National College of Ireland

Firearm detection using Yolov7

MSc Research Project
Data Analytics

Shaik Rizwana
Student ID: 22114611

School of Computing
National College of Ireland

Supervisor: Vikas Tomer

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Shaik Rizwana
Student ID:	22114611
Programme:	Data Analytics **Year:** 2023
Module:	MSc Research Project
Supervisor:	Vikas Tomer
Submission Due Date:	14/12/2023
Project Title:	Firearm detection using YOLOv7
Word Count:	7476
Page Count:	24

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:

Date: 14th December 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

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

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

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

Firearm detection using YOLOv7

Shaik Rizwana
22114611

Abstract

Firearms have been a constant issue and the biggest contributor to disrupting public safety worldwide. This is an important issue that cannot be overlooked. An autonomous visual gun detection model can help provide surveillance and monitoring to all public places. In earlier works, gun detection has always faced problems achieving the appropriate accuracy or speed in real time. A dependable gun detection model will allow a quicker response and propose safety measures. We look into different papers and their work for a robust gun detection model using Yolo algorithms. I utilized the Yolo algorithms with multiscale concatenation and prediction heads for our paper. We train and validate the Yolo variants on a curated gun image dataset acquired from various sources. The Yolo model for gun detection achieved 87% precision and 70% recall, making it a reliable and well-performing model for different images of firearms and their orientations. This detection model approaches the state-of-the-art in the targeted deep neural architectures for security applications. In a real-time scenario, the latest model for gun detection using Yolo enables automated surveillance and alert systems to detect firearm threats faster. The performance of this model is sufficient for the video and embedded application in CCTVs (Closed-circuit Television). The main challenges faced are the scenarios where illumination is not proper and partial visibility of the firearm makes it difficult for the model to detect the object. This has caused a few true negative and false positive scenarios.

Keywords: Yolov8, Convolutional neural network, Real-time detection, Security systems, Computer Vision

1 Introduction

When considered globally, guns and firearms are the major sources of violence-related crimes. A quick and accurate gun detection model has a good application scope and plays an important role in detecting threats at the scene at hand. With the data collected, one can automatically inform the respective public safety bodies like the Irish Garda prior to the crime. Deep learning models involving complex neural networks have shown that reliable real-time detection models can help security agencies respond quickly to a firearm situation. However, the models still need optimized architectures for the targeted domains. A study shows that there are a lot of unaccounted small handguns that have not been registered. According to the Small Arms Survey, there are around 857 million civilian-held firearms across 230 countries and territories worldwide (Karp, 2018). Due to the gun's small nature, hand-held guns or pistols are the easiest to conceal, raising the challenge of detecting it using the Closed Circuit Televisions(CCTV). Because of this, a quick and on-time weapon-based incident prediction is necessary in order to mitigate life-threatening situations when considering public safety. When manually monitoring the CCTVs, it is observed that there is a high chance that the security officer who is monitoring

the live footage will suffer from “video blindness” after 20 to 40 minutes of continuous observation (Velastin *et al.*, 2006).

1.1 Business Understanding

Detecting concealed or small weapons has always been the biggest challenge for the YOLO algorithms. One of the many difficulties is image or video quality. The image detection mainly depends on the quality of the image the model is being trained with. Few other problems, being addressed are the localization of data the feature of YOLO, which helps the model to train at a high speed instead of carefully weaving through them. There are many advantages of an autonomous weapon detection system. Let us look at the ones being highlighted in this paper,

Advanced Surveillance - At night, security professionals can better assess their surroundings with the help of automatic weapons detection. This will lessen the need for human oversight of the CCTV footage and alleviate the stress of constantly being in the same spot to ensure perfection.

Improved Safety – In times of trouble or peril, the security officers have the ability to notify the nearest garda stations for assistance promptly. This will allow us to take action in response to the provided circumstances.

This paper will help us understand how a model works when trained with small gun images and concealed weapons. The related papers in the literature review section also face the similar issue of detecting small weapons sometimes concealed under garments or bags. One of the major challenges this study has faced is the availability of computational power and resources (Diwan *et al.*, 2023). The size of the dataset and the requirement of extensive annotations for the images make this study a resource-intensive project to implement.
This study employs a series of procedures to obtain the answers to the research inquiries. The initial stage involves gathering the dataset and any additional customized datasets required for training the model. Two distinct datasets have been identified and will be utilized. The total number of photos is 8009. Following the selection of the datasets, a manual process was undertaken to ensure the accuracy of the labels. This involved choosing the top 20% of photos and verifying the bounding boxes. One of the datasets had a size of 416 x 416 pixels, which was enlarged to 640 x 640 pixels to accommodate the current YOLO model being utilized. Upon implementing the model, the decision rests with it.
If the gun is spotted under any circumstances, it is imperative to notify the nearest authorities promptly to implement preventive measures. The same is represented in the flow chart below.

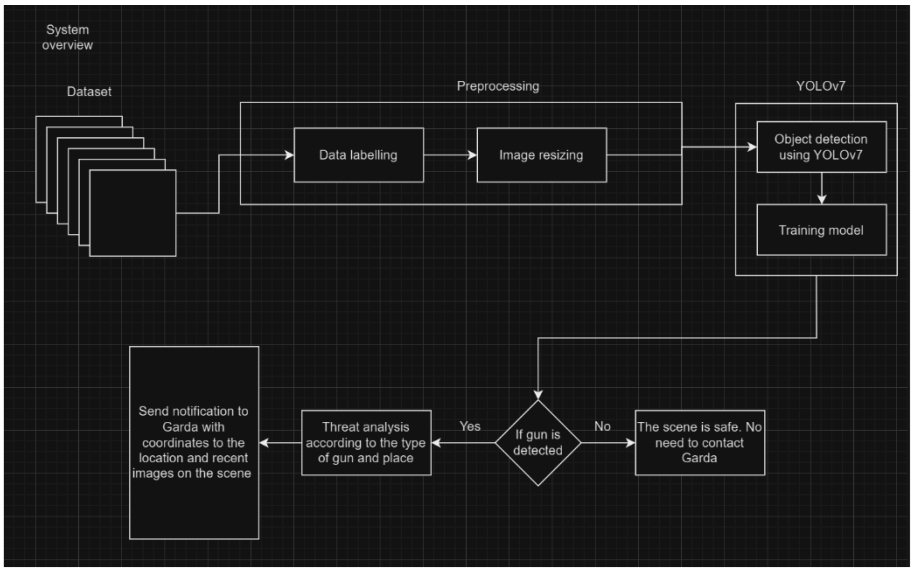

Figure 1: System design for this paper

In order to test and challenge the research questions, this paper will use YOLOV7 with two different datasets to see how a model works when trained with a custom object detection dataset. This is a comparative study on how YOLOV7 behaves on different datasets.

