Herrera (2018) devised a system to diminish the false positive rate. The most notable results were obtained by quicker R-CNN models that were trained on their database. It creates no false positives and has a 100% recall rate. Additionally, it fared better than the poor YouTube video.

Gelana and Yadav (2019) made an effort to implement a system that would lessen the frequency of active shooter situations, which have been increasing recently. The recommended approach uses feature extraction methods and a convolutional neural network classifier to categorise items as either a gun or not a gun. They achieved detection precision of 97.78%

Elmir, Laouar and Hamdaoui (2019) presented a method for spotting guns in recordings that may be used for surveillance and management. Their fouc was also to minimize the false positive detection. They worked with three distinct deep learning models. MobileNet SDD outperformed the other models. They advised to use better CPU to avoid issues faced during the study in the future.

Fernandez-Carrobles, Deniz and Maroto (2019) demonstrated a system based on the Faster R-CNN approach that can identify knives and firearms. In this work, the CNN-based architectures GoogleNet and SqueezeNet were contrasted. The authors claim that the Faster R-CNN technique has been used in a number of prior studies. With a AP_{50} of 85.44%, a SqueezeNet demonstrated the superior performance for gun detection. For knife detection.

For automatic gun or weapon identification, Jain, Vikram, Kashyap, Jain et al. (2020) deployed convolutional neural networks with Faster RCNN & SSD algorithms. The recommended technique makes use of two different kinds of datasets. Photos that were already recognised were included in one dataset, while manually labelled images were included in the other. Then, quicker RCNN and SSD algorithms are trained to identify guns using self- and pre-labeled picture datasets. Performance-wise, the SSD method performs better, averaging 0.736 fps. Contradictorily, faster RCNN has a rate of 1.606 seconds each frame. Contrarily, Faster RCNN outshines other models in terms of all the evaluation metrics.

Bhatti, Khan, Aslam and Fiaz (2021) proposed a deep learning-based weapon recognition technique for real-time camera video to implement binary classification system for pistols. Single-Shot multi-box Detection (SSD), Inception-V3, MobileNetV1, Inception-ResnetV2, and Faster-RCNN were among the region proposal and sliding window approaches they used to detect firearms. You Only Look Once v3 (YOLO3), Inception-ResnetV2 (FRIRv2), and You Only Look Once v4 (YOLO4). Three separate datasets, each with a unique collection of data and train-test ratio, were employed in the experiment. To assess the entire performance of the model, Accuracy alone is not as crucial as recall and precision. With a mean average accuracy of 91.73% and an F1-score of 91%, Yolov4 greatly beats all other algorithms.

Narejo, Pandey, Rodriguez, Anjum et al. (2021) trained the YOLOV3 algorithm to recognise a weapon using their own dataset. YOLOV3 outperforms YOLOV2 and conventional convolutional neural networks, according to training results. Additionally, YOLOV3 is computationally cheap. Furthermore, no sophisticated GPUs or processing resources were needed because their strategy used transfer learning to train their model.

Madhushree, Sowmya and Chennamma (2021) created a system that can successfully identify weapons in recordings, including firearms, knives, bazookas, and tanks. They have identified weaponry using Alex-net. They made use of photos from ImageNet to train their model. According to the statistics, the suggested method has a 95% accuracy for gun detection, a 93% accuracy for bazooka detection, a 94% accuracy for knife detection, and a 92% accuracy for tanker identification.

Velasco-Mata, Ruiz-Santaquiteria, Vallez and Deniz (2021) suggested a technique for spotting firearms in public places by observing people's postures in addition to looking for weapons to improve the efficiency of the weapon detection. Gray-scale pictures are used in this method to detect firearms, and the recommended technique improves handgun detection accuracy. Positive outcomes are obtained with the proposed network, with a maximum AP improvement of 17.5% over the baseline pistol detector (YOLOv3). Their study shows that a person's posture may be used to improve threat detection precision.

Hashmi, Haq, Fraz and Shahzad (2021) conducted a comparison study of the two models, known as YOLOV3 and YOLOV4, which are cutting-edge weapons detecting systems. For training purposes, they created a weapons dataset using images from Google Image Search and a number of other resources. Both versions were trained using this enormous dataset of weapons, and the outcomes of the tests were compared. In their article, they assert that YOLOV4 is superior to YOLOV3 in almost every evaluation metrics.

Noor, Isa et al. (2021) provided a cutting-edge plan for boosting Malaysia's usage of CCTV cameras. This study focuses on using the YOLOv4 Darknet framework to train an object identification model specifically to identify weapons like knives and firearms. To determine the effectiveness of this strategy, two phases of training were conducted. A single class custom object identification model was utilised in the first training, while a multiple class custom object identification model was employed in the second training. This device offers a solution and can recognise knives and pistols as weapons. The multiple class object identification model performs better identification and has higher mAP scores than the single class object identification model. The multiple class object identification model significantly outperformed the single class object identification model in terms of accuracy. Overall, the multi-class object identification model performs superior at detecting objects than the single class object identification model.

Singh, Anand, Sharma and Singh (2021) created an algorithm that can locate the weapon, identify it, and notify the necessary authorities. For this, they've used YOLOV4. This method incorporates automated and networked Internet of Things (IoT) smart devices to identify weapons. Microcontrollers and microprocessors for the Internet of Things will help YOLO model to send a specific alert in response to detection and notify the relevant authorities. If the probability is greater than the threshold value, the item with IoU (Intersection over Union) greater than the threshold value receives a bounding box. Acuuracy is measure with IoU. For the dataset, they used pictures from Google and Kaggle. The Kaggle dataset included pre-tagged data, but they had to manually label the data for the Google dataset. Additionally, it can be shown that the performance of the model is influenced by the sort of hardware used to execute it. The accuracy for photos of lesser quality is estimated to be approximately 75%, while that for images of better quality is predicted to be around 95%.

