Developing a Framework for Weapon and Mask Detection in Surveillance Systems

تطوير إطار عمل للكشف عن الأسلحة والأقنعة في أنظمة المراقبة

by

MOHAMMAD HUSNI ZAHRAWI

Dissertation submitted in partial fulfilment

of the requirements for the degree of

MSc INFORMATICS

at

The British University in Dubai

November 2022

DECLARATION

I warrant that the content of this research is the direct result of my own work and that any use made in it of published or unpublished copyright material falls within the limits permitted by international copyright conventions.

I understand that a copy of my research will be deposited in the University Library for permanent retention.

I hereby agree that the material mentioned above for which I am author and copyright holder may be copied and distributed by The British University in Dubai for the purposes of research, private study or education and that The British University in Dubai may recover from purchasers the costs incurred in such copying and distribution, where appropriate.

I understand that The British University in Dubai may make a digital copy available in the institutional repository.

I understand that I may apply to the University to retain the right to withhold or to restrict access to my thesis for a period which shall not normally exceed four calendar years from the congregation at which the degree is conferred, the length of the period to be specified in the application, together with the precise reasons for making that application.

M.Zahrawi

Signature of the student

ABSTRACT

Financial institutions, jewelry stores, hypermarkets and automated teller machines all experience yearly thefts of vast amount of money. Police have dismantled a few of the robbery attempts. Police successfully apprehend most of the robbers. The maintenance of safety and security around the globe is a difficult task for governments, particularly in a country like the UAE, which is home to more than 200 nationalities. This study examines the applications of neural network models in video surveillance systems for detecting weapons, thus preventing robberies. By expanding the dataset to include more classes and photos per class, the proposed model could perform better to be installed on outdoor surveillance systems. In this study, we will examine situations of weapons detectors, develop models using transfer learning approaches, and contrast them with other contemporary detectors like YOLOv5. We will develop our own unique dataset and contrast it with another dataset in terms of classes, image quality, and kind of items used for committing a robbery. Gun detectors in surveillance systems has a wide range of additional uses, from residentials units to the military.

Keywords—. AI, Deep learning, computer vision, Object Detection, gun detection, YOLO, SSD, weapon detection.

ملخص

تتعرض المؤسسات المالية ومحلات المجوهرات ومحلات السوبر ماركت وأجهزة الصراف الآلي للسرقة سنويًا لمبالغ ضخمة من المال. قامت الشرطة بتفكيك عدد من محاولات السطو. نجحت الشرطة في القبض على معظم اللصوص. يعد الحفاظ على الأمن والسلامة في جميع أنحاء العالم مهمة صعبة للحكومات ، لا سيما في بلد مثل الإمارات العربية المتحدة ، التي تضم أكثر من 200 جنسية. تبحث هذه الدراسة في تطبيقات نماذج الشبكات العصبية في أنظمة المراقبة بالفيديو لاكتشاف الأسلحة ، وبالتالي منع السرقات. من خلال توسيع مجموعة البيانات لتشمل المزيد من الفئات والصور لكل فصل ، يمكن أن يعمل النموذج المقترح بشكل أفضل ليتم تثبيته على أنظمة المراقبة الخارجية. في هذه الدراسة ، سوف ندرس حالات أجهزة الكشف عن الأسلحة ، ونطور نماذج باستخدام مناهج تعلم النقل ، ونقارنها مع أجهزة الكشف المعاصرة الأخرى مثل YOLOv5. سنقوم بتطوير مجموعة البيانات الفريدة الخاصة بنا ومقارنتها بمجموعة بيانات أخرى من حيث الفئات وجودة الصورة ونوع العناصر المستخدمة لارتكاب السرقة. تمتلك أجهزة الكشف عن الأسلحة في أنظمة المراقبة مجموعة واسعة من الاستخدامات الإضافية ، من الوحدات السكنية إلى الجيش.

DEDICATION

I dedicate my dissertation to whom who wish me success, and to each and every one who supported me during this exciting journey, and in specific to my beloved family, my wife, my kids and my colleagues who helped me from all aspects and plays significant role in my success.

ACKNOWLEDGEMENT

First and foremost, I thank my god Allah for handing me the strength, health, passion, knowledge, and patience to achieve this work. Also, my mother, father and sibling for encouraging me all the time.

I am pleased to acknowledge and deeply grateful and thankful to my supervisor, **Prof. Khaled Shaalan** for his tremendous assistance to get this work done. Your treasured prompt instructions, guidance and notes brought me all the way to complete this work.

In addition, I would like to acknowledge and genuinely thank **Prof. Sherief Abdullah** for the knowledge I have acquired to write and completed the proposal of this work.

Contents

LIST OF TABLES

LIST OF FIGURES

List of Abbreviations

AI: Artificial Intelligence

SSD: Single Shot Detection

YOLO: You Only Look Once

YOLOv4: You Only Look Once version 4

YOLOv5: You Only Look Once version 5

CCTV: Closed-Circuit Television

SLR: Systematic Literature Review

PRISMA: Preferred Reporting Items for Systematic Reviews and Meta-Analyses

CNN: Convolutional Neural Network

R-CNN: Regional Convolutional Neural Network

SVM: Support Vector Machine

PCA: Principal Component Analysis

ATM: Automated teller machine

RGB: Red, Green, and Blue

GPU: Graphics Processing Unit

VGG: Visual Geometry Group

SPP: Spatial Pyramid Pooling

COCO: Common Objects in Context

IoU: Intersection Over Union

AP: Average precision

mAP: Mean Average Precision

AR: Average Recall

TP: True Positive

TN: True Negative

FP: False Positive

FN: False Negative

ResNet: Residual Network

CONV: Convolutional layer

ANOVA: Analysis of Variance

VDSR: Very Deep Super Resolution

ROI: Region of Interest

1 CHAPTER ONE: INTRODUCTION

This chapter delivers an overview about dissertation subject, problem definition, Artificial Intelligence methods, research motivations, similar cases, and objectives of this research.

1.1 Background

Video Technology like video surveillance systems is absolutely essential for both indoor and outdoor security. Closed-circuit television (CCTV) cameras are frequently placed in airports, banks, train stations, hotels, schools, and business like hypermarkets, malls, etc. These CCTV cameras are used to monitor human behavior, deter theft and vandalism, investigate criminal damage, and ensure that the area around the entity is not subjected to excessive activity. The installation of closed-circuit television systems is inexpensive, enables the monitoring of large or multiple locations at once, eliminates the direct human contact, and operates nonstop around the clock. Despite this, one drawback of these closed-circuit television systems is that it still needs for security guards to view the events, record any unusual behaviors, and then transmit the videos for further analysis and examinations. However, it is impossible to manually observe every scene. Even if the scene had already happened, painstakingly searching through video recordings for specifics would take a lot of time.

According to data gathered in 2017 and released by the Small Arms Study (Karp 2018), over 85% of firearms are held by civilians in the world, while law enforcement agencies hold 2%, and 13% are held by military arsenals. To put it another way, there are around 400,000,000 firearms in the US, 120 weapons per 100 residents. According to the Small Arms Survey's findings (Karp 2018), it is clear how important it is to establish monitoring systems in sites like financial institutions and government buildings to stop robberies and other types of harm to the public. Nevertheless, security guards are complementary component for any real-time surveillance systems to watch the goings-on, gather and list reports, alert the police to any suspicious movements around the site, etc. Therefore, we suggest integrating AI into CCTV cameras to simulate the 24/7 work of human operators in order to lower operating costs and improve the dependability of surveillance systems.

With the advancement of computer vision technology and the rise of deep learning, intelligent monitoring system are being used in a variety of areas. For instance, in exam rooms, an intelligent visual system is employed to detect cheating among examinees based on their motions (Xu & Xiao 2021), additionally a surveillance system that can recognize faces and heads in video streams (Nguyen-Meidine et al. 2018). The intelligent monitoring system has been installed to identify any unusual activity, such as fighting, using a mobile phone, or fainting (Amrutha, Jyotsna & Amudha 2020). Another use of intelligent CCTV systems is for detection of workers in potentially dangerous areas, such as excavators operating in construction zones (Luo et al. 2020). Security cameras on the streetside use object detection algorithms to identify automobiles (Mohana & Ravish Aradhya 2019).

1.2 Problem Definition

Robbery is the attempt or taking of jewelry, cash, or valuable items illegally from institutions, banks, or any other custody. Robberies can devastate an entire city or even a whole country. Law enforcement units always respond swiftly to robberies, but bank employees claim that robberies occur quickly, with nearly two-thirds of them being completed in three minutes or less (*Bank Robbery | Page 2 | ASU Center for Problem-Oriented Policing | ASU* n.d.). A suggestion that needs to be considered is placing banks close to the police station. AI may be used in many different ways to enhance the surveillance system, making it a notable ally in the security field. In this study, we will discover how applying AI algorithms to a video surveillance system might help identify thefts and avert robberies, thereby enhancing community safety.

Deep learning techniques in computer vision can be applied to closed-circuit television systems that could identify, monitor, and alert police officers to any unusual activity. The identification and location of predefined objects in images or movies are made possible by object detection algorithms. In this research, we will train a model to recognize several types of firearms, including shotguns, assault rifles, handguns, knives, and balaclava masks as well. Moreover, teach the model to recognize items that people typically carry, like wallets, purses, and mobile phones, to distinguish them from guns.

The weapon detector may be used in CCTV cameras and drones to detect attacks on homes, ATMs, hospitals, and other places where security officers are not present. Olmos created the first handguns detector (Salido et al. 2021), in which the authors divided the case into two sections, weapons and background, and used images from the internet to construct training datasets and implement the RCNN algorithm, which is based on a sizable network VGG16 (Salido et al. 2021). Although the model was able to differentiate between wallets and cell phones and firearms, the findings are still quite impressive. To identify thieves and other criminals and abort robberies, we will examine two distinct scenarios:

Firearms and burglar mask detections: develop a model that can recognize knives, guns, and burglar masks. This model may be used in several places. The model is trained using variety images from the internet in a variety of colors, sizes, textures, and shapes. The model would also be able to recognize any type of robbery masks, in addition to that, weird objects like sports bags, and black costumes. The model works perfectly in banks, ATMs, and jewelry shops where typically no one is around at the midnight. The surveillance system would be linked to the nearest police station.

Home invasion detection: develop a model using state-of-art object detection techniques to detect any strangers, or intruders in the vicinity of the house or other secure location. The dataset will consist of images of family members or other individuals with permission to enter or be present in the area. A further goal of the proposed model is to distinguish between family members and intruders and thefts. Deep transfer learning will be used since training is costly, time-consuming, and resource intensive. Transfer learning is a strategy that involves freezing the majority of the model's layers to speed up training and reduce generalization error. By using transfer learning as a feature extractor in the intended weapon detection model, a deep learning classifier might be trained on top of it. A tiny hidden camera might even be installed in the peephole to detect strangers. Also, the intelligent CCTV could be placed to monitor the backyard as well as front yard. Intuitively, the dataset that utilized to develop the model is different based on intended place that want to be secured and monitored. One of the possible challenges is the shortage of images which may result in under-fitting or overfitting. Fortunately, transfer learning, which makes use of prebuilt built-in model with frozen layers, is crucial for training tiny datasets.

1.3 Proposed Solution

Object tracking and detection are described by getting the representations from data and used them for building deep learning models. Object detection is a subfield of computer vision that in charge of assigning bounding boxes or sketching a rectangle around the target object that want to be detected, which it is a gun or robber mask in our case. While image localization involves with in finding the coordinates of the target objects in an image. The concept of image localization will help with classification to introduce concept of object detection. Some traditional techniques are common for building object detectors model such as faster region-based CNN (faster-RCNN), single shot detection (SSD), and You Only Look Once (YOLO). While the SSD algorithm generally provides superior accuracy, YOLO is chosen whenever speed is valued above accuracy. The SSD and YOLO models will be introduced, and their structures will be thoroughly explained in this paper.

1. SSD: Google Inc. advanced the Single Shot MultiBox Detector, which was made available in 2016 (Liu et al. 2016). For object detection tasks, SSD produced excellent results in terms of accuracy and performance, averaging 94% mAP (mean Average Precision) at 59 frames per second on a common dataset (COCO) (Liu et al. 2016). The backbone model and the SSD head are the two primary components of an SSD. The backbone model, which is based on VGG architecture and typically without any of connected layers, is used as feature extractors. The SSD head is an additional element that is included in the backbone model to produce bounding boxes and forecast the names of the objects in an image. Confidence loss is a degree to which a model has confidence in the objectness of the bounding box. The distance between the training set's actual bounding box and the model's predicted bounding box is calculated using localization loss.

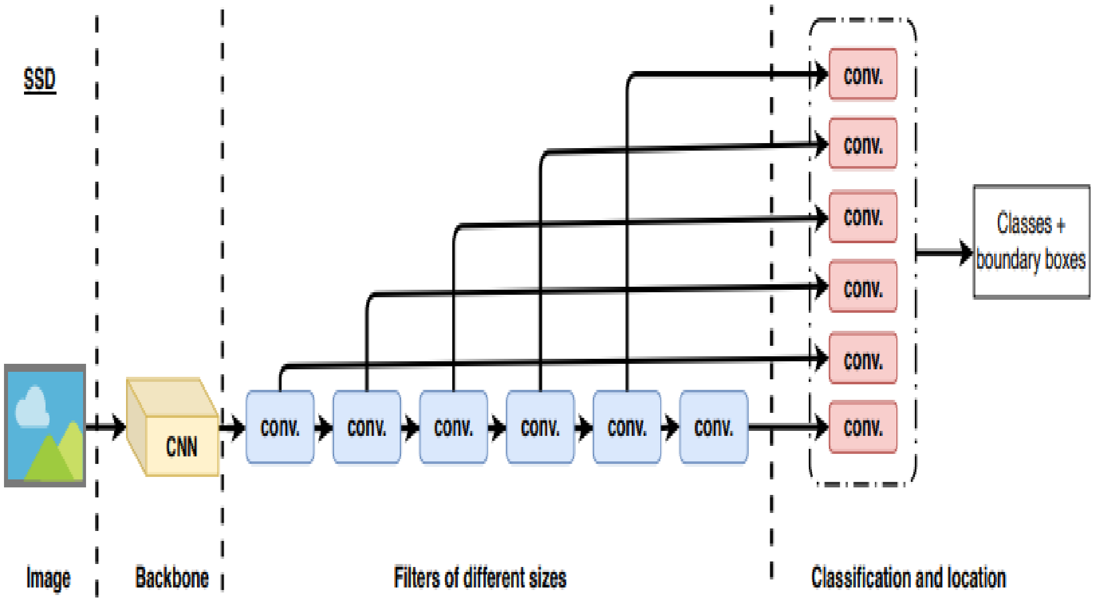


Figure 1: High level illustration of SSD

2. YOLO – Redmond et al. created the You Only Look Once (YOLO) algorithm for object detection (Redmon et al. 2016). YOLO takes the whole frame or image in a single sample and therefore predicts the coordinates of an intended object. The best feature of YOLO is its magnificent speed; it is incredibly quick and can process 45 frames per second. In order to locate and recognize items in the frame, YOLO divides the image into N grids that form comparable zones. GoogleNet served as an inspiration for the architecture of YOLO, which consists of 24 convolutional layers in total with fully connected layers linked at the end (Redmon et al. 2016). YOLO is available in several modifications, including YOLOv2, YOLO9000, YOLOv3, and YOLOv4+. In YOLOv2, batch normalization was added to address the problems with small object recognition and enhance mean average precision in the first version. As a quick and efficient model, YOLOv3 outperforms YOLOv2 in terms of accuracy. YOLOv3's model backbone is made up of 53-layer neural networks, it is known as DarkNet-53.

At regular intervals, frames are taken from CCTV cameras, and discrepancies between the frames are noted. A self-adaptive technique used for eradicating noise in data, it has a variety of adaptations to the size and amount the target objects and is beneficial in reducing operating time. Background subtraction method is valuable technique which useful for tracking and recognizing moving handguns in video surveillance systems. Recording analysis was the starting point for developing a method for detecting robber masks and firearms as well. We looked through CCTV footage in search of firearms used in crimes, violence, and property destruction. It should be noted that many CCTV videos have poor quality, blurriness, and low contrast due to old cameras. For a predetermined amount of time, firearms are visible, but lawbreakers continue to conceal them. One thing should be taken into account that the weapon detector model should be capable of handling low-quality input and small-caliber firearms. The model shouldn't require a supercomputer to operate in real-time.

Maintaining the least number of false positives is one of the most important considerations. In fact, the security guards start to overlook the alarms if the weapon detector raises alarms too frequently, rendering the surveillance system worthless. False alarms are annoying in CCTV systems since each one requires a security guard or employee to verify it, which wears them out. Minimizing false negative rates and undiscovered visible firearms is another crucial factor.

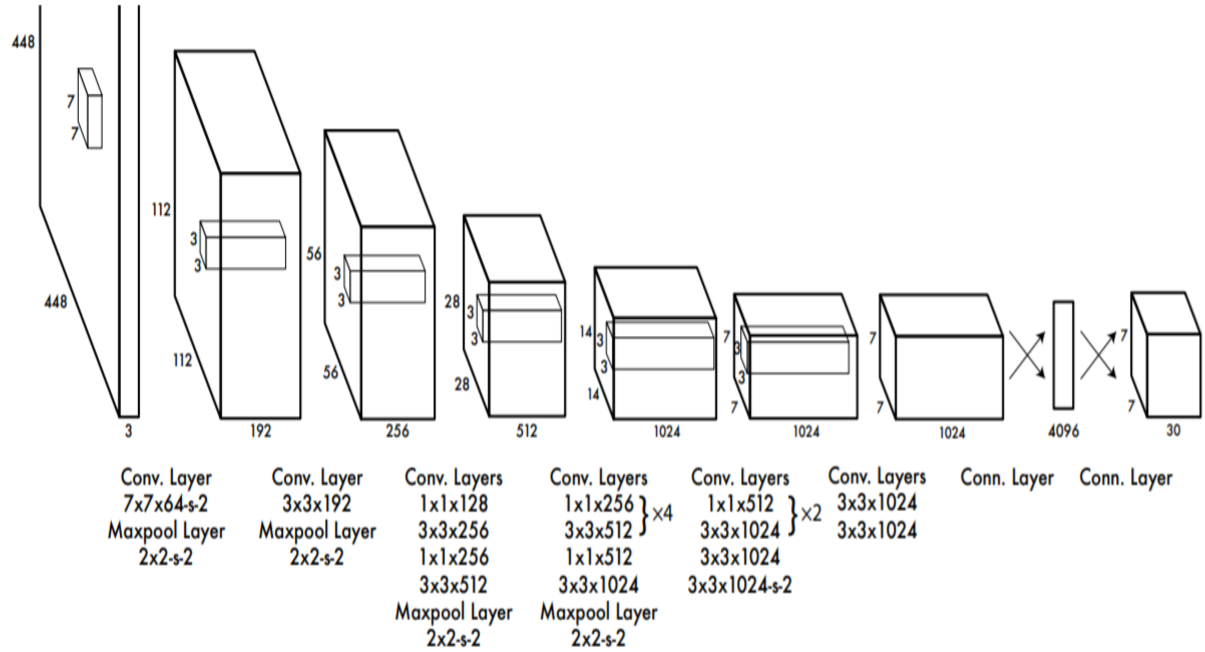


Figure 2: Design of YOLO network, it consists of 24 convo layers and two fully connected layers (Redmon et al. 2016).

1.4 Dataset Development.

To effectively detect dangerous objects in a variety of places, the weapon detector models should be trained on the specific dataset. The primary focus of this proposal is on video surveillance systems for homes, banks, and ATMs. Images of guns and other items used to rob banks, ATMs, and homes should also be included in the collection. To generate our own dataset, we will collect images from online resources and annotate them. One of the most popular datasets for anomaly identification in surveillance films was imported from the UCF website (Sultani, Chen & Shah 2018). The dataset comprises of 1900 surveillance films that have been categorized into 13 different classes, such as assault, robbery, explosion, gunshot, and so on. We alter the dataset for our situation and remove any non-relative classes, such as shoplifting, arson, traffic accidents, etc. We exclusively concentrate on bank robberies class. Each clip has a resolution of 240 x 320 pixels and a frame rate of 30 fps.

Though, the remaining data, after removing unrelated classes, is insufficient to train a weapon detector. As a result, Olmos et al. (Olmos, Tabik & Herrera 2018) from the University of Granada generated a dataset that has 3000 photos of various kinds of firearms in various settings.

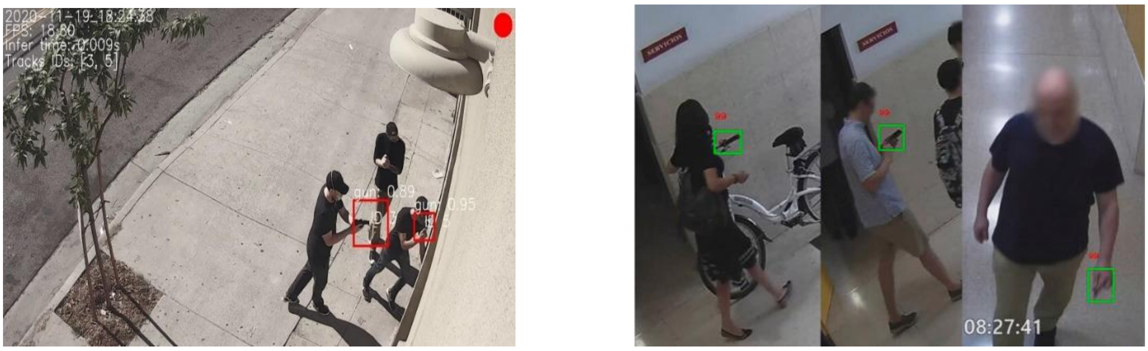


Figure 3: Real time gun detection in different places (Salazar González et al. 2020).

1.5 Pre-Processing

The objective of data preprocessing is to prepare the images for feeding the weapon detector by using a little of computer resources. One of the primary preprocessing steps is removing noise from the data that emerged during gathering and spreading through networks or a series of pulses. By using the Gaussian smoothing approach (Wink & Roerdink 2004), which is frequently used in graphics software, noise in images could be reduced. The input resolution that the proposed model is supposed to receive would be 240 × 320 pixels, so all images from the dataset should be resized accordingly. Additional data preparation techniques include:

A. Color transformation:

In many scenarios, it is helpful to convert RBG color to grayscale to save memory utilization on the CPU or GPU, subsequently shorten the training period, and speed up the model's ability to detect firearms and other target objects. RGB Images have more unnecessary colored pixels since the RGB image format carries more information than the greyscale format. Grayscale images are treated as one channel while train the model, while colored images are treated as three channels (red, green, and blue). In some cases, colors play a key role in identifying items, whereas in others, an object's shape and other qualities are more important.

B. Data augmentation:

The dataset is preprocessed to expand its size and expose the model to a larger variety of input object sizes and shapes. Random images in the dataset (or entire dataset) are chosen to be augmented by applying translation, scaling, flipping, rotation, cropping, and adding Gaussian noise. This technique is helpful to reduce the possibility of overfitting while the model is being trained.

C. Other techniques

Images could go through a variety of preprocessing stages before being fed into the model. In order to normalize image input, pixel values are scaled from 0-255 to 0-1. To achieve normalization, the average is subtracted from each pixel, and the result is then divided by the standard deviation. Faster convergence can be achieved by training a model. In some cases, it may also be advantageous to remove background colors to reduce noise. In other words, any changes to the dataset can be regarded as preprocessing procedures.