1.2 Research Question

Q1. To what extent are the most recent deep convolutional neural networks appropriate for precise and quick gun identification to facilitate automatic security alerts?

Q2. A comparative analysis using YoloV7 with two different datasets and trying to understand the performance metrics.

1.3 Objectives of the research question

1. To evaluate multiple state-of-the-art object detection models on two gun-based datasets.
2. Select and optimize the appropriate Yolo models with appropriate loss functions, training protocols, and test benchmarks.
3. To analyze performance metrics like IOU, accuracy, precision, recall rates, etc.
4. To provide insights on the model performance and compare the models.

1.4 Limitations

This study is limited to the RGB images of guns. Further studies can be performed on gun detection in infrared or low-light scenarios. Detection of partially visible objects is challenging and can be improved by training on extensive datasets.

1.5 Outline of the report

The paper is organized into the model selection process, proposed approach, experiments, results, and conclusions. We first provide the necessary background on deep neural networks for object detection.

2 Related Work

Guns have been the main source of terror in the daily lives of common people. Due to certain norms and regulations regarding the licensing or criteria of owning a gun, incidents like mass shootings, robberies, etc, are happening in society. This paper focuses on detecting Firearms through surveillance cameras, which can help alert the authorities on time-based on the threat level.

My main motivation to take up this project has been the recent emerging news of juveniles carrying guns to public places. Security issues have always faced issues when it comes to gun-related scenarios. With this paper, I have tried to re-create the work that has been done already using Yolov7 but also investigated the perspective of how Yolov7 models trained with custom datasets perform when it comes to transfer learning. I have used publicly available datasets by adhering to the ethics policy as they involve people holding guns and tried to create models based on transfer learning. The first two models were trained with custom datasets, and then I tried picking up one of the models that performed well to see how well it would detect guns from another dataset. Using the old weights it was trained on, it struggled to detect guns but worked on pictures it was never trained on. It was like showing a newborn baby what a gun looks like and then asking to identify different models without ever showing them beforehand. The results were not satisfactory as the dataset size was tiny. Given that this is an academic project with limited resources, if there is access to better resources in the future, we can look into training the models on bigger datasets and compare the performances. Let us now look at different papers that dealt with similar research questions and understand the limitations and research gaps which can help us answer our research questions -

(Chandan G *et al.*, 2018), this paper focuses on tracking and object detection using deep learning techniques like Mobile Nets, Single Shot Detectors (SSD), and OpenCV. An early paper from 2018 showed that different objects like trains, dogs, bicycles, and buses were detected with an accuracy ranging from 99.49% to 99.99%. This paves pathways to advance concepts in CNN like Faster-RCNN, YOLO, VGG16 etc. However, this accuracy is subject to how clear the picture is with solid-colored backdrops.

(Warsi *et al.*, 2019), with the custom-created dataset of different gun images, videos, and images from ImageNet, this paper has mainly used YOLOV3 as an alternative technology for faster RCNN. YOLOV3 was the first state-of-the-art algorithm in the YOLO series. Their model showed better results in 2 out of 4 videos used. This paper tried to tackle the issue with the backdrops by using videos and processing them into image frames. The model was quick to detect the weapon but could not reach high levels of accuracy.

(Fan *et al.*, 2021) This paper discusses how we can use YOLOV2 and R-CNN as an alternative to state-of-the-art algorithms in 2021. As we go further into the paper, it discusses how Faster R-CNN can be used in the first phase of feature detection and YOLOV2 as the final function to detect the object defining the bounding boxes. An algorithm named Kalman filter is used to deal with the low accuracy delivered by YOLOV2. Hence, a 78% accuracy was achieved when all these algorithms were used together. It still faced an issue with blurred images and vivid backdrops.

(Narejo *et al.*, 2021),This paper shows a comparative study between traditional CNN, YOLOV2, YOLOV3. As the latest algorithm among the three, YOLOV3 yields an accuracy of 98.89%. Instead of using the default Python language, the authors have used Java as the foundational language to implement these models. Even though the Java-based model yields almost 99% accuracy, the upcoming versions promise to perform better and faster due to the rise of updates in the YOLO models. The authors faced issues with detecting smaller objects and low-resolution pictures. This gave rise to the high localization errors with YOLOV3.

(Hashmi *et al.*, 2021), using a massive dataset with 7800 images, performs a comparative study between YOLOV3 and YOLOV4 to understand which algorithm works better with the same set of datasets. As YOLOV4 is an improved version that arose from YOLOV3, it is evident that YOLOv4 performs better. Compared to YOLOV3, YOLOV4 showed 85% of accuracy. It highlights the architectural changes and how YOLOV4 evolved using the DarkNet CP3 framework for feature detection.

(Dextre *et al.*, 2021) This paper deals with the usage of YOLOV5 to detect guns automatically through surveillance systems. Instead of any normal processor, the researchers have selected a specific System on Chip(SOC) produced by NVidia called Jetson AGX Xavier embedded in the Closed-Circuit Television(CCTV) cameras present on the premises. As YOLO is extremely fast in object detection, it performs well with an accuracy of 98% prediction. The limitation of the model is that it will face difficulty in identifying the images if they are diminutive.

(Wang *et al.*, 2022), with a dataset of 10231 images from three different sources custom-made and publicly available images, the model used in this paper is YOLOV4. The model has yielded an mAP of 81.75% and a precision of 84%. As this experiment is on CCTV-extracted images, the objects are small and have less resolution. Because of this, it is a challenging task for the model to detect the object. Also, due to the high number of false positives detected by the model, around 367, the model still has a scope of improvement.