Dhiyanesh, Rajkumar, Radha et al. (2021) demonstrated a probabilistic neural network-based convolutional neural network for enhancing the capacity to recognise objects in surveillance photographs (PNN). Additionally, this research intends to examine several frequently disregarded underlying concepts as well as the basic principles of perceptual and neural networks. They have demonstrated how deep learning generally produces extremely effective CNN architectures. Networks can match or even outperform many powerful techniques. CNN-PNN performs better at detecting objects in a dynamic environment than other methods. The simulation findings imply that the suggested approach is more accurate at detecting objects than earlier methods. Additionally, the proposed model fared better than earlier models in getting rid of the backgrounds.

Hnoohom, Chotivatunyu, Maitrichit, Sornlertlamvanich, Mekruksavanich and Jitpattanakul (2021) made an effort to help with the ongoing, long-running Thailand issue. To help the understaffed police force, they attempted to create a system that can identify the weapon accurately. The IMFDB Weapon Detection Dataset and ARMAS Weapon Detection dataset were the two publicly accessible datasets used in this investigation. They employed object identification techniques like EfficientDet-D0, Faster R-CNN Inception Resnet-V2 and SSD MobileNet-V1 In the experimental stage, the faster R-CNN Inception Resnet-V2 surpassed all other techniques; The maximum mAP was 0.540, while the average accuracy with 0.5 IoU and 0.75 IoU was 0.793 and 0.627, respectively.

For the purpose of locating the handgun, Garg and Singh (2021) compared several YOLO iterations. Deep neural network object identification struggles with tiny objects, including portable weapons. In contrast to categorization, object detection identifies objects but does not pinpoint their locations within an image. A picture with several objects in it is not categorised. The testing findings show that for recognising weapons held in hands, the YOLOV4 model obtains a mAP@0.5 of 98.79%, accuracy of 91%, and recall of 99%. (pistol). The YOLOv3 Model demonstrates precise detection of firearms in surveillance with a mAP@0.50 of 90.37%, Recall 80%, and Precision 93%. However, both of these models did well on images as well as videos.

2.4 Summary of Literature Review

In most of the earlier research in the realm of Weapon Detection, Deep learning models outperformed than the rest of the machine learning methods utilized. Most of the papers based on deep learning talked about different enhanced versions of R-CNN models, SSDs or YOLO models. When compared to other deep learning methods YOLO seems to be giving higher mAP and accuracy than others. Therefore, for this research YOLOv4 and Scaled-YOLOv4 were utilized.

3 Research Methodology

This study proposes a unconcealed firearms detection technique for identifying the existence of dangerous weapons by scanning pictures or video footage. A branch of machine learning and artificial intelligence called deep learning imitates how individuals learn. The main objective of deep learning is to give robots the ability to perceive the world similarly to humans. This research will employ the CRISP-DM approach. Cross-industry data mining process is referred to as CRISP-DM. Research will follow following phases throughout the process: i) Understanding the business need by gathering more knowledge on the study. ii) Selecting particular set of images according to usecase. iii) after selection of images annotating each images in format acceptable by the algorithm. iv) implementation of the models on annotated dataset v) evaluation of model based on multiple factors such as mAP, precision, F1-score, etc. vi) completed solution can then be applied to real life scenarios.

3.1 Business Understanding

Due to the enormous relevance it may have in regards to security and safety, security agencies have long sought to detect weapons in the real world. The primary goal of weapon detection is to safeguard national security against circumstances involving the use of any guns. Additionally, situations involving robberies or school shootings can also involve weapon detection. The project will concentrate on using pictures to train the models. Once finished, video footage may be added to witness the results. The suggested strategy will make use of YOLOv4 and Scaled-YOLOv4 designed by Bochkovskiy, Wang and Liao (2020) and Wang, Bochkovskiy and Liao (2021) respectively.

3.2 Data Understanding

There is not any particular dataset we can utilise for this investigation. This study will employ mixture of two datasets to get the great results. The initial data set for this study comes from Kaggle which includes 5687 images. Some of these images are already annotated and some are CCTV images without annotation. jubaerad (2020), Second dataset is Open Images V6 which is presented by Kuznetsova, Rom, Alldrin, Uijlings, Krasin, Pont-Tuset, Kamali, Popov, Malloci, Kolesnikov, Duerig and Ferrari (2020). This dataset consists population of nearly 9.6 million images with annotations for segmentation, object recognition, and classification.

Out of both of these datasets, only relevant images containing either handguns or common confusion objects are taken into consideration.

3.3 Data Preparation

As study will make use of multiple objects multiple datasets are used. Also, not all of the images in the dataset can be used as sometimes the dataset may contain irrelevant images to the study. For this study, mainly handgun images are focused. Reasons for choosing the handguns specifically are loose gun related laws about keeping firearm at individuals' household and multiple incidents reported in past where most used weapons are handguns (pistols or revolver). To test the model's accuracy, multiple confusion objects are also supplied like torch, remote control, stapler, etc.

