Weapon Detection System for Surveillance and Security

Abdullah Waqar

Enrollment No: 01-242202-001

Supervisor: Prof. Dr. Shehzad Khalid

A thesis is submitted to the Department of Computer Engineering, Faculty of Engineering Sciences, Bahria University Islamabad in the partial fulfillment of the requirements of a Master's degree in Computer Engineering.

Approval Sheet

Thesis Completion Certificate

Scholar's Name: **Abdullah Waqar**

Registration Number: **47103**

Enrollment Number: **01-242202-001**

Program of Study: **MS Computer Engineering**

Thesis Title: **Weapon Detection System for Surveillance and Security**

It is to certify that the above scholar's thesis has been completed to my satisfaction and, to my belief, its standard is appropriate for submission for examination. I have also con- ducted plagiarism test of this thesis using HEC prescribed software and found similarity index 18%. that is within the permissible limit set by the HEC for the MS degree thesis. I have also found the thesis in a format recognized by the BU for the MS thesis.

Principal Supervisor Signature: _____

Date: October 17, 2022

Name: **Prof. Dr. Shehzad Khalid**

Author's Declaration

I, ***Abdullah Waqar*** hereby state that my MS thesis titled

"Weapon Detection System for Surveillance and Security"

is my own work except, as cited properly and accurately in the acknowledgments and references, the material is taken from such sources as research journals, books, internet, etc. solely to support, elaborate, compare and extend the earlier work. Further, this work has not been submitted previously by me for taking any degree from this university. or anywhere else in the country/world. At any time if my statement is found to be incorrect even after my graduation, the University has the right to withdraw/cancel my MS degree.

Signature: _____

Date: October 17, 2022

Name: **Abdullah Waqar**

Dedication

I dedicate this piece of work to our parents, foremost, for their unfathomable support and to everyone who was there to guide us and help us through every thick and thin. Each and every person whom we met during this journey had a great impact on us and helped us in reaching at this point of success.

Acknowledgements

In preparing this thesis, I was in contact with many people, researchers, academicians, and practitioners. They have contributed toward my understanding and thoughts. In particular, I wish to express my sincere appreciation to my main thesis supervisor, Prof. Dr. Shehzad Khalid, for encouragement, guidance, critics and friendship. I am also very thankful to my co-supervisor Dr.Syed Muhammad Usman for his guidance, advice and motivation. Without their continued support and interest, this thesis would not have been the same as presented here.

Librarians at Bahria University also deserve special thanks for their assistance in supplying the relevant literature. My fellow postgraduate students should also be recognised for their support. My sincere appreciation also extends to all my colleagues and others who have provided assistance at various occasions. Their views and tips are useful indeed. Unfortunately, it is not possible to list all of them in this limited space. I am grateful to all my family members

ABSTRACT

Weapons are a critical and serious topic and has become a severe threat to current security needs. People who bring firearms into airlines, schools, and other secure locations pose a threat to public safety. In certain regions of the globe, mass shootings and gun violence are on the increase. These kinds of situations are time sensitive and may result in significant loss of life and property. Although CCTVs have been employed in many establishments but these require operators to continuously examines the video streams for weapons. The ability to identify suspicious activity is proportionate to their attention to each video stream shown on the screen, thus leading to a high rate of false positives which can become a liability to the daily operational needs of institutions. Therefore, the requirement for the deployment of video surveillance systems capable of recognizing firearms automatically has increased and plays an important role in intelligent monitoring. Several object detection models are available, which struggle to recognize firearms due to their unique size and form, as well as the varied colours of the background. This thesis presents a comprehensive literature review of recent vision-based approaches for automated detection of firearms from images and videos. The literature has broadly been categorized into classic vision/machine learning based approaches and deep learning based approaches. In this research, we further explored various deep learning alternatives for accurate fire detection. For region based detection, a deep learning based weapon detection system employing YOLO v5 for weapon detection that will be sufficiently resilient in terms of affine, rotation, occlusion, and size. The performance of our system was evaluated on a publicly available dataset and achieved the F1-score of 95.43%. Instance segmentation or pixel level segmentation was also performed which employs Mask-RCNN for the detection and segmentation of firearms. We achieved the detection accuracy (DC) of 90.66% and 88.74% Mean intersection over union (mIoU). The purposed methodology combined both techniques with different preprocessing methods along with various data augmentation techniques to improve the efficiency and accuracy of the system.

Table of Content

List of Figures

List of Tables

CHAPTER 1

INTRODUCTION

Security is a serious concern for the whole world in the modern era, as it is necessary to safeguard sensitive and valuable assets such as a person, house, community, and country. In the past centuries, humanity has achieved incredible things. Our daily life has changed drastically, and we enjoy the benefits of technology so much that we often forget about them. However, even with so much more conveniences, there are also more dangers. Guns are easier to obtain, and they are more dangerous. Especially in some countries where the access to weapons is legal, the amount of crimes related to them is high enough for it to be considered a risk. In 2020, the latest year for which complete data is available, a total of 45,222 people in the United States died as a result of gun-related incidents [57] and similar is the case in Pakistan.

The capability detecting people with guns in a crowded place, and therefore preventing a disgrace, is a desirable thing to have. This is currently achieved manually in majority of the instances through manual inspection of CCTV videos by CCTV operators and big deployments of police officers for VIP movements to prevent a possible attack. CCTV's purpose is to monitor the environment in order to prevent crime and social events. Its applicability is user dependent, for example, CCTV is used in street surveillance to monitor a number of activities, including discovering missing persons, detecting drug addiction, and identifying anti-social behaviour. Additionally, it is used to collect evidence of a crime and provide it to the appropriate authorities for prosecution. A CCTV system comprises of a camera and an operator that is unattended and deployed remotely. A CCTV camera captures and transmits the video to a base station's television screen, where the operator examines it for suspicious activity or evidence collecting. However, the operator's ability to identify suspicious activity is proportionate to his or her attention to each video stream shown on the screen. Due to the less number of operators to monitor the screen, the concurrent operation of several video feeds on the same screen, and the operating room's ambient settings, it is difficult for a CCTV operator to watch each video stream activity with total attention at all times. Due to the risk that they would miss detecting any aberrant behaviour that might have major security implications, it is

critical to discover such anomalies automatically in a timely way in order to avert any terrorist action. In rushy and safety-critical areas, like museums or stadiums, police officers stay in constant watch of video footage to try to prevent crimes. Their usual work is to keep a close look to each individual in search of weapons or general threats. This can be tiring, as it demands constant focus from police officers, who have to keep their concentration for several hours while looking at people non-stop. In fact, it demands so much focus, that some threats and weapons can pass under the security measures. This is why in special events there are big deployments of police officers, to prevent possible crimes.

Unfortunately, in certain cases even a large number of police presence is not enough for individuals to evade the security and commit a crime. Moreover, police cannot be everywhere, and that is what security cameras are for recording a specific section of a building or place, in case something happens, or someone with a weapon breaks in. These cameras are usually connected to the nearest police station and if an alarm signal is activated, or someone calls the police, the video feed can be reviewed to see what really happened. This is good when dealing with the problem of establishing the responsibility of a person in a crime, or identifying the thief, but this does not solve the problem of stopping the crime when it is being committed, or even better, before it even happens. This is where the necessity for automation arrives. Several automatic weapon detection systems have recently been proposed, trying to deal with the issue of terrorism, which may be one of the most serious problems in the topic of law enforcement for the next decade. A demonstration of automated weapon detection system is shown in Figure 1.1.

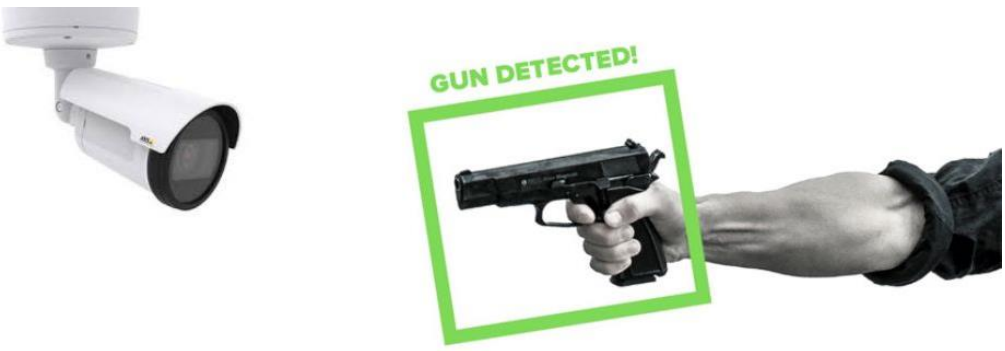

Figure 1.1 Demonstration of automated weapon detection using CCTV camera

Researchers have recently focused on this domain and have proposed a variety of approaches to automatically detect firearms in a video sequence. These approaches can be broadly categorized into traditional computer vision / machine learning based approaches and deep learning based approaches. Traditional techniques includes fuzzy classifier, edge detection techniques, interest point detector and color-based segmentation. These techniques have their drawbacks which includes less accuracy, low speed and manual extraction of feature space representation after application of tedious image processing steps. Whereas deep learning techniques such as deep Convolutional Neural Networks (CNN) are way much faster, highly intelligent, more accuracy and less number of false positives [3-6]. Based on a large quantity of labelled data, these models automatically learn the distinctive features of objects. We will discuss these techniques further in detail in upcoming sections.

1.1 Methods for Weapon Detection

There are several techniques that are widely used to detect weapons automatically on entry/exit of any important places, like museums or stadiums. Some of the techniques include metal detection machines which has metal detection sensors, some use Infra Rays sensors, some use IR sensors with image processing, X-Ray and millimetre wave detection. The classical example is the luggage control in airports and entrances to important events. All these techniques are used to detect concealed weapons as presented in Figure 1.2a. However, this method presents different limitations. They use metal detection, so they cannot detect non-metallic weapons. These methods are also very expensive, as they need the metallic sensors, and cannot be practically use in crowded places. They also lack precision, because they detect all metallic objects and therefore, they have a high rate of false positives. Other inconveniences are that they are slow and cannot be used in open spaces. In this research we are will be discussing about the technique used to detect visible weapons as shown in Figure 1.2b.

Figure 1.2 Sample images with human carrying (a) concealed weapons and (b) visible weapons

1.2 Deep Learning Techniques

Recent advancements in artificial intelligence (AI) and machine learning have enhanced the efficiency of image-based modelling and analysis (e.g., image segmentation, classification and real-time prediction) in various applications [7-10]. In addition, the development of nanoscale semiconductors has enabled a new generation of Tensor Processing Units (TPUs) and Graphical Processing Units (GPUs) to deliver data-driven approaches with exceptional computational power [11]. In addition, advanced cameras may be equipped with small edge GPU / TPU platforms for on-board processing to assist in the identification of weapons before a fatal event occurs. Deep learning has further overtaken the laborious techniques of manually extracting features in conventional image processing systems. Our objective is to create a model for the detection of weapons that is accurate, quick and sensitive. This model will be able to be implemented on a device that operates in the real world.

Researchers have developed several object identification algorithms since 2012, such as the region-based convolutional neural network (R-CNN) and its variations [13–15]. Joseph et al [16] proposed the YOLO (You Only Look Once) approach in 2016. Unlike conventional region-based approaches, YOLO is a single-stage algorithm that passes the image through a FCNN (Fully Convolutional Neural Network) only once, making it quite efficient for real-time applications. Compared to region-based technology, YOLOv2 [17] solves the comparatively

high localization error and poor recall by using batch-normalization and a classifier with a greater resolution. The third version of YOLO know as YOLOv3 [12] was launched with incremental enhancements in 2018. YOLOv4 (Accuracy and Optimal Speed of Object Detection) was introduced in 2020. Two-stage object recognition networks include Faster-RCNN and R–FCN, as well as a region proposal network. This kind of network has a slower in detecting the objects. A single-stage object identification network, similar to YOLO v3 and Single Shot Detector SDD, is thus proposed. A single forward CNN is used to predict the object's location and class.

1.3 Motivation

There are millions of events of shooting and thousands of people lost their lives, In 2020, the latest year for which complete data is available, a total of 45,222 people in the United States died as a result of gun-related incidents [57] and same is the case in Pakistan. Therefore, there is a dire need of automatic weapon detection system so that we can use it at the time of crime being committed instead of post crime analysis. It would also reduce the number of times a surveillance team fails to detect a weapon in a crowded place; and it would improve the response time of the police when a crime is committed and it is recorded by a surveillance camera, like the one that could be installed in a general store, or a gas station. Another important point is the scarcity of images that could be used as input for training the system. Most security videos are private, and labelled data is complex to gather. To deal with this problem, this project will use the concept of data augmentation to deal with the limited dataset available. Taking this into account, it seems appropriate to develop an automatic object detector capable of detecting weapons in images with a perspective similar to the one a surveillance camera has. As object detection is a complex task, a special effort needs to be made in this topic.

1.4 Problem Statement

Security is a serious concern for the whole world in the modern day, as it is necessary to safeguard sensitive and valuable assets such as a person, house, community, and country. For monitoring and surveillance tool in the fight against crime Closed Circuit Television (CCTV) are widely used. CCTV's purpose is to monitor the environment in order to prevent crime and social events. Its applicability is user dependent, for example in important places, like museums or stadiums, police officers stay on constant watch of video footage to try to prevent crimes. Their usual work is to keep a close look at each individual in search of weapons or general threats. This can be tiring, as it demands constant focus from police officers, who

have to keep their concentration for several hours while looking at people non-stop. This is where the necessity for automation arrives. Several automatic weapon detection systems are starting to appear, trying to deal with the issue of terrorism, which may be one of the most serious problems in the topic of law enforcement for the next decade. We are conducting this study to automatically detect weapons/handguns from an image or a video stream in the presence of various problems such as occlusion, complex backgrounds, different sizes, and angles of guns in a timely manner to avoid any harmful or terrorist act.

1.5 Challenges

Generally, automated weapon detection system faces various challenges which includes Automatic detection alarm systems need the pistol's precise placement in an image. Pistols may be held in a variety of ways with one or two hands, obscuring a considerable portion of the gun. There are many forms and shapes of weapons which may affect the model's accuracy and increase the number of false positives rate. Another problem is that there is no standard dataset for weapons is available and also there are very limited number of images are available. To overcome this problem, we will apply several augmentation techniques to the dataset.

1.6 Objectives

The aim of this research is to develop a system for monitoring an area's surveillance data. A person carrying a firearm in public is a significant sign of potentially harmful scenarios. Recently, the frequency of situations in which small groups or individuals use weapons to harm or kill people has increased. The following are the specific objectives:

1. To develop a deep learning based weapon detection system in real word scenarios.
2. To cater to the problem of a limited dataset through different types of preprocessing and data augmentations.
3. Reduce the number of false positive rate.
4. Apply pixel-level segmentation to achieve the highest level of accuracy in localizing the weapon.

1.7 Thesis organization

The rest of this document is structured as follows:

- Chapter 2 presents a review of previous work, along with a general review of detection methods, anomaly detection techniques and synthetic data usage.
- Chapter 3 represents the project's development structure and methodology of region based segmentation and as well as the instance segmentation.
- Chapter 4 includes the experimental results of our proposed methods as compared to various other techniques and competitors. It provides both qualitative and quantitative results for both the methods discussed in methodology.
- In chapter 5 the summary of the thesis is discussed along with the concluding remarks and the future work are presented to direct the research community.

CHAPTER 2

LITERATURE REVIEW

Creating an automated weapon identification system that operates in real-time and has a high level of accuracy, performance, can instantly produce an alert is the solution that will allow us to solve the issue of people carrying weapons into public spaces. Such detectors have several uses in security for the protection of human life, allowing authorities to respond swiftly before a significant event may occur. Therefore, the applications for weapon detectors may be a significant contribution to society for making cities considerably safer.

In this chapter we are going to present a comprehensive review of the literature. Varity of computer vision-based approaches were used towards the problem of weapons detection. The literature represented can be broadly categorized into categories (1) Traditional computer vision machine learning approaches and (2) Deep learning-based methodology.