1.6 Model Learning

In this part, we will examine the phases of building weapon detectors starting from representing images as matrices. Any deep learning model can be split into four major stages, build from scratch, or select a pre-trained model, fit the model, test the model on new data, and improve the model through experimentation. Because these strategies can differ depending on the dataset and the computational resources available, it is essential that researchers review the most recent methodologies in their research to choose the best one. As we have previously mentioned, YOLO and SSD algorithms will be used to train the proposed model. The choice of a cutting-edge DNN model for firearms detection depends on a variety of elements, including the size of the weapon, the speed-accuracy trade-off, and the available computing power. The key distinctions between YOLO and SSD in the context of object detection are shown in the table 1.

Table 1: Variations between SSD and YOLO.

SSD	YOLO
Uses a convolutional network once to calculate the feature map.	Runs on two fully connected layers and 24 convolutional layers.
Due to the ability to run it on a video card, SSD may be a better choice.	Although YOLO is extremely quick, it is a better choice when precision is not an issue.
Performance dropped when trying to find small size objects.	High efficiency with tiny size object detection.

One can enhance a deep learning model by adding layers, increasing the number of hidden neurons, altering the activation functions and optimization functions, changing the learning rate parameter, and train the model for longer time. Two types of loss were computed for the SSD model's loss function: localization loss and confidence loss. By utilizing Category Cross-Entropy, confidence loss assesses the level of trust in the bounding box predictions. On the other hand, localization loss is a represent the distance between the predicted box and the ground truth box. Three different types of losses are computed for YOLO model, classification loss, confidence loss, and localization loss. All YOLO's losses are mean squared error, except classification loss, which uses cross entropy function.

1.7 Best Model Selection

The effectiveness of models has been measured using a variety of methods. Analysts may not be able to evaluate various classifiers under identical conditions, using the same dataset and initial seed. A null hypothesis test is needed to determine whether one approach is statistically superior to another, one must do a null hypothesis test; this test might be parametric or non-parametric. Independence, normality, and heteroscedasticity must all be put to the test (García et al. 2010). As a result, it is evident that numerous algorithms used in computational intelligence used random beginning seeds to divide data into training and test sets along with independent conditions. Data that behaves normally or according to a Gaussian distribution is said to have a normal distribution. T-test with performance score is used to compare our proposed model to other models. occasionally, the distribution of the data dictates whether a parametric or non-parametric test should be used. Many statistical tests should be performed depending on how many models were used to address an issue, such as the ANOVA test used when comparing more than two models.

2 CHAPTER TWO: Systematic Literature Review (SLR)

This chapter examines applicable of state-of-the-art weapon detection methods in video surveillance systems. Furthermore, it does a thorough literature study and exposes the main challenges with this technology. This study will investigate the applications of artificial intelligence (deep learning techniques) in video surveillance systems to detect robberies and intruders. In order to carry out the systematic review, the following four research questions are posed:

Question 1: *How is AI implemented in video surveillance systems?*

Question 2: *What are most advanced techniques used for the weapon detection?*

Question 3: *How is the proposed model work with low-resolution frames?*

Question 4: *Is the proposed gun detection model recommended in bank area?*

2.1 Method

The review was carried out in four distinctive stages: starting from the determination of inclusion and exclusion criteria, then the sourcing of data and search techniques, and the evaluation of the review's quality, and finally the data coding and analysis. The details of these stages are revealed in the subsections that follow.

a. Inclusion and Exclusion criteria for the papers

Table 2 lists the inclusion and exclusion criteria for the articles in the systematic review.

Table 2: Inclusion / Exclusion criteria

Inclusion Criteria	Exclusion Criteria
Paper documents should be published between 2015 to 2022	The study is published before the year 2015
The article should be relevant to the video surveillance system or related topic.	The article is not related or relevant to the topic
Publish by recognized publishers, i.e. IEEE, Elsevier, and Springer.	The study was published by an unknown publisher.
The study’s full text should be accessible.	Duplicate and non-quality studies are excluded.
The article must be written in English.	Non-English papers.
The article with a high-quality journal ranking.	The article published by low-ranking journal.

b. Data Sources and Search strategies

The research papers for the systematic review were gathered and checked through search of different literature and academic resources:

- Scoups
- Semantic Scholar
- Science Direct
- Google Scholar

The process of searching and assessment of these studies was done in November 2022. The search terms used in academic engines were ("Weapon Detection" AND "Artificial Intelligence") OR ("Gun Detection" AND "Deep Learning") OR ("robberies detection" AND "Deep Learning"). By using the search terms that mentioned previously, the academic paper result was about 75122. Moreover, 5172 papers were noticed as duplicates, thus, they were excluded. The total number of remaining academic papers becomes 172. After that, the inclusion and exclusion criteria were used to filter out the academic research papers to end up by 21 papers. See the following table for final view.

Table 3: Data Sources Analysis

Data Source Name	Records Count	Classification
Scopus	4322	Primary Source
Science Direct	2140	Primary Source
Semantic Scholar	6160	Primary Source
Google Scholar	58,900	Secondary Source*

* Secondary source databases were principally used for double checking the number of citations, and availability of paper as PDF in some cases.

Preferred Reporting Items for Systematic Reviews and Meta-Analyses PRISMA is utilized as tool for writing a systematic review to achieve the highest possible quality in finding and selecting the right papers (Page et al. 2021). Despite the fact that PRISMA is not a tool for evaluating the caliber of systematic reviews, it may be helpful for critical evaluation of published systematic reviews. The next graph shows PRISMA model for our study.

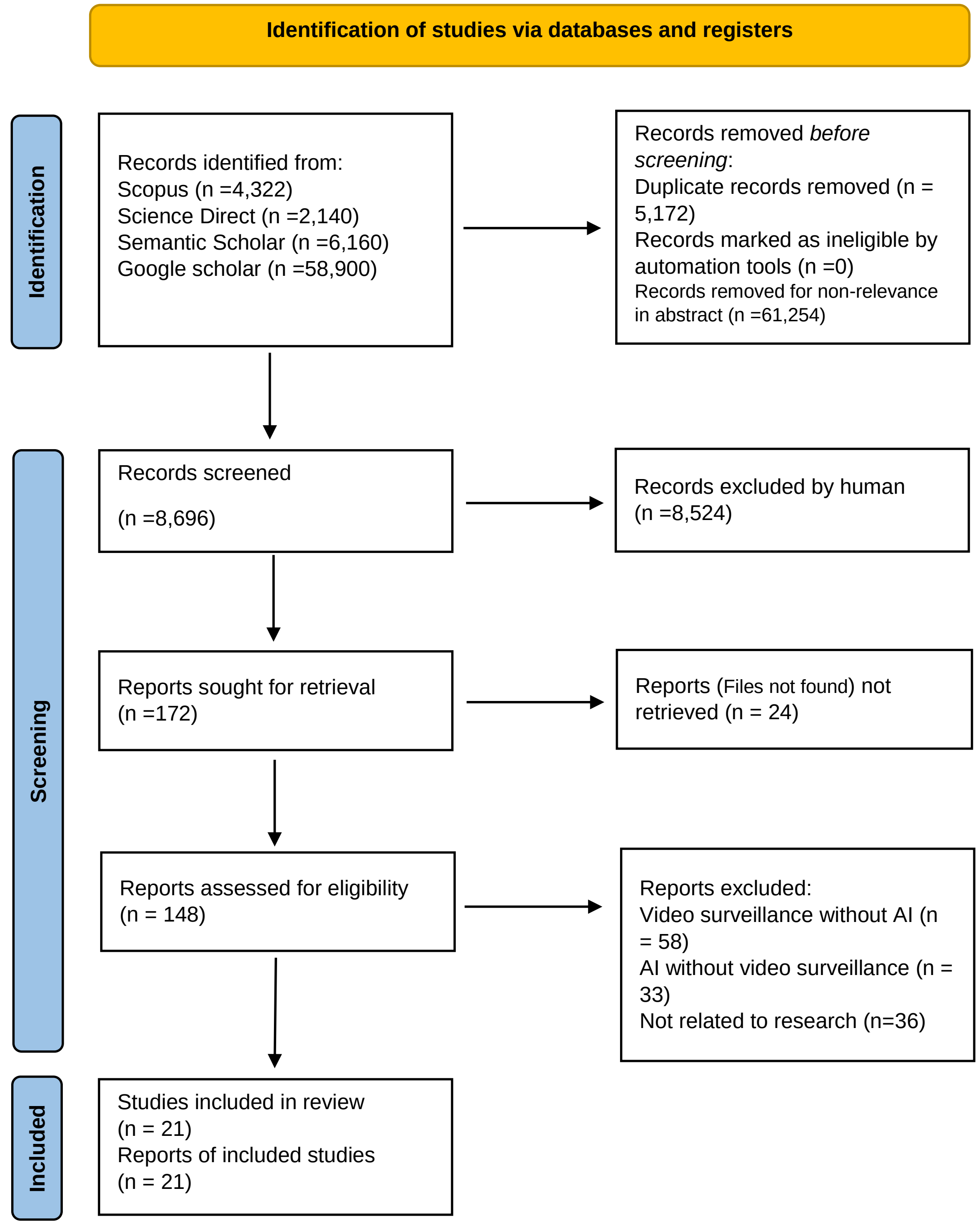

Identification of studies via databases and registers
Identification
Screening
Included
Records identified from:
Scopus (n =4,322)
Science Direct (n =2,140)
Semantic Scholar (n =6,160)
Google scholar (n =58,900)
Records removed *before screening*:
Duplicate records removed (n = 5,172)
Records marked as ineligible by automation tools (n =0)
Records removed for non-relevance in abstract (n =61,254)
Records screened
(n =8,696)
Records excluded by human
(n =8,524)
Reports sought for retrieval
(n =172)
Reports (Files not found) not retrieved (n = 24)
Reports assessed for eligibility
(n = 148)
Reports excluded:
Video surveillance without AI (n = 58)
AI without video surveillance (n = 33)
Not related to research (n=36)
Studies included in review
(n = 21)
Reports of included studies
(n = 21)

Figure 4: PRISMA Analysis Report

2.2 Quality Assessment

In order to improve the selection process of academic papers, the quality assessment method is introduced to assess study design of inclusion papers, those who meet inclusion criteria. Additionally, quality assessment measurements checklist has been added in next following table. The checklist questions evaluate the relevance, reliability, validity, and applicability of each research paper. The checklist items have linked with alignment with systematic review guidelines (Report 2007) . The quality assessment checklist aims to get rid of any bias from the studies. Nevertheless, there is no exact answer for what is the most effective quality assessment tool for the researcher that should stick with it (Report 2007). The specified checklist will be studied cross the (n=21) studies for additional quality assurance.

Table 4: Quality Assessment Questionnaires

No	Question
1	Are the objectives of the study clearly stated?
2	Does the study clearly state the relation between Artificial Intelligence techniques and robberies detection?
3	Are the claimed weapon and gun detection techniques clearly applied in the experiments?
4	Did the proposed solution or conclusion meet the objectives and solve the claimed problem?
5	Are the methods and techniques used in the experiment clearly stated?
6	Are the databases and datasets used in the study enough to perform the experiments and fulfill the objectives?
7	Do the results add to the literature?

Furthermore, a scale has been created to assess each study according to checklist items in table 4. Where “Yes” equivalent to “1”, “No” equivalent to “0 and partial equals to “0.5”. The main purpose of the scale is to deliver each study to a quantitative measure that can be relies on later (Report 2007).

Moving to result of quality assessment for systematic review, it shows around 50% (10 out of 21) of the studies achieve quality percentage of 100%; while 19% (4 out of 21) of the studies indicates a quality percentage of 92.85% and 24% (5 out of 21) achieve a quality percentage of 85.71%. As long as the overall quality assessment higher than 75%, means all selected papers are quality papers based on the checklist and measurement identified in the systematic review.

Table 5: Quality Assessment Results

Study	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Total	Percentage
S1	1	1	0.5	1	1	0.5	1	6	85.71%
S2	1	1	0	1	1	1	1	6	85.71%
S3	1	1	1	1	1	0.5	1	6.5	92.85%
S4	1	1	0	1	1	1	1	6	85.71%
S5	1	1	1	1	0.5	1	1	6.5	92.85%
S6	1	1	0.5	0.5	1	1	1	6	85.71%
S7	1	0.5	0.5	1	1	0.5	1	5.5	78.57%
S8	1	1	0	1	1	1	1	6	85.71%
S9	1	1	1	1	1	1	1	7	100%
S10	1	1	1	1	1	1	1	7	100%
S11	1	1	1	1	1	1	1	7	100%
S12	1	1	1	1	1	1	1	7	100%
S13	1	1	1	1	1	0.5	1	6.5	92.85%
S14	1	0	1	1	0.5	1	1	5.5	78.57%
S15	1	1	1	1	1	1	1	7	100%
S16	1	1	1	1	1	0.5	1	6.5	92.85%
S17	1	1	1	1	1	1	1	7	100%
S18	1	1	1	1	1	1	1	7	100%
S19	1	1	1	1	1	1	1	7	100%
S20	1	1	1	1	1	1	1	7	100%
S21	1	1	1	1	1	1	1	7	100%

2.3 Analysing Research Study and data Analysis

The research methods and quality assessment tool those applied in this study are summarized in more details in table 6:

Table 6: Data coding selected papers

Research Method	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21
Case Study		√		√			√	√										√		√	
Comparative causal mapping																					
Experiment	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Survey															√						
Focus Group																					
Discussion	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Unclear																					
Qualitative																					
Quantitative	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
Mixed																					
Unclear																					
Study Setting																					
Academic	√	√	√	√	√	√	√	√	√	√	√	√		√	√		√		√	√	
Industry		√		√			√	√			√		√		√	√		√	√	√	√
Interview																					
Dataset (Data Science)	√	√	√	√	√	√	√	√	√	√	√	√	√		√	√	√	√	√	√	√
Archival Record										√	√	√	√					√			
Observation																					
Questionnaire																					
Workshop																					
Focus Groups																					
Year of publication	2016	2021	2020	2020	2019	2020	2018	20180	2020	2019	2019	2015	2019	2018	2022	2022	2022	2020	2021	2020	2020
Citation	121	6	101	34	23	13	298	28	31	18	11	87	25	34	1	0	2	17	6	11	34
Journal Rating (SJR) -H - Index	127	406	406	20	127	13	46	NA	47	41	279	18	53	8	325	172	40	10	7	15	40

2.4 Literature Review

The use of AI in video surveillance systems has been the subject of numerous investigations. In numerous articles, the idea of weapon detection has been examined. Darker et al. first proposed the concept of firearms detection at crime scenes by studying the human poses that refer to a person was carrying a weapon (Darker et al. 2007). The UK-based MEDUSA project team carried out the first test in which they inserted a sensor for weapon detection into CCTV. Dee and Velastin's excellent overview of the most recent developments in self-operating surveillance systems is on exhibit (Dee & Velastin 2007). Aside from that, concealed weapon like handguns, swords, and knives could be detected using microwave radar waves. X-ray screening can also identify metal weaponry. These approaches' drawbacks include high costs, a lack of applicability in modern settings like banks, institutions, and universities, and most importantly harmful impacts on people's health. As a result, utilizing AI in video surveillance systems is faster, cheaper, easier, and healthier than using previous methods. Table 7 displays the paper content analysis for nine studies and contrasts the approaches to our proposed models.

Table 7: Paper Content Analysis.

#Paper	Algorithms	Detection type	Datasets	Findings
1 (Grega et al. 2016b)	Faster-RCNN	Pistols, knives, phones, credit cards, money.	Database-Sohas_weapon-Test.	This paper focuses on developing an AI model for recognizing small items that may confused with handguns like wallet, phone, in surveillance system.
2 (Salido et al. 2021)	R-CNN, RetinaNet, YOLOv3	Handguns.	Customized dataset	The study covers three CNN models (Faster R-CNN, RetinaNet, and YOLOv3) for building handguns detection in surveillance system.
3 (Pérez-Hernández et al. 2020)	Faster-RCNN.	Handguns, knives.	Database-Sohas_weapon-Test..	This study focuses on building a weapon detector for small weapons.
4 (Pérez-Hernández et al. 2020)	YOLOv2,	excavators' status, people, workers.	Customized dataset.	This study presents a real-time smart tracking system to identify people approaching potentially risky locations on a building site.
5 (Fernandez-Carrobles, Deniz & Maroto 2019)	Faster R-CNN,	Knives, Guns,	COCO dataset, customized dataset.	This study presents traditional gun detectors using R-CNN algorithms. COCO dataset has been used to train the proposed model alongside with many augmentation methods.
6 (el den Mohamed, Taha & Zayed 2020)	CNN.	Pistol,	Customized dataset.	The paper shows two distinctive techniques that contribute in building gun detection, AlexNet and GoogLeNet. The model in this study also took into account the poor images and low resolution frames.
7 (Muhammad, Ahmad & Baik 2018)	CNN	Fire	Chino's dataset, Foggia's video dataset.	This paper introduced a early fire detector using CCTV surveillance system, which it is very beneficial to disaster management system.
8 (Mumtaz, Sargano & Habib 2018)	Deep convolutional neural networks (CNN)	Quarrelsome, violence actions.	Hockey dataset, Movies dataset, ImageNet.	The paper suggested a model to keep people secure by keeping an eye on their behavior and alert for any fights or other acts of violence in public places. For the Movie and Hockey datasets, the deep CNN violence detector outperforms existing approaches with an accuracy close to 100%.
9 (Salazar González et al. 2020)	Faster R-CNN	Handguns,	UGR - Handgun dataset, Unity Synthetic Dataset,	This study mentions challenges of employing weapon detectors in real life scenarios, like low average precision, more time need to be

			Mock attack dataset.	allocated to improve the system, and high false positive rate.
10 (Romero & Salamea 2019)	CNN, VGG,	Handguns, pistols.	Customized dataset.	This research presents result of train a deep learning model used for robberies detection, by focusing on images where people are considered an important object to be focused on. Also, it train the model on gray scale data which lead to higher performance.
11 (Lim et al. 2019)	Multi-Level Feature Pyramid Network, M2Det.	firearms.	Granada dataset, UCF Crime dataset, custom dataset.	The M2Det method was used in this paper to develop a model for handgun recognition. It was trained on three datasets, however the customized dataset, when compared to other existing datasets, yielded the highest accuracy.
12 (Tiwari & Verma 2015)	SVM, PCA, neural network classifier.	Knives, handguns.	Gun Video Database, Knife Image Database.	The paper concentrates on knives and pistols recognition. Also, aims to deploy the model in surveillance CCTV cameras in houses.
13 (Olmos et al. 2019)	Faster R-CNN with VGG-16, Faster R-CNN with ResNet.	Pistols.	Handgun dataset	The paper introduces a symmetric dual camera system to get 3-D info to get rid of a background and thus improve the proposed model.
14 (Mosselman, Weenink & Lindegaard 2018)	Video data	Knives, guns,	n/a	The paper introduces a different way to detect robberies by using video data. The proposed method can used to detect robbers by their postures both and victims' as well.
15 (Yadav, Gupta & Sharma 2023)	Classical machine learning models, two stage deep learning models.	Weapons in general.	IMFDBs, Knives images database,	The paper shows a survey about implementation of one-stage deep learning models and two stage as well. Also, it covers many public datasets for weapon detection scenarios.
16 (Kambhatla & Ahmed 2023)	State-of-art-object detection methods.	Handguns, guns.	Customized dataset.	In this study, they have suggested a technique for finding visual weapons in images that makes use of the Harris interest point detector and color-based segmentation.
17 (Dwivedi, Singh & Kushwaha 2021)	CNN with VGG-16	Guns, bombs	Customized dataset.	This paper alleviates many limitations for weapon detectors presenting an algorithm to generate new images and another algorithm to preprocess images for quality improvement.
18 (Ahmed & Echi 2021)	Mask R-CNN, CNN.	Knives, machine guns, masked face, RPG.	Customized dataset.	The Hawk-Eye threat detector for real-time video surveillance was designed and put into use in this study.
19 (Lim et al. 2021)	M2Det, Faster R-CNN, YOLOv3, RetinaNet, CenterNet, Mask R-CNN	Handguns, guns,	Granada dataset	An improved deep multi-level feature pyramid network is proposed in this paper to handle the challenge of inferring firearms from a non-canonical standpoint.
20 (Mehta, Kumar & Bhattacharjee 2020)	YOLOv3	Fire, guns,	IMFDB dataset, UGR dataset, FireNet Dataset,	A real-time frame-based, effective fire and gun detection computer vision model with a high accuracy metric has been provided in this study. Also, it shows the detections per frame can be used on any GPU-based system and are suitable for real-time monitoring.

21 (Jain et al. 2020)	Faster R-CNN, SSD	weapons	Customized dataset	This study compares between SSD and Faster R-CNN algorithms in terms of speed and accuracy for weapon detection models. it proves that SSD is the best for speed, while faster R-CNN is better for the accuracy.

Early firearms detections during thefts or acts of violence while guarding banks, ATMs, and keeping an eye on public stations can save financial loss and reduce inconvenience for the general population. The above table presents an assortment of studies on firearms, violence, and fire detection for CCTV systems. The table presents several recognition categories like knife recognizer (Grega et al. 2016a), handgun detector (Salazar González et al. 2020),(Elmir, Y., Laouar, S.A. and Hamdaoui, L., 2019, April. Deep Learning for Automatic Detection of Handguns in Video Sequences. In JERI. n.d.),(Olmos, Tabik & Herrera 2018), (Lim et al. 2019), (Warsi et al. 2019), intruder sensor (Kanthaseelan et al. 2021), fire detector (Muhammad, Ahmad & Baik 2018), and aggression detector (Mumtaz, Sargano & Habib 2018). However, in a street scene, the CCTV camera is relatively far away from people, has low resolution, and occasionally shines weakly settings, which makes it a little challenging to detect small harmful objects. In bank indoor space, high resolution CCTV cameras placed close to the people make it simple to spot potentially dangerous items. Due to a shortage of train data, one option to increase accuracy is to create a dataset using the Unity engine (Salazar González et al. 2020).

2.5 SLR Results

According to selecting research papers highlighted in table 7 which published between 2015 and 2022 in the context of weapon detection technology, and this technology are supposed to be employed in banks, institutional buildings, hypermarkets, …, etc. this section shall focus on systematic literature review questions and answer them as a key result of the SLR.

Starting with the first SLR question "How is AI implemented in video surveillance systems?". as matter of fact, all research papers that are shown in table 7 described how surveillance systems could be improved by employing AI techniques. In this problem, neural networks alongside with machine learning classification techniques used to build weapon detectors to detect guns, pistols, masks, and thus deter criminals or even minimize losses. These models are based mainly on state-of-art-deep learning techniques such as SSD, YOLO, Faster R-CNN, neural network classifier, CNN, …, etc. However, the proposed models are trained using label images, where target objects in each image are labeled by humans to teach the model what kind of weapon and where is it. Finally, the proposed model is employed in CCTV cameras in the designated area that needs to be protected.

Moving to the second SLR question "What are most advanced techniques used for the weapon detection?". it is obvious that Faster R-CNN is the most common algorithm for weapon detection according to the previous table. Also, R-CNN, CNN, and Mask R-CNN are pretty used in this field. At the meantime, there are many modern techniques used in object detection such as Single Shot MultiBox Detector and YOLO. Both are quite popular, and YOLO has an advantage with tiny objects. Now moving to the third SLR question "How is the proposed model work with low-resolution frames?". Well, low-resolution frames are considered a major issue in weapon detection due to low quality of footage extracted from CCTV cameras. Many solutions could be applied to overcome poor image quality, one of these solutions is to train the proposed model on low resolution dataset, change the image sizes into one appropriate size, using super resolution techniques via VDSR network (Nasrollahi & Moeslund 2014) to improve image quality. Finally, moving to the fourth SLR question "Is the proposed gun detection model recommended in bank area?". By reviewing all models in selected papers in table 7, some models are applicable in the bank area because the model has been trained on the customized dataset. The dataset contains many images taken from bank footage, to let the model learn the bank environment. However, datasets that include images from hypermarket video footage, street footage, and home security camera footage are great datasets for developing gun detection for banks, but they are not good as footage from banks.