The datasets need to be annotated with Object Annotation in order to move on ahead. Bounding boxes, 3D cuboids, polygon segmentation, and semantic segmentation are just a few examples of the various sorts of Object Annotation techniques. Bounding box Object Annotation is a most commonly used method and it is also used by YOLO. As the study focuses on YOLO, bounding box technique is used for annotating objects. The labels.txt file includes all the classes that must be utilised for the labelling process is created before starting the object annotation process. Then, using open source applications like LabelImg or YoloLabel, each individual picture is manually tagged. The object bounding box annotation is done using YoloLabel because the YOLO model is utilised in this study. Figure 1 shows how the layout of YOLOLabel is and how all the classes are shown to the top right of the application.

There is a text file linked to each image with exact same name but different extention i.e. txt that provides the following information: 'class - center_x - center_y - width - height'. first digit represents the actual class of the object every class has different color associated with it, center_x is the horizontal position of the center of the bounding box divided by the whole width of the photo, center_y is the vertical position of the center of the bounding box divided by the whole height of the photo, the width of the bounding box divided by the whole width of image, and the height of the bounding box divided by the whole height of the photo. All these numbers are always between 0 to 1. All

Figure 1: YoloLabel

of the photographs must be individually labelled, making this the most time-consuming procedure. Although tools like YoloLabel or LabelImg might expedite the process, there is no other option except to work over each photo individually.

3.4 Modeling

Prior until recently, all have been focusing on sluggish, computationally intensive sliding window object identification methods that deliver quite subpar results. After development of new techniques, Neural networks built on a convolutional basis were the most popular and effective techniques. One of the deep learning techniques, a convolutional neural network (ConvNet/CNN), it takes an input picture and convolve over it to identify its characteristics. In the case of CNN, minimal dataset pre-processing is necessary. CNN is used to classify the image as well as locate each individual item in it.

For this work, a variety of R-CNN-based models, SSDs, AlexNet, GoogleNet and others, were investigated. The YOLO method was found to have the best accuracy and the least rate of false positives at this stage. As a result, this project will employ the use of YOLO.

The architecture of YOLOv4 is made up of the CSPDarknet53 backbone, the PANet neck, the YOLOv3 head, the spatial pyramid pooling as an extra module. A cutting-edge backbone called CSPDarknet53 can improve CNN's abilities to infer. Over CSP-Darknet53, the spatial pyramid pooling block is introduced to broaden the receptive field and isolate the most important context information. The PANet is utilised as a mechanism enabling parameter aggregation for various detector levels in place of the FPN which is used in YOLOv3.

Scaled-YOLOv4 uses improved backbone and neck uses CSP (Cross-stage-partial) connectors. It also employes mish activation function. During training phase, Scaled-YOLOv4 works with Exponential moving averages hence producing ema weights.

3.5 Evaluation

For the comparative analysis, similar metrics will be used: Recall, Average Loss function, Intersection over Union (IoU), Precision, Detection Time, Mean Average Precision (mAP), Confidence score, and F1-score. Before the model is evaluated, the following three crucial possibilities must be taken into account: When the model correctly predicts the positive class, it is said to be a true positive (TP). An incorrect discovery of an object in object detection is known as a false positive (FP). False Negative (FN) results when an object is there but the model fails to detect it. Object detection does not employ True Negatives (TN).

The computation of intersection over union (IoU) uses the intersection of the actual bounding box of the item with the predicted bounding box of the model. Bounding boxes must overlap exactly to get an IoU of 1.

$$\text{IOU} = \frac{\text{Area of Intersection of both Bounding Boxes}}{\text{Area of Union of Both Bounding Boxes}}$$

Average loss function (error) is inversely propositional to accuracy. the lesser the loss, greater the accuracy.

The trained model is evaluated to see if it is overfitting, underfitting, or a suitable match for the detection method using the Mean Average Precision (mAP). By default mAP@50 is used for most of the models which means mAP is calculated at IoU threshold of 0.5.

Your predictions' precision serve as a gauge of their accuracy. The proportion of your forecasts that are accurate, in other words.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Confidence score is score of particular class in prediction. More accurate methods produces higher confidence score.

Recall gauges how effectively a model finds every success. Recall is the percentage of all relevant results that the algorithm properly classifies, whereas precision refers to the proportion of relevant outcomes that an algorithm successfully classifies.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

How effectively accuracy and memory are balanced is determined by the F1 score. When the F1 score is high, precision and recall are also high, and vice versa.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN}$$

3.6 Deployment

The best model may be utilised to deploy in real-world scenarios following evaluation of the models based on the aforementioned metrics.

4 Design Specification

YOLO is a state-of-the-art object detection technique. Earlier systems used to split the image in multiple regions and then run a classifier on each region individually and then compare the results of each of the regions for the prediction of object. This involved running classifiers for each of the regions and for each iterations it may take multiple of neural network evaluations to get accurate object detection which is computationally expensive. YOLO does things differently as it looks at the given image only once and uses single neural network for predicting the object detection with the help of bounding boxes. YOLO makes use of Darknet. Darknet is publicly available neural network system written in C and CUDA for testing and training computer vision models Redmon (2013–2016). In this scenario, Microsoft's COCO (Common Objects In Context) dataset has been used to train the pre-trained model. For pre-training the model 80 different classes from COCO dataset were used. It requires more time to train a model from scratch. The process could occasionally take several weeks to complete in case of lower GPUs. A pre-trained model is aware of how each item should be classed because it has seen many different things. The weights for the YOLO model were created by utilising the COCO and ImageNet datasets to train the network. The Object Annotation techniques used by all YOLO implementations are identical.