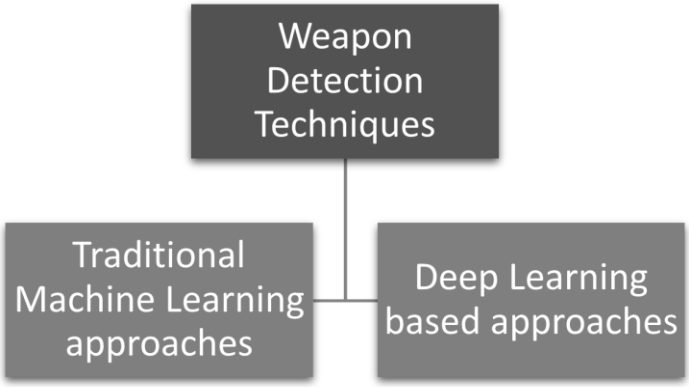

Figure 2.1 Categorization of research into Traditional and deep learning techniques

2.1 Traditional Machine Learning approaches

Variety of approaches has been proposed that has employed different computer vision based and traditional machine learning approaches. There techniques can be categorized into various groups. The taxonomy of these approaches is shown in Figure 2.2.

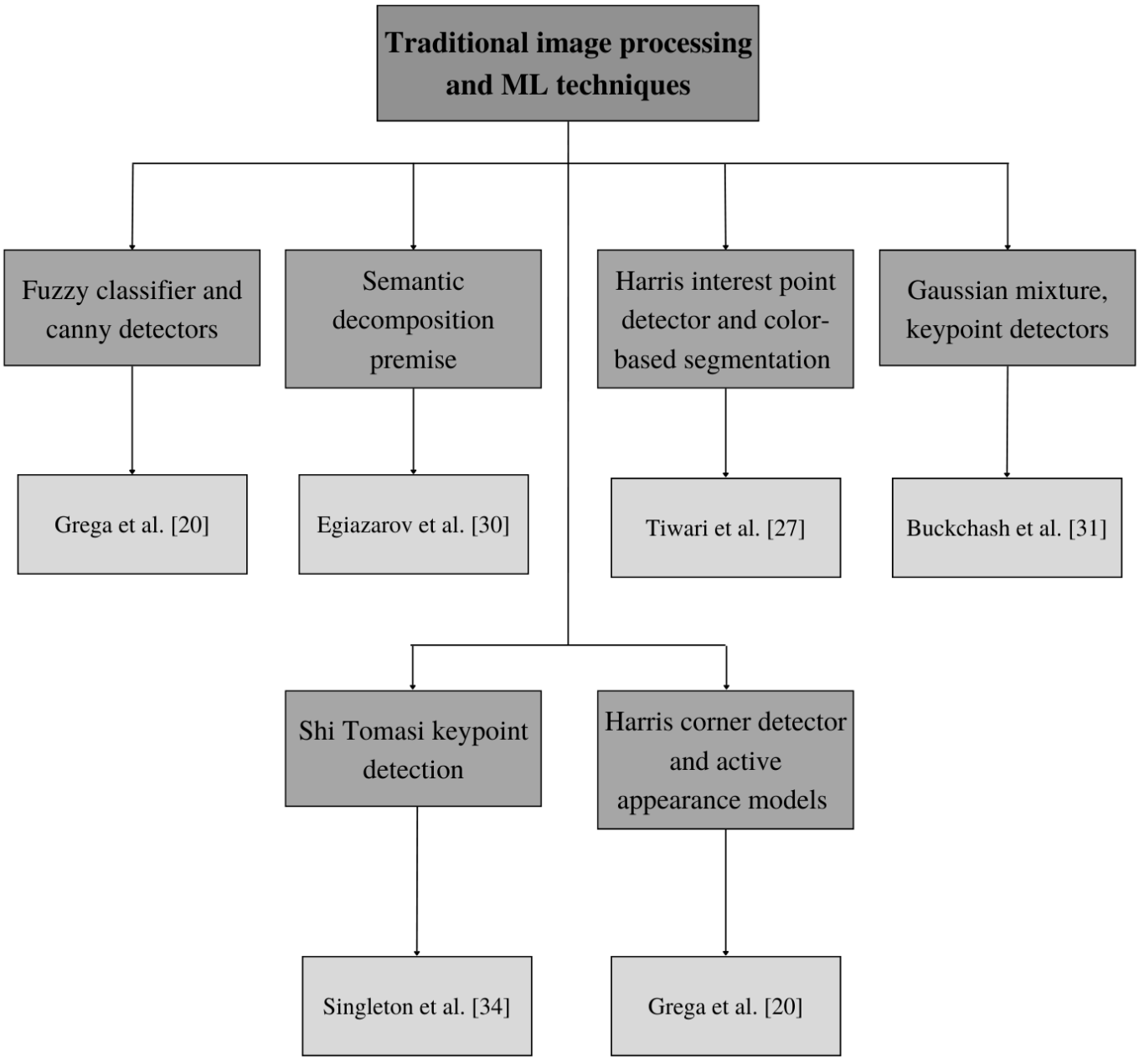

Figure 2.2 Taxonomy of traditional image processing and ML techniques employed for weapon detection system from images/videos.

Grega et al. [20] presented a machine learning-based and image processing technique for automatically detecting harmful situations in CCTV cameras. The identification of knives and firearms in a movie was accomplished using sliding window methods, a fuzzy classifier, and canny detectors. The algorithm's specificity and sensitivity are 96.69 % and 35.98 %, respectively. Egiazarov et al. [30] addressed the issue of recognizing firearms in video streams and images. They proposed an approach that used machine learning techniques and the semantic decomposition assumption of the problem. Instead of creating a single complicated learning module that would need a big amount of data and substantial computer resources, they recommended the use of numerous basic neural networks (NNs) that could be trained affordably to identify just particular components of a firearm and then combined to provide robust output. They assessed their model using data from the same training set (i.e., Google images), synthetic data that had been cleaned, and out-of-sample data (i.e., video frames). The findings demonstrated the dependability of the component part networks and the overall system's adaptability in combining the outputs of the various networks. When tested against negative samples, the network responsible for detection obtained a 95% accuracy rate but fared somewhat worse when it came to identifying weapons, obtaining an accuracy rate of 82%. The network responsible for identifying magazines attained an accuracy of 84% for negative samples and an accuracy of 81 % when detecting magazines (positive samples). The barrel-trained network properly identified 54% of the data with no guns in it and performed admirably well in recognizing barrels with a accuracy of 95%. The network responsible for stock identification achieved a 78% in accurately categorising negative samples and 82% accuracy in recognizing stocks.

Tiwari et al. [27] developed a system for visual gun detection in automated surveillance. To identify the gun in images, Harris interest point detector and color-based segmentation are utilised. Their suggested solution requires a significant amount of processing time, making it unsuitable for real-time applications. This technique also performs poorly with real-time movies with changing lighting. Buckchash et al. [31] presented a knife detection and classification system. Their suggested technique consists of three phases: foreground object detection using a Gaussian mixture, object localization using keypoint detectors, and object classification using Multiresolution Analysis. Castillo et al. [32] suggested an algorithm for detecting the presence of a handgun or knife in a picture and alerting people. Additionally, they attempted to minimize the frequency of false alerts generated by real-world surveillance recordings. Grega et al. [20] demonstrated a technique for detecting knives in photos. This

technique makes use of the Harris corner detector and active appearance models. The Harris corner detector is used to locate the knife's points. Following that, a model of the knife's active look is later produced utilizing these recommendations. The entire performance of this system is dependent on the precision of the harris corner detector. Handguns are recognized in [34] by retraining a MobileNet network and identifying areas in the picture for binary classification using Shi Tomasi keypoint detection, an improvement to Harris corner detection. Cropping off areas of the picture using the key point positions enables us to establish whether or not the crop includes a weapon.

2.2 Deep learning based approaches

Deep Learning is the specialized set of Machine Learning approaches with neural networks serving as its backbone. Deep Learning maps input to output using mathematical functions. These functions may extract unique patterns or information from the data, enabling them to build a link between input and output. Information is extracted by artificial neural networks (ANN) or neural networks (NN) in contemporary deep learning models. These neural networks are composed of a basic mathematical function that may be stacked and structured in layers to give them a sense of depth. Multiple visual analytic tasks, including as classifications, object identification, and tracking, have seen considerable performance improvements as a result of these methodologies. Currently research has focused on the introduction of lightweight neural architectures and the development of techniques that enhance the efficiency of such models. These techniques includes YOLO, Faster RCNN, MobileNet SSD and many others.

Using artificial neurons as building blocks, an MLP has an input layer to receive a signal, an output layer that retrieves the prediction or decision, and a hidden layer. MLPs with a single hidden layer are able to provide an approximation for any continuous function. A general representation of a machine learning algorithm can be seen in Figure 2.3.

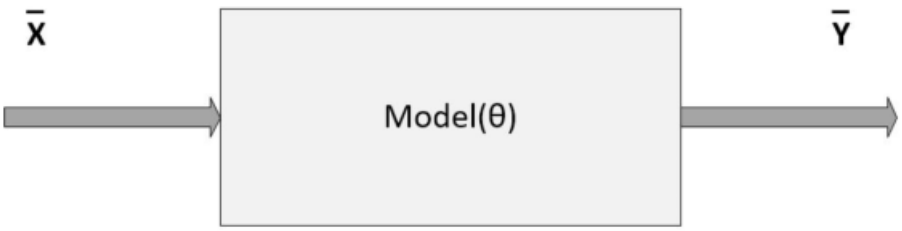

Figure 2.3 General representation of machine learning model

Nevertheless, more complex structures are needed when solving complex, non-linear problems like image analysis and processing. MLPs are more computationally expensive for image analysis because MLPs use one perceptron for each input. When dealing with images, this means the input layer needs one perceptron for each pixel, and that results in a huge amount of weights to be trained. CNN is the category of deep learning algorithm that is most commonly used for image analysis and recognition.

Convolutional Neural Networks (CNNs) are comparable to other Neural Networks (NNs), but their usage of a sequence of convolutional layers adds an additional layer of complexity. Convolutional layers are a fundamental component of CNNs. Convolutional layers consist of a collection of filters which are also known as kernels, that are applied to input images. A feature map is the output of the convolutional layer, which is a description of the input images with the filters applied. It is possible to stack convolutional layers to develop more complicated models that can learn more complex image features. Deep learning employs pooling layers, which are a form of convolutional layers. Pooling layers decrease the input's spatial dimensions, making it simpler to process and consuming less memory. Pooling also speeds up the training process by decreasing the number of parameters. Fully connected layers are among the most fundamental kinds of layers in a convolutional neural network (CNN). As the name indicates, every neuron in the top layer is linked to every neuron in the bottom layer. Typically, fully connected layers (FCL) are used near the ending of a CNN when the aim is to employ the previously learnt characteristics to produce predictions.

Variety of weapon detection and localization systems have been proposed that are based on different variants of deep learning models. The taxonomy of the approaches discussed in the literature is shown in Figure 2.4.

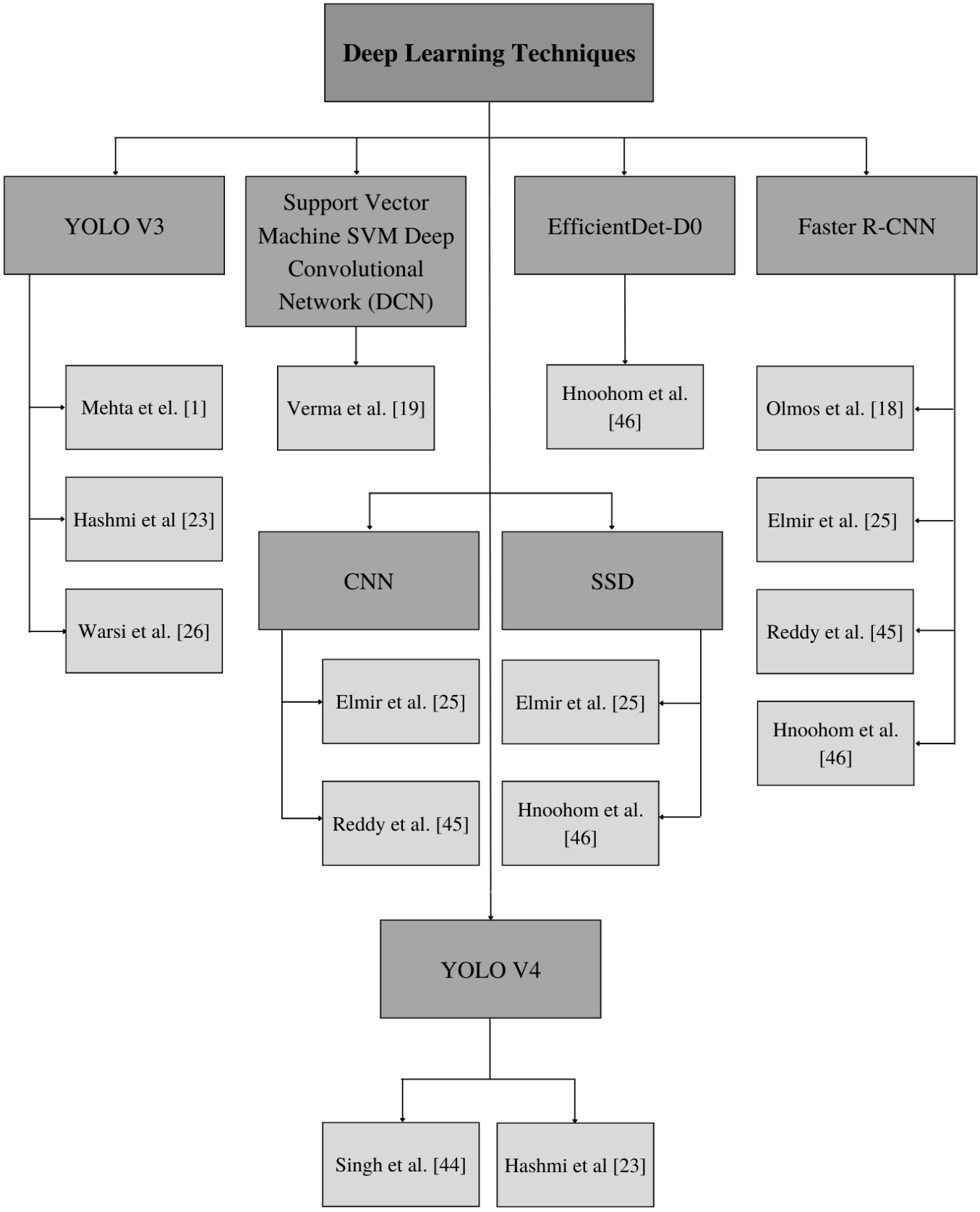

Figure 2.4 Taxonomy of deep learning techniques employed for weapon detection system from images/videos

Olmos et al. [18] presented a deep learning based approach for firearm recognition by considering it a two-class problem of weapon and background. They created a training database by manually labelling photos from the Internet and employing Faster-RCNN based on VGG16 as a detection model. In general, this model scored a satisfactory accuracy of 91.43 % when trained on a dataset of 6000 photos, but it incorrectly associates the gun with things that may be handled similarly, such as a smartphone, knife pocketbook, card and bill. Verma et al. [19] suggested an automated gun identification system based on Convolutional Neural Networks for crowded scenes (CNN). Through transfer learning, they employed Deep Convolutional Networks (DCN), a state-of-the-art Faster Region-based CNN model. The experiments used a dataset derived from a portion of the Internet Movie Firearm Database (IMFDb). The system detected and classified three kinds of firearms, shotguns, revolvers and pistols, with an accuracy of 89.9 % using Support Vector Machines (SVM).

Luvizon et al. [24] further extended the research by including the human's stance and look in order to get improved results for identifying movements and activities conducted on the scene. Given that the human stance has sufficient information to assess activity, it is predicted to be beneficial in solving the handgun detection challenge. The author presents a multitask framework for simultaneously estimating 2D and 3D position from still images and recognising human actions from video sequences. In a unified framework, these stances and visual information are combined to anticipate actions. While body position information has been frequently used to classical computer vision issues such as gesture and activity identification, it has not been widely applied to handgun detection. Luvizon et al. [24] shared a qualitative result of his research.