(Ashraf et al., 2022) When this paper is compared to other papers on the top, it is quite unique as it derives a comparative study between different algorithms for object detection and YOLOV5. As YOLO evolved, version 5 added a loss calculation function and algorithm to analyze blurred and low-resolution pictures from images and videos. This has helped to study how YOLO will perform with low-resolution pictures and blurred images, unlike all the papers on the top. This has resulted in an accuracy of 92% with the image dataset and 71% with the video dataset. The high number of false positives and recall rates can make the model fail to identify a few images.

(Bota-Ioana and Gavrilas, 2023) This paper trained a YOLOV7 model on various weapons, not just guns. The main motive of this paper was to use YOLOv7 and create a Graphical Use Interface (GUI) that anyone could use to check the predictions provided by the model. The paper was able to achieve a precision of 53%. To tackle YOLOV5's high recall rates, YOLOV7 has introduced an improvement in its feature selection phase. However, the main limitation of YOLO remains the same. Its precision and accuracy will drop as the pictures and frames blur. YOLOV7 has tried to reduce the dependency on the blurriness of the image, and with this paper, we can confirm that the precision has dropped from 92% to 66% when extreme blurriness is implemented.

(Pullakandam *et al.*, 2023), This paper's unique approach is quantizing the model's values to check if the model performs faster with better accuracy. As we go through the paper, we can see that YOLOV8 with quantized values performs faster than YOLOV5 and YOLOV8 without quantized values. Performance-wise, its accuracy and other parameters are performing less when compared to these models. The limitation lies in the trials of quantization. Due to quantizing the values, the weights on the algorithms' nodes are not well defined, resulting in poor performances.

(Dugyala *et al.*, 2023), This paper has proposed a novel algorithm to detect partially hidden or concealed weapons. It is known as Pixel-Level Semantic Feature Fusion-Deep Convolutional Network (PLSF-DCNN) with YOLOV8, the latest version in the YOLO series. This algorithm helps us to tackle all the limitations we have seen in the above papers, like blurred images, cropped images, or camera angles, processing less frame-rate videos, etc. We can still observe that the recall rates are quite high and the high number of false positives and recall rates can make the model fail to identify a few images.

(Fathima Safa and Suguna, 2023), This paper uses YOLOV8 and Mediapipe pose, an ML solution that defines the skeleton of a person in an image to understand how they are holding a gun and at what angle. They were successful in getting a precision of almost 94% with the model. In addition to this, they also calculated the threat level by measuring the body posture and angle of the shoulder that help the gun. The limitation of their work lies in the type of guns the model is trained on. It is limited to actual guns and not toy guns. This might raise a false alarm by the model.

(Ashish Ranjan *et al.*, 2023), This book is on different papers from ICDMAI, 2023. One of the papers called "YOLO Algorithms for Real-Time Fire Detection" focuses on the use of YOLO to detect fire instead of the firearm using a custom-built dataset containing 900 images. This opens the boundaries to not just firearms but incidents happening due to explosives like Molotov cocktails, etc. The model performs with a mAP of 74%. The different versions used for the comparative study are YOLOv3, YOLOV4 tiny, YOLOV4, YOLOV4-csp, YOLOV5 and YOLOR. YOLOV5 performed best when it came to precision, getting 81%, and YOLOR performed well at mAP with 74%. With all the outputs, we can understand that YOLOV5 has performed better than all the other models as a part of the comparative study. If there is a case of smoke that can hide or mask the fire, the model will face difficulties in detecting such cases.

(Uganya *et al.*, 2023), This paper uses tiny YOLO with a dataset of 60 robbery videos extracted from various sources and another dataset with 8327 images. After using Faster-Region-Based-CNN for the pre-processing, it has used tiny YOLO, a small and fast-paced model, to go through the train and test datasets. This yielded 96% of precision, 89% of recall, and mAP@0.50 at 92.33%. The paper aims to reduce the false positives and negatives as it can lead to wrong classifications of guns.

Let us look at the metrics and limitations of the above-discussed papers –

Papers (Year - Author)	Datasets used - size	Model Used	Results - Metrics used	Value	Limitations

(Chandan G *et al.*, 2018)	ImageNet dataset	SSD - Single shot detector	Precision	0.98	The precision obtained is contingent upon the presence of a uniform, solid-colored backdrop.
(Warsi *et al.*, 2019)	ImageNet Gun dataset - Videos	YOLOV3	mAP@0.50	0.81	While the model detected the weapon quickly, it struggled to achieve high accuracy in most cases.
			Precision	0.96	
			Recall	0.62	
			F1-score	0.75	
(Fan *et al.*, 2021)	ImageNet dataset	Faster R-CNN	IOU	0.756	The current technology struggles to identify weapons in foggy or complex images.
		YOLOV3	IOU	0.74	
(Narejo *et al.*, 2021)	ImageNet dataset	CNN	Accuracy	0.95	Diminutive objects and images with poor resolution have resulted in the occurrence of significant localization problems while employing YOLOV3.
		YOLOV2	Accuracy	0.967	
		YOLOV3	Accuracy	0.989	
(Hashmi *et al.*, 2021)	dataset - 7800 images	YOLOV3	mAP@0.50	0.77	The increased recall rate of YOLOV4 compared to YOLOV3 may result in reduced accuracy when applied to tiny and fuzzy pictures.
			Precision	0.84	
			Recall	0.71	
			F1-score	0.77	
		YOLOV4	mAP@0.50	0.84	
			Precision	0.85	
			Recall	0.78	
			F1-score	0.82	
(Dextre *et al.*, 2021)	Dataset - 7000	YOLOV5	mAP@0.50	0.987	One potential constraint of the model is its ability to detect pictures of a much-reduced size accurately.
			Precision	0.988	
(Ashraf *et al.*, 2022)	Image dataset	CNN and YOLOV5	Precision	0.995	The model's performance may be compromised due to a significant occurrence of false positives and a high recall rate, resulting in the failure to identify some photos accurately.
			Recall	0.84	
			F1-score	0.914	
(Bota-Ioana and Gavrilas, 2023)	Image dataset - 4062	YOLOV7	Precision	0.92	YOLOV7 attempted to reduce photo blurriness. This study found that extreme blurriness lowers accuracy from 92% to 66%.
(Dugyala *et al.*, 2023)	Image dataset	PELSF-DCNN + YOLOV8	mAP@0.50	0.954	False positives and a high recall rate may undermine
			Precision	0.968	
			Recall	0.942	