For this study YOLOv4 and Scaled-YOLOv4 are implemented. Both of them follow similar structure with fewer differences. For both the models, first all the images are annotated and uploaded on the Google drive. After object annotation, configuration files are created for each of the YOLO versions along with obj.names, obj.data, train.txt and test.txt. Once configuration files are uploaded from the local system, training for both the YOLO version is started. Once training is completed best weights from both the models are utilized to test the test pictures as well as video footage. Upon testing, both the models are evaluated using different metrics (see 2).

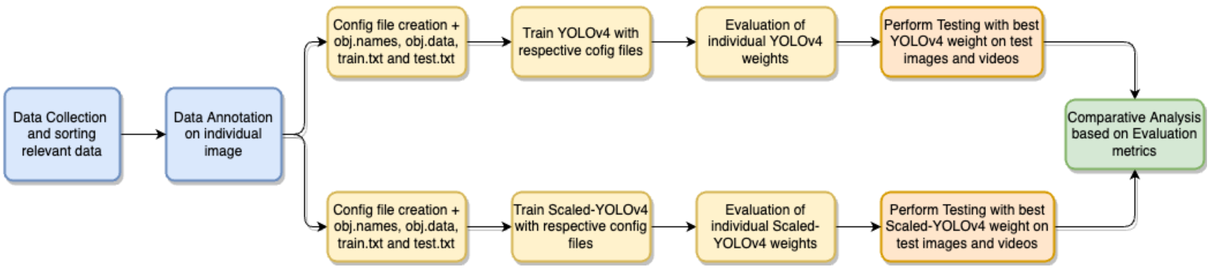


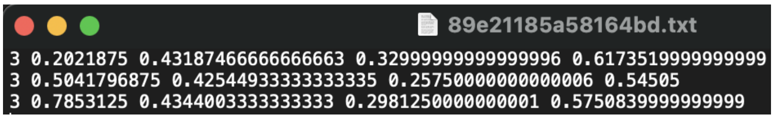
Figure 2: Design diagram of the project

5 Implementation

Majority of the time was spent on the Object Annotation process. At first, all the relevant images having multiple classes were imported in open source application named YOLO-Label [1]. Once all the images are imported, application ask location of obj.names file or classes.txt. These files are similar and they both carry all the classes present for the training. In this case, obj.names file contains Handgun, Torch, Stapler and Remote control. Each class is represented with different color. Upon selection of the class, color of the bounding box changes. If an image contains multiple objects then the txt file for

[1] https://github.com/developer0hye/Yolo_Label

that image will contain multiple rows each representing different object in the photo (see 3). After completion of object annotation each photo will have its own txt file with same name containing all the objects in that particular image.



89e21185a58164bd.txt

```
3 0.2021875 0.43187466666666663 0.32999999999999996 0.6173519999999999
3 0.5041796875 0.42544933333333335 0.25750000000000006 0.54505
3 0.7853125 0.4344003333333333 0.2981250000000001 0.5750839999999999
```

Figure 3: Text file containing multiple classes

Reason for using google colab was to make use of GPU provided by colab as local systems do not perform well for object detection methods with lack of GPUs. To enable the GPU provided by colab certain parameters are changed in Makefile of darknet before building it. These parameters include CUDNN, OPENCV, GPU, CUDNN_HALF, LIBSO. As darknet runs on CUDA and CUDACNN, CUDNN and CUDNN_HALF is set to 1. As GPU and Computer Vision Library is going to be used GPU and OPENCV is also set to 1. Once GPU is assigned for acceleration to the colab notebook, drive is mounted to the notebook and all the images from the local system are uploaded on the drive.

Five files are important in order to train the YOLO models:

1. **config.cfg** : config file contains necessary configurations for the implementation of model.
2. **obj.data** : this file comprises of number of all the classes, address of test.txt, train.txt, and obj.names. It also contains location of backup folder where all the weights files based on config files are saved.
3. **obj.names** : this file is similar to classes.txt which contains list of all the classes present in the dataset.
4. **train.txt** : this file comprises of list of all the files that will be used for the training set.
5. **test.txt** : this file comprises of list of all the files that will be used for the testing set.

5.1 Implementation of YOLOv4

To make config file for YOLOv4, yolov4-custom.cfg file is used. Height and Width parameters are set as multiple of 32. The higher the number of these parameters, the longer the time is taken for the training of the model. For this implementation, height and width are set to 416. batch variable denotes number of processed samples per iteration. For this implementation batch is set as 64. subdivisions variable denotes number of mini batches processed at once. classes is equal to no. of classes present in the dataset. Depending upon number of classes, max_batches and steps changes. filters are supposed to be $filters = (classes + 5) * 3$ so in this case, it will be 27 as we have total 4 classes. This needs to be done for all the 3 convolutional layers before yolo layer. max_batches should be $classes * 2000$ and it should never be less than 6000 so in this case max_batches values is set to 8000. steps are set to 80% and 90% of the max_batches so the steps will be 6400 and 7200. To avoid training models from the scratch as it takes a lot of time, pre-trained weights are used named 'yolov4.conv.137'. This is trained on Microsoft's COCO dataset already therefore for custom dataset it gives better and faster results.