Elmir et al. [25] evaluated three different models for firearms detection including Faster R-CNN, Mobile-Net and the Convolutional Neural Network CNN. They have validated their approaches and provided the comparison of the above three methods on separate datasets. The training data set comprises 9,261 images, 200 of which correspond to the 102 handgun classes. The training data set for the region task has 3,000 images. The detection and classification test data set comprises 608 images, of which half them has handguns. They trained models from a database for the region proposal approach using a sample of 420 images. They evaluated the first model using 608 images, the second model using 200 images, and the third model using 420 images. The performance of these models was evaluated, They achieved 55% of accuracy with CNN, 80% accuracy on Faster R-CNN, and 90% accuracy was archived using Mobile-Net. Lai et al. [35] utilise a Tensorflow-based version of Overfeat3, an integrated framework

that employs CNNs for classification, detection and localization. They achieve 93 % training and 89 % test accuracy on images from surveillance videos, films, and homemade movies.

Darknet YOLO is a convolutional neural network-based object identification system [10]. It was built first using Darknet, an open-source neural network framework written in C and CUDA. [22]. Traditional classifiers find potential areas and identify the intended item using sliding window approaches or selective search. Thus, locations with a high probability of having weapons are identified [21]. YOLO, in contrast to previous approaches, does not reuse a classifier for detection. This method merely examines the image once. The picture is segmented into many sub regions in order to do the detection. Five bounding boxes are examined for each sub-region, and the probability of each having an item is determined. YOLO performs detection 100 times faster than Faster RCNN. [21]. Since 2012, researchers have developed a range of object identification methods and architectures, including the region-based convolutional neural network (R-CNN) and versions [36 – 38]. Joseph Redmon [39] proposed the "YOLO" (You Only Look Once) approach in 2016. In contrast to classic region-based techniques, YOLO is a single-stage technique that passes the picture through a FCNN just once, making it relatively quick for real time applications. By using batch normalization and better resolution classifiers, YOLOv2 [40] overcomes the relatively high localization error and poor recall associated with region-based technology. YOLOv3 [41] was launched with gradual enhancements. YOLOv4, which stands for Optimal Speed and Accuracy of Object Detection, was introduced in 2020.

YOLO is a technique that provides real-time object identification using neural networks. This algorithm is often used because of how fast it is and how accurate it is. In diverse applications, it has been used to identify pedestrians, traffic lights, parking metres, and animals. "You Only Look Once" is the acronym for the expression YOLO. The process of object recognition in YOLO is approached as a regression issue, and the results offer information about the class probabilities of the images that were detected. CNNs are used by the YOLO algorithm to recognize objects in real time. Object detection simply takes a single forward propagation via a neural network, as the name indicates. This indicates that the entire image is predicted in a single algorithm run. The CNN is used to concurrently predict multiple bounding boxes and class probabilities.

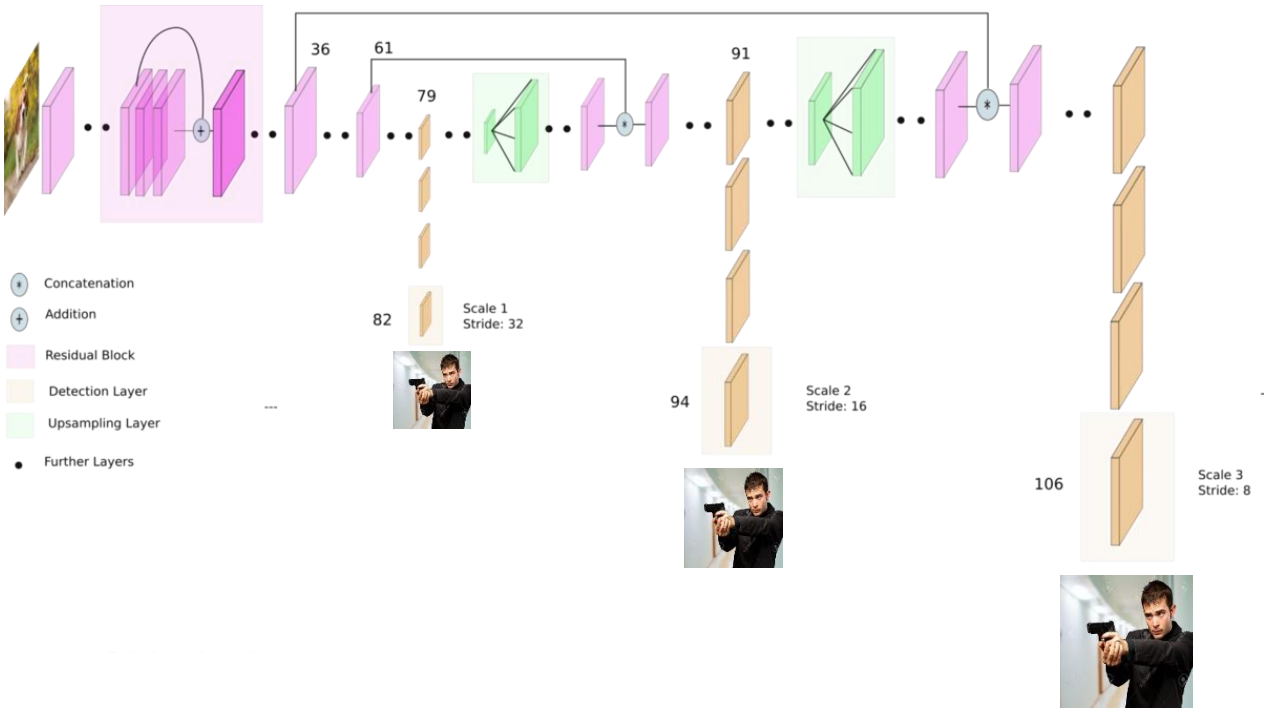

Figure 2.5 Demonstration of architecture and internal working of Yolov3 [1]

Mehta et al. [1] used YOLOv3 to detect weapons. Using the concept of a residual network, the author increases the precision of detecting objects. In terms of detecting speed, the single-stage technique works quite well. The architecture of YOLO v3 employed by Mehta et al. [1] for weapon detection can be seen in Figure 2.5. It utilises Darknet-53 minus the final three layers to build a 32-fold downsampled small-scale feature map from the source image. For instance, if the dimensions of the source image are 416 x 416, the feature map will have dimensions of 13 x 13 pixels. Large objects are detected using the small-scale feature map. It creates large-scale feature maps by upsampling the small-scale feature map and interconnecting it with a feature map from an earlier layer, as opposed to SDD's method of picking the earlier layers. This large-scale feature map including location data from previous layers and intricate features from deeper levels is utilized to recognize tiny objects. The source image is sampled eight, sixteen, and thirty-two times smaller to create each of the 3 scales of the feature maps. The concatenation process extends the dimensions of feature map, while the add operation only adds them without expanding the dimensions. They employed separate sigmoid functions to indicate multilabel classification for each bounding box rather than a softmax, which is often used to make predictions about single-label classifications. Thus, per bounding box, many categories, such as weapon or knife, may apply. This design is effective for spotting places where weapons simultaneously occur. To fulfil the requirements of firearm detection, the

number of detection object types is reduced to one at the network endpoint. They implied two different datasets for the validation of their proposed system. The first dataset namely UGR comprises of total 608 images from which half of them have guns and the remaining are without guns. The other dataset is known as IMFDB totally comprises 6000 images out of which 4000 images are with guns and the remaining 2000 without guns. The YOLOv3 model was trained and tested on both datasets separably. On the UGR dataset, they have reported the F1-score of 90.3% and for IMFDB they have reported 84.5% but they have not applied any preprocessing or data augmentations to further enhance the accuracy of the system.

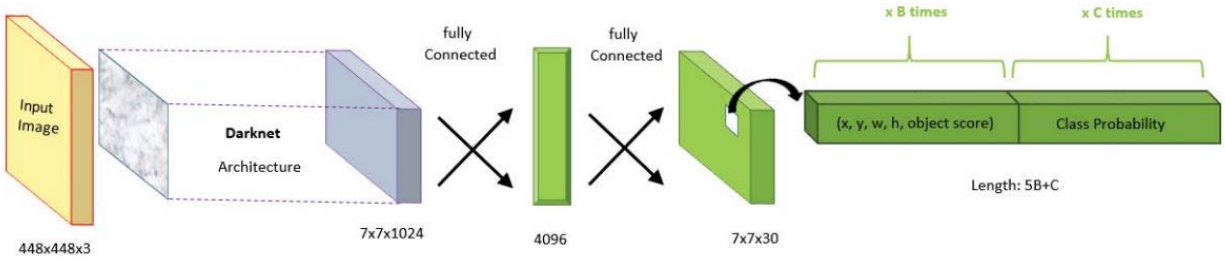

Figure 2.6 YOLO Architecture to Represent Convolutional Layer Implementation [23]

Hashmi et al [23] compared the performance of two versions of YOLO, YOLOv3, and YOLOv4. A new dataset containing 7800 images was obtained from different CCTV footage and Google images and was labeled manually. All the images in the dataset were resized to 448 x 448 x 3 and both model was trained and tested separately on the same dataset. After evaluating the result YOLO v4 managed to achieve a better F1-score of 82% as compared to 77% F1-score achieved by YOLO v3. Warsi et al. [26] proposed a system for visually detecting the presence of a firearm in real-time video surveillance. The suggested technique employs the YOLOv3 algorithm and compares the number of false-negative and false-positive predictions to those obtained using the Faster RCNN algorithm. They enhanced the findings by creating their own collection of pistols from all available angles and combining it with the ImageNet dataset. The YOLOv3 technique was used to train and evaluate the dataset. They validated the findings of YOLOv3 against Faster RCNN using 4 separate movies. The detector performed poorly at detecting pistols in scenarios with varying forms, sizes and rotations with an evaluated F1-score of 75%. Singh et al. [44] proposed YOLOv4 based algorithm for real-time security monitoring. YOLOv4 applies a single forward neural network to the whole image and splits it into regions, probabilities, and bounding boxes for each area. They utilized Kaggle Dataset

which was pre-labelled and also add images from google and labelled them manually and resized all the images to 416 x 416 pixels. The system achieved an accuracy of 70% after evaluation.

Reddy et al. [45] proposed two algorithms for the identification of hand-held weapons in real time. Faster Region-Based Convolutional Neural Networks (Faster R-CNN and Multi Contrast Convolutional Neural Networks (MC-CNN).

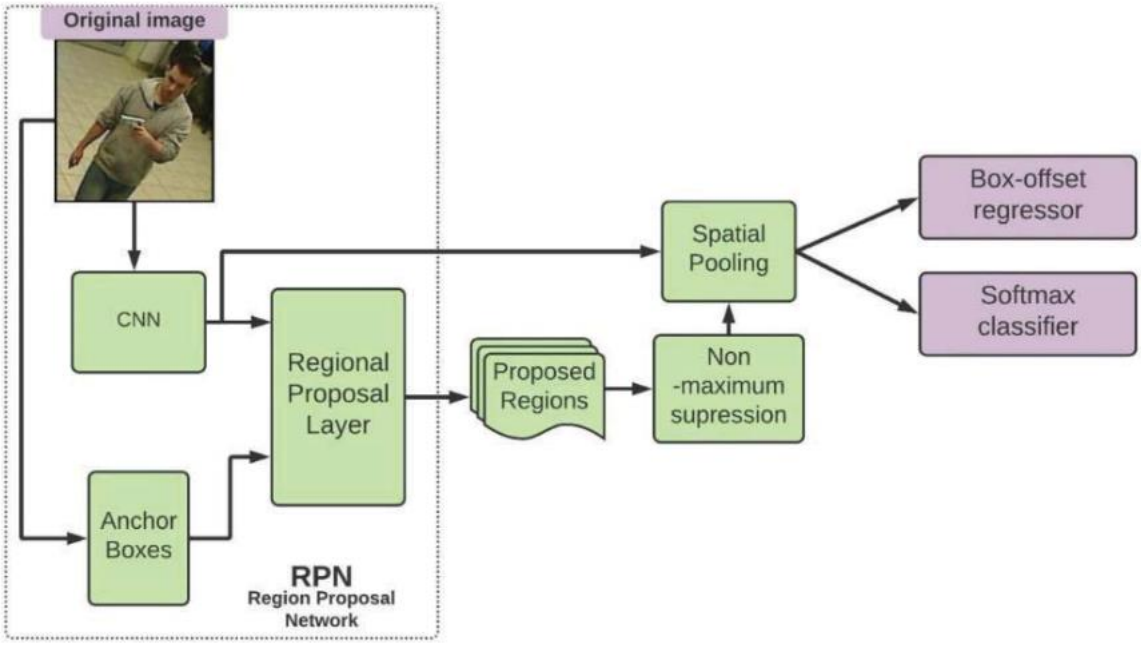

Figure 2.7 Faster-RCNN based framework for weapon-detection [45]

Figure 2.7 represents the two steps of the Faster-RCNN setup used by the author. Feature maps of the original picture are created using feature extraction networks (Inception, Inception Resnet -v, ResNet, VGG, etc.) in the first step. And the feature map from a particular intermediate convolutional layer is utilized by Region Proposal Network to recognise proposal sections with object classes scores and locations (RPN). Using a 2 class softmax layer and a robust loss function, this step simply outputs scores that indicate the likelihood of an object's presence or absence, as well as box regression for each proposition. In the second step, the positions of the proposed areas are utilised to crop features from the same feature map using an operation known as ROI pooling. The author created his own dataset of 788 images and divided it into ratio of 2:3 as train and test images. All the images were resized to 128 x 128

pixels and applied contrast enhancement techniques. After evaluating the results of both the techniques MC-CNN achieved 94.2% and Faster R-CNN achieved 92.5% of accuracy.

Hnoohom et al. [46] proposed a system to detect the weapons from a video stream automatically to reduce the screen-reading workload of police officers with a limited workforce. The Weapon detection methods proposed in this research are based on (1) Faster R-CNN Inception Resnet-V2 (2) EfficientDet-D0 and (3) SSD MobileNet-V1.

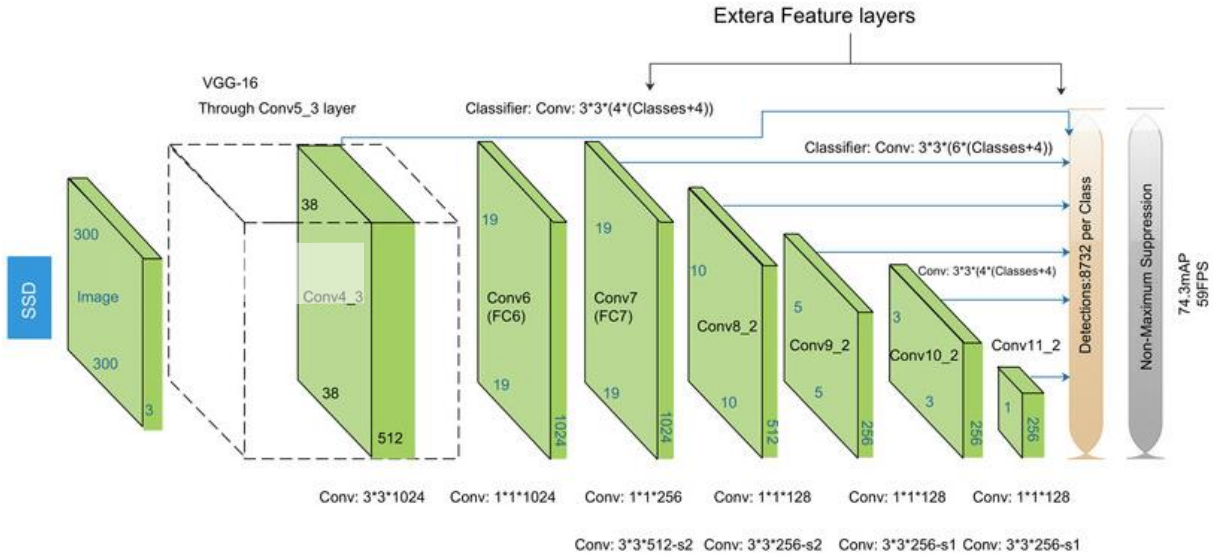

Figure 2.8 Architecture of Single Shot Detector SSD

Two-stage object detection networks include Faster-RCNN and R–FCN, as well as a Region Proposal Network (RPN). This kind of networt is slow in detecting an object. A single-stage object detection network, similar to YOLOv3 and Single Shot Detector SSD, is thus suggested. A single forward CNN is used to identify the object's location and class. It consists of three sections: Initially, the fundamental convolutional layers comprise of networks for feature extraction such as Inception, Inception Resnet -v, ResNet, VGG etc. To identify smaller objects, the intermediate convolutional layer of this section provides a large-scale feature map with more cells and lower receptive field sizes. Second, the extra convolutional layers are connected to the original convolutional network's final layer. This portion of the layers provides multi-scale feature maps with bigger receptive fields for the detection of larger objects. Thirdly, the prediction convolutional layers make use of a tiny convolutional kernel to make predictions about the positions of bounding boxes and confidence levels for different

categories. The author used two datasets, dataset 1 uses 3,000 images from ARMAS weapon detection dataset and was divided into 2400 train and 600 test images. Dataset 2 contains 4,940 images from Internet Movie Firearms Database (IMFDB) and was divided into 3952 train and 988 test images. After evaluating the models on dataset 1, the F1-scores of 73.56%, 67.42%, and 73.81% were achieved on Faster R-CNN Inception Resnet-V2 (2) EfficientDet-D0 and (3) SSD MobileNet-V1 respectively. On dataset 2, the F1-scores of 58.17%, 57.51%, and 61.08% were achieved respectively.