2.6 Research Methodology.

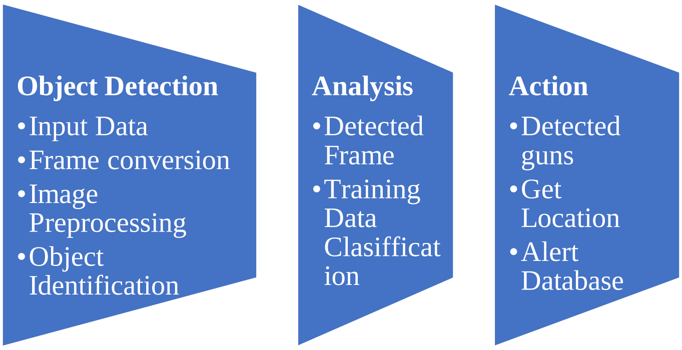


Figure 5: The flow of research methodology.

Our scheme of the research was divided into three groups, as shown in figure 5. The methodology for applying object detection techniques in various locations' video surveillance systems is proposed in this part. In this study, we want to develop a neural network-based model to detect any risky objects or specific objects that might be indicators of robberies or criminal activity. The two main steps in identifying dangerous objects are determining the bounding box of the target item (such as pistols) and categorizing the bounding box where items are present. Numerous object detection methods and deep transfer learning approaches have been used in various applications as was previously mentioned in the problem solution section. As previously stated, our method for detecting firearms applies transfer learning using SSD algorithms, as will be demonstrated in the subsections that follow.

3 CHAPTER THREE: CONCEPTUAL FRAMEWORK

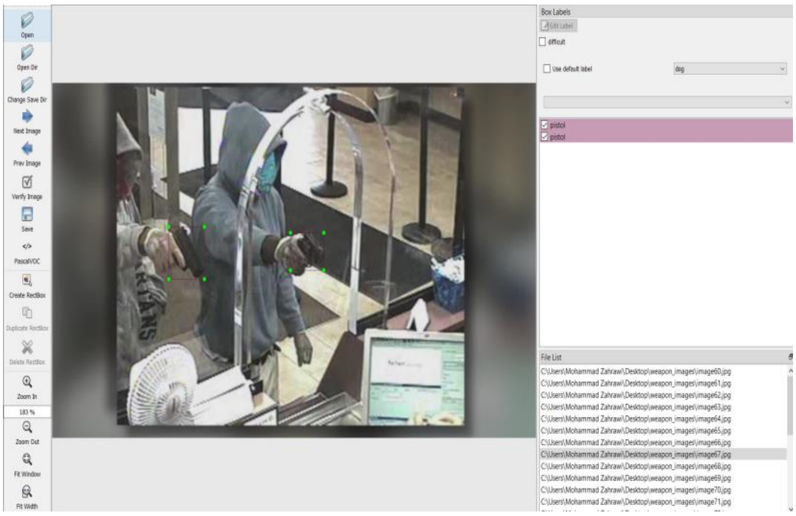
3.1 Labelling

Images labelling is one of the requirements of building object detection models, and models cannot be built without this process. It is annotating images with tags or labels by specifying a bounding box around each object that wanted to be detected or discovered. Usually, those who add labels or tags to images are called annotators or labelers. The annotated images are used for supervised machine learning tasks, used to teach model shapes and colors of intended objects. Mislabel images and flaws labels can lead to a lower accuracy model and therefore decrease the likelihood of implementing the model in real-life scenarios especially those that have an impact on human life such as cancer detection.

There are various ways of data labeling, according to the use cases. In an object-detection task, we are looking for the location of certain objects (weapons in our case) as well as their classes. Hence, we create bounding boxes around the target items in the images. Meanwhile, in image classification tasks, we are looking for classes of images only regardless of the location of the main features. There are many images labeling tools are available on Github, which are easy to use and completely free such as labelImg, labelme (*GitHub - wkentaro/labelme: Image Polygonal Annotation with Python (polygon, rectangle, circle, line, point and image-level flag annotation).* n.d.), CVAT (*GitHub - openvinotoolkit/cvat: Powerful and efficient Computer Vision Annotation Tool (CVAT)* n.d.), hasty.ai (*GitHub - openvinotoolkit/cvat: Powerful and efficient Computer Vision Annotation Tool (CVAT)* n.d.), labelbox (*GitHub - Labelbox/labelbox: Labelbox is the fastest way to annotate data to build and ship computer vision applications.* n.d.), in this research we use labelImg to annotate weapons and other similar objects.

labelImg is a commonly used open-source tool. It is only appropriate for object localization or detection tasks, and it is uniquely able to draw rectangle boxes about considered objects. Even with restrictions, we would like to recommend this graphical annotation tool mainly focused on drawing rectangles boxes which simplifies the tool as much as possible. Also, it has all the essential functionality and convenient keyboard shortcuts. This tool can save and load as well tags in three standard annotation formats: VOC, YOLO, and PASCAL.

As we mentioned earlier, this process is very important and delicate to achieve great results. However, many projects took a very long time to be completed because of poor labelling quality and ambiguous defect definitions, which leads to shorten lifetime of model and lower reliability. Therefore, in order to construct a dataset with high quality tags, it is very crucial to spend time in the project's early phase to treat defect definitions and validate labelling.



□ difficult

□ Use default label dog

☑ pistol
☑ pistol

C:\Users\Mohammad Zahrawi\Desktop\weapon_images\image60.jpg
C:\Users\Mohammad Zahrawi\Desktop\weapon_images\image61.jpg
C:\Users\Mohammad Zahrawi\Desktop\weapon_images\image62.jpg
C:\Users\Mohammad Zahrawi\Desktop\weapon_images\image63.jpg
C:\Users\Mohammad Zahrawi\Desktop\weapon_images\image64.jpg
C:\Users\Mohammad Zahrawi\Desktop\weapon_images\image65.jpg
C:\Users\Mohammad Zahrawi\Desktop\weapon_images\image66.jpg
C:\Users\Mohammad Zahrawi\Desktop\weapon_images\image67.jpg
C:\Users\Mohammad Zahrawi\Desktop\weapon_images\image68.jpg
C:\Users\Mohammad Zahrawi\Desktop\weapon_images\image69.jpg
C:\Users\Mohammad Zahrawi\Desktop\weapon_images\image70.jpg
C:\Users\Mohammad Zahrawi\Desktop\weapon_images\image71.jpg

Figure 6: Graphical user interface of labelImg tool.

Labelling images can be time-consuming, and in some scenarios, it is necessary to have knowledge-domain experts such as neurologists for brain tumor detection, which cannot be labelled by anyone who lacks knowledge. In our case, weapons and other related objects can be labelled by people without the need to any specific experience or knowledge in any domain.

3.2 OpenCV

An open-source software package called OpenCV is used for computer vision and deep learning. OpenCV can bring machine perception into commercial products by preparing infrastructure for computer vision applications. ItSeez and Willow Garage are now maintaining OpenCV, originally developed by Intel in 1999. It supports Android, Linux, Windows, iOS, and Mac OS X. It can work with many programming languages such as C++, C, Python, and Java (Android) interfaces. It provides around 2500 algorithms (*About - OpenCV* n.d.)(*Home · opencv/opencv Wiki · GitHub* n.d.).

OpenCV could be used in many tasks such as face recognition, object detection, organizing individual actions in videos, shaping 3D models of items, creating 3D point clouds from stereo cameras (Chaurasia & Mozar 2022), searching for similar images from an image database, following eye movements, enhanced images quality, improving the visibility level of foggy image and video, and establishing markers to overlay it with augmented reality, etc.

3.3 Object Recognition

A set of related computer vision algorithms that refer to identifying objects in images and videos. The ability to recognize objects is a major outcome of neural networks and machine learning methods. We can quickly identify individuals, items, and particular things when we glance at a photo or watch a movie. The aim is to help a computer learn how to recognize and understand images the way humans do. Object recognition is a fundamental feature of self-driving cars, permitting them to recognize a stop sign or distinguish between an individual and a lamppost (Alam, Mehmood & Katib 2018). It's also useful for disease detection in bioimaging, industrial inspection, and robotic vision, among other things (*DSpace at My University: OBJECT RECOGNITION USING IMAGE PROCESSING AND DEEP LEARNING IN MATLAB"* n.d.).

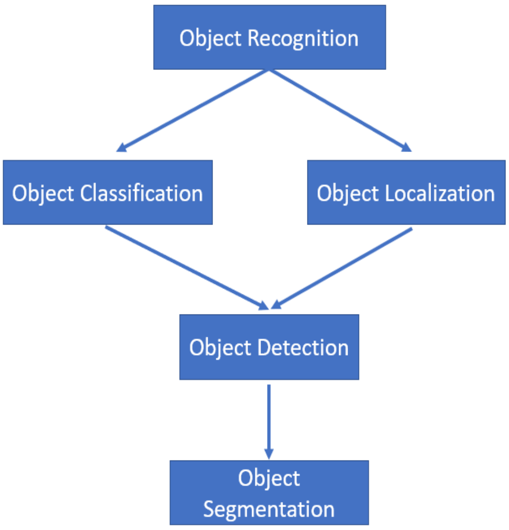


Figure 7: Flowchart shows object recognition computer vision missions

As shown in figure 7, object detection is a subcategory of object recognition, where the intended object is not only identified but also located in an image. It is also possible to identify, and even locate several objects in the same image/frame. Scene understanding in Object recognition is a complicated procedures in which a large number of items, in various locations and lighting conditions, and from various perspectives, are gathered at the same time.

In a naturalistic scene, objects can be displayed in different contexts and poses, making them computationally difficult to recognize any position, orientation, or distance to the viewer or other object. Object recognition work by comparing the features (representations) of items stored in memory with features extracted from the images (Mozer 2001).

3.3.1 Image Classification

Image classification is one of the crucial tasks in object recognition, it is concerned with identifying the class of one object in an image, predicting what is in the image and what level of confidence is expressed as a probability. The aim is to categorize the image by associating one tag (single-label classification) or more (multi-label classification) tags to a given image. Usually, Image Classification indicates to images in which only one item exists and is analyzed (Wang, Su & Zurich 2019).

The most common classification task is the single label where the model outputs a vector with many values equal to a number of classes, and the value indicates to the probability of the image belonging to this class. As we mentioned earlier, the output of the classification model is a vector where A Softmax activation function is used to ensure the total values sum up to one and the max of the values is taken to make the model's output. Figure 7 presents simple architecture of convolutional neural networks for image classification purpose.

Another type of image classification in terms of a number of tags is multi-label classification, where each image can contain more than one tag, and some images can have all tags at once. Multi-label classification tasks are broadly used in the medical imaging area, where a patient may have several diseases that can be diagnosed using visual data such as X-rays. The performance of Image classifier models must be compared to other models. There are several well-known metrics that are used in Image Classification such as precision, recall, and F1 score. Scores for precision and recall are heavily dependent on the type of problem that is trying to solve. Recall is very important indicator for problems related to medical image analysis, for instance, the detection of pneumonia from medical images cannot show false negatives to prevent the patient from being diagnosed as healthy when actually sick (Allaouzi & Ben Ahmed 2019). However, precision is crucial when you need to minimize false positives, for example, detection of email spam, where users confront serious problems when important emails are classified as spam (Wu et al. 2005).

3.3.2 Object Localization

A high-level computer vision system involves finding the location of the object in a given image. Occasionally, however, Objects in an image are analyzed in more detail for their pose or certain regions. In our case, we are more interested in detecting weapon poses alongside weapons and other objects related (Heitz et al. 2009).

It is a method that produce a list of object categories that are visible in the image, alongside with an axis-aligned bounding box that describes the position and scale of each object. The primary distinction between localization and detection is that object detection searches for all items and their borders, whereas localization only looks for the most obvious object in an image. Therefore, splitting the image into multiple images and then running a neural network on all of them to detect objects.

3.3.3 Object Detection

This task uses properties of object localization and Image classification together, the detection model provides us with the bounding box's coordinates, as well as the class label and related width and height. The output box with our preferred threshold is a result of using a non-max suppression technique (*Object Detection vs Object Recognition vs Image Segmentation - GeeksforGeeks* n.d.). These techniques can deal with multi-class classification and localization, as well as multiple occurrences of objects.

One of the challenges of using object detection is the bounding boxes are always rectangular. Therefore, if the object contains a portion of curvature, it will not help define the shape of the object. Also, some metrics, such as an object's area and perimeter, cannot be reliably estimated using object detection (*Object Detection vs Object Recognition vs Image Segmentation - GeeksforGeeks* n.d.). Object detection techniques are widely used in different fields, they can be used for vehicle detection, people counting, pedestrian detection, and tracking many things like a person in a restricted area, football in a match, and animal behaviour.

Deep neural networks have considerably increased object-detecting operation. On computer vision datasets, neural network topologies such as GoogLeNet, VGG, and ResNet were utilized to build transfer learning models for object detection, segmentation, and tracking (Simonyan & Zisserman 2015)(Namphol, … & 1996 1996). In image analysis, convolutional neural networks have been shown to enhance performance, particularly in the domains of object detection and tracking (Cheong & Park 2017). Based on each location having an object of interest, bounding box proposals were created from the image (Dong et al. 2016)(Kang et al. n.d.). Each box was categorized into one of the object classes based on the features extracted from it. Using feature extraction and classification algorithms, object detection will have a low error rate (Ouyang, international & 2013 n.d.). Neural networks would be trained using the variant features retrieved and bitrate compressions of the images (Han et al. n.d.).

In our case, we proposed to deploy object detection models into video surveillance systems for weapon detection. We are dealing with a live video on CCTV, not normal images, which makes our project more challenging to be done. Also, many factors should be considered while dealing with video such as motion blur, object occlusion, drastic appearance, and location change of the same object as time passes (Wang et al. n.d.).

As with any other deep learning models, two ways to start building an object detector model either by creating and training a custom object detector or using a pretrained object detector. The customized object detector can be built from scratch by designing a network architecture that learns features and representations extracted from objects of interest. You also need to collect a huge collection of annotated images to train the CNN. A custom object detector can produce interesting results. However, you need to manually configure hyperparameters of neural network architecture including layers, weights, which needs a lot of time and training data.

In contrast, many object detection models take advantage of transfer learning techniques by loading a pre-trained model from a particular point and then fine tune it for our case (weapon detection). Due to the pre-trained models' extensive training on thousands or millions of photos, this method can be trained more quickly. In situations in which transfer learning is applied to a new project, one can accomplish much better performance than if they only trained with small amounts of data. ImageNet, AlexNet, and Inception are examples of transfer learning models. Whether you build a customized object detector or use a pre-trained model, you will need to determine what kind of object detection network: a single-stage network or a two-stage network.

3.4 Classification and Detection Approach

There are several methods for producing region proposals, but the sliding window approach is the most straightforward. The sliding window method is slow because the filter glides across the entire frame. Therefore, we have the following two main methods used in classification and detection models:

A. Sliding Window/Classification Models

In the approach of the sliding window, a box of appropriate size, say m x n is selected to slide over the target image (Giusti et al. 2013). Searching over the whole image for objects is an exhaustive process. It is not only necessary to search in the image for all possible places, but also to conduct searches on different scales due to the fact that models are usually trained on a specific range.

The sliding windows approach can be divided into four main phases starting by scan images at all scales and locations, then extracting features over windows, after that run the classifier on all locations, and lastly fusing multiple detections in 3-D positions and scale space. The sliding window approach is computationally expensive (Bhatti et al. 2021), because of the search with multiple aspect ratios and particularly if the step or stride value is small for big-size images.

Binary classifiers are the core component of the task. weapon classifiers check whether a weapon is in the image based on the part of the image that overlaps with the window. After that, a window is slid, and the process is repeated until the window has passed through the entire area in the image. This process can be characterized as brute force with a lot of local decisions.

Features are extracted from raw image data, which are assumed to be informative, helpful, condensed, non-redundant, and helpful. These features are fed into the classifier, then A decision is made based on the features given. There are several feature extraction systems and feature representations such as pixel-based representations, color based representations, and gradient based representations.

B. Region Proposal/Object Detection Models

In computer vision, the latest state-of-the-art techniques show some hopeful models (e.g. R-CNN (*CVPR 2014 Open Access Repository* n.d.)) that explain the remarkable performance regarding detection accuracy. However, this method requires many computing resources to generate and classify many proposed regions. Selective search is a frequent method employed in object detection for producing object proposals (Uijlings et al. 2013). The computational cost is high, especially when extracting deep features from many proposed regions. It is noted that semantically significant regions can be spotted at deeper layers.

The region proposal generates many candidates of bounding boxes to assess the likelihood of a prospective object or intriguing information based on the type of object detection (Ma et al. 2017). Examples of region proposal methods are R-CNN (Girshick et al. 2014b), SPP-NET (He et al. 2014), Faster R-CNN (Ren et al. 2017), R-FCN (Dai et al. n.d.).

The proposed R-CNN model contained three modules, region proposal, feature extractor, and classifier. R-CNN generates 2000 region proposals through a selective search algorithm, those regions are transformed into squares and fed into a CNN network that yield 4096-dimensional feature vector. AlexNet model was used as feature extractor for R-CNN that won in the ILSVRC-2012 image classification competition (Russakovsky et al. 2014).

The SVM classifier feeds features extracted from the target picture into the output dense layer to determine if the target object is present in that region (Girshick et al. 2014b).

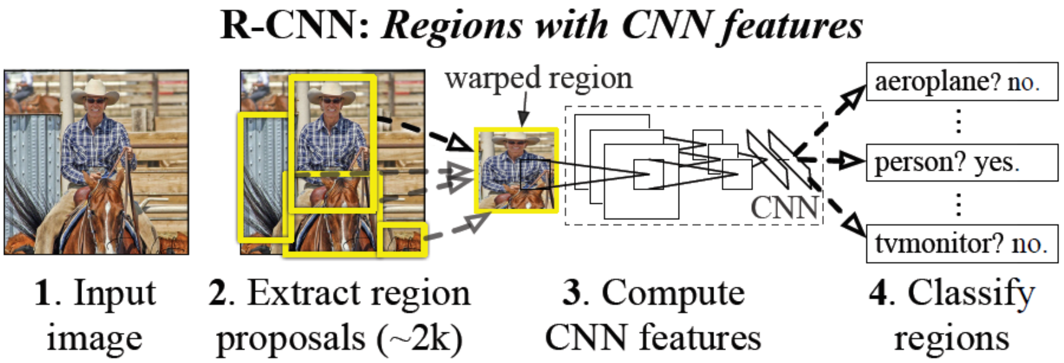


Figure 8: Summary of the R-CNN Model Architecture (Girshick et al. 2014a).

There are some limitations to implementing R-CNN models such as taking a long time to train the model by classifying 2000 region proposals for each image. It cannot be deployed in real-time applications as it needs around 47 seconds for each image (Girshick et al. 2014a).

Ross Girshick proposed an extension to R-CNN to overcome speed issues by releasing Fast R-CNN (Ren et al. 2017). Fast R-CNN is quite like R-CNN, here the difference is by feeding the input image instead of region proposals to the CNN network to create a feature map. Thereafter, region proposals are recognized from the feature map, reshaping them into squares and by adopting the RoI pooling layer, they have restructured again into fixed shapes, so that they can be fed into fully connected layers, therefore, a softmax layer is used to forecast the class of the proposed region.

The previous two models R-CNN and Fast R-CNN employ a selective search algorithm to identify the region proposals. Selective Search algorithms are quite sluggish and take a long time and it have an impact on the model (Ren et al. 2017). Shaoqing Ren et al recommends a more advanced object detection technique that eliminate the selective search method and allow the network to learn the region proposal (Ren et al. 2017).

Faster R-CNN is quite like Fast R-CNN in terms of input which the data is fed into a convolutional network that generates a convolutional feature map. Instead of locating the area proposals on the feature map using a selective search technique, a neural network is utilized to predict them. After that, the Region of Interest (RoI) pooling layer is utilized to enlarge the proposed regions, then categorized the proposed region and compute the bounding box's offset value (Saikia et al. 2021).

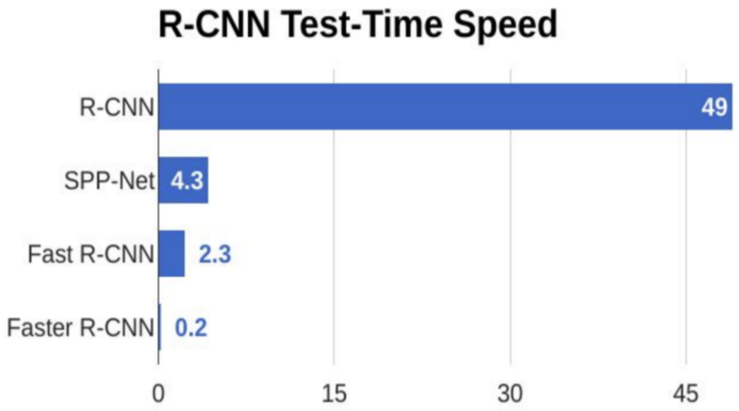


Figure 9: Comparison of test-time speed of object detection algorithms (Saikia et al. 2021).

Faster R-CNN is obviously faster than its predecessors, as shown in the graph above. As a result, Fast R-CNN can even be deployed to detect objects in real-time. The entire system of Fast R-CNN is shown in figure 10.

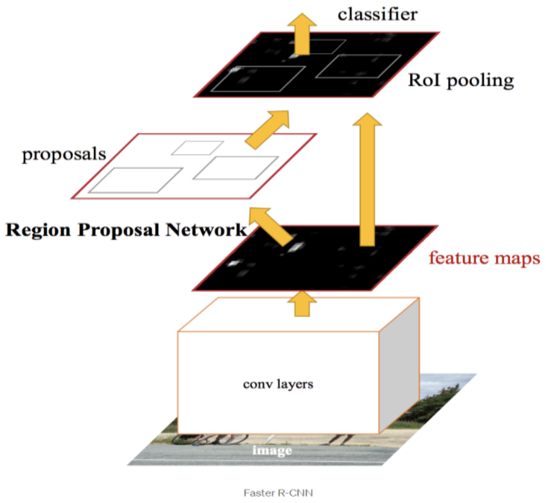


Figure 10: Fast R-CNN architecture (Ren et al. 2017).

3.5 Backbone Networks

In this paper, the VGG-16 neural network, Inception V2 network, MobileNet V2 network, ResNet50 V1 Network, and Darknet network are introduced.

3.5.1 VGG16

VGG-16 is a type of convolutional Neural Network (CNN) known as ConvNet, created by Karen Simonyan and Andrew Zisserman from the Visual Geometry Group at the University of Oxford (Simonyan & Zisserman 2015). VGG-16 net has input and output layers and several hidden layers. In VGG-16 architecture, there are 13 convolutional layers, 5 Max Pooling layers, and other 3 Dense layers which total 21 layers, but only 16 layers have learnable parameters (convolutional and dense layers) (Verma & Dhillon 2017). VGG16 is a large network with roughly 138 million parameters (Karen Simonyan∗ & Andrew Zisserman+ 2018).

The input layer in VGG-16 receives an image with dimensions 224x224x3. The first two convolutional layers have 64 filters each of 3x3 size with stride 1, followed by a maxpool layer of 2x2 filter of stride 2. Then the number of filters doubled to 128 filters for the next two layers with the same size filter 3x3 and stride 1. In the last two convolutional layers, the number of filters jumped to 512. Three dense layers (fully connected layers) follow the stack of convolutional layers (Verma & Dhillon 2017). There are 4096 neurons in each of the first two dense layers and 1000 neurons in the last dense layer matching the number of classes in the ImageNet dataset. The Softmax function is applied on the last layer for categorical classification. Figure 11 shows the architecture of the VGG-16 network (Morales et al. 2019).