5.2 Implementation of Scaled-YOLOv4

For Scaled-YOLOv4, yolov4-csp.cfg file is used. Just like YOLO-v4, width and height are set to 416 (multiple of 32). Except for a few of the variables, all the remaining settings are identical to YOLOv4 configuration. letter_box variable is set to 1 to keep the aspect ration of the sample image. Training from scratch takes huge amount of time so in this case also pre-trained weights trained on Microsoft's COCO dataset is used. For Scaled-YOLOv4, 'yolov4-csp.conv.142' is trained on COCO dataset.

For YOLOv4 as well as Scaled-YOLOv4, obj.data file is similar. Only difference between them is that the location of weights file changes according to the config file. train.txt, test.txt and obj.names will be similar in both versions of YOLO as we are working with similar set of data and need to perform comparative analysis.

5.3 Common things in the Implementation of both models

Google colab platform's free subscription is sometimes not reliable. If user is idle for some time, allocated GPU acceleration gets disconnected leaving the models untrained or half-trained. To avoid such situation, Google Colab's Pro subscription is used. Though it gives better quality GPU and RAM for the users, it is not guaranteed that the experience will be seamless. Even with Pro subscription sometimes GPU gets disconnected.

train.txt and test.txt is created with the help of python code give 90% to training data and 10% to test data. These files only contain the names of all the files required for training or testing.

Once training is started, chart is observed for loss function (error). Ideal loss function value can be from 0.05 to 0.2. Depending upon the complexity of the dataset loss function can be in the range of 0.2 to 0.5. Also while training, mAP is observed constantly. If the mAP is increasing and the loss function is decreased continuously model is properly getting trained.

Once the training is completed, for every 1000 iterations weight file is saved. To get the best weights with highest mAP or IoU 'detector map' command can be used. Best weight file is also present along with other iterations' weights but sometimes it is possible that other weight files produce higher mAP. Therefore, it is better to double check the mAP of each file before testing it on the test data. Threshold for both the model is set at 50% i.e. mAP@0.50 which will force the models to not include predictions with confidence score below 50%.

6 Evaluation

As both the models follow darknet architecture performing comparative analysis on both the model is relatively simple. Research was done with many iterations but in the end results from only main iterations are included. Each of the experiment gives different insight to the research.

6.1 Experiment 1: Training with Open Images

First experiment is performed on Images from Open Images. Four distinct classes were downloaded from Open Images. After splitting train and test data, model is trained with pre-trained weights trained on COCO dataset.

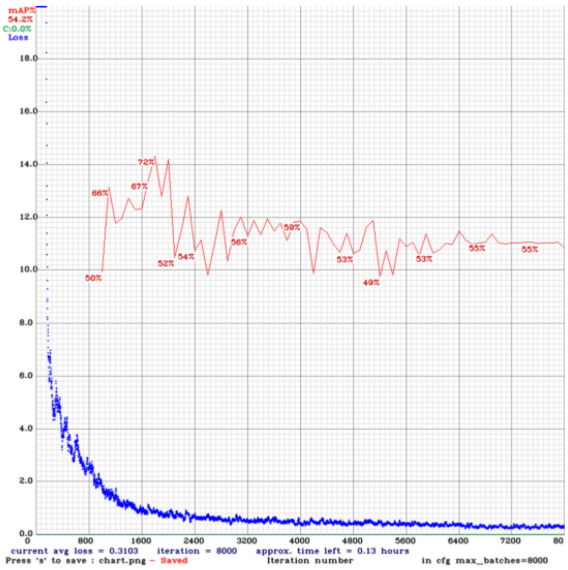


Figure 4: Exp 1: mAP and avg loss graph

Figure 4 depicts mAP and average loss for experiment 1. It was observed that, loss was continuously decreasing and there was fluctuations in mAP over the course of training. Highest mean average precision (mAP@0.50) of 71.58% was observed in yolov4-weapon-4_best.wights file. This weight file is used to train next model to improve the model's overall performance further.

6.2 Experiment 2: Training with Kaggle annotated dataset

Second experiment was performed with data annotated with YOLO-label. Kaggle dataset contained many objects like rifles, SMG, rocket launcher, knife, handgun, etc. only images containing revolver or pistols (handgun) were taken into consideration for this study from this dataset. On top of it, some of the confusion objects are also added in this dataset as the earlier model had four classes. Before uploading kaggle dataset in data/obj folder, open images were copied back to the original location and then kaggle annotated images were uploaded in data/obj folder. This model is trained with best weight file (yolov4-weapon-4_best.wights) from earlier experiment to improve the accuracy of YOLOv4 model.

The training process was seen to have a mean average precision (mAP@0.50) of 100% throughout the process 5, in addition to recall, precision, and an F1-score of 1 for each weight file with no False positives. This suggested that the model was over-fitted. The model can be trained using both datasets together with pre-trained weights from the COCO dataset to prevent this over-fitting problem. We could have taken any file having lesser mAP than 100 to check the model accuracy. As all the weights showed exactly the same mAP, this was clearly the case of over-fitting.