2.3 Analysis

In the previous section we have provided comprehensive review of the recent literature of the work that has been done for the weapon detection. A variety of deep learning and machine learning approaches were used with range datasets. However, we have observed that most researcher has used customized dataset which are not publicly available which makes the comparison difficult. This section provides the analysis of the techniques that are discussed in literature review. The comparison of the various approaches were presented, as we can see that variety of researchers used CNN, YOLO, Faster RCNN and many other. However, the best results are obtained by Reddy et al. [45] and achieved a reasonable F1-score as they have preprocessed the images, but the speed of detection was an issue as the technique that they used is slower as compared to latest approaches. The highest F1-score is 94.2% which is achieved by Reddy et al. [45]. However, there is still room for improvement in terms of F1-score and as well as the detection speed. Table 2.1 presents the preprocesssing techniques along with feature extraction and classifier employed by different researchers, it also highlights the dataset that they have used along with the F1-score that they have achieved. Another visualization of the comparison of the F1-score is presented in the form of graph in Figure 2.9 below.

Table 2.1 Comparative analysis of various recently proposed weapon detection systems

Sr. no	Paper	Preprocessing	Technique / Classifier	Dataset	F1-score (%)
1	Verma et al. [19]	Resize images to 224 × 224 x 3	Support Vector Machine SVM Deep Convolutional Network (DCN)	IMFDb	89.9
2	Olmos et al. [18]	-	Faster R-CNN	6000 images	91.43
3	Mehta et al. [1]	-	YOLO V3	UGR	90.3
				IMFDb	84.5
4	Hashmi et al [23]	Resize all images to 448 x 488 x 3	YOLO V3	7800 images	77
			YOLO V4		82
5	Elmir et al. [25]	32 x 32	CNN	608 images	55
			Fast R-CNN	200 images	80
			Mobile-SSD	420 images	90
6	Warsi et al. [26]	Resize all images to 448 x 488 x 3	YOLO V3	own dataset	75
7	Singh et al. [44]	Resize all images to 416 x 416	YOLO V4	own dataset	70
8	Reddy et al. [45]	Resize images to 128 × 128	MC-CNN	788 images	94.2
			Faster R-CNN		92.5
9	Hnoohom et al. [46]	-	EfficientDet-D0	3000 images	73.81%, 67.42%, 73.56%
			SSD MobileNet-V1		
			Faster R-CNN Inception Resnet-V2	4940 images	61.08%, 57.51%, 58.17%

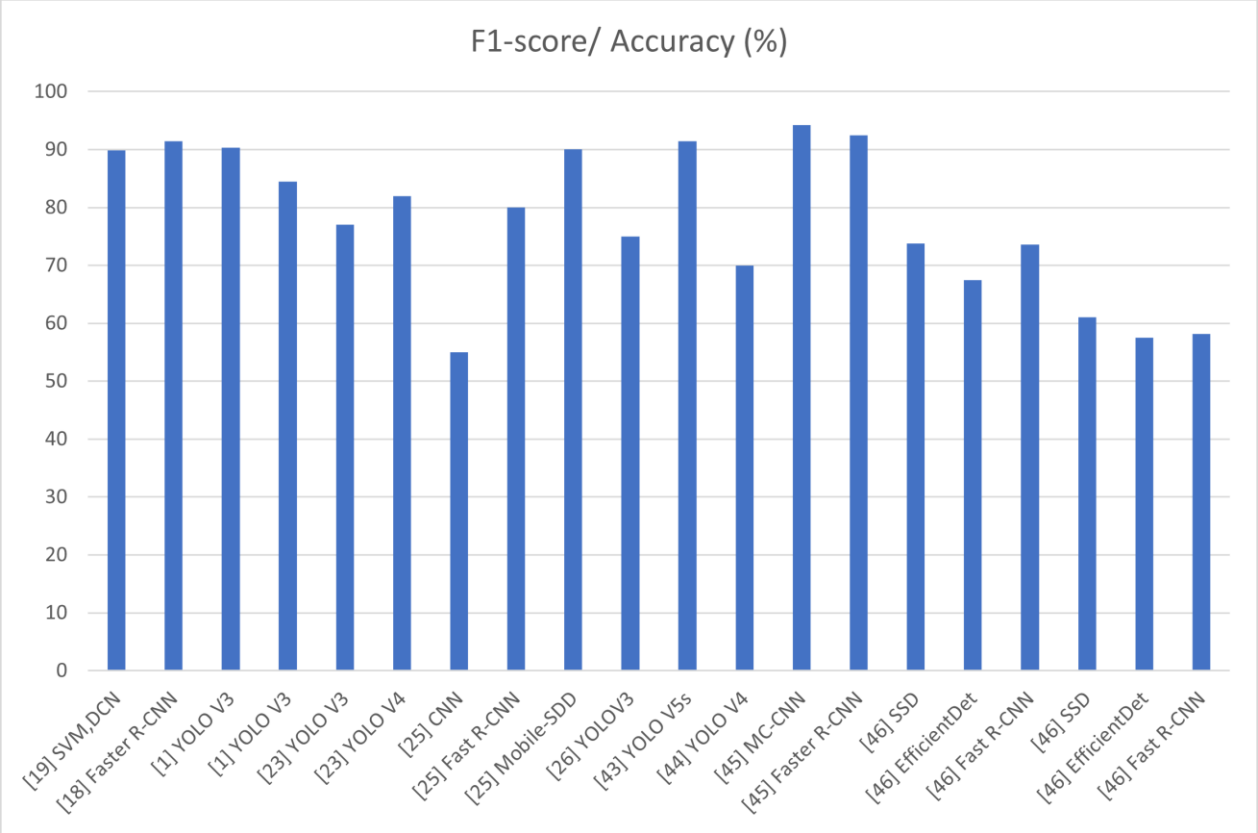

Figure 2.9 Comparison of F1-scores of stat-of-the-art techniques

CHAPTER 3

METHODOLOGY

Gun shots are very common across the world now a days which results in injuries and loss of life. Therefore, there is a dire need of a system that can automatically detect weapons in busy and rushy areas. This chapter explains organization of the research work, internal working of the techniques used, and how the results are obtained. The detection and localization of firearms in images and live video stream is done at two different levels. These techniques can be broadly categorized into 2 categories: 1) Region based segmentation and 2) Instance or pixel level segmentation. Region based segmentation quickly detects and localizes the presence of weapon from the image in the form of bounding box around the weapon. However, to accurately estimate the area of the weapon in an image or video stream, pixel level segmentation is required. The sample outputs for both region based segmentation and as well as pixel level segmentation is depicted in Figure 3.1.

Figure 3.1 Sample results of (a) region based segmentation and (b) pixel level segmentation technique

3.1 Region based Segmentation

The flow chart of our region based segmentation technique is depicted in Figure 3.2 below. First, we started off with the acquisition of dataset and then applied preprocessing and augmentation techniques on the acquired dataset. Afterwards the training of the modal was done and then tested on the unseen/test images. The detail of each step is discussed below in the coming section.

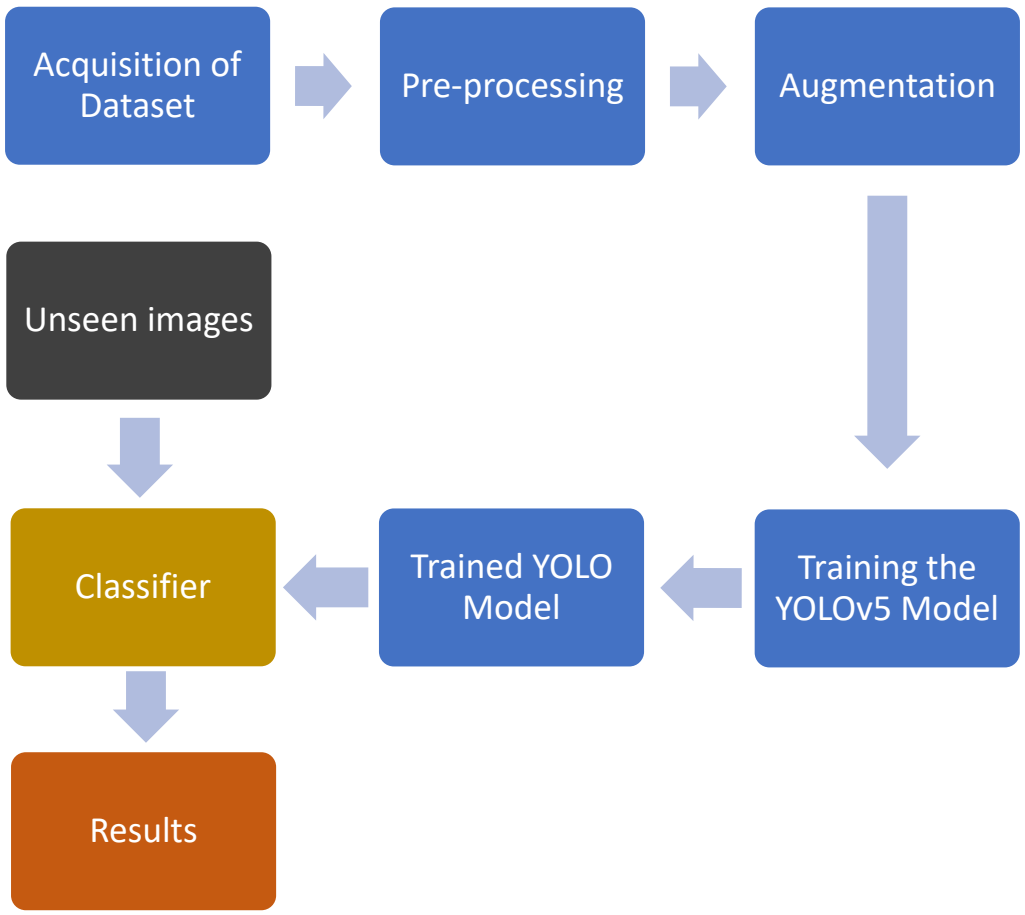

Figure 3.2 Flow of Work for region based segmentation

3.1.1 Acquisition of Dataset

This research aims to develop a fast and accurate system to detect weapons from a video stream or any image. Therefore, we used a publicly available dataset which contains 2.971 images which was published by University of Granada. We split our dataset into three sets, 70% training set (which contains 2,080 images), 20% for validation set (which contains 594 images), and lastly 10% for testing set (containing 297 images), Table 3.1 shows the division of dataset. Figure 3.3 contains samples taken from the dataset.

Table 3.1 Division of dataset into training, validation and testing sets

Dataset	Training Set	Validation Set	Testing Set
No. of Images	2,080	594	297

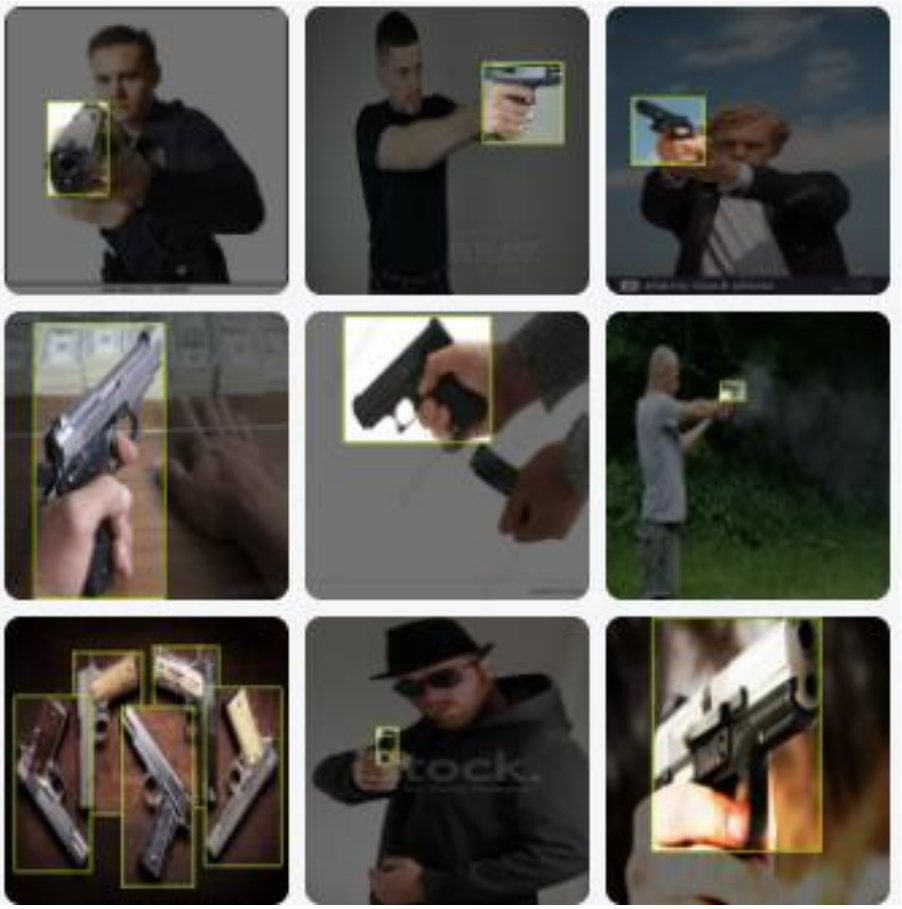

Figure 3.3 Samples from the dataset

3.1.2 Preprocessing techniques applied on the dataset

The process of cleaning and formatting raw data so that it may be used by a machine learning model is referred to as "data preprocessing." It is the first and most important phase in model training. When developing a deep learning application, it is not always the case that we encounter clean and well-structured data. In addition, it is necessary to filter and prepare data prior to performing any action on it. The images in our dataset were pre-processed before training the model. The pre-processing techniques were applied using an online tool known as RoboFlow. Following Pre-processing techniques were applied to the dataset i.e. Resize, Auto-Orient and Grayscale. Table 3.2 shows the types and range of the preprocessing techniques.

Table 3.2 Preprocessing techniques applied to the dataset

Pre-processing type	Pre-processing
Auto-Orient	Applied
Resize	Stretch to 416x416
Grayscale	Applied

3.1.3 Augmentation techniques applied on the dataset

Solutions for the data limitation include massive data collecting and the use of other pre-trained networks. Using image fusion, geometric alteration and other approaches, data augmentation generates new images. It replicates the appearance of things in a variety of situations, and approximates actual data using existing and incomplete data. After training neural networks many times, it is observed that data augmentation makes the training results more resilient, improves prediction accuracy, accelerates convergence, and reduces training process time. This enhances the accuracy and consistency of the training. Our dataset was augmented using crop, flip, rotate, blur, noise, and cutoff, among others, and created three outputs per training sample. Table 3.3 shows the augmentation techniques applied to the dataset.

Table 3.3 Augmentations techniques applied to the dataset

Augmentation type	Ranges
Crop	0 - 20% Zoom
Flip	Vertical, Horizontal
90° Rotate	Counter-Clockwise and Clockwise
Rotation	Between -15° and +15°
Noise	Up to 5% of pixels
Cutout	3 boxes with 10% of image size each
Blur	Up to 10 pixels

After applying all the pre-processing and augmentation techniques the total number of images obtained were 7,131, from which 6,240 images were used for training, 594 images for validation and 297 images for test set. A sample of the dataset after applying the above-mentioned augmentations is shown in Figure 3.4 below.