			F1-score	0.955	the model's ability to identify some photographs.
(Pullakandam *et al.*, 2023)	Image dataset	YOLOV5	mAP@0.50	0.891	This limitation is inherent to quantization. Quantizing the data causes algorithms to have undefined weights on their nodes, resulting in poor performance.
			Precision	0.924	
			Recall	0.842	
		YOLOV8	mAP@0.50	0.901	
			Precision	0.926	
			Recall	0.814	
(Fathima Safa and Suguna, 2023)	Image dataset	YOLOV8	mAP@0.50	0.909	Weapons used to train the model limit research. This restriction covers real firearms, not imitations. This caused a misleading model warning.
			Precision	0.939	
			Recall	0.869	
(Ashish Ranjan *et al.*, 2023)	DIAT-fireDS dataset (900 images)	YOLOV3	mAP@0.50	0.59	Smoke can obscure fires, making it difficult for the detection model to identify and recognize them effectively.
			Precision	0.73	
			Recall	0.48	
			F1-score	0.58	
		YOLOV4	mAP@0.50	0.73	
			Precision	0.78	
			Recall	0.73	
			F1-score	0.75	
		YOLOV5	mAP@0.50	0.65	
			Precision	0.81	
			Recall	0.7	
			F1-score	0.74	
		YOLOR	mAP@0.50	0.74	
			Precision	0.72	
			Recall	0.75	
			F1-score	0.73	
(Uganya et al., 2023)	8327 images and 60 videos	tiny YOLO	mAP@0.50	0.923	The paper tries to eliminate false positives and negatives, which can lead to gun classification errors.
			Precision	0.96	
			Recall	0.89	
			F1-score	0.92	
(Wang et al., 2022)	10231 images	YOLOV4	mAP@0.50	0.818	The model has the need for improvement due to its 367 false positives.
			Precision	0.84	
			Recall	0.77	
			F1-score	0.8	

Table 1: A succinct table for comprehending the metrics and constraints of the utilized models.

An extensive review of the works over the years has led us to understand several challenges Yolo algorithms face. Few include precision and false positives, hardware optimizations, image quality, etc. As we look through Yolov-1 to Yolov-8, it is not just the accuracy scores that depict how well the model works. Different metrics like precision score at 50% and 95% and reduction in false positives can be used to show how well a model performs. If the image quality increases, there are fewer chances of getting localization errors. The major problems were

detecting objects that were small in the image and the quality of the image. It is still a challenge with the latest model, yolov8.

Apart from accuracy, this paper has also investigated mAP(Mean Accuracy Precision), false positives, and recall rates of the models created using different datasets. Accuracy alone cannot determine how well the Yolov7 models are performing; mislabeling the objects can lead to severe issues with security.

3 Research Methodology

The proposed methodology involves applying the YOLOV7 model with a normalization algorithm to understand how the model performs regarding small object detection. Let us look at the architecture design on the YOLOV7 algorithm and try to understand how it works.

3.1 Architecture of Yolov7

Let us understand how YOLOV7 works. YOLOV7 (You Only Look Once Version 7) is a state-of-the-art convolutional neural network model for real-time object detection in images and videos. This section provides an in-depth overview of the YOLOV7 architecture, key components, and object detection pipeline to illustrate how it achieves high accuracy and speed for detecting objects in a wide range of applications.

The YOLOV7 model builds on prior YOLO architectures but introduces new design improvements for better generalization capabilities, accuracy on small objects, and overall speed. The core of the architecture utilizes a fully convolutional neural network for object detection. The key components are:

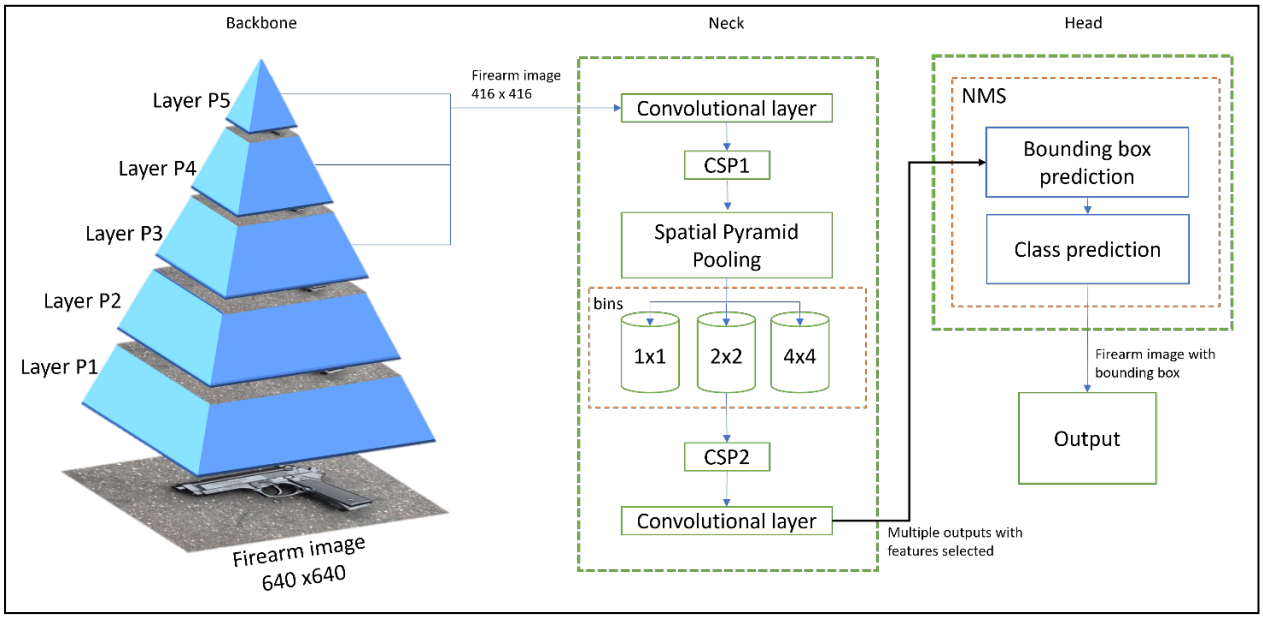

Figure 2: Architecture of Yolov7

Backbone –

1. *Input image*: The raw input image is first resized to a standard size of 640x640 pixels. This normalizes all images to a fixed size.