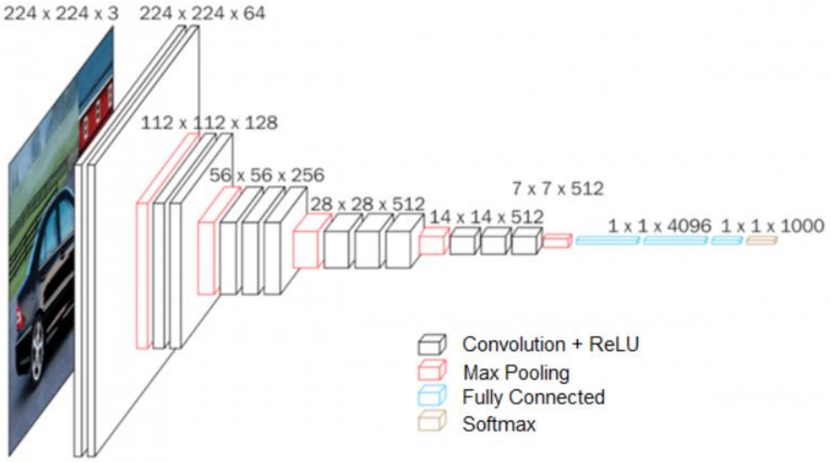


Figure 11: VGG-16 architecture (more details).

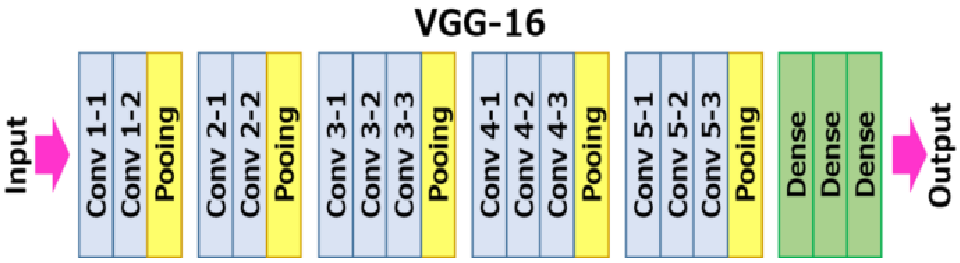


Figure 12: VGG-16 architecture.

3.5.2 Inception V2

There is a straightforward way of designing better object detection models either by adding more layers or adding more neurons in each layer. However, the number of parameters increased in the model means more computing resources need to train the model. A solution for this is, inception nets are employed in neural networks to reduce computation expenses for deeper networks and overcome overfitting issues. The idea is to use various kernel filter sizes within the CNN concatenated into a single output layer instead of stacking them sequentially (Szegedy et al. 2015).

Inception nets are used to act as multi-level feature extractors by performing convolutions by three filters with different shapes (1x1, 3x3, 5x5) and maxpooling operations as well. The outcomes are concatenated and fed into the next layer. By comparing to CNN architecture, the network becomes wider, not deeper. The computational cost can be reduced further by introduced 1x1 convolution before 3x3 and 5x5 layers and after maxpooling layer.

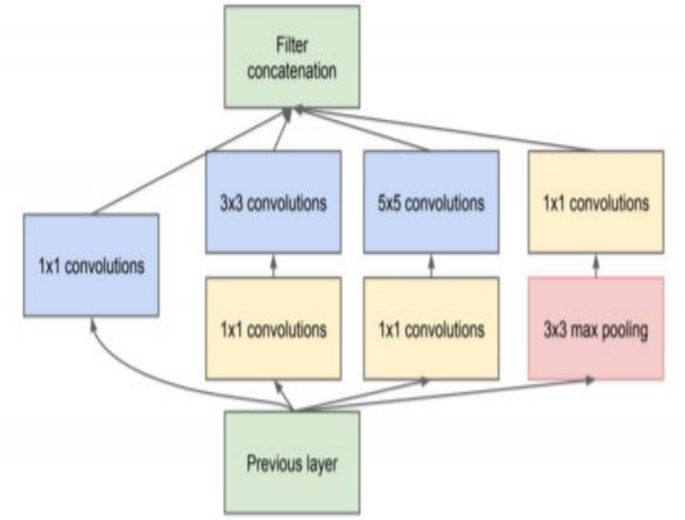


Figure 13: Inception module with dimension reductions (Szegedy et al. 2015).

Inception-v2 works in a similar way to inception-v1 with some changes to make the model more efficient and faster. One of the changes is to replace 5x5 convolutions with two 3x3 convolutions in order to reduce computational cost as shown in figure 14. Since 3x3 filter is 2.28 cheaper to compute than 5x5 filter, therefore stacking two 3x3 convolution filters results in improving performance (Szegedy et al. 2016). Another change is employing asymmetric convolutions, where A 3x3 filter can be replaced by a 1x3 filter followed by a 3x1 filter

Inception-V3 is quite like Inception V2 with minor changes. Inception V3 uses RMSprop optimizer, batch normalization, using 7×7 convolution filters, and labeling smoothing regularization (Szegedy et al. 2016).

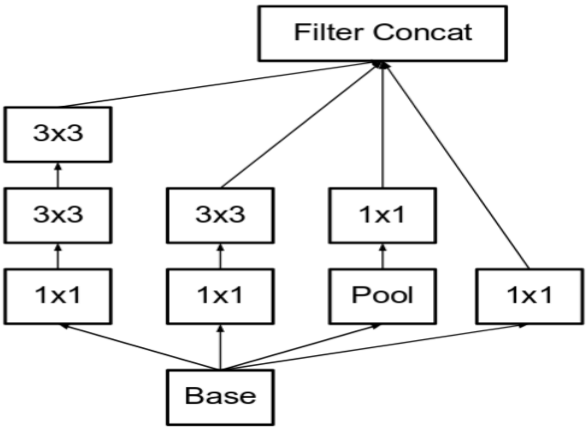


Figure 14: The original inception module's leftmost 5x5 convolution is represented now by two 3x3 convolutions.

3.5.3 MobileNet

A lightweight deep neural network, it is designed to be used in mobile applications and embedded vision systems as well. The core module of MobileNet is Depthwise Separable Convolutions, which is used to decrease the number of parameters compared with normal convolution nets and improve the accuracy (Howard et al. 2017).

Pointwise convolution and Depthwise convolution make up the depthwise separable convolution. The pointwise convolution filter uses a 1x1 filter to project the channel's output from the depthwise convolution onto a new channel space. The depthwise convolution filter performs one convolution on each input channel (Howard et al. 2017). The following figure compares standard convolution with depthwise separable convolution.

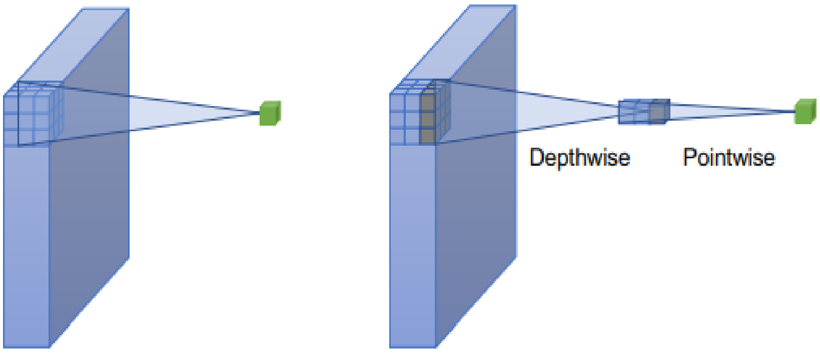


Figure 15: Standard convolution and depthwise separable convolution (Guo et al. 2019).

Resolution Multiplier and Width Multiplier are two hyperparameters introduced in MobileNet, width multiplier allows the model to trade off the latency against speed. While the resolution layer works on minimizing computational cost (Howard et al. 2017).

3.5.4 ResNet50 V1

A widely known 50-layer deep convolutional neural network called ResNet, which stands for Residual Networks. it serves as the groundwork for numerous computer vision applications. ResNet model was the winner of the ImageNet challenge in 2015. As matter of fact, the error rate could be reduced by using more layers in deep neural networks, it works for adding a few layers. However, there is a common problem that appears when adding too many layers called a Vanishing / Exploding gradient, where the gradient vanishes to 0 or becomes too large, therefore the training and test error rate increase (He & Sun 2014).

Deep residual learning (ResNet) was introduced to address the vanishing gradient problem. ResNet uses a skip connection technique between layers as shown in figure 16 (Jian 1996). This skip connection adds the output of previous layers to the output of stacked layers by skipping some layers in between. skip connection technique allows us to train an extraordinary deep neural network with 150 layers.

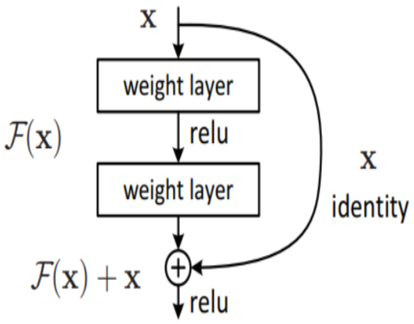


Figure 16: Skip Connection

There are many different ResNet models with same concept but a different number of layers, such as ResNet-18, ResNet-34, ResNet-101, ResNet-110, ResNet-152, ResNet-164 etc. The ResNet50 model architecture consists of 4 stages as shown in figure 17. The height and width of the input layer should be multiples of 32. The default input size is 224 x 224 x 3 where 3 represents channel width. The convolution filter 7x7 and 3x3 maxpool layer perform on the input layer with stride 2. Then, the network enters Stage 1, which consists of 3 residual blocks containing 3 layers each, a total of 9 layers. The size of filters used in all three layers of the block in stage 1 is 64, 64, and 128. The arrows denote the skip connections. As we move from one stage to another, the input size is halved while the channel width is doubled. The ResNet-50 has over 23 million trainable parameters. The network ends with an average pooling layer followed by a 1,000 fully connected (fc) layer with softmax activation.

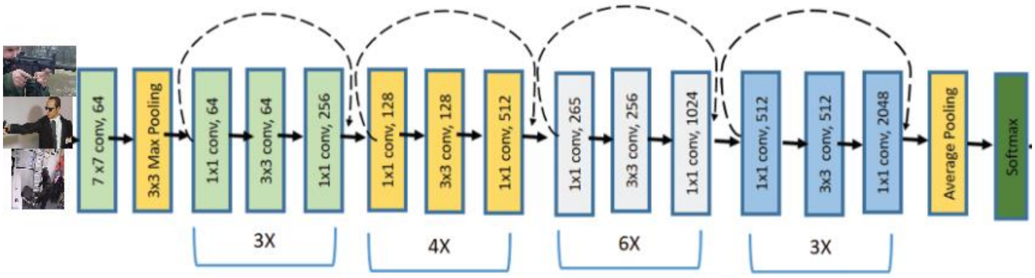
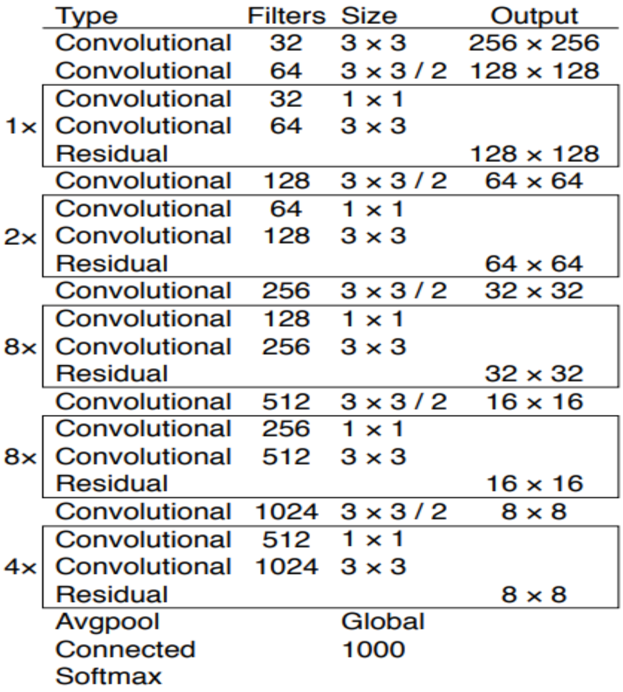


Figure 17: Block diagram of the Resnet50 network

3.5.5 Darknet

A convolution neural network is used as a feature extractor for YOLO models. Darknet-53 is more powerful than Darknet-19. Darknet-53 consists of 53 layers, it uses successive 3x3 convolutional kernels, 1x1 convolutional kernels to reduce the number of parameters, and residual skip connections like ResNet architecture (Redmon & Farhadi 2018). Darknet-53 achieved the highest floating-point operations per second. This indicates that the network structure makes better use of the GPU, which results in faster evaluation. The table 8 shows the architecture of Darknet-53(Redmon & Farhadi 2018).

Table 8: Darknet-53 architecture.



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

3.6 Object Detection Algorithms

As we show in the previous section what the backbone network is, and how they are used to encode the images input into certain feature representations. In this paper, we use SSD and YOLO algorithms combined with one backbone network to build a weapon detector. The object detection models used in building weapon detectors are SSD MobileNet V2, Yolov4, SSD ResNet V1, SSD Inception V2, Yolov5.

3.6.1 SSD

SSD model stands for the single shot detector, it is much faster than the Faster R-CNN model (Liu et al. 2016), it does not need to create boundary boxes to classify target objects. SSD employs several enhancements, including multi-scale features and default boxes, to improve the accuracy and beat up Faster R-CNN's accuracy. SSD model is composed of two main processes, first extracting feature maps and applying convolutional filters to identify intended items (Liu et al. 2016).

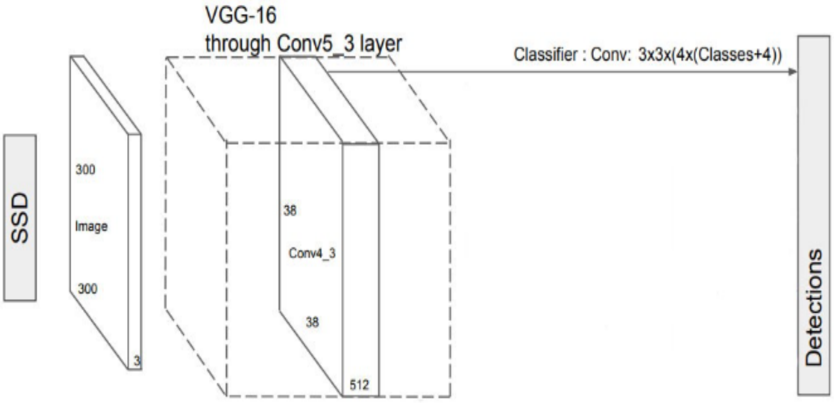


Figure 18: VGG-16 backbone network in Single Shot detection (SSD) model architecture (modified) (Liu et al. 2016).

SSD model employs VGG16 to extract feature maps. Afterward, it detects target objects by Conv4_3 layer as shown in figure 18. The size of the Conv4_3 layer is 38 x 38 and makes four predictions for each cell irrespective of its depth, which means 38 x 38 x 4 predictions. Each prediction composes of a boundary box and a number of scores as same as a number of classes plus one (for no object) (Liu et al. 2016).

Subsequently, the SSD model provides 3x3 convolution filters for each cell to evaluate either the scores for each prediction or to produce a small set of default bounding boxes. These default bounding boxes are important to SSD models as same as anchor boxes to Faster R-CNN. However, during the training phase, the SSD model uses the Intersection over Union (IoU) metric to match the ground truth box with the predicted box. IoU with the truth > 0.5 will result in a positive label for the box. A SSD model draw bounding boxes in every cell in the image, with multiple different sizes, at some different scales. Therefore, compared to other models, the SSD model creates a significantly higher number of bounding boxes, and almost all of them are classified as negative examples (Liu et al. 2016).

As shown in figure 19, there are extra feature layers after VGG-16 neural networks that scale down the size of layers. The variable size of extra feature layers is important to capture both large and small target objects.

As matter of fact, the number of negative matches with IoU<0.5 is much larger than positive matches, which leads to the classes' imbalance, therefore it has a bad impact on the training phase. So, the model employs a hard negative mining technique to overcome the classes imbalance problems. In hard negative mining (Wan et al. 2016), only a small number of negative examples with the highest loss score is added to the training set. These negative examples are useful to the model to learn background space (Wan et al. 2016).

Finally, non-max suppression method is used to combine overlapping boxes into one final single box for each detected object (Hosang, Benenson & Schiele 2017). In simple words, if six boxes with similar dimensions contain the same object, non max suppression would keep the box that has the highest level of confidence and removes the others. SSD architecture looks like this:

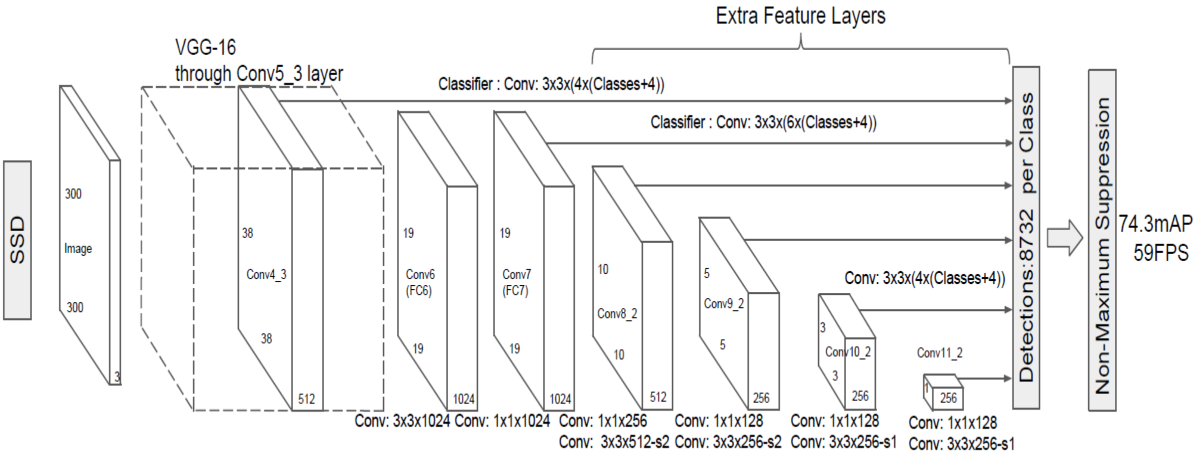


Figure 19: Architecture of SSD algorithm (Liu et al. 2016).

3.6.2 YOLO

As mentioned in section three of chapter one, YOLO stands for You Only Looks Once, and it is one of the single-stage object detectors. The first version was released in 2015 by Redmon et al (Zhiqiang & Jun 2017). Subsequently several versions have been published known as YOLOv2, YOLOv3, …, and YOLOv7 released in July 2022. This research paper only focused only on YOLOv4 and a small version of YOLOv5.

YOLOv4 was introduced in April 2020 by Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao (Bochkovskiy, Wang & Liao 2020). It is regarded as one of the most developed real-time object detection techniques available at the moment. As reported by its paper, YOLOv4 is 12% faster than YOLOv3, and 10% more accurate. The new architecture of YOLOV4 is constructed with CSPDarknet53 as a backbone, which supports CNN's learning capacity. The CSPDarknet53 is based on DenseNet. DenseNet was created to link layers in convolutional neural networks to address vanishing gradient issues and decrease the number of network parameters.

YOLOv4 utilized a Bag of Freebies that enhance the operation of the network with no increasing inference time. Most techniques in Bag of Freebies are related to data augmentation and these techniques are CutMix, Mosaic data augmentation, Class label smoothing, and Self-Adversarial Training (SAT). Starting with the CutMix technique deals with the problem of information loss by removing pixels from one image and replacing it with pixels from another image (Yun et al. 2019). Moving to mosaic augmentation creates a single image by combining 4 images, each of the four images is resized, then taking a random cutout of the combined image to obtain the final mosaic image. While the label smoothing technique deals with models who suffer from overconfidence (Müller, Kornblith & Hinton 2019).

YOLOv4 utilizes techniques that inference time marginally and can drastically enhance the accuracy of the object detector called “Bag of specials”. These techniques are also associate with incrementing of the receptive field, the implementation of attention, feature assimilation like skip-connections & FPN, and post-processing like non-maximum elimination (Bochkovskiy, Wang & Liao 2020).

Activation functions are designed to transform features while they move via the network. YOLOv4 utilizes the Mish activation function in the backbone Activation functions are designed to transform features while they move via the network. YOLOv4 utilizes the Mish activation function in the backbone to push the feature to the right and left. The following figure shows a graph of the Mish function (Non-monotonic 2020).

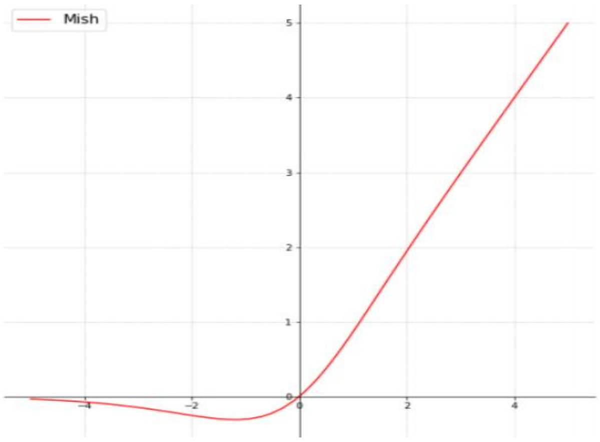
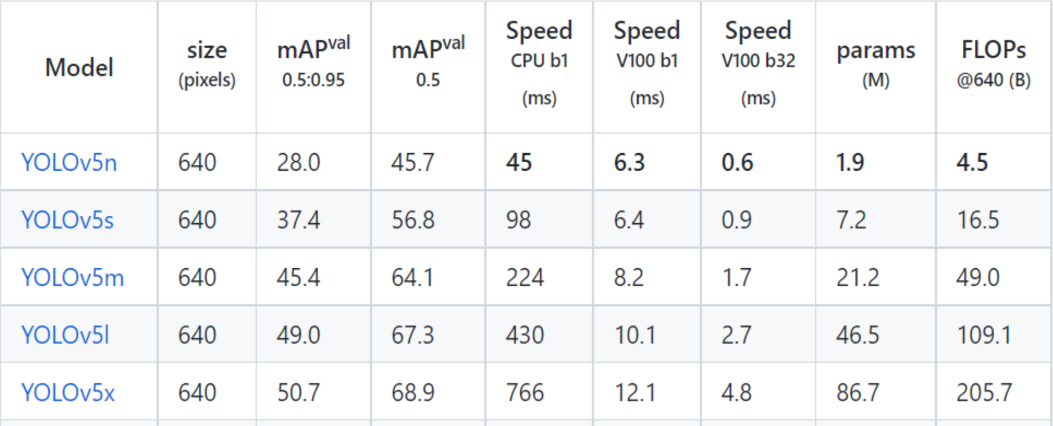


Figure 20: Graph of Mish Function.

Moving to YOLOv5, several network architectures of YOLOv5 are easier to use, have a relatively small model size (YOLOv5s), and their accuracy is comparable to the YOLOv4 model. However, deep learning practitioners still have concerns about employing YOLOv5 because it is less inventive than YOLOv4 (Jiang et al. 2022) despite the fact the environment is simple to set up, the training phase is very fast and simpler to put into production. YOLOv5 is written in the PyTorch framework which makes it easier to develop and maintain.