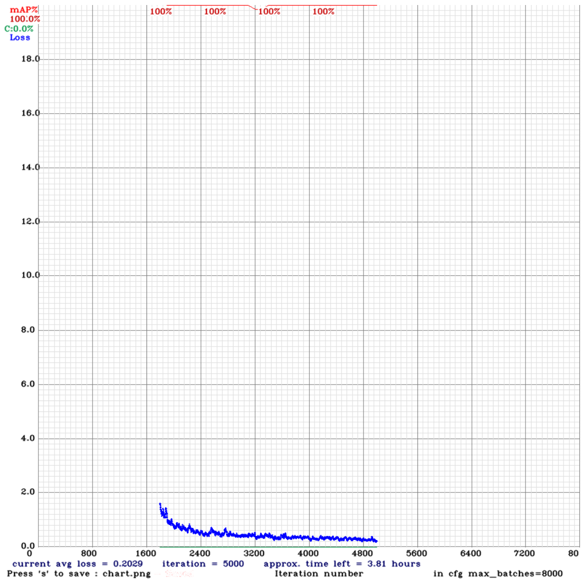
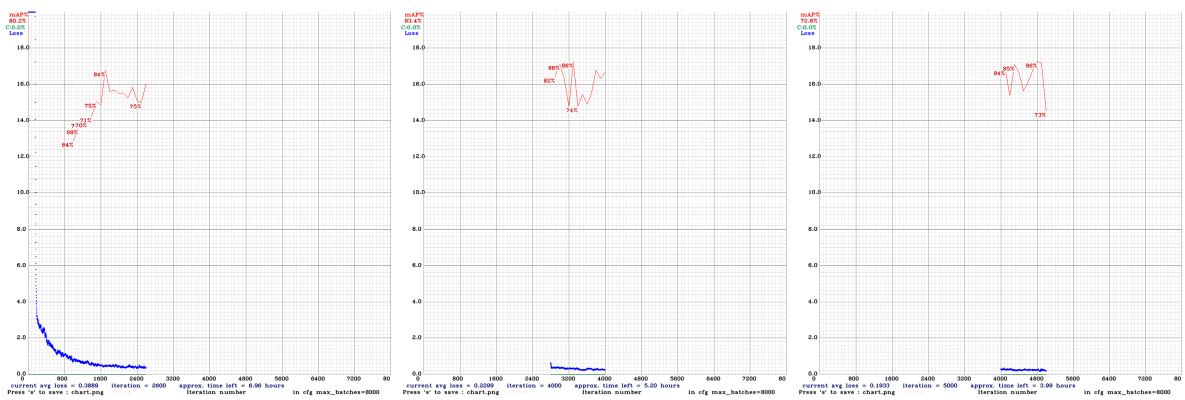


Figure 5: Exp 2: mAP and avg loss graph

6.3 Experiment 3: Training YOLOv4 with combined dataset

As kaggle annotated data is already present data/obj folder. Open images data is copied back in data/obj combining both the datasests. For this experiment, the model is trained with pre-trained weights trained on COCO dataset (yolo4.conv.137.weight) as the earlier experiment was over-fitted. After splitting train and test data, obj.data file is produced with a new directory for the backup of weight files. Training was interrupted twice due to disconnection of GPU. As a result, there were three different charts produced for the same model. These charts were downloaded after each hindrance as every time process is restarted, charts are overwritten. After training for few hours, training was stopped manually as there was no significant difference spotted for mean average precision (mAP@0.50) and average loss function was observed to be 0.1933 and not declining further (see 6).



(a) Chart 1 (b) Chart 2 (c) Chart 3

Figure 6: Exp 3: mAP and avg loss graph :YOLOv4

As training was stopped after 4000 iterations, five distinct weight files were created excluding best and last weight file. The mAP of every weight file is evaluated with 'darknet detector map' command. The weight files are evaluated on number of metrics including precision, recall, F1-score. Mean average precision is average of average precisions by all the classes present. By keeping track of all these metrics and highest mean average precision (mAP@0.50), it was observed that 'yolov4-weapon-4_best.weights' performed better than other weight files with mean average precision (mAP@0.50) of 86.19%.

6.4 Experiment 4: Training Scaled-YOLOv4 with combined dataset

For the final experiment, training and testing files are kept exactly the same along with obj.names. obj.data file is modified to change the location of backup weights for Scaled-YOLOv4 model. This model was trained with 'yolov4-csp.conv.142' weight file which was pre-trained on COCO dataset. After starting the training process, average loss function was significantly higher than it was observed in YOLOv4 model training process. After more than 1500 iterations average loss error was declined below 20. Till 1600 iteration it was there in the same territory. As a result, lowest mean average precision (mAP@0.50) of around 14% was consequently noted at this time. After 1600 iterations, there was sudden rise in the mAP till it reached 83% by 2400 iterations. Average loss error function was seen to be fluctuating significantly during the course of the training process in Scaled-YOLOv4.

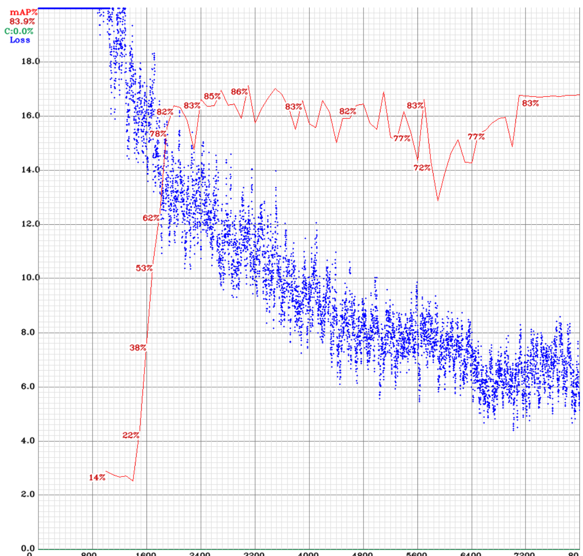


Figure 7: Exp 4: mAP and avg loss graph : Scaled-YOLOv4

After the training procedure was finished, it was found that the average loss error function was 5.244, which is much greater than that of the YOLOv4 model. The highest mAP@0.50 of 85.60% was produced by 'yolov4-csp-weapon_best.weights' file which was better than exponential moving average weight file having mAP of 83.98% (yolov4-csp-weapon_ema.weights).