2. *Initial layers*: The resized image passes through some initial convolutional and downsample layers. These extract low-level features like edges, colors, gradients, etc.
3. *Residual blocks*: The backbone uses a series of residual blocks with skip connections. This allows gradients to flow better during training. Each block applies convolutions to extract higher-level features.
4. *Downsampling*: Downsampling layers are applied between some residual blocks to reduce the spatial resolution and progressively increase the receptive field. This encodes a more semantic, global context.
5. *Backbone output*: After passing through 53 convolutional layers, the backbone outputs feature maps from c1, c2, c3, c4, c5 stages. These encode visual features at different scales.
6. *Neck input*: The multi-scale feature maps are fed into the neck module to process further and prepare for detection.

The CSPDarknet53 backbone applies convolutions and downsampling to extract semantic features from the input image at multiple scales. This is done in stages via the residual blocks. The resulting feature representations encode the visual context needed to detect objects in the subsequent stages.

Neck –

The role of the neck in YOLOV7 is to enrich the backbone features before they are fed into the detection head for making predictions. The YOLOV7 neck uses a CSP-SPP module. Let's break it down:

CSP - Cross Stage Partial connections - This takes the backbone output from different stages and concatenates them together. So you get multiscale features from the c3, c4, and c5 stages.

SPP - Spatial Pyramid Pooling - This takes an input tensor and pools it into different bin sizes using max pooling. For example, pooling into 1x1 bins, 2x2 bins, 4x4 bins, etc. This allows objects of different scales to be detected.

The CSP-SPP neck has a few convolutional and CSP layers:

- Conv layer - Reduces channel dimensions.
- CSP1 - Cross stage partial layer.
- SPP - Spatial pyramid pooling into different bin sizes.
- CSP2 - Another CSP layer to combine scales.
- Conv - Reduce channels again.

The CSP connections concatenate features from different backbone stages. This gives multiscale information. The SPP layer pools these features into different bin sizes to output features tuned to different scales. Additional CSP and convolutional layers integrate the multiscale features.
This enriched set of neck features is then fed to the detection head for predicting objects of different shapes, sizes, and contexts.

Head –

The role of the YOLOV7 head is to convert the feature maps from the backbone and neck into the desired output predictions for detection. It has three separate branches, each with a few layers:

1. *Bounding Box Regression* – This branch predicts the 4 bounding box offsets relative to grid cells. Having 3 heads with upsample layers allows for predicting boxes at different scales thanks to getting input features from different backbone stages.
2. *Object Confidence* – This branch predicts the confidence level that an object exists in each grid cell. The upsample allows it to match spatial sizes across the different input scales for confidence prediction.
3. *Class Probabilities* – This branch predicts the probability distribution over all the classes per grid cell. As we need a high spatial context for classification, it only uses the high-resolution backbone features. Each detection head branch uses convolutional layers to transform features into the desired spatial predictions. The upsample layers expand the feature maps as needed.

The head uses feature maps from multiple layers for predictions at different scales. These three heads allow YOLOV7 to directly predict objects and class probabilities from full images in a single evaluation.

3.2 Understanding the Dataset

For this research paper, we are going to use two different datasets. One of the datasets is a Common Object in Context (COCO) benchmark dataset provided by Roboflow. The other is a custom dataset created from various sources and publicly available.

1. *Dataset 1*(Olmos *et al.*, 2018)-
    a. Size – 3087 images
    b. Image size – 416 x 416 pixels

    As the COCO provides this dataset, there was no issue with missing labels and corrupted images. All the 3087 images were imported onto the Roboflow platform to test the YOLOV7 model provided by the ultralytics.

2. *Dataset 2*(Wang *et al.*, 2022) –
    a. Size – 4922 images
    b. Image size – 640 x 640 pixels

    This dataset belonged to an old YOLOv3 model. The labels with bounding boxes were in XML format. The latest YOLOV7 model does not support XML format. It supports a simple text format describing the class, 0, and the dimensions of bounding boxes with the anchor free coordinates to values of the image's center in x and y values.

    We have the following components in the label files –

    1. The class of the object is represented numerically.
    2. The values of bounding boxes are x_center, y_center, width, and height.

    This dataset was challenging to use as the labels had an outdated format in XML files. I designed an algorithm to convert that into the latest text format suitable for YOLOV7.

This involved normalizing the bounding box values to match the format and parsing through XML files to extract the related data and save it as another text file with the same name.

We are going to train, test, and validate YOLOV7 with these two datasets to compare the results. Preprocessing methods for an image dataset may not vary. However, this plays a key function to get the desired outcomes. For the train, validation, and test split, I used the Roboflow portal. The data split is - train: validate test – 70%:10%:20%. This is randomly done by the Roboflow(Lin *et al.*, 2022).

4 Design Specification

All YOLO models face issues with the detection of small objects. In order to avoid this, steps have been taken, such as normalizing the bounding boxes and scaling the image to 640 x 640 pixels. Let us now look at the gun detection algorithm designed.

Here is a mathematical representation of the above-designed algorithm –

Input: Image I of size 640 x 640 x 3
Output: Bounding boxes B of detected guns

1. **Preprocess** I:
   - $I_{norm} = I / 255$
   - Normalizes pixels from 0-255 to 0-1

2. **Forward Pass** through YOLOV7:
   - Model Y consists of backbone φ, neck ν, and detection head δ
   - Detection head has branches:
     δ_{boxes} for boxes
     δ_{conf} for confidence
     δ_{class} for classes

3. **Get Predictions**:
$P_{boxes} = \delta_{boxes}(\nu(\varphi(I_{norm})))$
Dimensions are $80\ x\ 80\ x\ (4 * 3)$

$P_{conf} = \delta_{boxes}(\nu(\phi(I_{norm})))$
Dimensions 80 x 80 x 3

$P_{class} = \delta_{class}(\nu(\phi(I_{norm})))$
Dimensions 80 x 80 x 80

4. **Parse Predictions**:
   - Decode 80x80x12 box predictions
   - Extract 80x80x3 confidence scores
   - Extract 80x80 gun class probabilities

   5. **Apply Threshold**:
   - Extract boxes with confidence > 0.25

   6. **NMS**:
   - IOU threshold = 0.45

7. **Return**:
   - Set of bounding boxes B detecting guns

Table 2: Step-by-step algorithm for the code designed.