YOLOv5 is flexible to use, there are many models available in the YOLOv5 family starting from the smallest and fastest (YOLOv5n) to the largest and sharpest (YOLOv5x). The following table shows different versions of YOLOv5. YOLOv5n is a nano model, which is the fastest model but the least accurate. It is the best option for mobile applications. In this research, our default model is YOLOv5s which is a small model with 7.2 million parameters.

Table 9: Different models in YOLOv5 (ultralytics/yolov5: YOLOv5 in PyTorch > ONNX > CoreML > TFLite n.d.)



Model	size (pixels)	mAP^{val} 0.5:0.95	mAP^{val} 0.5	Speed CPU b1 (ms)	Speed V100 b1 (ms)	Speed V100 b32 (ms)	params (M)	FLOPs @640 (B)
YOLOv5n	640	28.0	45.7	**45**	**6.3**	**0.6**	**1.9**	**4.5**
YOLOv5s	640	37.4	56.8	98	6.4	0.9	7.2	16.5
YOLOv5m	640	45.4	64.1	224	8.2	1.7	21.2	49.0
YOLOv5l	640	49.0	67.3	430	10.1	2.7	46.5	109.1
YOLOv5x	640	50.7	68.9	766	12.1	4.8	86.7	205.7

4 CHAPTER FOUR: WEAPON DETECTION

4.1 Dataset Construction

In surveillance applications, it is expected that the input images are of relatively high quality (Dodge & Karam 2016). Processing time and high precision are important factors for the evaluation of real-time weapon detectors. Accurate, relevant, and high-quality images play a crucial role in the building and development any of computer vision model. Research has been conducted on armed robbery and states that the most frequent weapon used was firearms (Mouzos, Carcach & Australian Institute of Criminology. 2001).

Banks were the target of 29 robberies; 24 of the robbers had firearms, and none had knives (Gill & Matthews 2005). Although firearms remained the most preferred weapon choice, other types of robbery offenders preferred knives. As weapons appeared to be essential parts of the robbery. Robberies can be aborted by detecting different kinds of weapons. let's move on to the datasets used in our scenario. The images utilized in the weapon detector should be cleaned, preprocessed, and appropriately annotated in order to achieve high precision. The process of collecting images and labeling them was delicate and tough as well. Images for robberies, burglaries, and criminal trespassing were collected from the internet, extracted from video, Github repositories, and movies.

In our study, we focus on four classes only, pistol, knife, rifle, and robber masks. We include revolvers and handguns in pistol class. Also, we include shotguns in rifle class. However, there are many objects that are most likely to be confused with a pistol, such as wallets, cell phones, selfie sticks, money, etc. so, it would be a great idea to label them as a non-weapon class, hence minimize false positives and false negatives, therefore boosting the overall accuracy. We have made two datasets, which are explained below.

A. Dataset1

This is the first dataset, we have four classes here, pistol, knife, rifle, and robber masks. We have 1160 images in total, the majority of images belong to the pistol class because almost 95% of weapons used in robberies are either pistols or revolvers (Bhatti et al. 2021). Dataset was separated into train and test with split size shown in table 10. Figure 21 displays the distribution of the four different classes in the dataset1 used in the experiments.

Table 10: Data Distribution.

No.	Category	Total Data	Training Data	Test Data	Split Size
1	Dataset1	1160	920	240	79%
2	Dataset2	5000	4250	750	85%

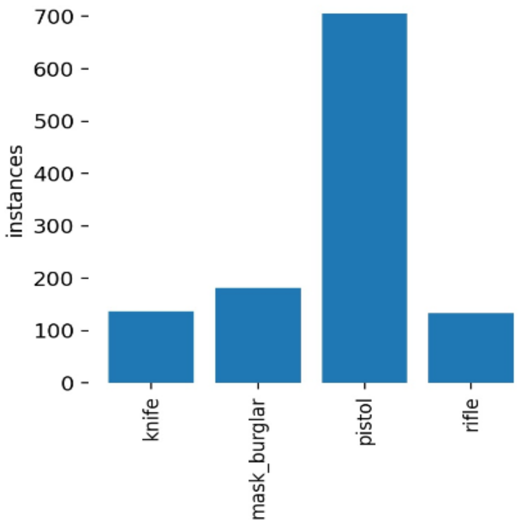


Figure 21: Class Distribution in dataset1.

B. Dataset2

This dataset is used to build a weapon detection binary classifier so two classes were made, knife and pistol. We have 5000 images in total, with 4250 images train set and 750 in the test set. Figure 22 displays the distribution of the two different classes in dataset2 used in the experiments.

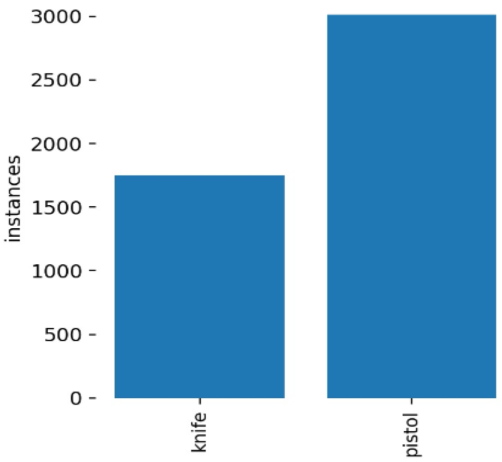


Figure 22: Class Distribution in dataset2.

C. Preprocessing

While building a machine learning model, data preparation and filtering stages need a large amount of processing time. Data cleansing, standardization, processing, filtering, scaling, and feature selection for traditional machine learning models are all part of the pre-processing process (*Data Preprocessing for Supervised Leaning* n.d.). Pre-processing stages carried out on the acquired images led to the creation of the final training dataset.

Preprocessing is a crucial step for improving object detection precision and accuracy as well. The first step is resizing all images into one size, then applying the mean normalization to achieve faster convergence. The next step is assigning rectangular boxes to target objects, which is called annotation or labeling. In the annotation process, a bounding box is drawn around each target object in images. The width, height, x, and y coordinate of the bounding box are saved as xml format for the SSD model or text document for YOLO models.

D. Images quality challenges

Generally, low-quality images are one of the core problems of deep learning models. Almost all CCTV footages in our dataset seem to be grainy, blurry, monochromatic, and poor quality. The main reason CCTV footages are of low quality is that the amount of data require to record in higher quality would be huge, hence more data storage is needed. In addition to that, outdated cameras, weak internet connection, and lens condition are factors responsible for low-quality images.

Beside to poor-quality images, many challenges appear regarding the quality of the dataset. One of these challenges is Background Clutter where the target object may blend into the environment and make them difficult to identify. Also, occlusion issue when a small portion of the target object is visible. Furthermore, viewpoint variation challenge where a target object can be rotated in any direction with respect to the observer or camera. Lastly, scale variation when an object appears in images in different sizes.

4.2 Models

In our research, we are going to build weapon detection models by using transfer learning and fine-tuning pre-trained models. The models namely are SSD Inception V2, SSD MobileNet V2, SSD ResNet50 V1, YOLOv4, and YOLOv5. All models are trained on Colab notebook using Tensorflow and GPU accelerated. Starting with SSD Inception V2 models, it uses VGG-16 as a feature extractor, the batch size is 32 and the input layer receives images of resolution 300x300. The num_step parameter is 10,000 defines how many training steps it will run. The model uses RMSprop as an optimizer, it is designed for deep neural networks.

The Second model is SSD MobileNet V2, the input layer takes images with size 320 x 320. The num_step parameter is 10,000 and the batch size is also 32. In contrast with SSD ResNet 50, the input layer receives the images with a resolution of 640 x 640, the batch size is 8, which is small due to the high resolution of the input layer. Yolov4 and Yolov5 are popular single-stage object detectors. The YOLOv4 model is made up of a feature extraction process to identify features like shapes, edges, or motion in the image as well as detection heads for object localization in images. All pre-trained networks are trained using the COCO dataset, which has 80 different object classes (Lin et al. 2014).

4.3 Metrics

This section explains the metrics used to assess the performance of object detection models. AP, IOU, and mAP are object detection metrics used to evaluate how good object detection models are. Before diving deep into metrics for object detection, some definitions need to be cleared. Starting with IoU (Intersection over Union) which represents the shared area between the predicted bounding box and the ground-truth bounding box dividing by the area of their union. IoU value ranges between 0 and 1 where 0 means there is no overlapping between two boxes, and 1 indicates perfect overlap.

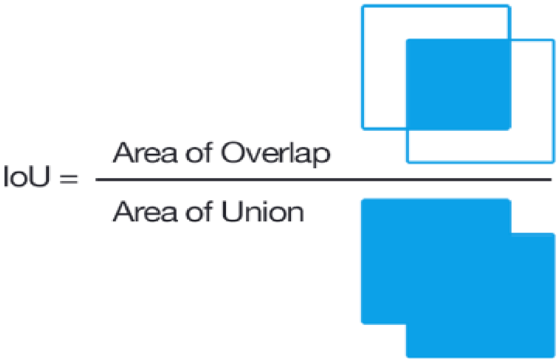
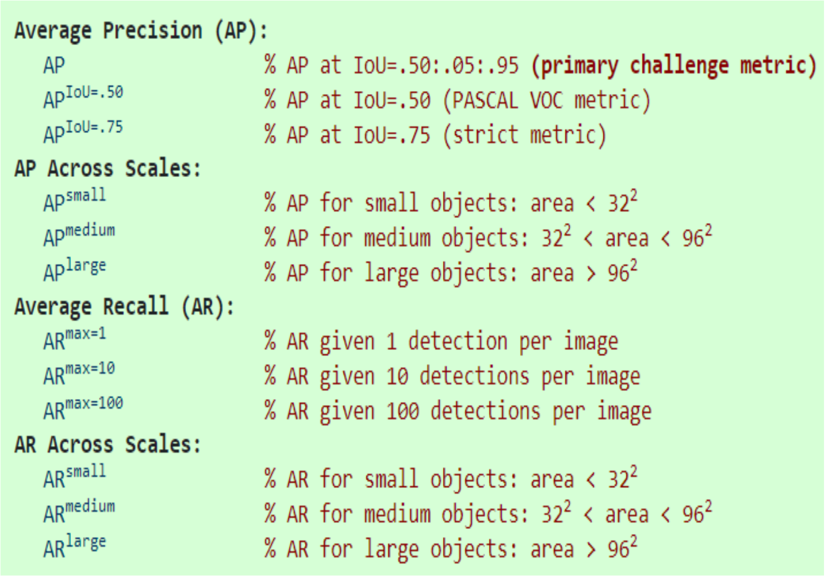


Figure 23: Intersection over Union

The average precision (AP) was calculated in the Pascal VOC challenge by interpolating precision at 11 different points, and an intersection over union (IoU) of 0.5 is only considered. Meanwhile, Google Open Images Challenge employs mean average precision metric to evaluate the object detection model by predicting tight bounding boxes around target objects of 500 classes (*Open Images evaluation protocols* n.d.). In contrast with the COCO challenge (Lin et al. 2014), it utilized every point interpolation and calculated mean average precision across several thresholds, with IoU ranging from 0.5 to 0.95. The following table explains the 12 metrics used for assessing the performance of object detectors on the COCO dataset.

Table 11: Metrics for COCO dataset (COCO - Common Objects in Context n.d.).



Average Precision (AP):	
AP	% AP at IoU=.50:.05:.95 **(primary challenge metric)**
$AP^{IoU=.50}$	% AP at IoU=.50 (PASCAL VOC metric)
$AP^{IoU=.75}$	% AP at IoU=.75 (strict metric)
AP Across Scales:	
AP^{small}	% AP for small objects: area < 32^2
AP^{medium}	% AP for medium objects: 32^2 < area < 96^2
AP^{large}	% AP for large objects: area > 96^2
Average Recall (AR):	
$AR^{max=1}$	% AR given 1 detection per image
$AR^{max=10}$	% AR given 10 detections per image
$AR^{max=100}$	% AR given 100 detections per image
AR Across Scales:	
AR^{small}	% AR for small objects: area < 32^2
AR^{medium}	% AR for medium objects: 32^2 < area < 96^2
AR^{large}	% AR for large objects: area > 96^2

AP is the average precision across 10 intersections over Union (IoU) for all classes. There is no distinction between AP and mAP, and likewise AR and mAR in metrics used by COCO. As shown in table 11, AP and AR are also calculated across the different sizes of target objects. The area of target objects is calculated as the number of pixels in the segmentation image.

Moreover, a confidence score is a probability that a predicted box contains an object. More weapon items will be missed by the detector as the confidence score threshold increases, which means more false negatives and thus low recall and high precision. Whereas the detector will get more false positives if the confidence score is low, hence low precision and high recall.

4.4 losses

A mathematical equation known as a loss function is utilized to generate loss values through the training phase. A model's performance is evaluated during training based on the loss (L) that it generates for each batch of samples. The loss function monitors how far the predicted value is from the ground truth value. High loss means predicted values are far away from ground truth values, and vice versa.

The YOLO loss function is broken into three parts:

- The classification loss: when the model detects a weapon item, the classification loss is calculated by squaring the class conditional probabilities in every cell. Here is the formula for classification loss where ‖z_i^{obj} denotes if the object exists in cell *i* (Redmon et al. 2016).

$$\sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

- The localization loss: it is responsible for errors between the predicted boundary box and ground truth box in terms of locations and sizes. It describes how well the predicted bounding box covers an object. The formula for localization loss (Redmon et al. 2016) is shown below where ‖ij^{obj} represents that the *j*th bounding box predictor in cell *i* is responsible for the prediction.

$$\lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\textbf{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

- The confidence loss: is a categorical cross-entropy loss for categorizing the detected items. The objective of this part is to make sure that the proper label is given to each detected object. The following formula describes the confidence loss in the YOLO algorithm (Redmon et al. 2016).

$$\sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \quad + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

Where $\mathbb{1}_{ij}^{noobj}$ is the complement to $\mathbb{1}_{ij}^{obj}$ when the target object is not detected in the box. λ_{noobj} denotes a factor used to weigh down the loss when detecting background.

5 CHAPTER FIVE: RESULTS

The Results from training and evaluation of weapon detection models are presented in this chapter. Here, we trained in a total of 10 weapon detection models, five for each dataset. The results are presented in tables and various graphical representations as well, for visual comparison purposes and are easy to understand.

5.1 Training Results

The training was performed with a different number of training steps for each model, SSD Inception, SSD MobilNet, and SSD ResNet models were trained for 10,000 training with batch size range from 8 to 32 depending on the dimension of the input images. Meanwhile, YOLOv4 has trained for 6000 – 8000 training steps only to prevent overfitting problems. YOLOv5 was trained on dataset1 for only 270 epochs and on dataset2 for 832 epochs since no improvement was observed in the last 100 epochs. Training results are shown below in figure 24 & 25.

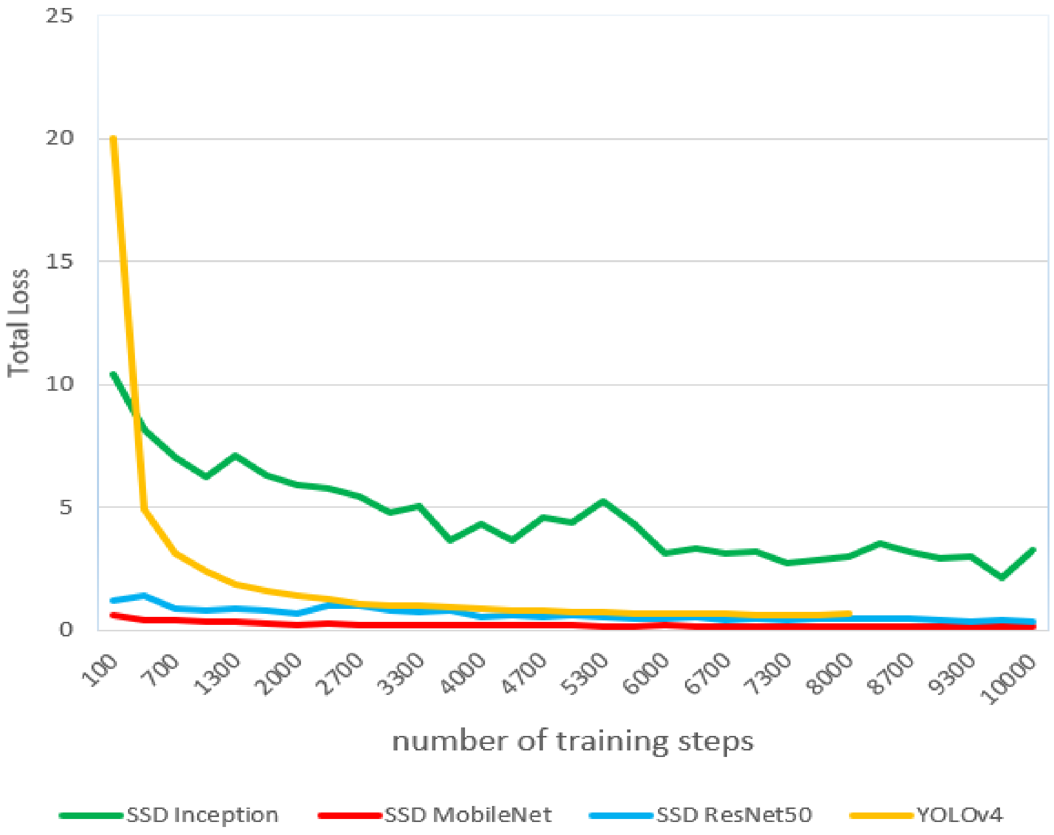


Figure 24: Training results: total loss of the object detection models on dataset1.

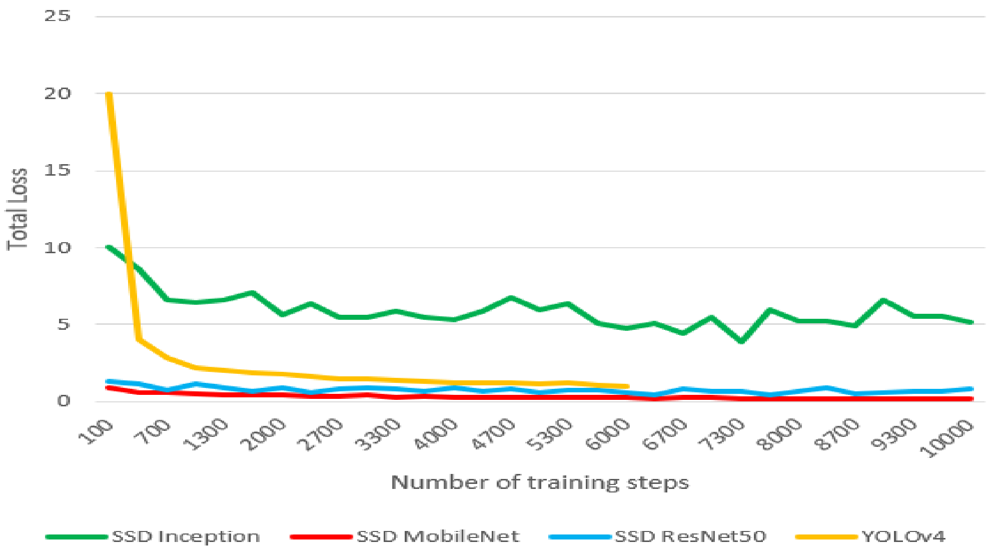
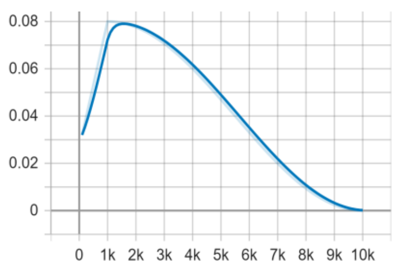


Figure 25: Training results: total loss of the object detection models on dataset2.

The behavior of total loss drops of models for both dataset1 and dataset2 is quite similar. As the loss starts to decrease through training, the model accuracy starts to increase until it stagnated. After 10,000 training steps, the value of the total loss of the SSD MobileNet model on dataset1 is around 0.16, and the learning rate vanished to zero. However, the model with the highest total loss is SSD Inception about 3.26 on the same dataset.

While configuring a neural network, the most important hyperparameter is the learning rate which determines how much to change the model based on estimated error every time the model weights are updated. Figure 26-a and 26-b show the learning rate behavior for the SSD MobileNet model on dataset1 and the SSD ResNet50 model on dataset2 respectively. The comparison of learning rate values of SSD models is presented in figure 27.



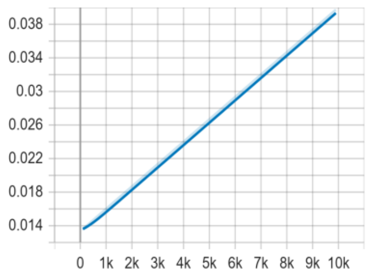


Figure 26-a: Behavior of learning rate for SSD MobileNet model.

Figure 26-b: Behavior of learning rate for SSD ResNet50 model.

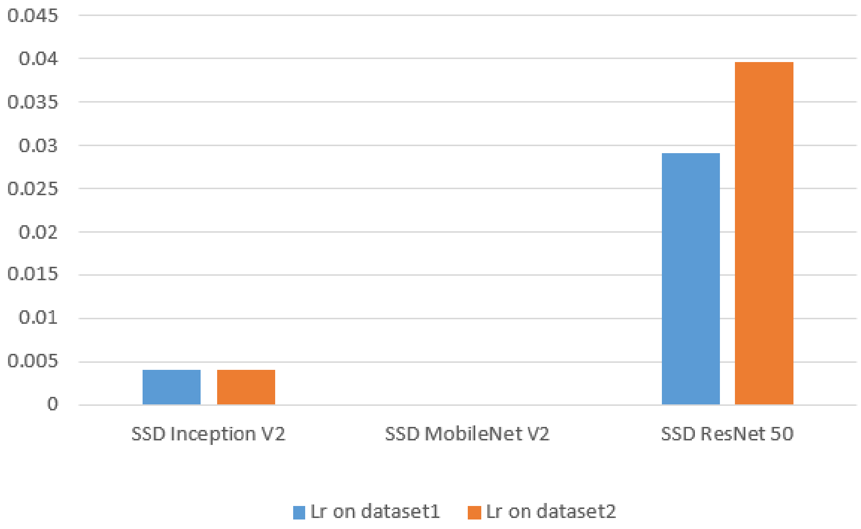


Figure 27: Learning rate values for various SSD models.

The total loss is combined of classification loss, localization loss, and regularization loss in SSD Models. However, the loss function for YOLO models composes of classification loss, localization loss, and confidence loss (objectness loss). Figures 28 and 29 show the loss function and mAP chart for the YOLOv4 model throughout the training phase on dataset1 and dataset2 respectively.