Figure 3.4 Sample of the dataset after applying the augmentations

3.1.4 Region segmentation using YOLOv5

We used YOLO for several reason; This technique increases the speed of detection since it can identify objects in real-time. It has a very high precision because YOLO is a prediction method that yields precise results with very little background errors. The method has great learning capabilities, allowing it to learn the representations of objects and apply them to object identification. The Darknet framework was developed using C and CUDA programming languages. YOLOv5 was built on top of the Pytorch framework, not long after the previous version of YOLOv4, was made public. With a few exceptions, YOLOv5 is the subsequent member of the YOLO family. It is composed of object identification models. Starting with very tiny models capable of providing real-time FPS and progressing to extremely big and precise models intended for cloud GPU installations. It contains about everything one

could need. There are five version of YOLOv5 in total. Starting from YOLOv5 nano (smallest and quickest) through YOLOv5 extra-large (biggest and slowest). Below is a brief explanation of each of these.

YOLOv5n is a newly launched nano model that is the smallest in the series, designed for edge and IoT devices, and also supports OpenCV Deep Neural Network DNN. It is around 4 MB in FP32 format and less than 2.5 MB in INT8 format. YOLOv5s is the family's smallest model, with around 72,000,000 (7.2 million) parameters, and is appropriate for executing inference on a CPU. YOLOv5m is a moderate-sized model with 21,200,000 (21.2 million) parameters. It is maybe the best model for a variety of datasets since it strikes a nice mix between precision and speed. Then comes YOLOv5l, which is the largest member of the YOLOv5 family, with 46,500,000 (46.5 million) characteristics. It is useful for datasets in which tiny items must be detected. The YOLOv5x is the biggest of the five versions and also has the greatest mAP of the five. Despite being slower than the others and containing 86.7 million parameters. Table 3.4 provides a more comprehensive summary of all models, including the inference speed on CPU and GPU, as well as the number of parameters for an image size of 640 pixels. Figure 3.5 below shows the comparison of YOLOv5 models in a graphical view.

Table 3.4 Comparison of YOLOv5 Versions

Model Name	Params (Million)	Accuracy (mAP 0.5)	CPU Time (ms)	GPU Time (ms)
YOLOv5n	1.9	45.7	45	6.3
YOLOv5s	7.2	56.8	98	6.4
YOLOv5m	21.2	64.1	224	8.2
YOLOv5I	46.5	67.3	430	10.1
YOLOv5x	86.7	68.9	766	12.1

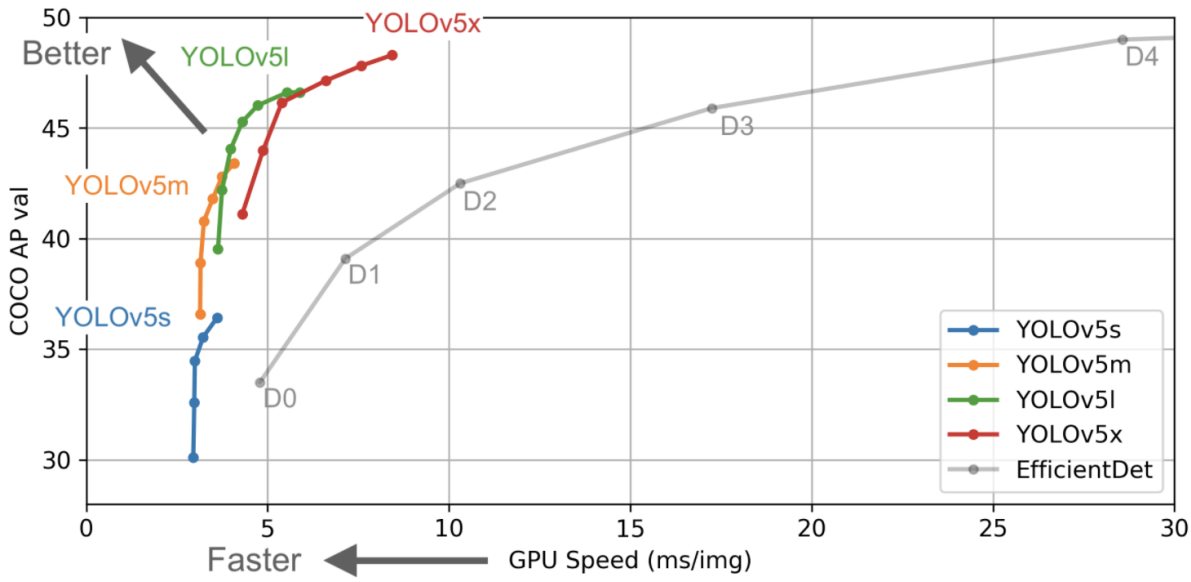

Figure 3.5 Comparison of YOLOv5 Versions in graphical form [58]

YOLOv5 uses a different approach than other object identification algorithms, which typically make use of preset anchor boxes in accordance with the MS COCO dataset. In reality, prior versions of YOLO, such as YOLOv2, solely employed k-Means clustering for this purpose. However, YOLOv5 employs a genetic algorithm to build anchor boxes. This procedure, known as auto anchor, recalculates the anchor boxes to suit the data if the default ones are inadequate. This is used in combination with the k-Means method to generate k-Means evolved anchor boxes as represented in Figure 3.6. This is one of the reasons why YOLOv5 performs so well even when applied to a wide variety of datasets.

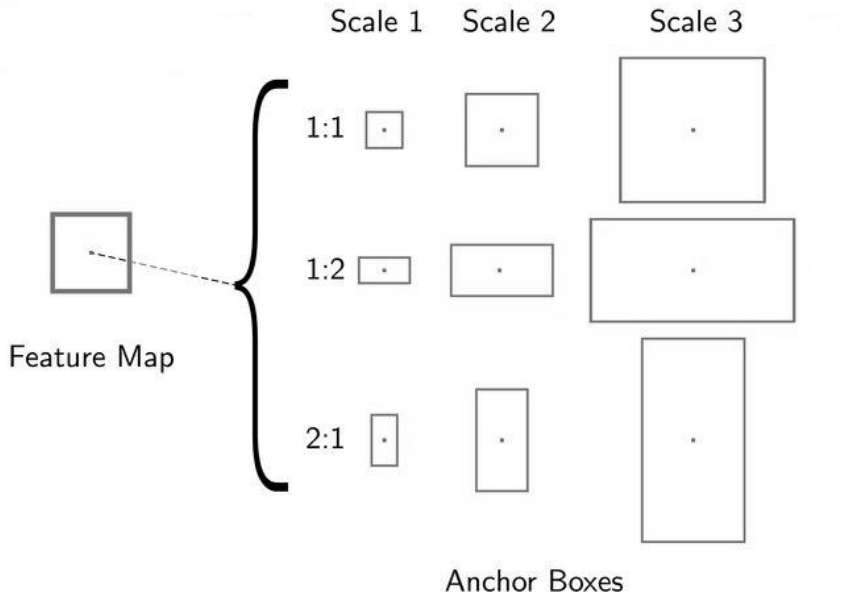

Figure 3.6 YOLOv5 anchor boxes for object detection

Each of these 5 versions of YOLOv5 has their own advantages. Our issue is very much time sensitive, as we have to detect the weapon in a timely manner to raise the alarm for the law enforcement agencies so that they can stop the crime before any serious damage happens. So based on our requirement we choose YOLOv5s as it is one of the smallest model in the YOLOv5 family with around 72,000,000 (7.2 million) parameters and it is ideal for running inference on the CPU and due to this it is the fastest among all.

3.1.4.1 Architecture of YOLOv5s

We have used a state-of-the-art object detection algorithm YOLOv5s for our research work. As typical other single-stage object detectors, it is composed of three major components: the Backbone, Neck, and Head. The internal workings and structure of the model are shown in Figure 3.7.

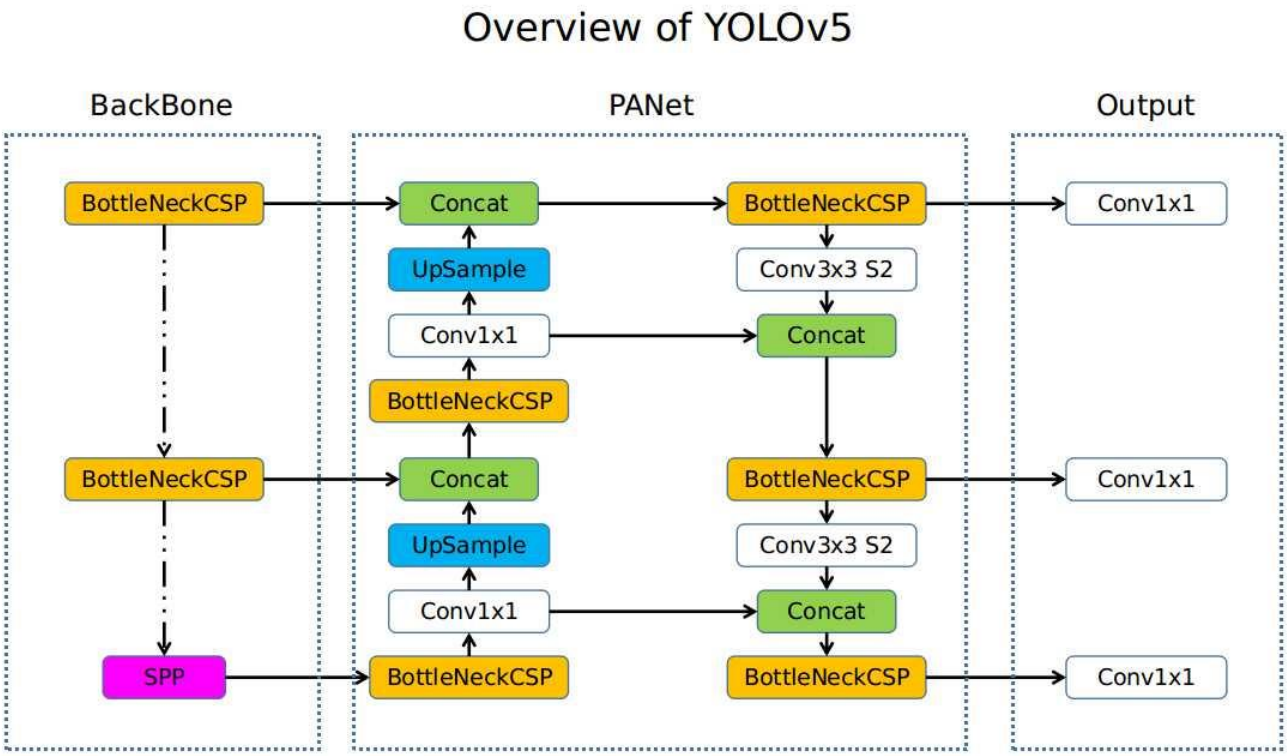

Figure 3.7 Architecture and structure of YOLOv5

Backbone: In YOLO v5, the backbone utilized is Cross-stage Partial Networks (CSP) which extracts important features from an image. The purpose of CSPNet is to eliminate computational bottlenecks by uniformly spreading computation throughout all convolutional layers, the objective is to maximize the usage rate of each calculation unit. **Neck:** The Neck is

mostly used to generate feature pyramids; it aids in the model's generalization when objects are scaled. So it is responsible for identifying the same objects which are in a variety of scales and sizes. The Feature pyramids aid the model in performing effectively on unseen data. For the Neck, YOLO v5 employs Path Aggregation Network (PANet) to generate feature pyramids. **Head:** The head is mainly utilised to complete the process of detection. It creates final output vectors with objectless scores, class probabilities, and bounding boxes by applying anchor boxes to features.

In any deep neural network, the activation function is critical; YOLO v5 employs Sigmoid and Leaky ReLU in our model. The final detection layer employs the sigmoid whereas the middle/hidden layers utilizes the leaky ReLU activation function. Because SGD employs tiny selections and reduces unnecessary and wasteful calculations, our YOLO v5 uses it for training.

3.1.5 Training the model

After applying preprocessing and data augmentations the training data is passed to the model for extracting and learning the features. We trained our modal using Google Colab, a cloud-based Jupyter notebook environment. Most importantly, it is completely self-contained, and the notebooks you create may be modified concurrently by members of your team - exactly as you can with Google Docs projects. A variety of well-known machine learning libraries are available on Colab and may be instantly loaded into your notebook. It provides Nvidia k80 12GBs GPU with the clock speed of 0.82 GHz and 12 GBs of RAM. We used the batch size of 12 and 300 epochs were used to train our model.

3.2 Instance or Pixel Level Segmentation

Pixel level segmentation is the process by which you assign a class to each pixel in an image. Pixel level image segmentation offers a far more granular understanding of an image. The primary use of pixel level image segmentation is to build a computer-vision-based application that delivers a high degree of accuracy. Region based segmentation quickly detects and localizes the presence of weapon from the image in the form of bounding box around the weapon. However, to accurately estimate the area of the weapon in an image or video stream, pixel level segmentation is required. Firstly, we acquired the dataset and labeled the images using VGG Image annotator and then we trained our model using Mask RCNN algorithm.

3.2.1 Acquisition and annotation of Dataset for Mask R-CNN

The same dataset is used for mask R-CNN as well that was used for YOLOv5 in the earlier sections, which contains 2.971 images and was published by the University of Granada. For mask R-CNN we had to label the dataset differently (pixelwise). For this purpose, we used an online tool known as VGG Image Annotator. We uploaded our dataset onto it and then selecting the polyline option for labelling the image pixelwise which was suitable for our model. A snapshot of the VGG Image Annotator is shown in Figure 3.8 below.