The overall pipeline for detecting objects with YOLOV7 is as follows:

1. *Input* - The input image is resized to a standard resolution like 640 x 640 x 3 pixels. Data augmentation is also used during training for regularization, which is notated by I/255. This step will help the image to scale up to 640 x 640 pixels.
2. *Forward Pass* - The resized image is passed through the YOLOV7 model to generate raw predictions. This takes a single pass through the network.
    a. *Backbone(φ)* – The backbone extracts features using the 53 CNN channels and passes it on to the neck.

b. *Neck(v)* – Neck works on extracting and pooling different features extracted from the backbone and selects the ones that are necessary.
c. *Head(δ)* – This is the object prediction phase.

3. *Parse Predictions (Post-process)* - The predictions are decoded by:
    a. Parsing bounding boxes across different scales. In the algorithm above, we see that the pixels are 80 x 80 x (4 x 3), which is our x_center, y_center, width and height.
    b. Applying activations like sigmoid to scale confidences and probabilities. From the algorithm, we are going to get 80 x 80 x 3 confidence scores for the required object in the image.
    c. Thresholding boxes based on confidence. 80 x 80 is the number of possibilities the model is going to generate. This number represents the likelihood of an object belonging to the required class, i.e., 'Gun.'
4. *Threshold* – The confidence score detected is going to be more than 0.25. This removes all the weaker detections.
5. *Non-Max Suppression* - NMS removes duplicate detections by suppressing overlapping boxes. The IOU(Intersection over Union) threshold is set up to 0.45. The boxes that overlap more than 0.45 are suppressed from predictions.
6. *Draw Bounding Boxes* - The final object detections are drawn as bounding boxes on the original input image as 'B.'

The pipeline of the code is represented in a flow chart below.

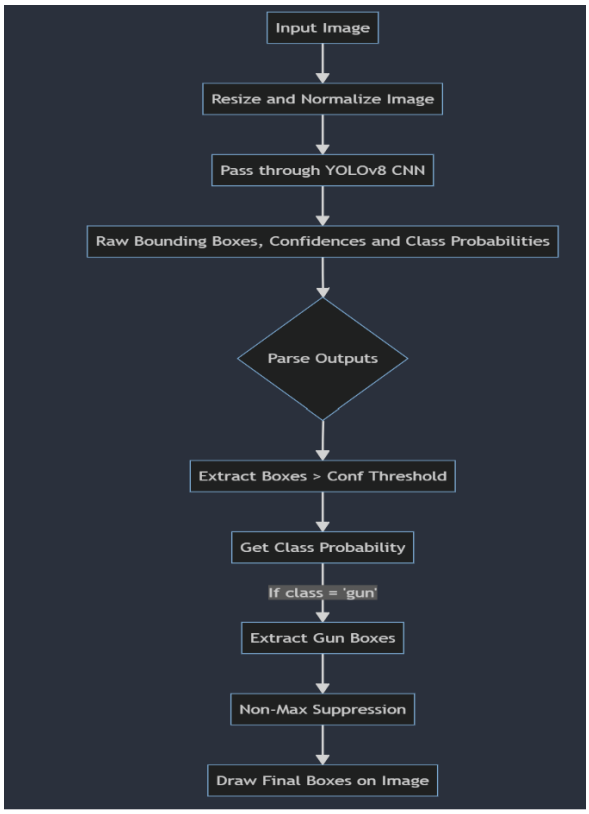

Figure 3: Pipeline for detecting guns using YOLOV7

This pipeline allows for end-to-end detection of multiple objects in a single evaluation pass. The model can process images at 20-50 FPS on a standard GPU for real-time detection. The unified architecture and training process enables high accuracy on benchmarks like COCO.

To achieve state-of-the-art object detection performance, YOLOV7 introduces architectural improvements like the CSPDarknet53 backbone, CSP-SPP neck, and multi-scale predictions. The unified, fully convolutional design enables real-time detection speeds suitable for autonomous driving, robotics, and video surveillance applications. Extensive benchmarks demonstrate YOLOV7's capabilities for detecting a wide range of objects accurately and rapidly by modeling the detection problem as a direct spatial prediction task using convolutional neural networks.

5 Evaluation and Discussion

To implement this custom training of a Yolov7 model, I used Google Colab and its free GPU. Since this research looks at a comparative study of how Yolov7 performs on two different datasets, The training and testing of these datasets has taken around 5 to 6 hours of computational time. Google Colab provided a T4 GPU, which allocated a 12GB Nvidia Tesla. Even though the 12GB GPU was enough, the extensive computational time caused the constant failure of the models' training. We can refer to the table below to understand the time taken to train the models for both datasets.

Dataset	*Model*	*Size*	*Batch size*	*Epochs*	*Computational time*
Dataset1	Yolov7	3087	16	50	3.5 hours
				100	5 hours
Dataset2	Yolov7	4922	16	50	4 hours
				100	6 hours

Table 3: Computation time taken by the datasets.

After the extensive training and validation of the Yolov7 model for both datasets, it is time to compare the test results in different scenarios. We are also going to look at how transfer learning can help us understand the Yolov7 model. In algorithms related to computer vision, metrics like True Positive(TP), True Negative(TN), False Positive(FP), and False Negative(FN) are used to understand whether the labels are correct or not. Let us understand a few metrics to describe the output using these metrics.

1. Precision - $\frac{TP}{TP+FP}$ (1)
2. Recall - $\frac{TP}{TP+FN}$ (2)
3. F1 – Score - $2 \times \frac{precision \times recall}{precision + recall}$ (3)
4. mAP@50% - $\frac{1}{N}\sum_{i=1}^{N} AP_i$ (4)

   The letter 'N' denotes the total number of classes in this inquiry which is 'Gun'. In this study, the primary focus of the object detection task is on a single class, namely 'handgun.' The Intersection over Union (IoU) is defined as the ratio of the predicted bounding box's and ground truth bounding box's shared area to the combined area of both bounding boxes from equation (5).
5. IOU(Intersection over union) - $\frac{Area\ of\ Intersection}{Area\ of\ Union}$ (5)
6. Yolov7-1 – The Yolov7 model trained with dataset 1.
7. Yolov7-2 – The Yolov7 model trained with dataset 2.