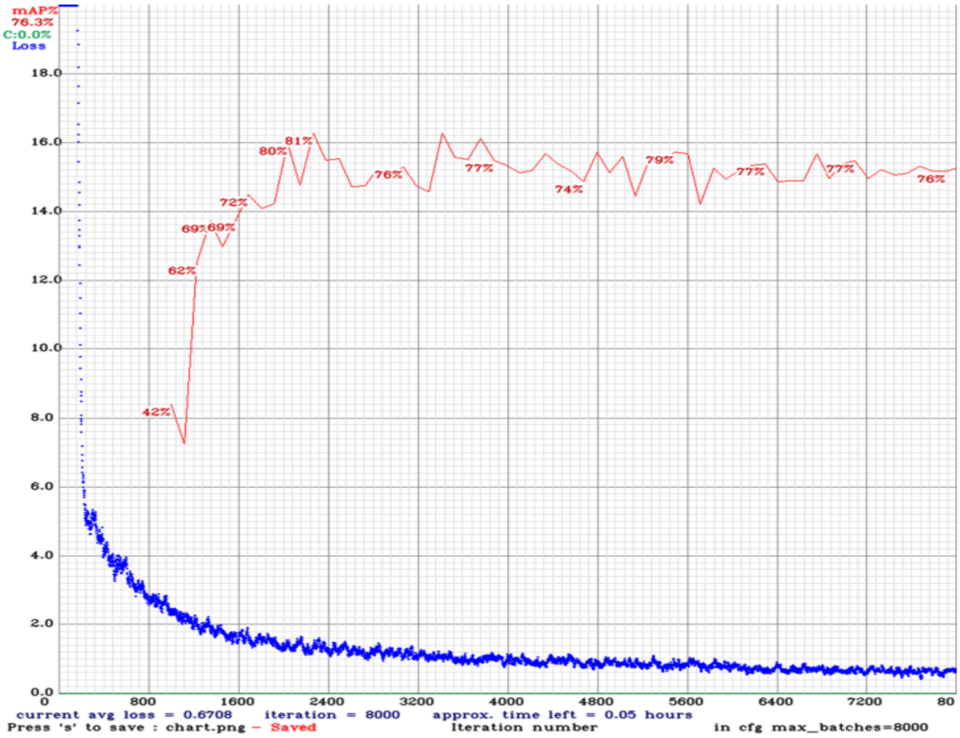


Figure 28: Chart of loss and mAP from YOLOv4 model training on dataset1.

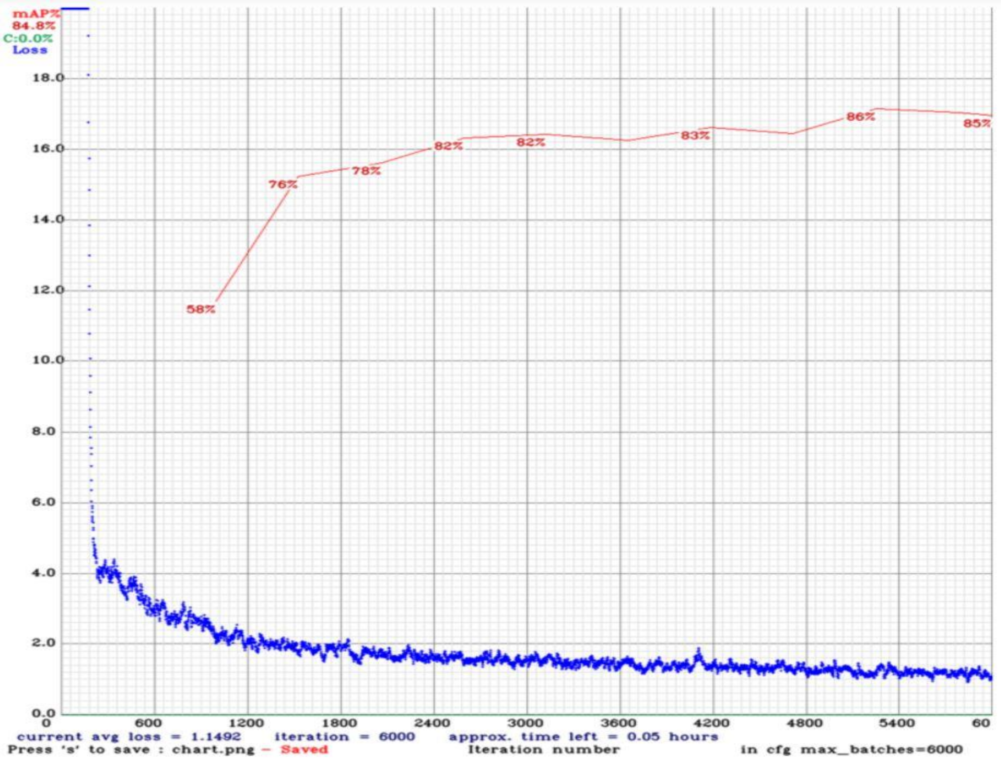


Figure 29: Chart of loss and mAP from YOLOv4 model training on dataset2.

The blue curve represents the training loss, and it decreases while the number of iterations increases. The red line is the mean average precision when the intersection-over-union threshold is 0.5.

5.2 Evaluation Results

To assess the accuracy of the new weapon detection models, test data were created from images of various weapons in different scenarios. In this section, we present the evaluation results for all experiments in tables and graphical representations.

5.2.1 overview

All object detection models were evaluated on mAP@0.5(Pascal VOC), mAP@0.5:0.95(COCO), and mAP@0.75(strict metric) (Xue et al. 2021). Tables 12 & 13 show evaluation results on dataset1 and dataset2 respectively. COCO standard metric (AP@IoU = 0.5:0.95) represented as average precision for IoU thresholds from 0.5 to 0.95 with a step size of 0.05 (Ren et al. 2017).

A few more evaluation metrics have been used in this paper, like AP for evaluating detection for different weapon sizes inside the image determining whether the model is good for only large weapons, small weapons, or both. SSD models are evaluated on three different sizes, small, medium, and large where small objects have an area (h*w) less than 322 pixel scale, medium objects have an area more than 322 and less than 962, and large objects have an area more than 962. However, the average precision for the IoU threshold of 0.5 (AP@IoU = 0.5) is used to evaluate weapon detection models based on YOLO algorithms. The average recall metric is also used in evaluation given a fixed number of detections per image.

According to table 12, the SSD AP of the MobileNet model surpasses the other two SSD models in the detection of small weapons for dataset1 and dataset2 as shown in table 13. For the average recall (AR) metric, the SSD ResNet50 model performs better than SSD MobileNet and SSD Inception models in the detection of small weapons on dataset1, while the SSD MobileNet model performs better on dataset2. However, for the detection of medium and large objects, SSD MobileNet achieves better than other SSD models in both dataset1 and dataset2. The primary metric used in comparing weapon detection models is AP@IoU = 0.5, by comparing the models, it is obvious YOLOv4 surpasses other models with AP 0.795 on dataset1, while the least performance model is SSD Inception with AP around 0.375. In contrast with dataset2, YOLOv5s is the best model in weapon detection with average precision of 0.82, and the lowest performance model is SSD Inception with AP 0.279.

Table 12: Evaluation results on dataset1.

Metric/model	area	Max Detection	SSD Inception V2	SSD MobileNet	SSD ResNet 50	YOLOv4	YOLOv5
AP@IoU = 0.5:0.95	All	100	0.215	0.469	0.364	-	-
AP@IoU = 0.5	All	100	0.375	0.717	0.577	0.795	0.724
AP@IoU = 0.75	All	100	0.206	0.529	0.392	-	-
AP@IoU = 0.5:0.95	Small	100	0.001	0.247	0.201	-	-
AP@IoU = 0.5:0.95	Medium	100	0.087	0.311	0.243	-	-
AP@IoU = 0.5:0.95	Large	100	0.254	0.511	0.397	-	-
AR@IoU = 0.5:0.95	All	1	0.261	0.488	0.399	-	-
AR@IoU = 0.5:0.95	All	10	0.329	0.574	0.523	-	-
AR@IoU = 0.5:0.95	All	100	0.361	0.581	0.548	-	-

AR@IoU = 0.5:0.95	Small	100	0.033	0.256	0.344	-	-
AR@IoU = 0.5:0.95	Medium	100	0.255	0.367	0.414	-	-
AR@IoU = 0.5:0.95	Large	100	0.400	0.636	0.584	-	-

Table 13: Evaluation results on dataset2.

Metric/model	area	Max Detection	SSD Inception V2	SSD MobileNet	SSD ResNet 50	YOLOv4	YOLOv5
AP@IoU = 0.5:0.95	All	100	0.177	0.411	0.247	-	-
AP@IoU = 0.5	All	100	0.279	0.674	0.406	0.771	0.82
AP@IoU = 0.75	All	100	0.193	0.384	0.251	-	-
AP@IoU = 0.5:0.95	Small	100	0	0.015	0.001	-	-
AP@IoU = 0.5:0.95	Medium	100	0.013	0.148	0.060	-	-
AP@IoU = 0.5:0.95	Large	100	0.213	0.479	0.292	-	-
AR@IoU = 0.5:0.95	All	1	0.195	0.436	0.285	-	-
AR@IoU = 0.5:0.95	All	10	0.237	0.499	0.389	-	-
AR@IoU = 0.5:0.95	All	100	0.268	0.534	0.428	-	-
AR@IoU = 0.5:0.95	Small	100	0.000	0.114	0.050	-	-
AR@IoU = 0.5:0.95	Medium	100	0.055	0.351	0.233	-	-
AR@IoU = 0.5:0.95	Large	100	0.32	0.584	0.479	-	-

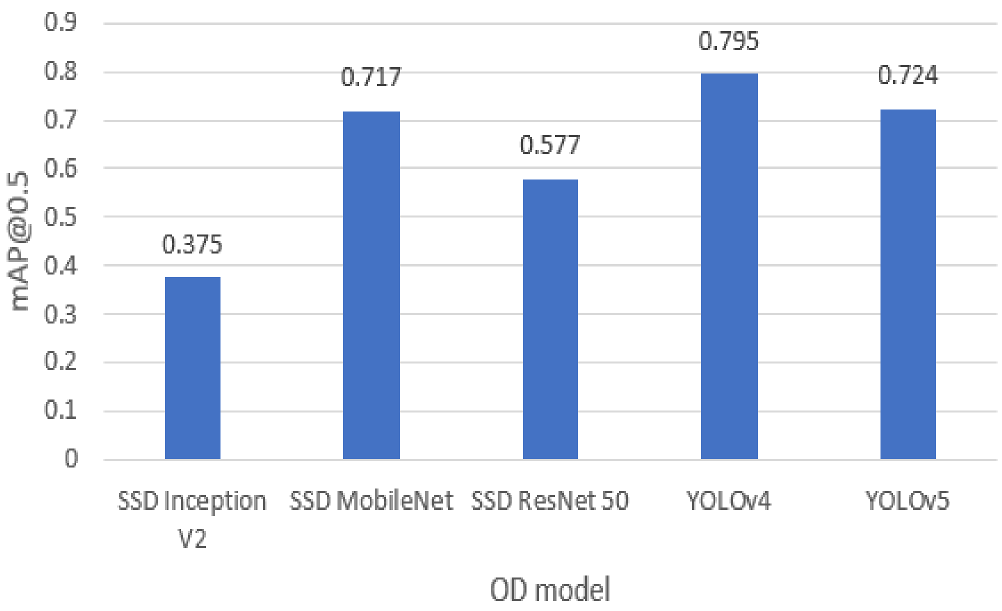


Figure 30: Weapon detection models comparison on dataset1 using mAP@0.5 metric.

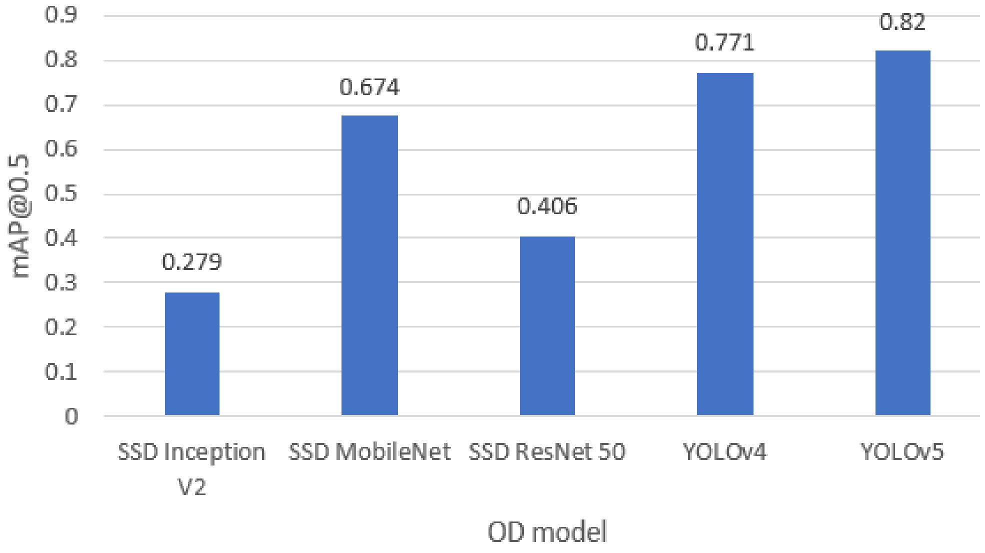


Figure 31: Weapon detection models comparison on dataset2 using mAP@0.5 metric.

As mentioned previously, there are different model sizes for YOLOv5. In this paper, we trained both datasets on YOLOv5s which “s” stands for small size model. Meanwhile, the x-large YOLOv5 model (YOLOv5x) achieves higher performance than YOLOv5s with mAP around 0.762 on dataset1.

5.2.2 Confusion Matrix

The confusion matrix comprises of four main numbers used to define the performance of the classification problems. These four numbers are true positive (TP), true negative (TN), false positive (FP), and false negative (FN). The confusion matrix displays the ways in which your weapon detection model is confused when it makes a prediction. Since YOLO models outperform SSD models, we present the confusion matrix of weapon detection for the YOLOv5 model on dataset1 and dataset2 as shown in figure 32 and figure 33 respectively.

Dataset1 contains 240 test images with four classes namely knife, mask burglar, pistol, and rifle. And dataset2 contains 750 test images with two classes, pistol, and knife. According to the confusion matrix in figure 32, the most successful weapon detection rates were obtained for the mask burglar class at 0.84, while the least successful rates were for the knife class at 0.4. Similarly, the most successful weapon detection rates in figure 33 were obtained for the pistol class at 0.84, and the least successful rates were for the knife class at 0.47.

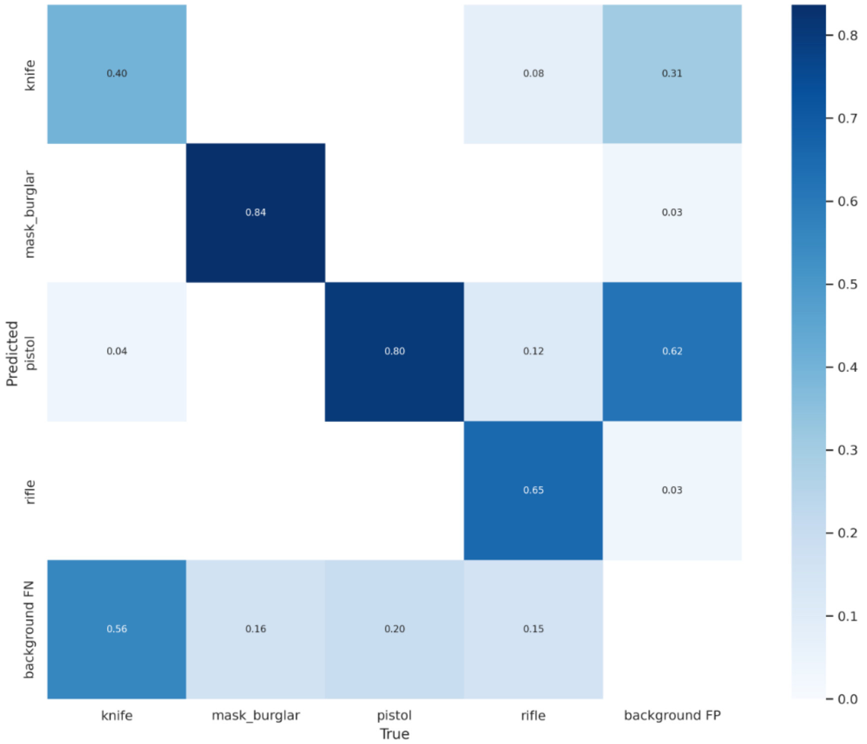


Figure 32: Confusion matrix of weapon detection for YOLOv5 on dataset1.

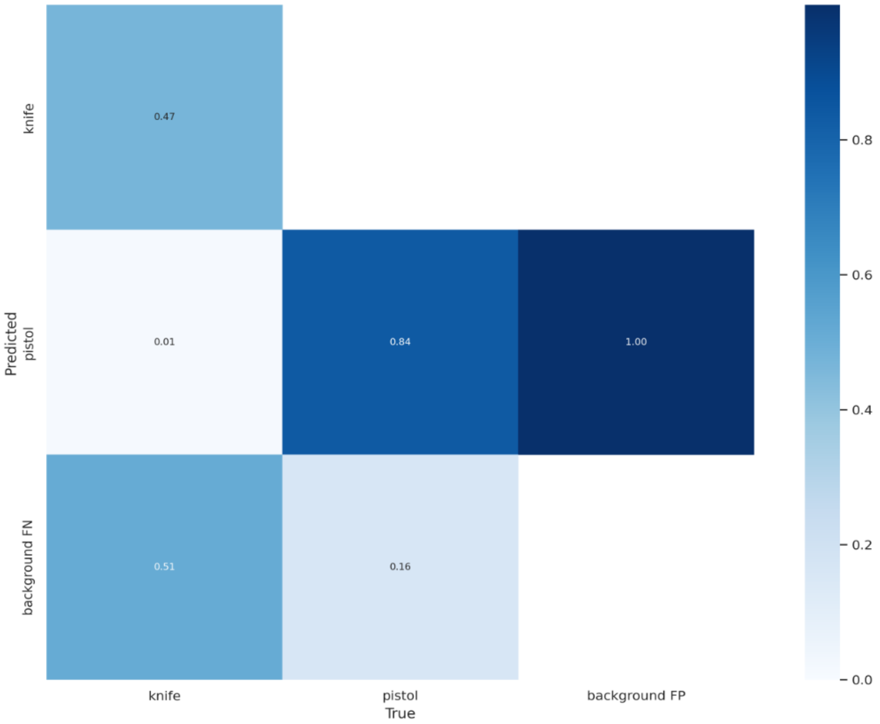


Figure 33: Confusion matrix of weapon detection for YOLOv5 on dataset2.

It can be seen that around half of the knife weapons in both datasets are confused with the background. Also, around a fifth of pistol images in both datasets were misclassified with the background. In dataset1, 0.12 of rifle images were misclassified with pistol class, while a small portion of rifle images (0.08) was misclassified with knife class, and 0.04 of knife images were misclassified with pistol class.

In the confusion matrix, background FP indicates the model may think detection is a weapon when it is not, and background FN indicates the model misses the detection of real weapons. By looking at previous confusion matrices, background FP refers to the model incorrectly predicting the weapons, and background FN refers to the model incorrectly predicting the negative class. We can summarize our weapon detection model in the bank using a 2x2 confusion matrix that depicts all four possible outcomes.

True Positive: Reality: A theft threatened. CCTV system said: Robber. Outcome: CCTV is a hero.	**False Positive:** Reality: No theft threatened. CCTV system said: Robber. Outcome: police and customers are angry at bank for false alarm.

False Negative: Reality: A theft threatened. CCTV system said: No Robber. Outcome: robbers stole the bank.	**True Negative:** Reality: No theft threatened. CCTV system said: No Robber Outcome: everyone is fine

Figure 34: Understanding Confusion Matrix.

5.2.3 Recall x Precision

Precision and recall are widely used as model evaluation metrics. Precision implies how accurate the object detection model is for detecting weapons. Therefore, it evaluates the accuracy of predicted positive samples (Bruce & Bruce 2017). The precision is the ratio between the number of weapon items correctly detected to the total number of items that are either correctly or incorrectly detected as a weapon. Also, it is referred to as the Positive Predictive Value.

Meanwhile, recall is another metric used to evaluate the strength of the weapon detection model to detect positive outcomes (weapons). The recall is the ratio between a number of weapon items correctly detected to the total number of weapon items. Also, it is referred to as the sensitivity of a model. Both metrics give precious information, the main goal is to increase the recall without decreasing the precision (Chawla 2009).

The formulas for evaluating precision and recall are given below:

- Precision = $\frac{TP}{TP+FP}$ ....1 (Fawcett 2006)

- Recall = $\frac{TP}{TP+FN}$ ....2 (Fawcett 2006)

The trade-off between precision and recall is depicted by the a precision-recall curve for different classes. High recall and high precision are both denoted by a high area under the curve, where high precision and high recall are associated with a low false positive and false negative rate respectively. Figure 35 & 36 present precision recall graph for the YOLOv5s model on dataset1 and dataset2 respectively.

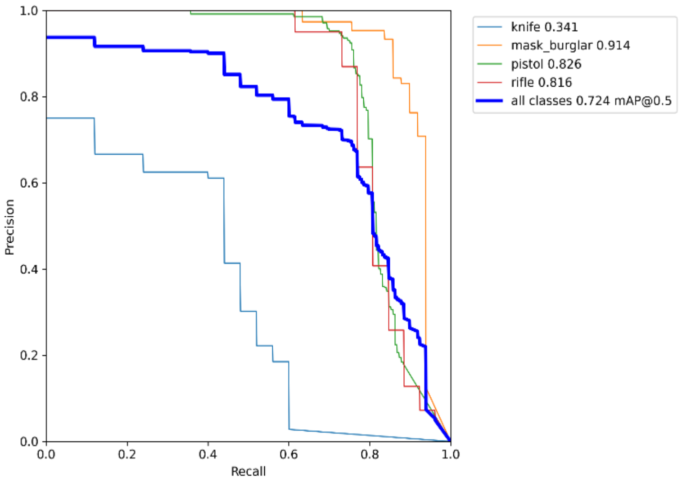


Figure 35: Precision recall graph for YOLOv5s on dataset1.

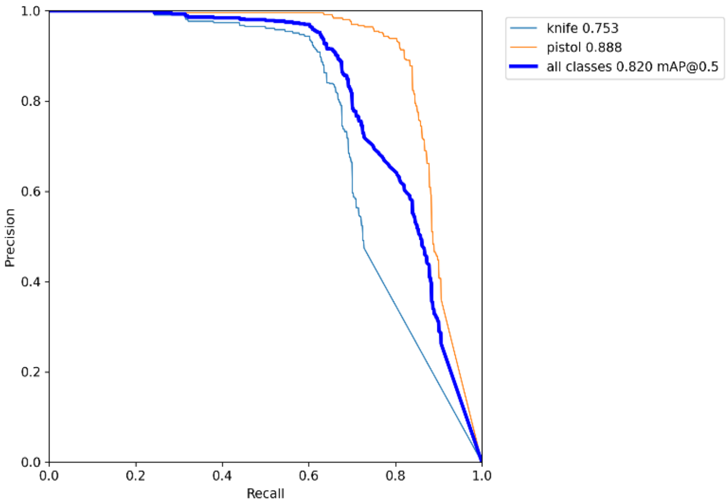


Figure 36: Precision recall graph for YOLOv5s on dataset2.

According to figure 35, it can be seen the model is returning accurate results for detecting burglar mask items, with mAP 0.914. The model also performs quite well on detecting pistol items and rifles as well, with mAP around 0.82. Meanwhile, the model is poorly detecting knife items with mAP 0.341 in real-time. Moving on to figure 36, it is obvious that the performance model of detecting pistol items is higher than detecting knife class due to the quality of knife images in the dataset.

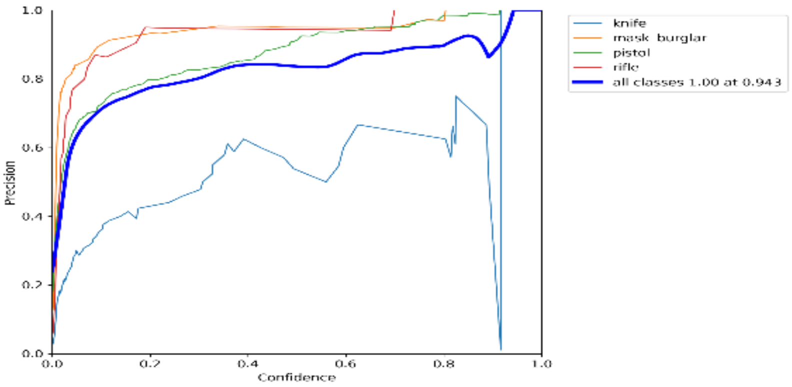
5.2.4 F1 x Confidence

In some applications, we desire to maximize either precision or recall at the expense of other metrics. For instance, in a weapon detection scenario, we would probably want a recall near 1.0, which means we want to catch all robbers and thieves who carry weapons.