6.5 Discussion

The outcomes of each experiment listed above are compared in this section. All experiments performed above using both versions of YOLO models were compared at IoU threshold of 50%. Accuracy, Precision and F1-score are calculated at confidence threshold of 25%. Every weight file has different constitution of images having different classes predicting different average precision for each of the weight file.

Experiments performed using YOLOv4 model produces lower average loss error function than Scaled-YOLOv4. It also produced smoother average loss graph than Scaled-YOLOv4.

Overall metrics of experiment 1 were notably improved in experiment 3 by training model on larger dataset. The model trained in experiment 2 was over-fitted. Therefore, further analysis are only done on experiment 3 and 4.

	YOLOv4	**Scaled-YOLOv4**
Precision	79%	81%
Recall	74%	71%
F1-score	77%	76%
Total True Positives	31	30
Total False Positives	8	7
Total False Negatives	11	12
Average IoU	63.86%	61.23%
mean average precision (mAP@0.50)	86.19%	85.60%
average loss function error	0.1933	5.2449
Total Detection Time	2 seconds	2 seconds

Table 1: Comparative analysis of YOLO models in experiment 3 & 4

Table 1 depicts the comparative analysis of both the YOLO models. YOLOv4 outperforms Scaled-YOLOv4 in almost all the metrics other than precision.

(a) Confidence score of YOLOv4

(b) Confidence score of Scaled-YOLOv4

Figure 8: Confidence score comparison of YOLO models

The mAP curve in YOLOv4 was also observed to be smoother having lower number of spikes and variations.

Between YOLOv4 and Scaled-YOLOv4, the mean average accuracy value does not significantly change.

In majority of cases, the confidence score of YOLOv4 was always greater than Scaled-YOLOv4 for every class of the object identification, regardless of the quality of the picture, when models were evaluated against test photos (see 8).

However, when models were evaluated against higher-quality video footage, YOLOv4 outperformed Scaled-YOLOv4 in terms of performance and confidence score.Conversely, for poorer quality movies, YOLOv4 was more likely to react incorrectly to minor amounts of video noise. As opposed to YOLOv4, Scaled-YOLOv4 generates more accurate predictions with a lower confidence score in lower quality videos.

The following are the main conclusions drawn from the study:

- The loss error graph for Scaled-YOLOv4 was totally distorted. Whereas, the loss error graph for YOLOv4 was smoother.
- Overall confidence score of YOLOv4 was consistently higher than Scaled-YOLOv4 in both pictures and videos.
- YOLOv4 is more sensitive to noise in the videos than Scaled-YOLOv4 which results in predicting wrong predictions at times.
- YOLOv4 surpassed Scaled-YOLOv4 in almost all the metrics.
- With the inclusion of new data between experiments 1 and 3, the model's performance and accuracy were considerably improved. This accuracy may be further enhanced with more amount of data.

7 Conclusion and Future Work

Main focus of the study is to predict the weapons in images as well as video files. Both implementations of model using YOLOv4 and Scaled-YOLOv4 were satisfyingly able perform this task. Also both models employed in this study were able to successfully predict confusion objects along with key ability to predict actual firearm (handgun). Therefore, they can be applied for the use of weapon detection in real world scenarios. Amongst YOLOv4 and Scaled-YOLOv4, YOLOv4 performs slightly better with mean average precision (mAP@0.50) of 86.19%, loss error function of 0.1933, Precision of 79%, and F1-score of 77%.

Future work will focus on further lowering false negatives and positives. System can be integrated with the alerting system to alert concerned authorities in case of weapon detection with high confidence score. We may also attempt to use more training data to gain better results. Additional weaponry can be incorporated as this study solely considers handguns. We may also add more number of common confusion object to add further complexity. We may anticipate more precise outcomes with improved CCTV footage quality.

Acknowledgement

I would like to extend my honest gratitude and thanks towards my supervisor Mr. Vladimir Milosavljevic for his constant guidance, encouragement, and feedback.

References

Bhatti, M. T., Khan, M. G., Aslam, M. and Fiaz, M. J. (2021). Weapon detection in real-time cctv videos using deep learning, *IEEE Access* **9**: 34366–34382.

Bochkovskiy, A., Wang, C.-Y. and Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection.

Buckchash, H. and Raman, B. (2017). A robust object detector: application to detection of visual knives, *2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, IEEE, pp. 633–638.

Darker, I., Gale, A., Ward, L. and Blechko, A. (2007). Can cctv reliably detect gun crime?, *2007 41st Annual IEEE International Carnahan Conference on Security Technology*, IEEE, pp. 264–271.

Darker, I. T., Gale, A. G. and Blechko, A. (2008). Cctv as an automated sensor for firearms detection: Human-derived performance as a precursor to automatic recognition, *Unmanned/Unattended Sensors and Sensor Networks V*, Vol. 7112, International Society for Optics and Photonics, p. 71120V.

Delgado, R. (2013). From edison to internet: A history of video surveillance, *Business 2 Community* .

Dhiyanesh, B., Rajkumar, S., Radha, R. et al. (2021). Improved object detection in video surveillance using deep convolutional neural network learning, *2021 Fifth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, IEEE, pp. 1–8.