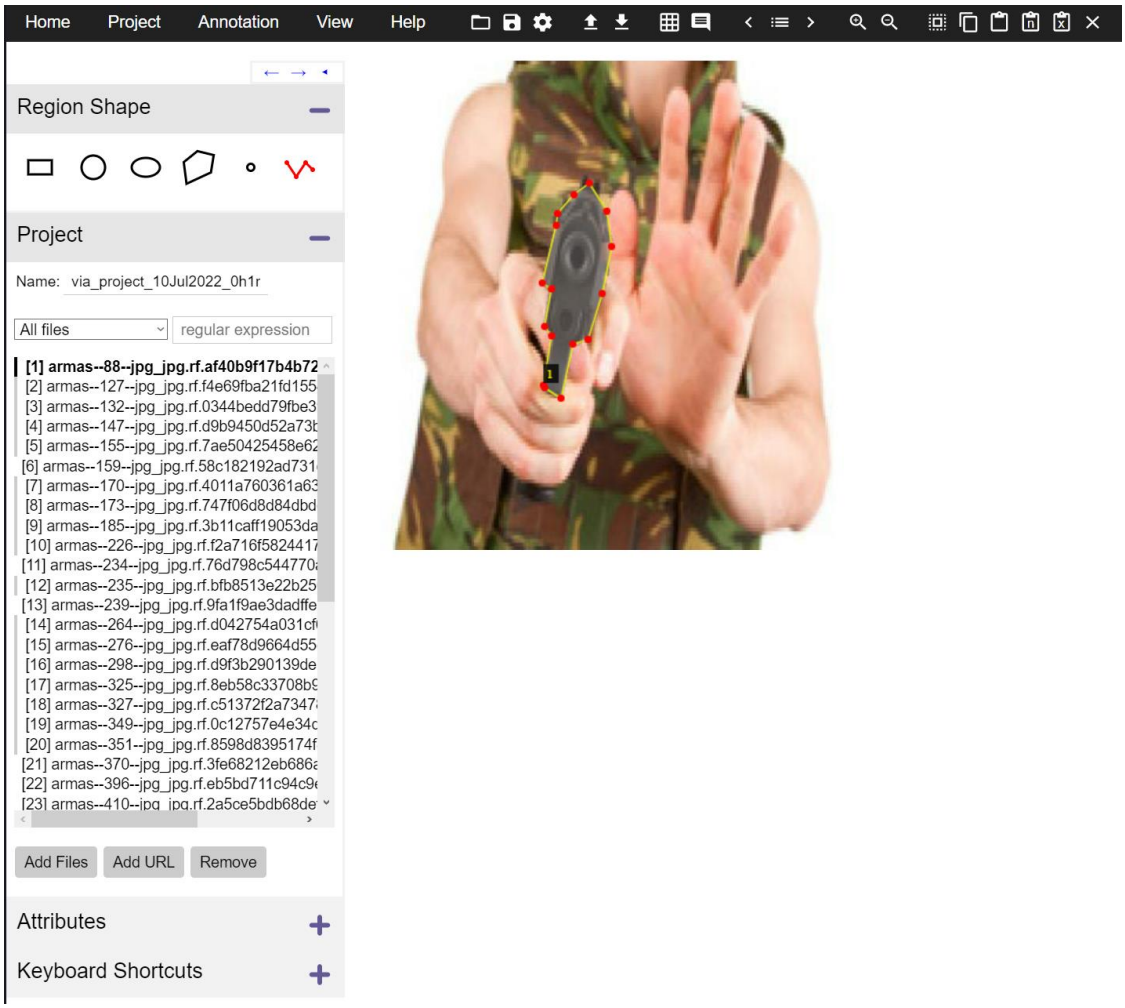

Figure 3.8 Snapshot of VGG Image Annotator

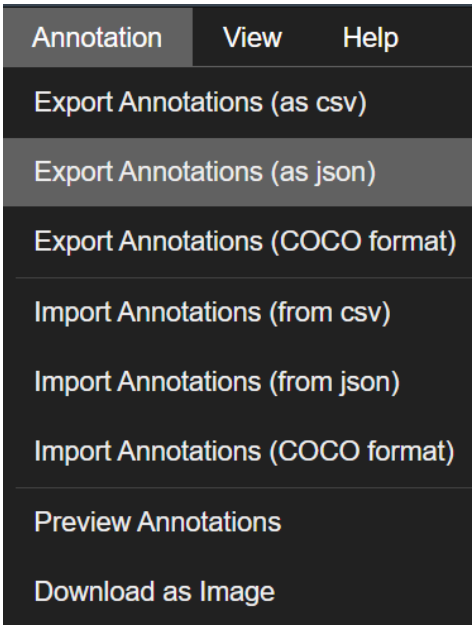

Figure 3.9 Exporting Json file from VGG Image Annotator

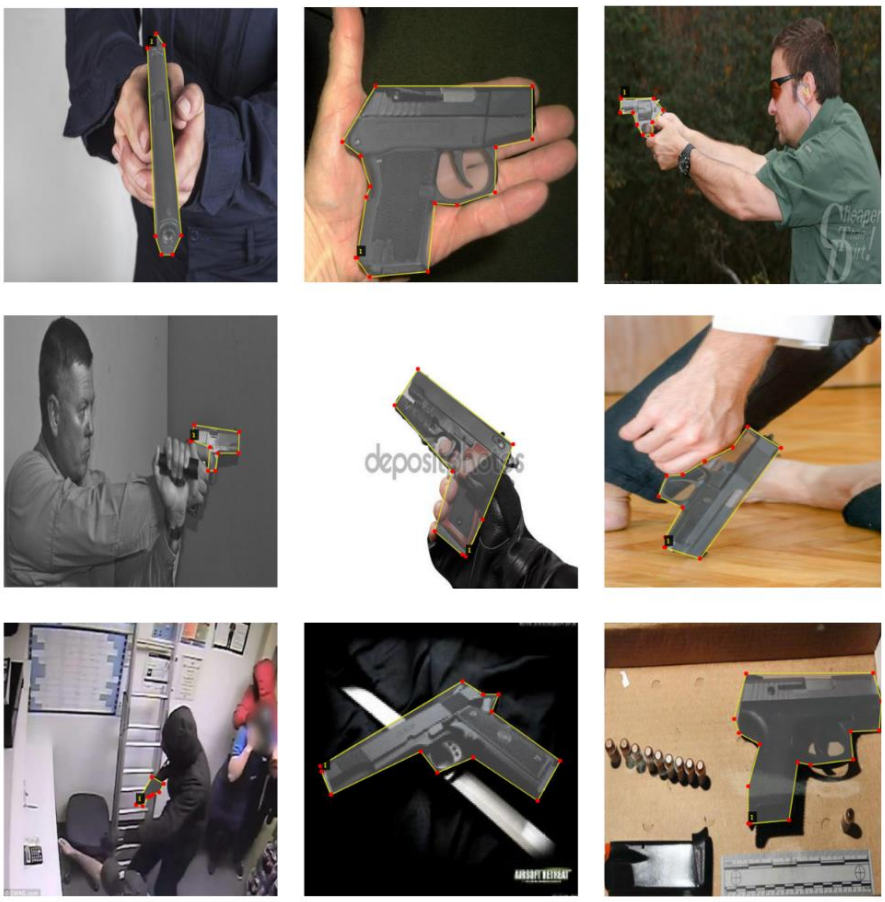

Figure 3.10 Samples of images after annotation

After all the images are annotated, click on “Annotation” from the top menu and select Export Annotations as json. We need the json file to train our model which has all the labels of all the images we annotated. Figure 3.10 shows samples of images after annotation.

3.2.2 Instance segmentation using Mask RCNN

We used the Mask R-CNN algorithm to implement the pixel level segmentation technique. It is the most advanced CNN for pixel level segmentation. This variation of a Deep Neural Network (DNN) recognizes objects in an image and creates a segmentation mask of superior quality for each occurrence. Its artificial neural network is tuned to analyze pixel input and is used for image recognition and processing. Therefore, CNNs are the essential building blocks for the image segmentation problem in computer vision (CNN segmentation). Faster R-CNN has two outputs, a class label and a bounding-box offset, for each candidate object; we add a third branch that produces a mask over the object. However, the additional mask output is separate from the box and class outputs, allowing the extraction of a far more precise spatial arrangement of an object. Mask R-CNN is an expansion of Faster R-CNN that adds a branch for predicting an object mask (Regions of Interest ROI) in combination with the branch for bounding box identification.

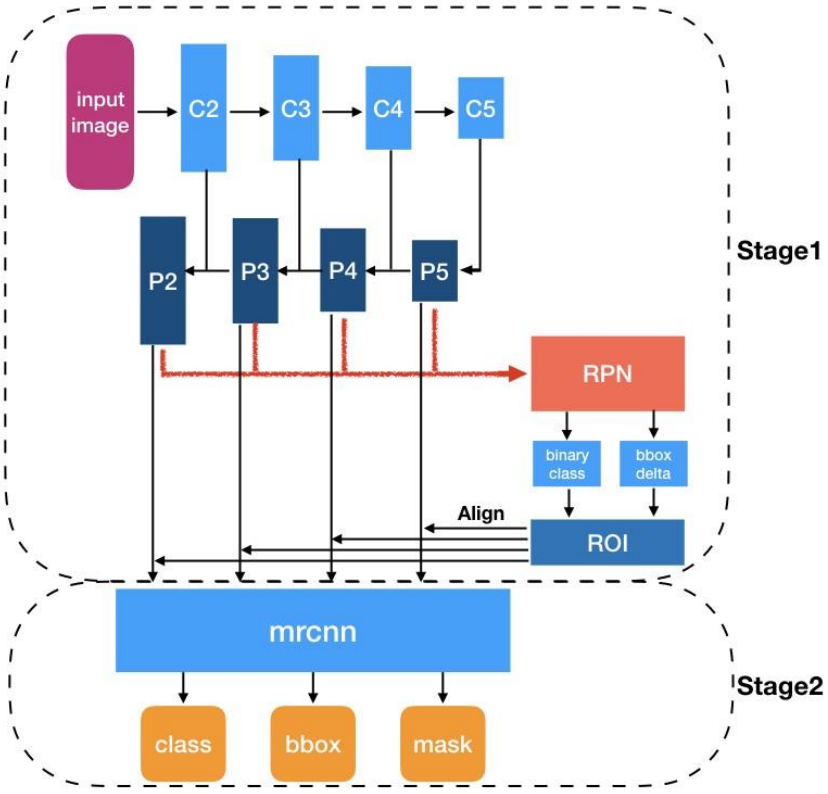

Figure 3.11 Illustration of Mask RCNN structure [54]

Mask R-CNN employs the two-step technique, with the first stage that uses Region Proposal Network (RPN). In addition to predicting the box offset and class in the second stage, Mask R-CNN also produces a binary mask for each RoI. This is in contrast to most modern systems, which rely on mask predictions for classification (e.g. [50] – [52]). Our method adheres to the spirit of Fast R-CNN [47], which utilizes concurrent bounding-box classification and regression (resulting in a simplification of the original RCNN multi-stage process. [53]).

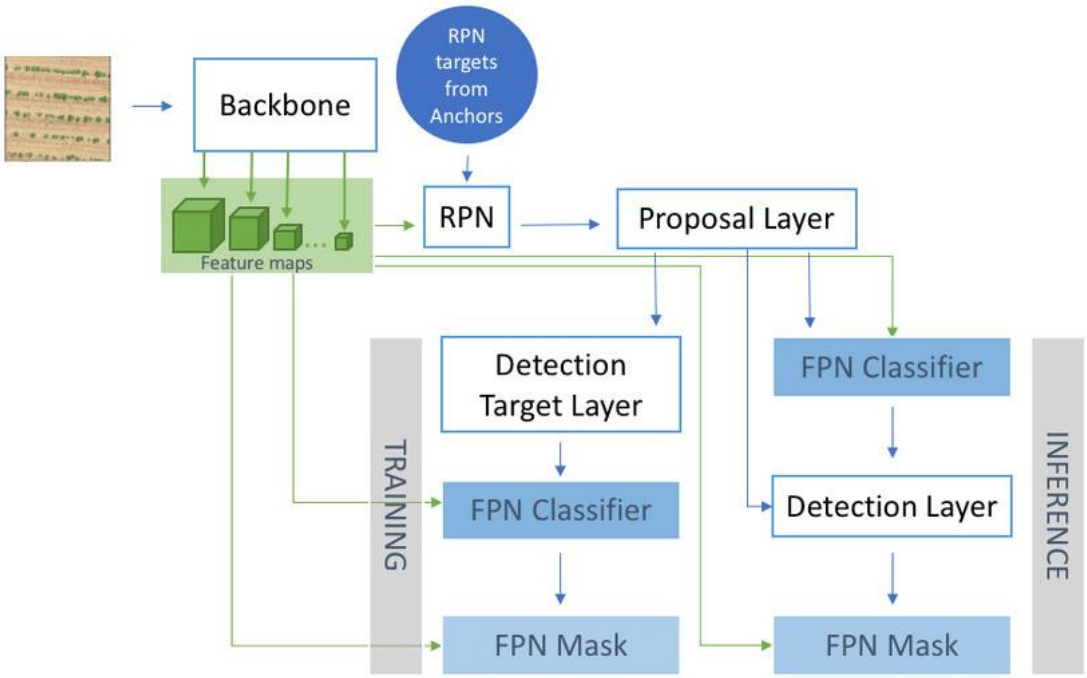

Figure 3.12 Architecture of Mask R-CNN [54]

The RGB scorce image is sent to the Backbone network, which is responsible for feature extraction at varying sizes. This Backbone is a ResNet-101 with prior training [54]. Each block in the ResNet design generates a feature map, with the resultant collection serving as an input to other blocks: the Feature Pyramid Network (FPN), as well as the Region Proposal Network (RPN). The RPN receives additional input in the form of the targets for the RPN, which are formed from the collection of anchor boxes. A collection of ratios is applied to these scales, its basic scales are related to the feature map forms and are customized for each feature map.. Last but not least, anchors are produced for each individual pixel of each feature map. The Proposal Layer is connected to the RPN which consists of a filtering block that retains only relevant recommendations from it. The RPN generates scores for each anchor based on the chance that it would be categorized as negative, positive or neutral and the proposal Layer starts by retaining the highest scores in order to choose the best anchors. Delta coordinates that have been predicted by the RPN are connected to the anchors that have been chosen. Then, the

Non-Maximum Suppression (PNMS) algorithm [55] is used to eliminate anticipated RPN boxes that overlap.

The next step is the Detection Target Layer on the training route, as seen in Figure 3.12. This is not a network layer, but rather an additional filtering step for the Regions of Interest (ROIs) generated by the proposal Layer. Nonetheless, this layer utilizes the ground truth boxes to calculate the overlap with the ROIs and sets them to true if IoU > 0.5 and false otherwise. In the end, these ROIs are resampled and subsampled into a larger collection of ROIs, however the resampling is done randomly such that a ratio of the total to positive ROIs may be determined. As the connection with ground truth boxes is formed in this block and the concept of anchors is eliminated, the output of this layer consists of ROIs with associated ground truth masks, instance classes, and the delta with ground truth boxes for positive ROIs. For elements corresponding to negative ROIs, the ground truth boxes are padded with 0 values. The produced ground truth features that correlate to the supplied ROI features will be used to train the FPN using ground truth. The Classifier and the Mask Graph are the two components that make up the Feature Pyramid Network, often known as FPN. As they are simply a collection of areas with their associated pixel coordinates, the input to these layers will be referred to as Areas of Interest (AOIs). As seen in Figure 3.12, the type of these AOI may change over the training and inference phases. Both variants of the FPN (Mask and Classifier) have the same sequence of blocks, which are comprised of a series of ROIAlign and convolution layers with different objectives. As indicated at the beginning of this section, the ROIAlign method must pool all feature maps from FPN inside the AOIs by discretizing them into a set of predetermined square pooled size bins without creating pixel misalignment, as opposed to standard pooling processes. The output of ROIAlign is a set of uniformly sized squared feature maps that are fed into the convolution layers in batches of size equal to the number of AOIs. The result of these deep layers in the Classifier is a classifier head that includes logits and probabilities for each item in the collection to be an object and belong to a given class, along with refined box coordinates that should be as near as feasible to the ground truth boxes utilised at this stage. In the instance of the Mask network, this step yields a set of square-shaped masks that will be scaled to match the shape in pixels of the corresponding bounding box provided by the classifier.

Detection Layer is a block which is responsible for filtering the predictions produced by the proposal Layer based on the likelihood scores per image and class derived from the FPN classifier graph in inference mode. Low probability-scoring AOIs are eliminated. Finally, only the finest AOIs are chosen to extract their masks using the block FPN Mask Graph. Each of these blocks undergoes training with a loss proportional to its function. Predicting the coordinates of boxes is correlated with a smooth L1-loss, instance classification with categorical cross-entropy loss and binary mask segmentation with binary cross-entropy loss. Utilizing the Adam optimizer [56] to reduce these loss functions. We picked Mask R-CNN over other segmentation methods due to the following benefits:

- Simplicity: Training Mask R-CNN is straightforward.
- Performance: Mask R-CNN exceeds all previous single-model entrants on every challenge.
- Efficacy: The approach is very effective and only adds a little amount of overhead to Faster R-CNN.
- Adaptability: Mask R-CNN is readily applicable to different jobs. In the same framework, it is feasible, for instance, to employ Mask R-CNN for human posture estimation.

Fast or Faster R-CNN is lacking the pixel level alignment, which is the most important component of Mask R-CNN. It utilizes the 2 step approach in which the first stage uses RPN. In addition to predicting the box offset and class in the second stage, Mask R-CNN also produces a binary mask for each RoI. This is in contrast to the majority of modern systems, which rely on mask predictions for classification. In addition, the Faster R-CNN framework makes it simple to create and train a mask R-CNN, which enables a variety of customizable architectural designs. The mask branch imposes a little computing burden, allowing for a quick system and rapid testing.

3.2.3 Training of Mask RCNN

Our Mask RCNN is implemented on the deep learning development framework Tensorflow and Keras. We again used google colab for training of mask R-CNN as it is a free notebook for Artificial Intelligence developers. A variety of well-known machine learning libraries are available on Colab and may be instantly loaded into your notebook. The details of the environment on which the model was trained is shown in Table 3.5 below.

Table 3.5 Details of the environment on which the model was trained

Attribute Name	Attribute Value
CPU	Intel(R) Xeon(R)
CPU Clock Speed	2.30GHz
RAM	12 GBs
GPU	Nvidia K80
GPU Memory	12 GBs
GPU Memory Clock	0.82GHz

CHAPTER 4

EXPERIMENTAL RESULTS AND DISCUSSION

We present the various experiments to show the efficacy of both proposed approaches compared to the competitors and both of these approaches will be evaluated in this chapter. The focus of our research was on both, detecting the weapon and also to apply pixel level segmentation to it. We trained both the models in the same environment i.e. Google Colab which is a free notebook for Artificial Intelligence developers. A variety of well-known machine learning libraries are available on Colab and may be instantly loaded into your notebook. We used the same dataset for both the techniques, the dataset was publicly available dataset which contains 2,971 images which was published by University of Granada. A sample from the dataset is shown in Figure 4.1.