However, we may accept low precision if the cost of consequences of false alarms is not high. In situations where we want to find the best value of precision and recall, we can combine them using the F1 score metric (Luo et al. 2020). The F1 score is a harmonic mean of the precision and recall, where the value of the F1 score is between 0 and 1. The formula for the F1 score is:

$$\text{F1} = 2 \cdot \frac{precision\ x\ recall}{precision\ + recall} \text{ ………… 3 (Fawcett 2006)}$$

The confidence score shows how probable the bounding box contains the target object and how confident the weapon detection model is about it. The confidence score will be zero when no weapon items are detected in the bounding box. For a reminder, the IoU value can be obtained by dividing the shared area between the predicted bounding box and the ground-truth bounding box by the area of their union. When ground-truth and predicted bounding boxes have the same area and location, the match is perfect. A bounding box tends to be more restrictive for a higher IoU thresholds, thus higher confidence score. In contrast, a bounding box is more flexible for lower IoU threshold, in other words, minor overlap is considered correct detection. Figures 37 and 38 show precision-confidence curves and recall – confidence curves for the YOLOv5 model on dataset1 and dataset2 respectively.



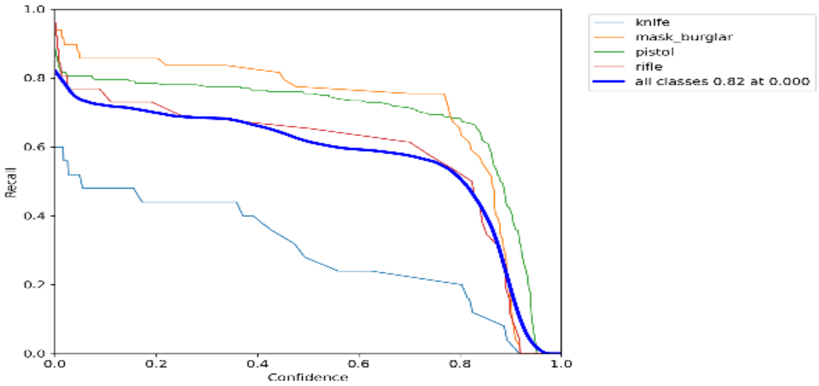


Figure 37: Trade-off between precision and recall by varying confidence for YOLOv5 model on dataset1.

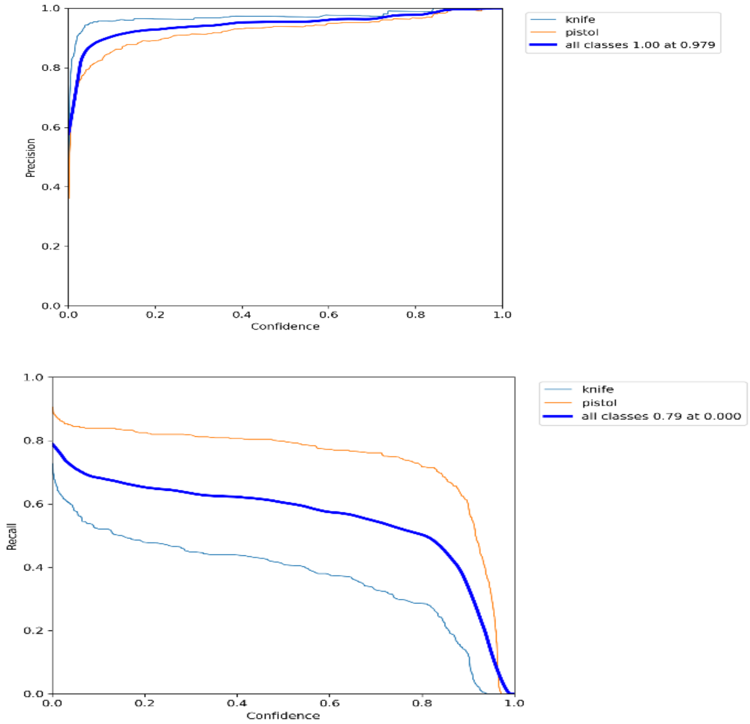


Figure 38: Trade-off between precision and recall by varying confidence for YOLOv5 model on dataset2.

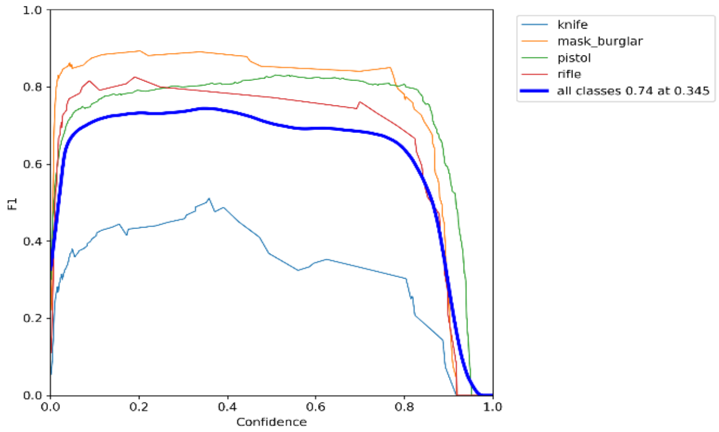


Figure 39: F1-Confidence curve presents best F1 score of 0.74 with a confidence threshold of 0.345 on dataset1.

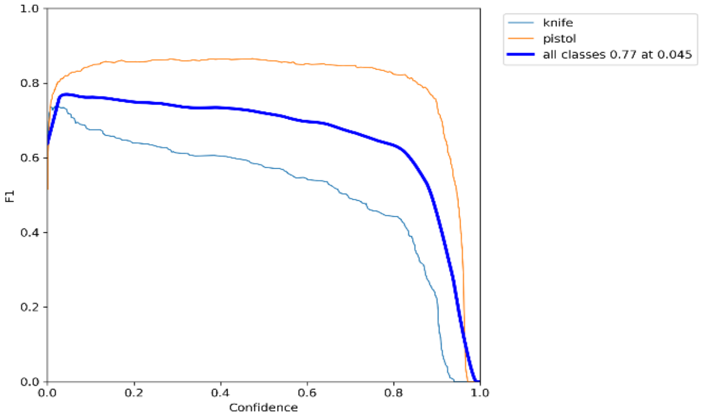


Figure 40: F1-Confidence curve presents best F1 score of 0.77 with a confidence threshold of 0.045 on dataset2.

The outcome of an object detector is represented by a bounding box, class, and confidence score. As observed in figures 37 and 38, the precision is increased when the confidence score goes up, and simultaneously recall decreases. The detection is considered valid (positive) when its confidence is higher than a confidence threshold. Otherwise, it is negative detection. The number of TP and FP decreased when the confidence threshold increased. Conversely, the number of false negatives increased as the confidence threshold increased, thus decreasing the recall metric.

According to the F1 confidence curve shown in figure 39, the confidence value that optimizes the precision and recall is 0.345 for the YOLOv5 model on dataset1, and the best F1 score for all classes is 0.74. Similarly, to figure 40, the best confidence value that optimizes precision and recall is 0.045 for the same model on dataset2, and the best F1 score is 0.77.

5.3 Detection Results

The proposed model fails to detect weapon items in images as shown in the following figures.

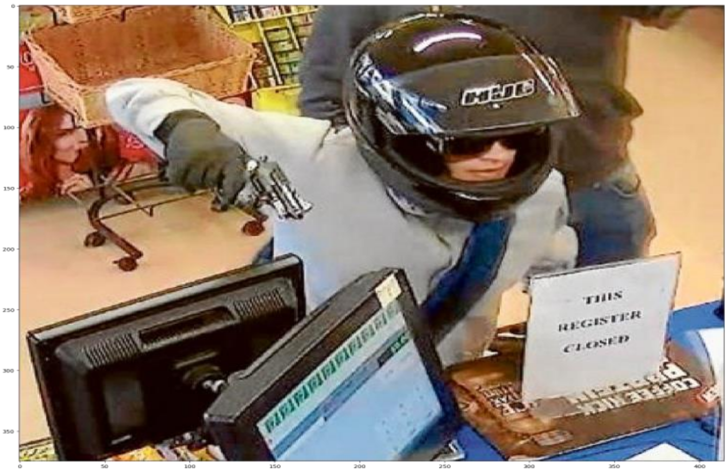


Figure 41: Object detection model could not find the pistol because of low resolution of image.

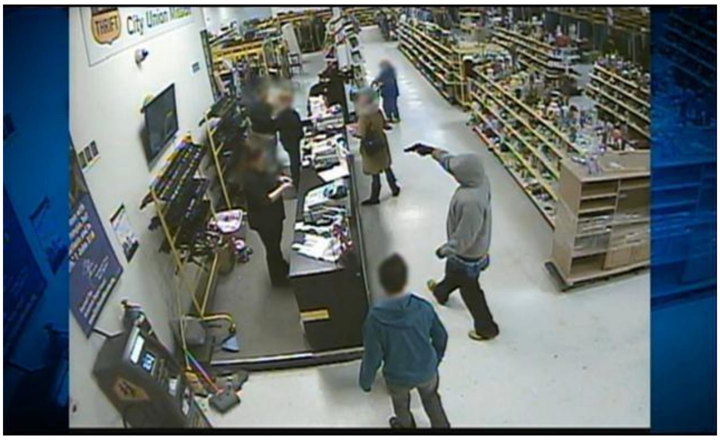


Figure 42: Object detection model could not find the pistol because of small size of pistol.

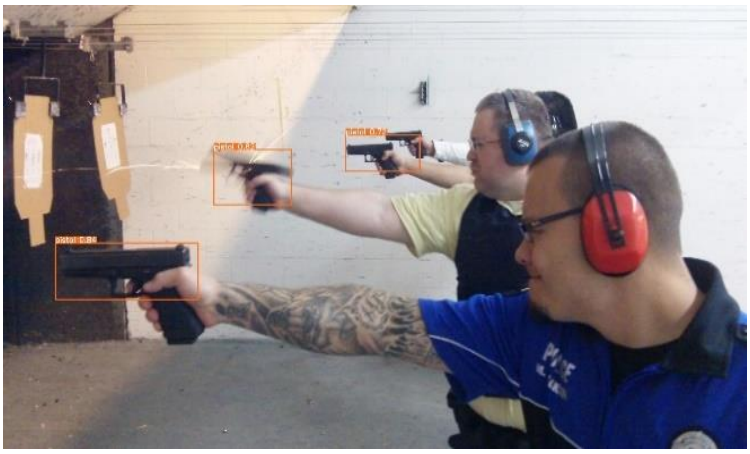


Figure 43: Object detection model could not find all pistols in the image.

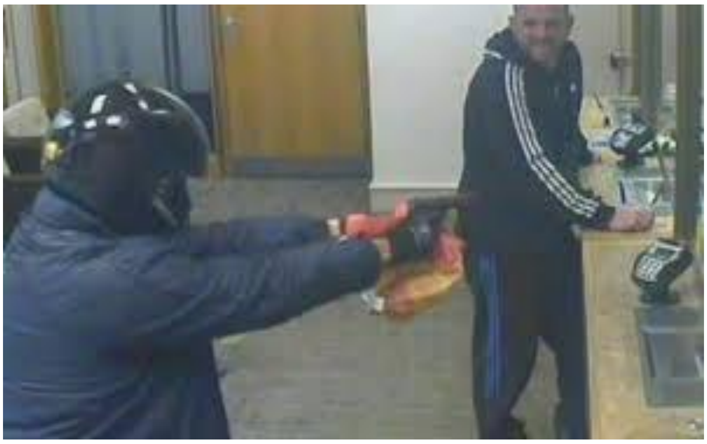


Figure 44:Object detection model could not find the pistol because of low resolution of image.

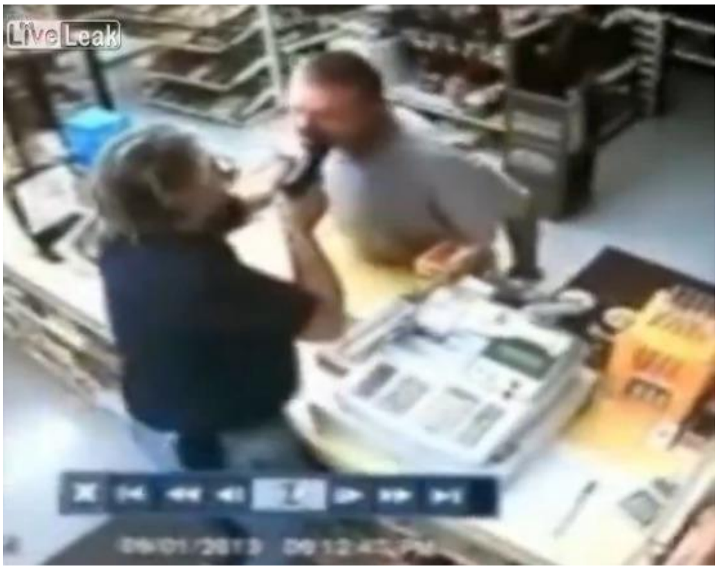


Figure 45: Object detection model could not find the pistol because of low resolution of image.

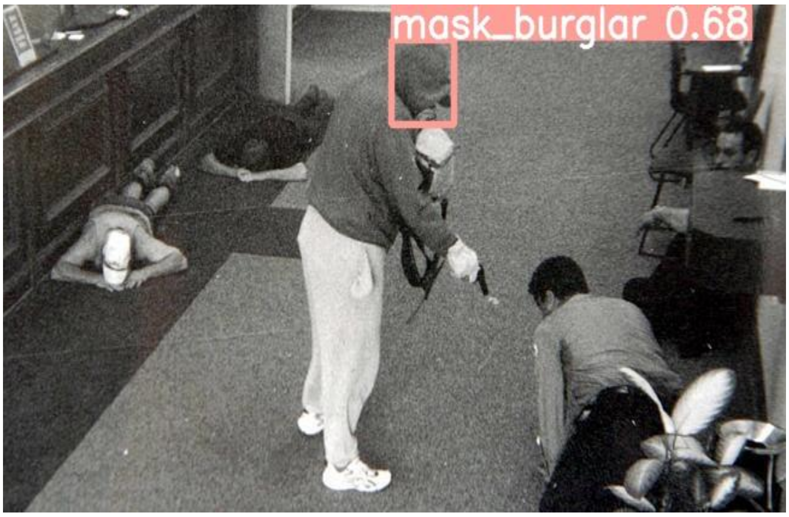


Figure 46: Object detection model could not find the pistol because of small size of pistol.

The following figure shows weapon detection results of footages from monitoring cameras in real life scenarios.

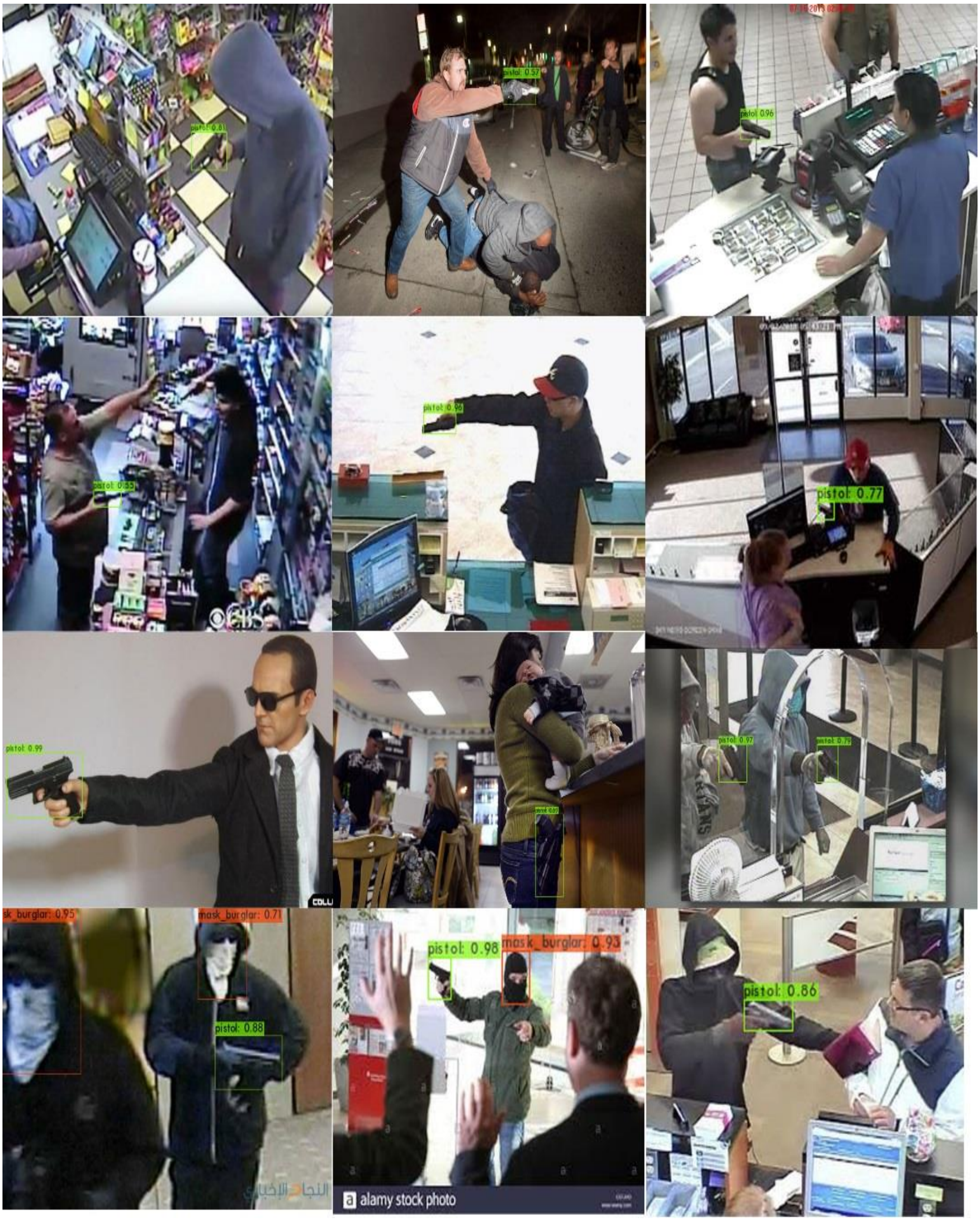


Figure 47: Detection Results of weapon detection model on sample images were taken from CCTV camera, each image shows a bounding box, class and confidence score beside a target object.

5.4 Discussion

Weapon detection models were trained using several configurations based on model architecture. Many factors have an impact on model performance and results such as different image resolutions, batch size, training steps, rescaling, and data augmentation options. The ideal parameters for each model should be investigated using a hyperparameter optimization pipeline to balance out or improve the results of weapon detectors.

Moving on to the evaluation section, weapon detectors are evaluated based on the main purpose of the model. If we want to use the proposed model as a weapon recognizer without taking boxes overlapping into account, we can merely minimize the IoU threshold to close to zero while keeping the false negative number low and the false positive number high as much as possible. In other words, the recall score will be high and precision low. However, in the case of building a weapon detector instead of a weapon recognizer, we take the location of a predicted box into account. It would be recommended to evaluate the model using the COCO evaluation metric (AP@IoU = 0.5:0.95). This metric describes how the model works in the area of overlapping that we are most interested in.

Different models and different classes all need certain confidence thresholds to boost the performance and thus maximize it. Figures 36 & 37 offer a tradeoff between confidence, precision, and recall finding the best confidence threshold for each class and model. It is kind of hard to select a model based on a single metric, thus many metrics were used in the evaluation process.

6 CHAPTER SIX. CONCLUSION AND FUTURE WORK

6.1 Choice of detector

As demonstrated in section two chapter four, different detectors perform well on different datasets, proving that one weapon detector does not fit all datasets. To wrap up the results and come to a conclusion, we conclude that the most suitable weapon detector for the task would be YOLOv4 for dataset1 and YOLOv5s for dataset2. YOLO models perform better than SSD models in term of detection speed and average precision. YOLO models are better options when the weapon size is small.

6.2 Future Work

In this research, the most advanced object detection methods were not employed such as DINO (Zhang et al. 2022), and YOLOv7 (Wang, Bochkovskiy & Liao 2022) due to availability. They would be employed in the future to find if they considerably boost performance. Another point that should be taken into consideration is image quality, most images used in this research are low resolution and some images are blurry. In addition to that, many images contain target weapons with white backgrounds, while few images were taken from surveillance cameras of market robberies, bank robberies, and home burglaries. For future work, we should focus on extracting images from bank surveillance cameras or any other related places.

In this research paper, we focused on indoor surveillance cameras in banks, supermarkets, malls, …, etc. However, it would be a great idea to use weapon detectors as a precautionary system to prevent robberies by implying the model in outdoor surveillance cameras. In addition to that, we have to focus in the future on decreasing false positive and false negative numbers by increasing the number of images or classes. Adding new classes to the dataset such as money, wallets, cellphones, bills, purses, and any other items handle in the same way as the weapons.

6.3 Conclusion

This work presented a study of several weapon detection models (SSD and YOLO) applied to the video surveillance system. The two main objectives of the study were to compare the effectiveness of five different models (SSD Inception, SSD MobileNet, SSD ResNet50, YOLOv4, and YOLOv5) and to observe the enhancement of the model by training on two datasets with a varied amount of classes and images.

The evaluation of the results in the five experiments was performed on 240 test images in dataset1 and 750 test images in dataset2. According to the results in chapter four, we found weapon detectors using YOLOv4 outperform YOLOv5 and SSD models on dataset1 with a mean average precision of 0.795. Meanwhile, YOLOv5 is the best option for dataset2 with mean average precision of around 0.82. Finally, the model performance is poor in detecting knife items unlike pistols in both datasets.

Availability of Data and material

All data can be downloaded from the following GitHub repository. https://github.com/Mohammad-H-Zahrawi/Projects/tree/main/Weapon%20Detection

CONFLICT OF INTEREST

MOHAMMAD H. Zahrawi declares that they have no conflict of interest.