Elmir, Y., Laouar, S. A. and Hamdaoui, L. (2019). Deep learning for automatic detection of handguns in video sequences., *JERI*.

Fernandez-Carrobles, M., Deniz, O. and Maroto, F. (2019). Gun and knife detection based on faster r-cnn for video surveillance, *Iberian conference on pattern recognition and image analysis*, Springer, pp. 441–452.

Garg, R. and Singh, S. (2021). Intelligent video surveillance based on yolo: A comparative study, *2021 International Conference on Advances in Computing, Communication, and Control (ICAC3)*, IEEE, pp. 1–6.

Gelana, F. and Yadav, A. (2019). Firearm detection from surveillance cameras using image processing and machine learning techniques, *Smart innovations in communication and computational sciences*, Springer, pp. 25–34.

Grega, M., Matiolański, A., Guzik, P. and Leszczuk, M. (2016). Automated detection of firearms and knives in a cctv image, *Sensors* **16**(1): 47.

Hashmi, T. S. S., Haq, N. U., Fraz, M. M. and Shahzad, M. (2021). Application of deep learning for weapons detection in surveillance videos, *2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2)*, IEEE, pp. 1–6.

Hnoohom, N., Chotivatunyu, P., Maitrichit, N., Sornlertlamvanich, V., Mekruksavanich, S. and Jitpattanakul, A. (2021). Weapon detection using faster r-cnn inception-v2 for a cctv surveillance system, *2021 25th International Computer Science and Engineering Conference (ICSEC)*, IEEE, pp. 400–405.

Jain, H., Vikram, A., Kashyap, A., Jain, A. et al. (2020). Weapon detection using artificial intelligence and deep learning for security applications, *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*, IEEE, pp. 193–198.

jubaerad (2020). Weapons in Images.
URL: *https://www.kaggle.com/datasets/jubaerad/weapons-in-images-segmented-videos*

Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., Duerig, T. and Ferrari, V. (2020). The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale, *IJCV* .

Madhushree, B., Sowmya, K. and Chennamma, H. (2021). Automatic weapon detection in video using deep learning, *Data Engineering and Intelligent Computing*, Springer, pp. 503–510.

Narejo, S., Pandey, B., Rodriguez, C., Anjum, M. R. et al. (2021). Weapon detection using yolo v3 for smart surveillance system, *Mathematical Problems in Engineering* **2021**.

Noor, W. E. I. B. W., Isa, N. M. et al. (2021). Object detection: Harmful weapons detection using yolov4, *2021 IEEE Symposium on Wireless Technology & Applications (ISWTA)*, IEEE, pp. 63–70.

OLIVEIRA (2022). Washington mom, 41, reveals heartbreak of having to turn her own son, 17, into police after finding his chilling diary which 'revealed plans to murder her and shoot up his school' : She says he's turned his life around and is planning to go to college.
URL: *https://www.dailymail.co.uk/news/article-11036951/Woman-shares-harrowing-tale-calling-police-son-feared-planning-school-shooting.html*

Olmos, R., Tabik, S. and Herrera, F. (2018). Automatic handgun detection alarm in videos using deep learning, *Neurocomputing* **275**: 66–72.

Ratcliffe, J. (2006). *Video surveillance of public places*, Citeseer.

Redmon, J. (2013–2016). Darknet: Open source neural networks in c, `http://pjreddie.com/darknet/`.

Singh, A., Anand, T., Sharma, S. and Singh, P. (2021). Iot based weapons detection system for surveillance and security using yolov4, *2021 6th International Conference on Communication and Electronics Systems (ICCES)*, IEEE, pp. 488–493.

Tiwari, R. K. and Verma, G. K. (2015a). A computer vision based framework for visual gun detection using harris interest point detector, *Procedia Computer Science* **54**: 703–712. Eleventh International Conference on Communication Networks, ICCN 2015,

August 21-23, 2015, Bangalore, India Eleventh International Conference on Data Mining and Warehousing, ICDMW 2015, August 21-23, 2015, Bangalore, India Eleventh International Conference on Image and Signal Processing, ICISP 2015, August 21-23, 2015, Bangalore, India.
URL: *https://www.sciencedirect.com/science/article/pii/S1877050915014076*

Tiwari, R. K. and Verma, G. K. (2015b). A computer vision based framework for visual gun detection using surf, *2015 International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO)*, IEEE, pp. 1–5.

Velasco-Mata, A., Ruiz-Santaquiteria, J., Vallez, N. and Deniz, O. (2021). Using human pose information for handgun detection, *Neural Computing and Applications* **33**(24): 17273–17286.

Wang, C.-Y., Bochkovskiy, A. and Liao, H.-Y. M. (2021). Scaled-yolov4: Scaling cross stage partial network, *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, pp. 13029–13038.

Wikipedia contributors (2022a). Christchurch mosque shootings — Wikipedia, the free encyclopedia. [Online; accessed 19-March-2022].
URL: *https://en.wikipedia.org/w/index.php?title=Christchurch_mosque_shootings&oldid=1077367011*

Wikipedia contributors (2022b). Orlando nightclub shooting — Wikipedia, the free encyclopedia. [Online; accessed 21-March-2022].
URL: *https://en.wikipedia.org/w/index.php?title=Orlando_nightclub_shooting&oldid=1075445595*

Zhang, S., Wang, C., Chan, S.-C., Wei, X. and Ho, C.-H. (2014). New object detection, tracking, and recognition approaches for video surveillance over camera network, *IEEE sensors journal* **15**(5): 2679–2691.