Now let us dive into the different experiments and interpret the results using equations (1) to (5).

5.1 Experiment 1:

We are going to look at the test results for both datasets when used the models trained for 100 epochs with batch size 16. The below test results are for a lesser confidence level score which is 0.4 and is using the best.pt weights generated by the train and validate data.

5.1.1 COCO annotated Dataset with Yolov7

Let us look at the results for dataset 1 from COCO when tested with YOLOv7 model.

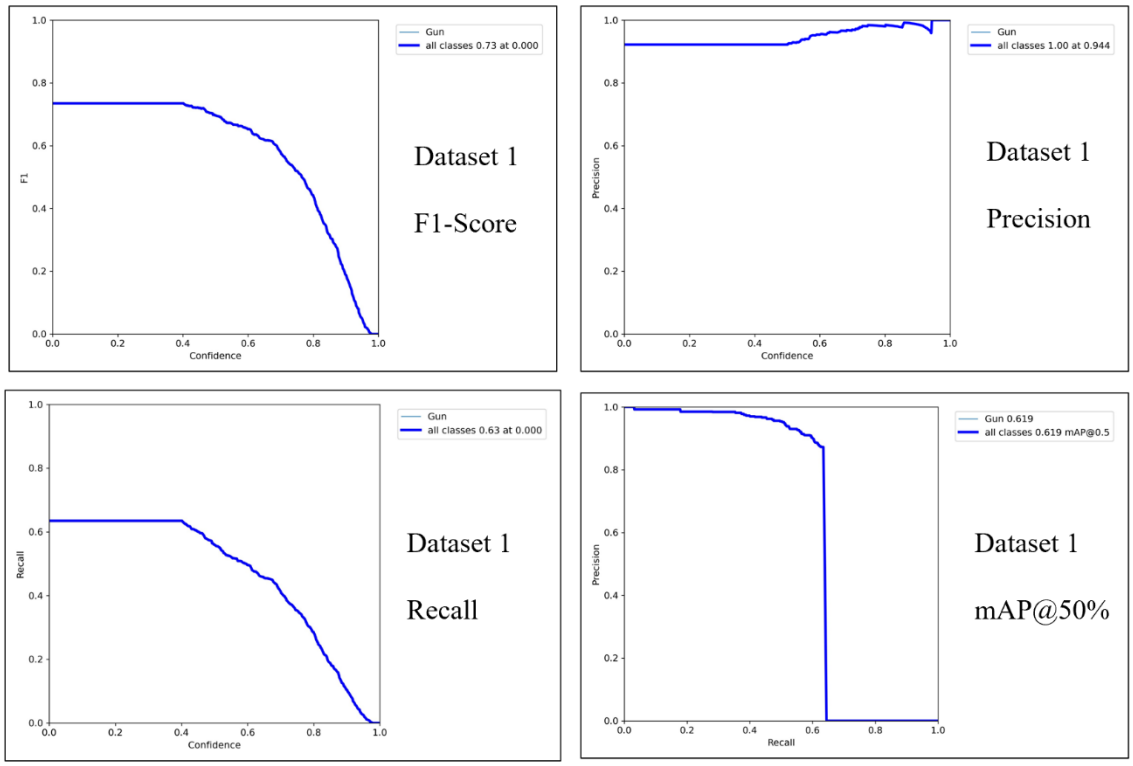

Figure 4: The result graphs from Dataset 1, including F1-score, precision, Recall, and mAP@50%

The image below shows the label predictions performed by the model.

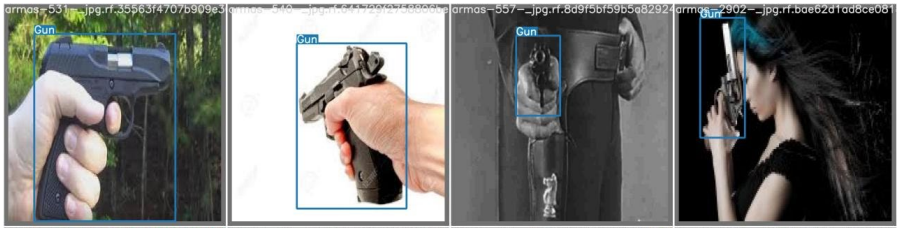

Figure 5(a): Test labels batch

Figure 5(b): Prediction labels for the above image

As we can see it is has missed labelling a few images. This can be because of the low mAP we have got from the training.

5.1.2 Custom Dataset with Yolov7

Let us look at the results for dataset 2, which is the custom dataset when tested with the YOLOv7 model.