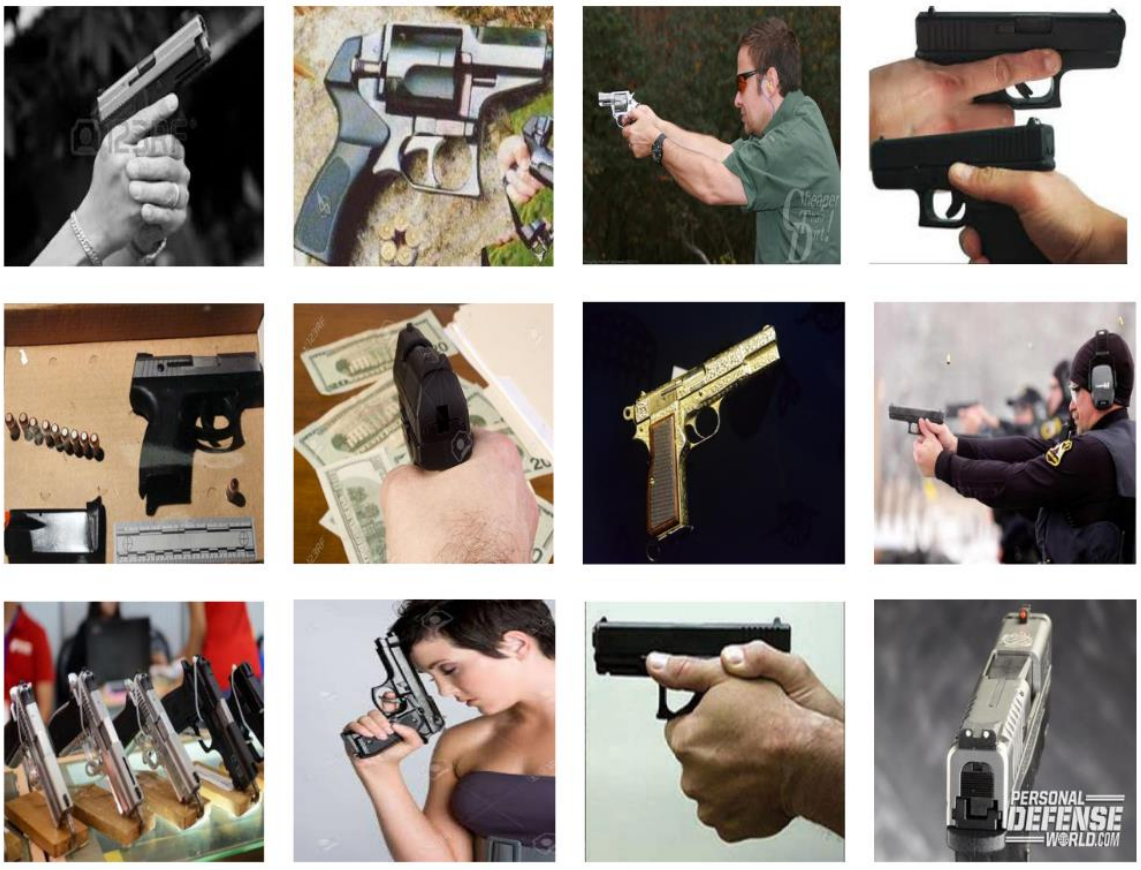

Figure 4.1 Sample images from the dataset

4.1 Performance metrics

We have employed various performance matric techniques to evaluate both of the our proposed approaches. We investigated two parameters i.e. Precision and Recall. As defined below:

$$Precision = \frac{TP}{(TP + FP)} \quad 4.1$$

$$Recall = \frac{TP}{(TP + FN)} \quad 4.2$$

True Negative (TN)

- The actual value was negative, and the model predicted a negative value
- The predicted value matches the actual value

False Negative (FN) – Type 2 error

- The actual value was positive, but the model predicted a negative value
- The predicted value was falsely predicted
- Also known as the Type 2 error

True Positive (TP)

- The actual value was positive, and the model predicted a positive value
- The predicted value matches the actual value

False Positive (FP) – Type 1 error

- The actual value was negative, but the model predicted a positive value
- The predicted value was falsely predicted
- Also known as the Type 1 error

For any test image, if the classifier misses any object, we call it is a False Negative (FN), otherwise True Positive (TP). In case there's no weapon existing but it predicts fire then it's called False Positive (FP). The confusion matrix for region based segmentation is shown in Table 4.1 and the confusion matrix for pixel level segmentation is shown Table 4.2.

Table 4.1 Confusion matrix for region based segmentation

	Actual YES	**Actual NO**
Predicted YES	282	16
Predicted NO	11	0

Table 4.2 Confusion matrix for pixel level segmentation

	Actual YES	**Actual NO**
Predicted YES	262	16
Predicted NO	11	0

4.2 Experimental evaluation of region segmentation technique

To evaluate the experimental result for the region based segmentation approach we calculated the F1-score using the confusion matrix. We determined the F1-score using following equation.

$$F1 - score = \frac{2\,TP}{(2\,TP + FP + FN)} \qquad 4.3$$

After applying above mentioned formulas, we obtain following results as shown in Table 4.3 below, we obtain an average value of precision is 94.63%, the average value of recall is 96.25%, and the F1- Score of 95.43%. This means that the model's performance is excellent. In terms of latency, the YOLO family is superior to other algorithms such as SSD, EfficientDet and Faster RCNN. The YOLO is frequently utilised in real-world applications because to its higher inference speed. The delay of YOLOv5 is 24 milliseconds. The Qualitative results of region based segmentation of weapons based on proposed approach is also presented in Figure 4.2.

Table 4.3 Experimental results of region segmentation approach

Precision	Recall	F1-score
0.9463	0.9625	0.9543

Figure 4.2 Results obtained by our YOLOv5s model

4.2.1 Performance comparison with existing state-of-the-art methods

We performed several other techniques on the same dataset which we used, including YOLOv3, YOLOv4 and Faster RCNN to compare result with the proposed approach. The comparison of the F1-scores of those techniques is shown in Table 4.5. The comparison of F-1 scores obtained from different experiments and concludes that YOLO V5 outperforms over other approaches that include YOLO V3, YOLO V4 and Faster RCNN. The performance of the proposed method was also compared with existing approaches mentioned in the literature review, shown in Table 4.6. The acquired results demonstrate the higher performance of the proposed method. Several approaches used deep learning, while others employed image processing techniques. The suggested model employs a convolutional neural network, which is more efficient and accurate in contrast to earlier models. After developing this model, we can simply use real-time classification outcomes in the real world.

Table 4.5 Comparison of results obtained from different experiments

Algorithm	**F1-score (%)**
YOLOv3	87.8
YOLOv4	87.1
Faster RCNN	88.2
YOLOv5 (Proposed approach)	**95.43**

Table 4.6 Performance comparison of the proposed approach against existing state-of-the-art deep learning based methods

Sr. no	Author	Technique / Classifier	F1-score / Accuracy (%)
1	**Proposed**	**YOLOv5s**	**95.43**
2	Verma et al. [19]	Support Vector Machine SVM Deep Convolutional Network (DCN)	89.9
3	Olmos et al. [18]	Faster R-CNN	91.43
4	Mehta et al. [1]	YOLO V3	90.3
			84.5
5	Hashmi et al [23]	YOLO V3	77
		YOLO V4	82
6	Elmir et al. [25]	CNN	55
		Fast R-CNN	80
		Mobile-SDD	90
7	Warsi et al. [26]	YOLO V3	75
8	Singh et al. [44]	YOLO V4	70
9	Reddy et al. [45]	MC-CNN	94.2
		Faster R-CNN	92.5
10	Hnoohom et al. [46]	EfficientDet-D0	73.81%, 67.42%, 73.56%
		SSD MobileNet-V1	
		Faster R-CNN Inception Resnet-V2	61.08%, 57.51%, 58.17%

As presented in Table 4.6 above our proposed approach achieved the best F1-score of 95.43%. The second and third best F1-score of 94.2 and 92.5 using MC-CNN and Faster R-CNN respectively, was achieved by Reddy et al. [45], followed by Olmos et al. [18] who achieved the F1-score of 91.43 using Faster RCNN. A graphical representation of the comparison of the F1-score of the proposed method against the existing deep learning based approaches mentioned in the literature is shown in Figure 4.3.

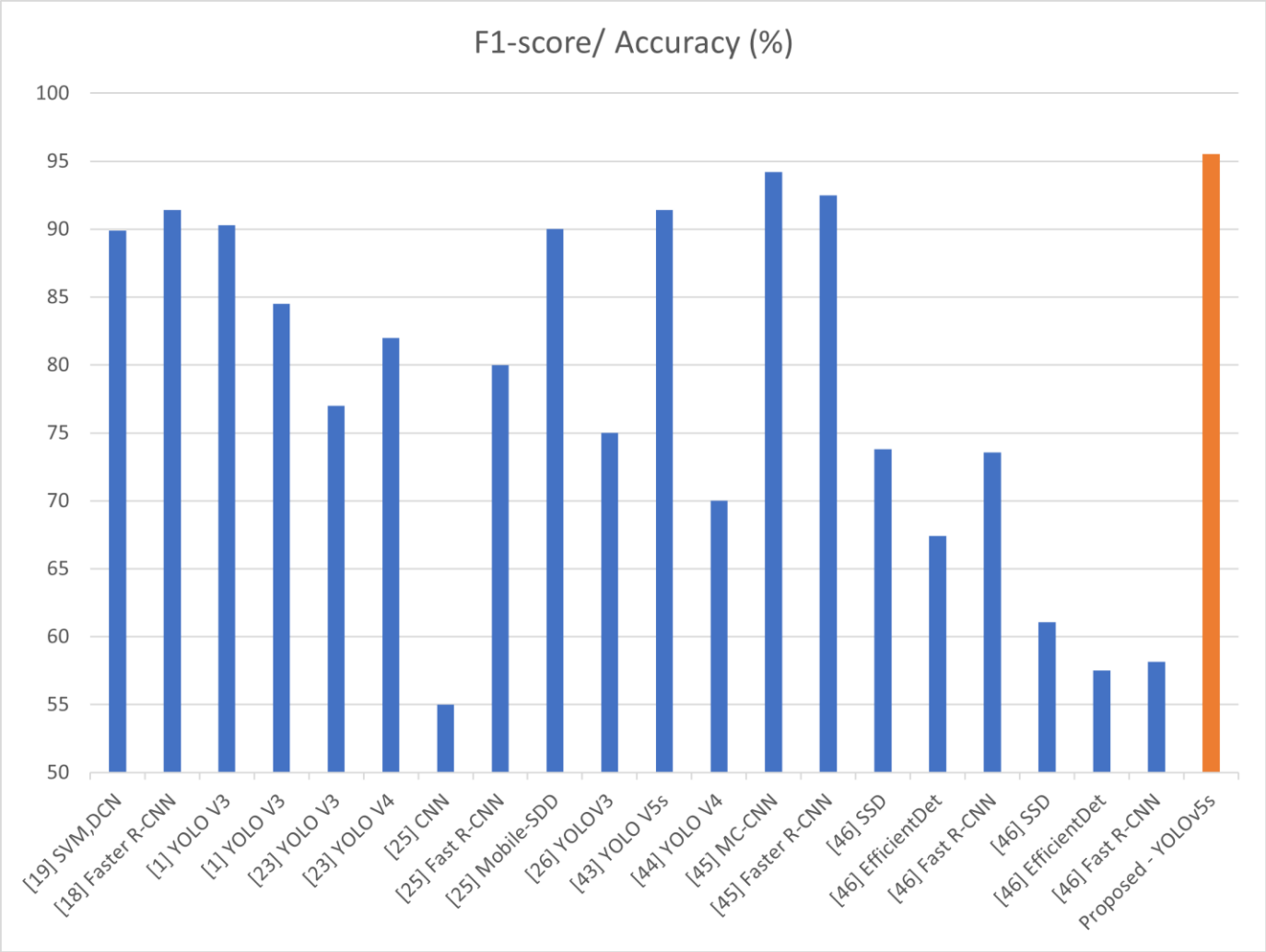

Figure 4.3 F1-score of proposed system against the existing methods

4.3 Experimental evaluation of instance segmentation technique

Our robust Mask R-CNN-based model is capable of properly detecting and segmenting guns impacted by shadowing effects and some degree of firearm overlap . It is a combination of two different approaches, which are referred to as Faster R-CNN and R-CNN. As a measure for segmentation, these findings were determined using the mean intersection over union (mIoU) as a metric. This metric not only considers the correspondence between the ground truth masks and their respective predictions, but also effects the accuracy of the final mask on a pixel-by-pixel basis. Using VGG image annotator, the gathered images for our datasets have been manually annotated with great accuracy. This enables the quality of model predictions to be evaluated by comparing them to image annotations.

We evaluated the performance of our model based on two substandards: the accuracy of location prediction and the accuracy of classification. IoU (Intersection of Union), the ratio between the actual area of interest and the anticipated location area, was used to assess the location prediction accuracy. On the basis of the Confusion Matrix, precision and recall, mAP (mean Average Precision), were used to measure classification accuracy. Precision is the ratio of those that were correctly classed as True to those that were correctly classified as True, while Recall is the ratio of those that the model projected to be True to those that are really True. We will use equation 4.1 and 4.2 to calculate each value. To measure the accuracy of the model we used the following:

$$AC = \frac{TP + TN}{TP + FP + TN + FN} \qquad 4.4$$

True Positive TP refers to a result that indicates that the model accurately predicted the positive class. A True Negative TN is a comparable outcome in which the model forecasts the negative class accurately. A False Positive (FP) is a result when the model incorrectly forecasts the positive class. And a False Negative FN occurs when the model incorrectly predicts the negative class.

The shapes of the areas derived by the RPN may vary. Therefore, we apply a pooling layer and transform each area into a uniform shape. Next, these areas are run through a fully connected network in order to predict the bounding boxes and class label. Up to this point, the stages are almost identical to how the Faster R-CNN algorithm works. The key distinction between the two architectures are as follows. Mask R-CNN generates the segmentation mask in addition to this. To do this, we first calculate the area of interest to decrease the calculation time. We calculate the Intersection over Union (IoU) between all the anticipated areas and the ground truth boxes. We can calculate IoU as follows:

$$IoU = \frac{\text{Area of the intersection}}{\text{Area of the union}} \qquad 4.5$$

Now, in order for us to consider that area a region of interest, the IoU must be either larger than 0.5 or equal to 0.5. Otherwise, we will ignore this area. We carry out this procedure for each of the areas, and thereafter, we narrow our focus to only the sets of regions where the IoU is higher than 0.5. We evaluated the mIoU (Mean Intersection over Union) using following:

$$mIoU = \frac{1}{k+1} \sum_{i=1}^{k} \frac{Pii}{\sum_{j=0}^{k} Pij + \sum_{j=0}^{k} Pji + Pii} \qquad 4.6$$

Where k is the total number of output classes that may be generated by the model, pij is the number of pixels that should have been assigned to category I but were instead assigned to category j. pii represents the number of successfully categorized pixels, whereas pij and pji represent pixels that have been incorrectly classified. Table 4.2 shows the confusion matric of Mask R-CNN. After applying equation 4.1 and 4.2, we obtain following results as shown in Table 4.7 below, we obtain an average value of precision is 94.24%, the average value of recall is 95.97%. Table 4.8 shows the accuracy of detection and as well as the Mean Intersection over union mIoU for the accuracy of the mask. We have achieved the detection accuracy of 90.66% and mIoU of 88.74%. The qualitative results of pixel level segmentation using mask RCNN are shown in Figure 4.4.

Table 4.7 Precision and Recall for Mask R-CNN

Model	Precision	Recall
Mask R-CNN	0.9424	0.9597

Table 4.8 Accuracy and intersection over union of mask R-CNN

Model	Detection Accuracy (AC)	Mean Intersection over Union (mIoU)
Mask R-CNN	90.66 %	88.74 %

Figure 4.4 Results obtained by our Mask R-CNN model

4.4 Discussion

In section 3, we discussed about the 2 architectures, YOLOv5s and Mask R-CNN. We trained both the models separately and evaluated their results. The combined results of proposed techniques are shown in Table 4.9 below.