References

1. *About - OpenCV*. (n.d.) [online]. [Accessed 28 May 2022]. Available at: https://opencv.org/about/.
2. Ahmed, A. A. & Echi, M. (2021). Hawk-Eye: An AI-Powered Threat Detector for Intelligent Surveillance Cameras. *IEEE Access*. Institute of Electrical and Electronics Engineers Inc., vol. 9, pp. 63283–63293.
3. Alam, F., Mehmood, R. & Katib, I. (2018). D2TFRS: An Object Recognition Method for Autonomous Vehicles Based on RGB and Spatial Values of Pixels. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*. Springer, Cham, vol. 224, pp. 155–168.
4. Allaouzi, I. & Ben Ahmed, M. (2019). A Novel Approach for Multi-Label Chest X-Ray Classification of Common Thorax Diseases. *IEEE Access*. Institute of Electrical and Electronics Engineers Inc., vol. 7, pp. 64279–64288.
5. Amrutha, C. V., Jyotsna, C. & Amudha, J. (2020). Deep Learning Approach for Suspicious Activity Detection from Surveillance Video. *2nd International Conference on Innovative Mechanisms for Industry Applications, ICIMIA 2020 - Conference Proceedings*. Institute of Electrical and Electronics Engineers Inc., pp. 335–339.
6. *Bank Robbery | Page 2 | ASU Center for Problem-Oriented Policing | ASU*. (n.d.) [online]. [Accessed 20 November 2021]. Available at: https://popcenter.asu.edu/content/bank-robbery-page-2.
7. Bhatti, M. T., Khan, M. G., Aslam, M. & Fiaz, M. J. (2021). Weapon Detection in Real-Time CCTV Videos Using Deep Learning. *IEEE Access*. Institute of Electrical and Electronics Engineers Inc., vol. 9, pp. 34366–34382.
8. Bochkovskiy, A., Wang, C.-Y. & Liao, H.-Y. M. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection [online]. [Accessed 20 July 2022]. Available at: http://arxiv.org/abs/2004.10934.
9. Bruce, P. & Bruce, A. (2017). *Practical statistics for data scientists : 50 essential concepts*. First edition. Sebastopol CA: O'Reilly.
10. Chaurasia, M. A. & Mozar, S. (eds). (2022). Contactless Healthcare Facilitation and Commodity Delivery Management During COVID 19 Pandemic. Singapore: Springer Singapore (Advanced Technologies and Societal Change).
11. Chawla, N. V. (2009). Data Mining for Imbalanced Datasets: An Overview. *Data Mining and Knowledge Discovery Handbook*. Springer, Boston, MA, pp. 875–886.
12. Cheong, J. Y. & Park, I. K. (2017). Deep CNN-Based Super-Resolution Using External and Internal Examples. *IEEE Signal Processing Letters*. Institute of Electrical and Electronics Engineers Inc., vol. 24(8), pp. 1252–1256.
13. *COCO - Common Objects in Context*. (n.d.) [online]. [Accessed 11 September 2022]. Available at: https://cocodataset.org/#detection-eval.
14. *CVPR 2014 Open Access Repository*. (n.d.) [online]. [Accessed 5 June 2022]. Available at: https://openaccess.thecvf.com/content_cvpr_2014/html/Girshick_Rich_Feature_

Hierarchies_2014_CVPR_paper.html.

15. Dai, J., Li, Y., He, K., information, J. S.-A. in neural & 2016, undefined. (n.d.). R-fcn: Object detection via region-based fully convolutional networks. *proceedings.neurips.cc* [online]. [Accessed 5 June 2022]. Available at: https://proceedings.neurips.cc/paper/2016/hash/577ef1154f3240ad5b9b413aa7346a1e-Abstract.html.
16. Darker, I., Gale, A., Ward, L. & Blechko, A. (2007). Can CCTV reliably detect gun crime? *Proceedings - International Carnahan Conference on Security Technology*, pp. 264–271.
17. *Data Preprocessing for Supervised Leaning*. (n.d.) [online]. [Accessed 24 June 2022]. Available at: https://publications.waset.org/14136/data-preprocessing-for-supervised-leaning.
18. Dee, H. M. & Velastin, S. A. (2007). How close are we to solving the problem of automated visual surveillance? *Machine Vision and Applications 2007 19:5*. Springer, vol. 19(5), pp. 329–343.
19. el den Mohamed, M. K., Taha, A. & Zayed, H. H. (2020). Automatic gun detection approach for video surveillance. *International Journal of Sociotechnology and Knowledge Development*. Information Resources Management Association, vol. 12(1), pp. 49–66.
20. Dodge, S. & Karam, L. (2016). Understanding How Image Quality Affects Deep Neural Networks [online]. [Accessed 20 June 2022]. Available at: http://arxiv.org/abs/1604.04004.
21. Dong, C., Loy, C. C., He, K. & Tang, X. (2016). Image Super-Resolution Using Deep Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. IEEE Computer Society, vol. 38(2), pp. 295–307.
22. *DSpace at My University: OBJECT RECOGNITION USING IMAGE PROCESSING AND DEEP LEARNING IN MATLAB"*. (n.d.) [online]. [Accessed 28 May 2022]. Available at: http://14.99.188.242:8080/jspui/handle/123456789/10473.
23. Dwivedi, N., Singh, D. K. & Kushwaha, D. S. (2021). Employing data generation for visual weapon identification using Convolutional Neural Networks. *Multimedia Systems 2021 28:1*. Springer, vol. 28(1), pp. 347–360.
24. Elmir, Y., Laouar, S.A. and Hamdaoui, L., 2019, April. Deep Learning for Automatic Detection of Handguns in Video Sequences. In JERI. (n.d.) [online]. [Accessed 19 November 2021]. Available at: https://www.researchgate.net/publication/332798483_Deep_Learning_for_Automatic_Detection_of_Handguns_in_Video_Sequences.
25. Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*. North-Holland, vol. 27(8), pp. 861–874.
26. Fernandez-Carrobles, M. M., Deniz, O. & Maroto, F. (2019). Gun and Knife Detection Based on Faster R-CNN for Video Surveillance. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer, vol. 11868 LNCS, pp. 441–452.
27. García, S., Fernández, A., Luengo, J. & Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational

intelligence and data mining: Experimental analysis of power. *undefined*, vol. 180(10), pp. 2044–2064.

28. Gill, M. & Matthews, R. (2005). Robbers on robbery: offenders' perspectives. *Crime At Work*. Palgrave Macmillan UK, pp. 11–28.
29. Girshick, R., Donahue, J., Darrell, T. & Malik, J. (2014a). R-CNN: Regions with CNN features. *Proceedings of the ieee conference on computer vision and pattern recognition* [online]. [Accessed 5 June 2022]. Available at: http://gwylab.com/pdf/rcnn_chs.pdf.
30. Girshick, R., Donahue, J., Darrell, T. & Malik, J. (2014b). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, pp. 580–587 [online]. [Accessed 21 September 2022]. Available at: http://arxiv.
31. *GitHub - Labelbox/labelbox: Labelbox is the fastest way to annotate data to build and ship computer vision applications.* (n.d.) [online]. [Accessed 27 May 2022]. Available at: https://github.com/Labelbox/labelbox.
32. *GitHub - openvinotoolkit/cvat: Powerful and efficient Computer Vision Annotation Tool (CVAT)*. (n.d.) [online]. [Accessed 27 May 2022]. Available at: https://github.com/openvinotoolkit/cvat.
33. *GitHub - wkentaro/labelme: Image Polygonal Annotation with Python (polygon, rectangle, circle, line, point and image-level flag annotation).* (n.d.) [online]. [Accessed 27 May 2022]. Available at: https://github.com/wkentaro/labelme.
34. Giusti, A., Cire¸san, D. C., Jonathan, C., Luca, M., Gambardella, M. & Schmidhuber, J. (2013). Fast image scanning with deep max-pooling convolutional neural networks. *ieeexplore.ieee.org* [online]. [Accessed 4 June 2022]. Available at: https://ieeexplore.ieee.org/abstract/document/6738831/.
35. Grega, M., Matiolański, A., Guzik, P. & Leszczuk, M. (2016a). Automated detection of firearms and knives in a CCTV image. *Sensors (Switzerland)*. MDPI AG, vol. 16(1).
36. Grega, M., Matiolański, A., Guzik, P. & Leszczuk, M. (2016b). Automated Detection of Firearms and Knives in a CCTV Image. *Sensors 2016, Vol. 16, Page 47*. Multidisciplinary Digital Publishing Institute, vol. 16(1), p. 47.
37. Guo, Y., Li, Y., Wang, L. & Rosing, T. (2019). Depthwise convolution is all you need for learning multiple visual domains. *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*. AAAI Press, pp. 8368–8375.
38. Han, C., Duan, Y., Tao, X., … M. X.-I. T. on & 2020, undefined. (n.d.). Toward variable-rate generative compression by reducing the channel redundancy. *ieeexplore.ieee.org* [online]. [Accessed 29 May 2022]. Available at: https://ieeexplore.ieee.org/abstract/document/8952753/.
39. He, K. & Sun, J. (2014). Convolutional Neural Networks at Constrained Time Cost [online]. [Accessed 19 June 2022]. Available at: http://arxiv.org/abs/1412.1710.
40. He, K., Zhang, X., Ren, S., pattern, J. S.-I. transactions on & 2015, undefined. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. *ieeexplore.ieee.org* [online]. [Accessed 5 June 2022]. Available at: https://ieeexplore.ieee.org/abstract/document/7005506/.

41. Heitz, G., Elidan, G., Packer, B. & Koller, D. (2009). Shape-based object localization for descriptive classification. *International Journal of Computer Vision*, vol. 84(1), pp. 40–62.
42. *Home · opencv/opencv Wiki · GitHub*. (n.d.) [online]. [Accessed 28 May 2022]. Available at: https://github.com/opencv/opencv/wiki.
43. Hosang, J., Benenson, R. & Schiele, B. (2017). Learning non-maximum suppression [online]. [Accessed 14 June 2022]. Available at: http://arxiv.org/abs/1705.02950.
44. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M. & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications [online]. [Accessed 17 June 2022]. Available at: http://arxiv.org/abs/1704.04861.
45. Jain, H., Vikram, A., Mohana, Kashyap, A. & Jain, A. (2020). Weapon Detection using Artificial Intelligence and Deep Learning for Security Applications. *Proceedings of the International Conference on Electronics and Sustainable Communication Systems, ICESC 2020*. Institute of Electrical and Electronics Engineers Inc., pp. 193–198.
46. Jian, K. H. X. Z. S. R. (1996). Deep Residual Learning for Image Recognition arXiv:1512.03385v1. *Enzyme and Microbial Technology*, vol. 19(2), pp. 107–117 [online]. [Accessed 19 June 2022]. Available at: http://image-net.org/challenges/LSVRC/2015/.
47. Jiang, P., Ergu, D., Liu, F., Cai, Y. & Ma, B. (2022). A Review of Yolo Algorithm Developments. *Procedia Computer Science*. Elsevier, vol. 199, pp. 1066–1073.
48. Kambhatla, A. & Ahmed, K. R. (2023). Firearm Detection Using Deep Learning. *Lecture Notes in Networks and Systems*. Springer Science and Business Media Deutschland GmbH, vol. 544 LNNS, pp. 200–218.
49. Kang, K., Li, H., Yan, J., Zeng, X., … B. Y.-… on C. and & 2017, undefined. (n.d.). T-cnn: Tubelets with convolutional neural networks for object detection from videos. *ieeexplore.ieee.org* [online]. [Accessed 29 May 2022]. Available at: https://ieeexplore.ieee.org/abstract/document/8003302/.
50. Kanthaseelan, K., Pirashaanthan, P., A.A.P, J. J., Sivaramakrishnan, A., Abeywardena, K. Y. & Munasinghe, T. (2021). CCTV Intelligent Surveillance on Intruder Detection. *International Journal of Computer Applications*. Foundation of Computer Science, vol. 174(14), pp. 29–34.
51. Karen Simonyan∗ & Andrew Zisserman+. (2018). VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION Karen. *American Journal of Health-System Pharmacy*, vol. 75(6), pp. 398–406 [online]. [Accessed 14 June 2022]. Available at: https://click.endnote.com/viewer?doi=arxiv%3A1409.1556&token=WzQxNTM2OSwiYXJ4aXY6MTQwOS4xNTU2Il0.1ERLUjt9JINMPEYW74pwJFdB7PA.
52. Karp, A. (2018). Civilian-held Firearms Numbers 1 Estimating gloBal Civilian-HElD FirEarms numBErs.
53. Lim, J., Al Jobayer, M. I., Baskaran, V. M., Lim, J. M., Wong, K. & See, J. (2019). Gun detection in surveillance videos using deep neural networks. *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference,*

APSIPA ASC 2019. Institute of Electrical and Electronics Engineers Inc., pp. 1998–2002.

54. Lim, J. Y., Al Jobayer, M. I., Baskaran, V. M., Lim, J. M. Y., See, J. & Wong, K. S. (2021). Deep multi-level feature pyramids: Application for non-canonical firearm detection in video surveillance. *Engineering Applications of Artificial Intelligence*. Pergamon, vol. 97, p. 104094.
55. Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Verlag, vol. 8693 LNCS(PART 5), pp. 740–755.
56. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y. & Berg, A. C. (2016). SSD: Single shot multibox detector. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Verlag, vol. 9905 LNCS, pp. 21–37.
57. Luo, H., Liu, J., Fang, W., Love, P. E. D., Yu, Q. & Lu, Z. (2020). Real-time smart video surveillance to manage safety: A case study of a transport mega-project. *Advanced Engineering Informatics*. Elsevier Ltd, vol. 45.
58. Ma, J., Ming, A., Huang, Z., Wang, X. & Zhou, Y. (2017). Object-Level Proposals. *Proceedings of the IEEE International Conference on Computer Vision*. Institute of Electrical and Electronics Engineers Inc., vol. 2017-October, pp. 4931–4939.
59. Mehta, P., Kumar, A. & Bhattacharjee, S. (2020). Fire and Gun Violence based Anomaly Detection System Using Deep Neural Networks. *Proceedings of the International Conference on Electronics and Sustainable Communication Systems, ICESC 2020*. Institute of Electrical and Electronics Engineers Inc., pp. 199–204.
60. Mohana & Ravish Aradhya, H. V. (2019). Simulation of object detection algorithms for video survillance applications. *Proceedings of the International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2018*. Institute of Electrical and Electronics Engineers Inc., pp. 651–655.
61. Morales, A., Fierrez, J., Sánchez, J. S. & Ribeiro, B. (eds). (2019). Pattern Recognition and Image Analysis. Cham: Springer International Publishing (Lecture Notes in Computer Science), vol. 11867.
62. Mosselman, F., Weenink, D. & Lindegaard, M. R. (2018). Weapons, Body Postures, and the Quest for Dominance in Robberies: A Qualitative Analysis of Video Footage. *Journal of Research in Crime and Delinquency*. SAGE Publications Inc., vol. 55(1), pp. 3–26.
63. Mouzos, J., Carcach, C. & Australian Institute of Criminology. (2001). Weapon involvement in armed robbery. Australian Institute of Criminology, p. 44.
64. Mozer, M. C. (2001). Object Recognition: Theories. *International Encyclopedia of the Social & Behavioral Sciences*. Pergamon, pp. 10781–10785.
65. Muhammad, K., Ahmad, J. & Baik, S. W. (2018). Early fire detection using convolutional neural networks during surveillance for effective disaster management. *Neurocomputing*. Elsevier, vol. 288, pp. 30–42.
66. Müller, R., Kornblith, S. & Hinton, G. (2019). When Does Label Smoothing Help?

[online]. [Accessed 21 July 2022]. Available at: http://arxiv.org/abs/1906.02629.

67. Mumtaz, A., Sargano, A. B. & Habib, Z. (2018). Violence detection in surveillance videos with deep network using transfer learning. *Proceedings - 2018 2nd European Conference on Electrical Engineering and Computer Science, EECS 2018*. Institute of Electrical and Electronics Engineers Inc., pp. 558–563.
68. Namphol, A., … S. C.-I. transactions on & 1996, undefined. (1996). Image compression with a hierarchical neural network. *ieeexplore.ieee.org*, vol. 32(1), pp. 326–338 [online]. [Accessed 29 May 2022]. Available at: https://ieeexplore.ieee.org/abstract/document/481272/.
69. Nasrollahi, K. & Moeslund, T. B. (2014). Super-resolution: A comprehensive survey. *Machine Vision and Applications*. Springer Verlag, vol. 25(6), pp. 1423–1468.
70. Nguyen-Meidine, L. T., Granger, E., Kiran, M. & Blais-Morin, L. A. (2018). A comparison of CNN-based face and head detectors for real-time video surveillance applications. *Proceedings of the 7th International Conference on Image Processing Theory, Tools and Applications, IPTA 2017*. Institute of Electrical and Electronics Engineers Inc., vol. 2018-January, pp. 1–7.
71. Non-monotonic, M. A. S. R. (2020). MISH : Activation Function.
72. *Object Detection vs Object Recognition vs Image Segmentation - GeeksforGeeks*. (n.d.) [online]. [Accessed 29 May 2022]. Available at: https://www.geeksforgeeks.org/object-detection-vs-object-recognition-vs-image-segmentation/.
73. Olmos, R., Tabik, S. & Herrera, F. (2018). Automatic handgun detection alarm in videos using deep learning. *Neurocomputing*. Elsevier B.V., vol. 275, pp. 66–72.
74. Olmos, R., Tabik, S., Lamas, A., Pérez-Hernández, F. & Herrera, F. (2019). A binocular image fusion approach for minimizing false positives in handgun detection with deep learning. *Information Fusion*. Elsevier, vol. 49, pp. 271–280.
75. *Open Images evaluation protocols*. (n.d.) [online]. [Accessed 22 September 2022]. Available at: https://storage.googleapis.com/openimages/web/evaluation.html#object_detection_eval.
76. Ouyang, W., international, X. W.-P. of the I. & 2013, undefined. (n.d.). Joint deep learning for pedestrian detection. *openaccess.thecvf.com* [online]. [Accessed 29 May 2022]. Available at: http://openaccess.thecvf.com/content_iccv_2013/html/Ouyang_Joint_Deep_Learning_2013_ICCV_paper.html.
77. Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M. & Moher, D. (2021). Updating guidance for reporting systematic reviews: development of the PRISMA 2020 statement. *Journal of Clinical Epidemiology*. Elsevier Inc., vol. 134, pp. 103–112.
78. Pérez-Hernández, F., Tabik, S., Lamas, A., Olmos, R., Fujita, H. & Herrera, F. (2020). Object Detection Binary Classifiers methodology based on deep learning to identify small objects handled similarly: Application in video surveillance. *Knowledge-Based Systems*. Elsevier, vol. 194, p. 105590.
79. Redmon, J., Divvala, S., Girshick, R. & Farhadi, A. (2016). You only look once:

Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, vol. 2016-December, pp. 779–788.

80. Redmon, J. & Farhadi, A. (2018). YOLOv3: An Incremental Improvement [online]. [Accessed 19 June 2022]. Available at: http://arxiv.org/abs/1804.02767.
81. Ren, S., He, K., Girshick, R. & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. IEEE Computer Society, vol. 39(6), pp. 1137–1149.
82. Report, E. T. (2007). *Guidelines for performing systematic literature reviews in software engineering*. *Technical report, Ver. 2.3 EBSE Technical Report. EBSE*.
83. Romero, D. & Salamea, C. (2019). Convolutional Models for the Detection of Firearms in Surveillance Videos. *Applied Sciences 2019, Vol. 9, Page 2965*. Multidisciplinary Digital Publishing Institute, vol. 9(15), p. 2965.
84. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C. & Fei-Fei, L. (2014). ImageNet Large Scale Visual Recognition Challenge [online]. [Accessed 5 June 2022]. Available at: http://arxiv.org/abs/1409.0575.
85. Saikia, K., Kumar Muchahari, M., Ali, A., Nayak, B., Kr Muchahari, M. & Kumar, P. (2021). AUGMENTED REALITY BASED ONLINE APPLICATION FOR E-SHOPPING. *researchgate.net*, vol. 12(3), pp. 212–232.
86. Salazar González, J. L., Zaccaro, C., Álvarez-García, J. A., Soria Morillo, L. M. & Sancho Caparrini, F. (2020). Real-time gun detection in CCTV: An open problem. *Neural Networks*. Elsevier Ltd, vol. 132, pp. 297–308.
87. Salido, J., Lomas, V., Ruiz-Santaquiteria, J. & Deniz, O. (2021). Automatic handgun detection with deep learning in video surveillance images. *Applied Sciences (Switzerland)*. MDPI AG, vol. 11(13).
88. Simonyan, K. & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR.
89. Sultani, W., Chen, C. & Shah, M. (2018). Real-world Anomaly Detection in Surveillance Videos App We should not only judge whether there is an accident in the image, but also determine the location of the accident accurately. [online]. [Accessed 23 November 2021]. Available at: http://arxiv.org/abs/1801.04264.
90. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, vol. 07-12-June-2015, pp. 1–9.
91. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J. & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, vol. 2016-December, pp. 2818–2826.
92. Tiwari, R. K. & Verma, G. K. (2015). A Computer Vision based Framework for Visual

Gun Detection Using Harris Interest Point Detector. *Procedia Computer Science*. Elsevier, vol. 54, pp. 703–712.

93. Uijlings, J. R. R., Van De Sande, K. E. A., Gevers, T. & Smeulders, A. W. M. (2013). Selective Search for Object Recognition. *International Journal of Computer Vision 2013 104:2*. Springer, vol. 104(2), pp. 154–171.

94. *ultralytics/yolov5: YOLOv5 🚀 in PyTorch > ONNX > CoreML > TFLite*. (n.d.) [online]. [Accessed 20 July 2022]. Available at: https://github.com/ultralytics/yolov5.

95. Verma, G. K. & Dhillon, A. (2017). A Handheld Gun Detection using Faster R-CNN Deep Learning. *ACM International Conference Proceeding Series*. Association for Computing Machinery, pp. 84–88.

96. Wan, S., Chen, Z., Zhang, T., Zhang, B. & Wong, K. (2016). Bootstrapping Face Detection with Hard Negative Examples [online]. [Accessed 14 June 2022]. Available at: http://arxiv.org/abs/1608.02236.

97. Wang, B., Tang, S., Xiao, J., Yan, Q., Visual, Y. Z.-J. of & 2019, undefined. (n.d.). Detection and tracking based tubelet generation for video object detection. *Elsevier* [online]. [Accessed 30 May 2022]. Available at: https://www.sciencedirect.com/science/article/pii/S1047320318302876.

98. Wang, C.-Y., Bochkovskiy, A. & Liao, H.-Y. M. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors.

99. Wang, S., Su, Z. & Zurich, E. (2019). Metamorphic Testing for Object Detection Systems.

100. Warsi, A., Abdullah, M., Husen, M. N., Yahya, M., Khan, S. & Jawaid, N. (2019). Gun Detection System Using Yolov3. *2019 IEEE 6th International Conference on Smart Instrumentation, Measurement and Application, ICSIMA 2019*. Institute of Electrical and Electronics Engineers Inc.

101. Wink, A. M. & Roerdink, J. B. T. M. (2004). Denoising Functional MR Images: A Comparison of Wavelet Denoising and Gaussian Smoothing. *IEEE Transactions on Medical Imaging*, vol. 23(3), pp. 374–387.

102. Wu, C. T., Cheng, K. T., Zhu, Q. & Wu, Y. L. (2005). Using visual features for anti-spam filtering. *Proceedings - International Conference on Image Processing, ICIP*, vol. 3, pp. 509–512.

103. Xu, T. & Xiao, H. (2021). Application of SSD core detection algorithm in intelligent visual monitoring of examination room. *Journal of Physics: Conference Series*. IOP Publishing Ltd, vol. 2037(1).

104. Xue, B., Huang, B., Wei, W., Chen, G., Li, H., Zhao, N. & Zhang, H. (2021). An Efficient Deep-Sea Debris Detection Method Using Deep Neural Networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. Institute of Electrical and Electronics Engineers Inc., vol. 14, pp. 12348–12360.

105. Yadav, P., Gupta, N. & Sharma, P. K. (2023). A comprehensive study towards high-level approaches for weapon detection using classical machine learning and deep learning methods. *Expert Systems with Applications*. Pergamon, vol. 212, p. 118698.

106. Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J. & Yoo, Y. (2019). CutMix:

Regularization Strategy to Train Strong Classifiers with Localizable Features [online]. [Accessed 21 July 2022]. Available at: http://arxiv.org/abs/1905.04899.

107. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L. M. & Shum, H.-Y. (2022). DINO: DETR with Improved DeNoising Anchor Boxes for End-to-End Object Detection.

108. Zhiqiang, W. & Jun, L. (2017). A review of object detection based on convolutional neural network. *Chinese Control Conference, CCC*. IEEE Computer Society, pp. 11104–11109.