Table 4.9 Combined results of both proposed techniques

	YOLOv5	**Mask R-CNN**
Precision	94.53 %	94.25 %
Recall	96.25 %	95.97 %
Accuracy	-	90.66 %
F1-Score	95.43 %	-
Mean Intersection over Union	-	88.74 %

As we can see in Table 4.9 above, both the algorithms preformed quite well. YOLOv5 algorithm after evaluating the model, achieved the F1-score of 95.43 %. Whereas Mask R-CNN after evaluation achieved the accuracy of 90.66 % and Mean Intersection over Union (mIoU) of 88.74 %. In section 2 literature review we discussed about the previous studies and techniques used to detect weapons that includes faster R-CNN [19], YOLOv3 [1][23][26], YOLOv4 [23][44], SDD [25][46] and many more. Whereas our proposed methodology outperformed the existing techniques in terms of accuracy. YOLOv5 gives the bounding boxes on the objects detected in the input image, it is one of the fastest and lightest deep learning object detection model. Whereas Mask R-CNN gives pixel level segmentation on the objects detected in the input image. Figure 4.5 below shows the results of both the techniques.

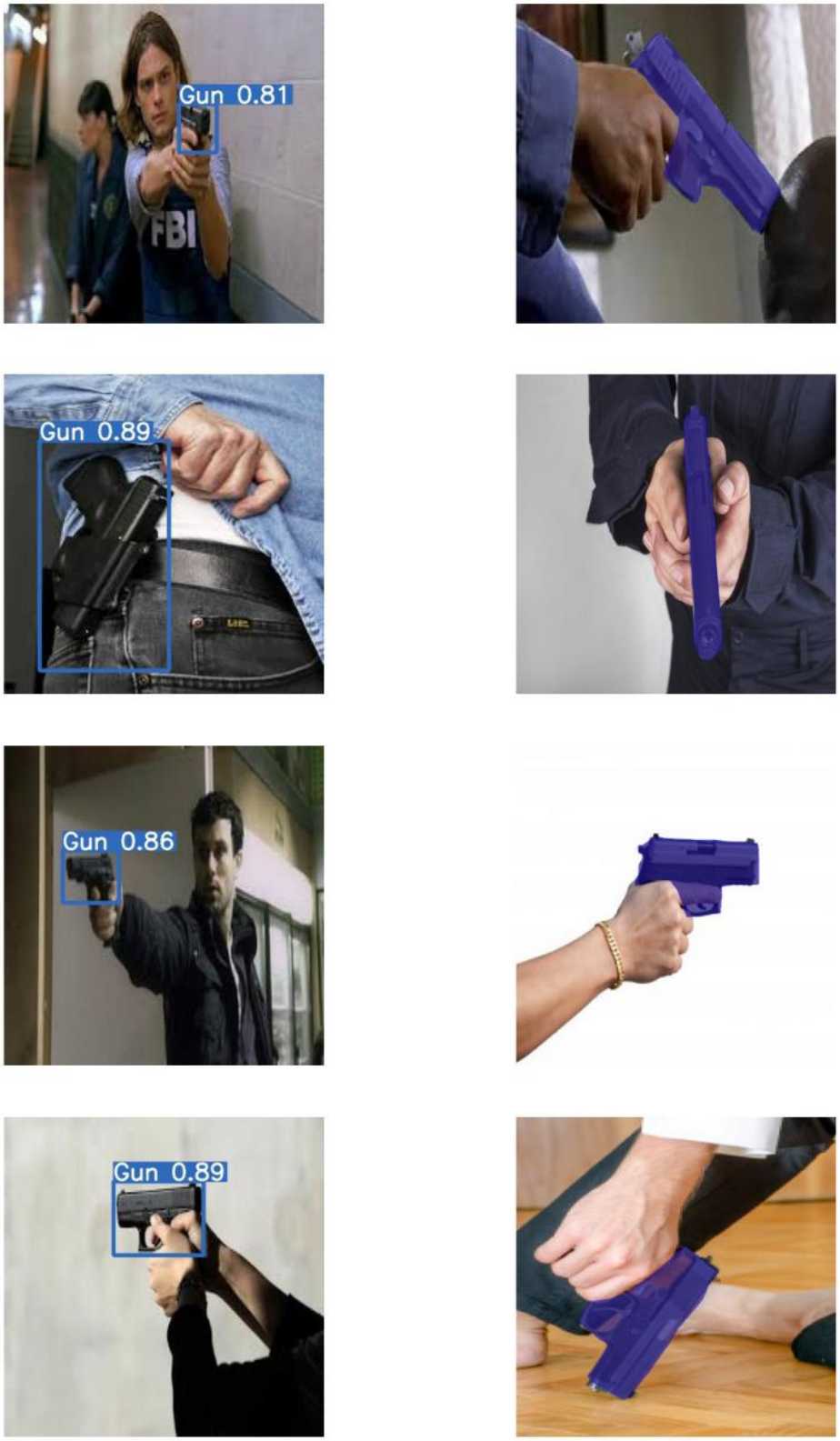

Figure 4.5 Results of both the proposed approaches

CHAPTER 5

CONCLUSION

In this research work, we presented a model for detecting firearms with high speed that may be employed in real-time surveillance situations using alarm-based systems. We made use of the most up-to-date models known as YOLOv5s, which offered highly efficient results and a high rate of speed. We further used the pre-processing and augmentation techniques of cropping, cutout, blurring, rotating, flipping etc to improve our F1-score. Our YOLOv5s model, which was trained on the 2,971 image dataset that was provided by the University of Granada, produced the most encouraging results of all the models we tested. Our model upon evaluating achieved 95.43% of F1-score, 94.53% precision and 96.25% recall. The objective was to minimize the false positive and it is clear from the results that YOLOv5 has a good detection performance even in low quality images and videos. Our proposed methodology outperformed the existing researches so far.

We also applied Mask R-CNN technique to detect and segment the weapons in the images. We demonstrated the architecture of Mask-RCNN model, which comprises of object detection, object localization, and instance segmentation. The same dataset was used to train Mask R-CNN as it was used in YOLOv5. Upon evaluating our model, it achieved the accuracy of 90.66 %, Mean Intersection over Unio (mIoU) 88.74 %, precision of 94.25% and recall of 95.97%. In terms of image segmentation and instance segmentation, the state-of-the-art solution is referred to as Mask R-CNN, which is a Convolutional Neural Network (CNN). A Region-Based Convolutional Neural Network known as Faster R-CNN served as the foundation for the development of Mask R-CNN. Under Mask R-CNN, there are two primary forms of image segmentation: (a) Semantic Segmentation and (b) Instance Segmentation.

We trained both the models in the same environment i.e. Google Colab which is a free notebook for Artificial Intelligence developers. A variety of well-known machine learning libraries are available on Colab, and each of these libraries can be loaded into your notebook quickly. It provides Nvidia k80 12GBs GPU with the clock speed of 0.82 GHz and 12 GBs of RAM. It is a cloud-based Jupyter notebook environment. Most importantly, it is completely

self-contained, and the notebooks you create may be modified concurrently by members of your team - exactly as you can with Google Docs projects.

The purpose of this research is to provide an efficient real-time weapon detection deep learning model with a high accuracy metric. We evaluated several deep learning algorithms and methodologies for the early identification of firearms. According to our research, deep learning algorithms have shown outstanding results in terms of both speed and accuracy, which should assist law enforcement agencies to minimize threats as efficiently as possible. This system may be used in cooperation with an alarm system to detect handguns effectively. This technology will aid police departments in different areas to detect the weapons automatically and respond to it in swift time to prevent harmful incident. With the increasing availability of low-cost storage, video infrastructure, and improved video processing technologies, intelligent surveillance systems would totally replace the present infrastructure. Eventually, inexpensive computers, high-end technology, video infrastructure, and improved video processing will allow robot-based digital monitoring systems to completely replace the present surveillance systems.

REFERENCES

[1] Mehta, P., Kumar, A. and Bhattacharjee, S., 2020, July. Fire and gun violence based anomaly detection system using deep neural networks. In *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)* (pp. 199-204). IEEE.

[2] Tiwari, R.K. and Verma, G.K., 2015. A computer vision based framework for visual gun detection using harris interest point detector. *Procedia Computer Science*, *54*, pp.703-712.

[3] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*, pp.1097-1105.

[4] Zhang, Q., Yang, L.T., Chen, Z. and Li, P., 2018. A survey on deep learning for big data. *Information Fusion*, *42*, pp.146-157.

[5] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. and Berg, A.C., 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, *115*(3), pp.211-252.

[6] Zhao, Z.Q., Zheng, P., Xu, S.T. and Wu, X., 2019. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, *30*(11), pp.3212-3232.

[7] S. S. Mousavi, M. Schukat, E. Howley, Traffic light control using deep policy-gradient and value-function-based reinforcement learning, IET Intelligent Transport Systems 11 (7) (2017) 417–423.

[8] S. S. Mousavi, M. Schukat, E. Howley, Deep reinforcement learning: an overview, in: Proceedings of SAI Intelligent Systems Conference, Springer, 2016, pp. 426–440.

[9] R. Sarcinelli, R. Guidolini, V. B. Cardoso, T. M. Paixao, R. F. Berriel, P. Azevedo, A. F. De Souza, C. Badue, T. Oliveira-Santos, Handling pedestrians in self-driving cars using image tracking and alternative path generation with fren´et frames, Computers & Graphics 84 (2019) 173–184.

[10] S. Mousavi, M. Schukat, E. Howley, A. Borji, N. Mozayani, Learning to predict where to look in interactive environments using deep recurrent learning, arXiv preprint arXiv:1612.05753.

[11] Y. E. Wang, G.-Y. Wei, D. Brooks, Benchmarking TPU, GPU, and CPU platforms for deep learning, arXiv preprint arXiv:1907.10701.

[12] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.

[14] R. Girshick, "Fast r-cnn," in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster rcnn: Towards real-time object detection with region proposal networks," in Advances in Neural Information Processing Systems, 2015, pp. 91–99.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.

[17] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7263–7271

[18] Olmos, R., Tabik, S. and Herrera, F., 2018. Automatic handgun detection alarm in videos using deep learning. *Neurocomputing*, *275*, pp.66-72.

[19] Verma, G.K. and Dhillon, A., 2017, November. A handheld gun detection using faster r-cnn deep learning. In *Proceedings of the 7th International Conference on Computer and Communication Technology* (pp. 84-88).

[20] Grega, M., Matiolański, A., Guzik, P. and Leszczuk, M., 2016. Automated detection of firearms and knives in a CCTV image. *Sensors*, *16*(1), p.47.

[21] Redmon, J. and Farhadi, A., 2018. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

[22] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

[23] Hashmi, T.S.S., Haq, N.U., Fraz, M.M. and Shahzad, M., 2021, May. Application of Deep Learning for Weapons Detection in Surveillance Videos. In *2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2)* (pp. 1-6). IEEE.

[24] Luvizon, D.C., Picard, D. and Tabia, H., 2018. 2d/3d pose estimation and action recognition using multitask deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5137-5146).

[25] Elmir, Y., Laouar, S.A. and Hamdaoui, L., 2019, April. Deep Learning for Automatic Detection of Handguns in Video Sequences. In *JERI*.

[26] Warsi, A., Abdullah, M., Husen, M.N., Yahya, M., Khan, S. and Jawaid, N., 2019, August. Gun detection system using YOLOv3. In *2019 IEEE International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)* (pp. 1-4). IEEE.

[27] Tiwari, Rohit Kumar, and Gyanendra K. Verma. "A computer vision-based framework for visual gun detection using harris interest point detector." Procedia Computer Science 54 (2015): 703-712.

[28] González, J.L.S., Zaccaro, C., Álvarez-García, J.A., Morillo, L.M.S. and Caparrini, F.S., 2020. Real-time gun detection in CCTV: An open problem. *Neural networks*, *132*, pp.297-308.

[29] Akcay, S., Kundegorski, M.E., Willcocks, C.G. and Breckon, T.P., 2018. Using deep convolutional neural network architectures for object classification and detection within x-ray baggage security imagery. *IEEE transactions on information forensics and security*, *13*(9), pp.2203-2215.

[30] Egiazarov, A., Mavroeidis, V., Zennaro, F.M. and Vishi, K., 2019, November. Firearm detection and segmentation using an ensemble of semantic neural networks. In *2019 European Intelligence and Security Informatics Conference (EISIC)* (pp. 70-77). IEEE.

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

[32] Castillo, A., Tabik, S., Pérez, F., Olmos, R. and Herrera, F., 2019. Brightness guided preprocessing for automatic cold steel weapon detection in surveillance videos with deep learning. *Neurocomputing*, *330*, pp.151-161.

[33] Tiwari, R.K. and Verma, G.K., 2015, January. A computer vision based framework for visual gun detection using SURF. In *2015 International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO)* (pp. 1-5). IEEE.

[34] Singleton, M., Taylor, B., Taylor, J. and Liu, Q., 2018, July. Gun identification using tensorflow. In *International Conference on Machine Learning and Intelligent Communications* (pp. 3-12). Springer, Cham.

[35] Lai, J. and Maples, S., 2017. Developing a real-time gun detection classifier. *Course: CS231n, Stanford University*.

[36] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.

[37] R. Girshick, "Fast r-cnn," in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.

[38] S. Ren, K. He, R. Girshick, and J. Sun, "Faster rcnn: Towards real-time object detection with region proposal networks," in Advances in Neural Information Processing Systems, 2015, pp. 91–99.

[39] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.

[40] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7263–7271.

[41] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018.

[42] Li, P. and Zhao, W., 2020. Image fire detection algorithms based on convolutional neural networks. *Case Studies in Thermal Engineering*, *19*, p.100625.

[43] Ashraf, A.H., Imran, M., Qahtani, A.M., Alsufyani, A., Almutiry, O., Mahmood, A. and Habib, M., 2021. Weapons Detection for Security and Video Surveillance Using CNN and YOLO-V5s.

[44] Singh, A., Anand, T., Sharma, S. and Singh, P., 2021, July. IoT Based Weapons Detection System for Surveillance and Security Using YOLOV4. In 2021 6th International Conference on Communication and Electronics Systems (ICCES) (pp. 488-493). IEEE.

[45] Reddy, R., Vallabh, K.G. and Sharan, S., 2021, May. Multiclass Weapon Detection using Multi Contrast Convolutional Neural Networks and Faster Region-Based Convolutional Neural Networks. In 2021 2nd International Conference for Emerging Technology (INCET) (pp. 1-8). IEEE.

[46] Hnoohom, N., Chotivatunyu, P., Maitrichit, N., Sornlertlamvanich, V., Mekruksavanich, S. and Jitpattanakul, A., 2021, November. Weapon Detection Using Faster R-CNN Inception-V2 for a CCTV Surveillance System. In 2021 25th

International Computer Science and Engineering Conference (ICSEC) (pp. 400-405). IEEE.

[47] Girshick, R., 2015. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

[48] Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.

[49] Long, J., Shelhamer, E. and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).

[50] Pinheiro, P.O., Collobert, R. and Dollár, P., 2015. Learning to segment object candidates. Advances in neural information processing systems, 28.

[51] Dai, J., He, K. and Sun, J., 2016. Instance-aware semantic segmentation via multi-task network cascades. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3150-3158).

[52] Li, Y., Qi, H., Dai, J., Ji, X. and Wei, Y., 2017. Fully convolutional instance-aware semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2359-2367).

[53] Girshick, R., Donahue, J., Darrell, T. and Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

[54] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[55] Chen, X. and Gupta, A., 2017. An implementation of faster rcnn with study for region sampling. arXiv preprint arXiv:1702.02138.

[56] Kingma, D.P., 2015. &Ba J.(2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[57] "What the data says about gun deaths in the U.S." *pewresearch.org*, para. 3. [Online]. Available: https://www.pewresearch.org/fact-tank/2022/02/03/what-the-data-says-about-gun-deaths-in-the-u-s/. [Accessed: May. 28, 2022].

[58] "YOLOv5 New Version - Improvements And Evaluation" *roboflow.com*, para. 1. [Online]. Available: https://blog.roboflow.com/yolov5-improvements-and-evaluation/. [Accessed: May. 30, 2022].