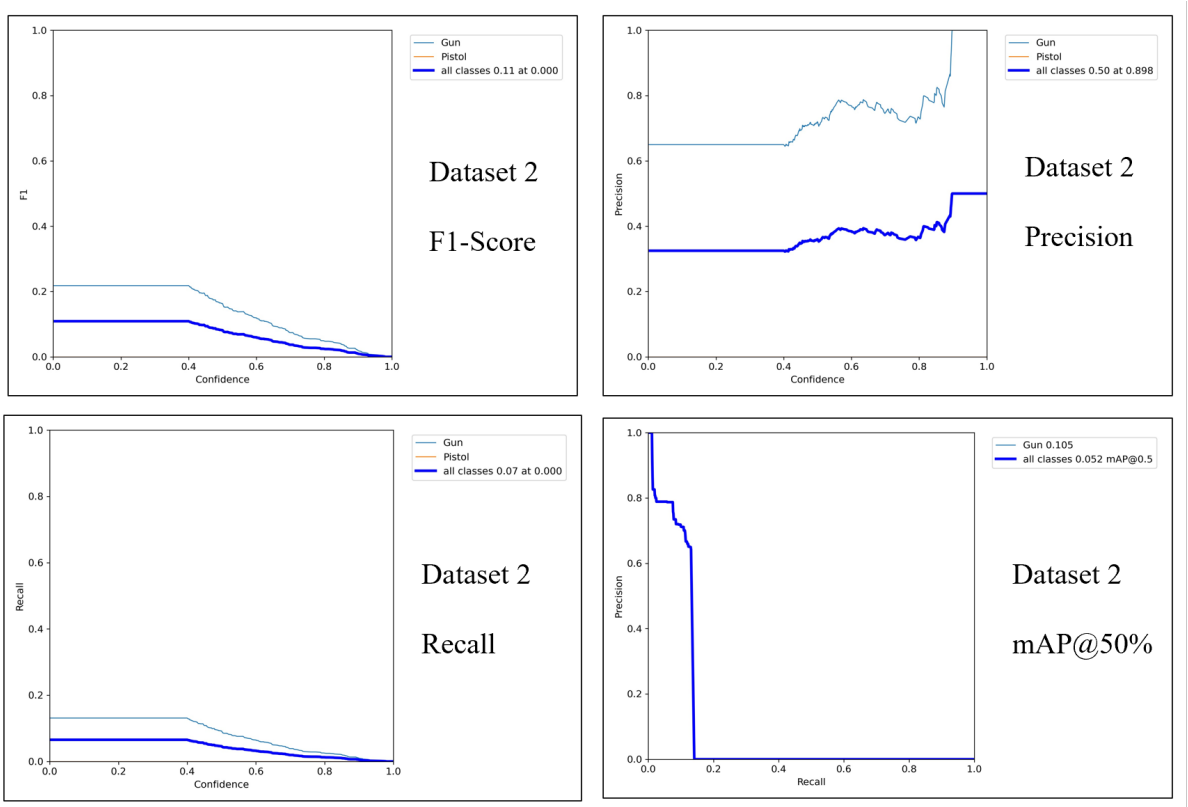

Figure 5: The result graphs from Dataset 2, including F1-score, precision, Recall, and mAP@50%

The image below shows the label predictions performed by the model.

Figure 6(a): Test labels batch

Figure 6(b): Prediction labels for the above image

As we can see it is has missed labelling a most of the images. This can be because of the low mAP and recall rate we have got from the training. There is a high chance that because the images are too pixellated, the model was poorly trained. It is similar to the condition faced by (Bota-Ioana and Gavrilas, 2023)

5.2 Experiment 2: Custom Dataset with Transfer Learning

As a part of the experiment, I tried training the Yolov7-1 model Dataset 2. Yolov7-1 was trained on the COCO annotated dataset. After a long run of one hour, let us look at the results.

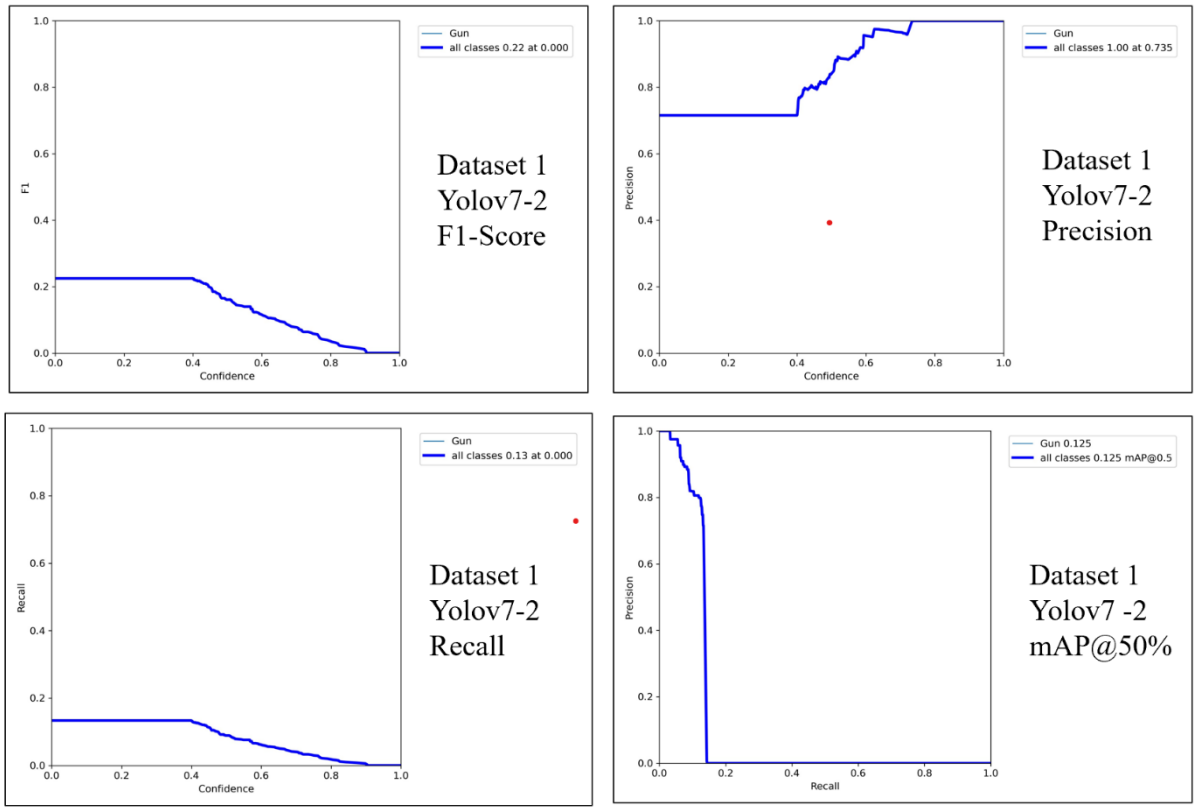

Figure 7: The result graphs from Dataset 2, including F1-score, precision, Recall, and mAP@50%

The results are similar to the results obtained when tested with Dataset 2 in Figure 5. Now, there can be several known reasons for the low mAP and precision. It is majorly due to the pixellated nature of the image and lack of a large dataset for training.

5.3 Discussion

As the results shown in Table 4, the Yolov7 trained with the COCO dataset has outperformed the Yolov7 trained with a custom dataset. The high precision indicates a low probability of false positives, which is critical in situations where misclassifications might have serious consequences(Bota-Ioana and Gavrilas, 2023). The model's ability to recognize firearms is satisfactory, as evidenced by the obtained recall and F1-score. We can see the same highlighted in green in table 4.

Dataset	Size(test)	Model	Labels	Precision	Recall	F1-Score	mAP@0.50
Dataset1	603	Yolov7 - 1	698	0.872	0.635	0.70	0.619
		Yolov7 - 2	698	0.715	0.133	0.22	0.125
Dataset2	984	Yolov7 - 2	984	0.65	0.131	0.11	0.105

Table 4: Overall performance of the model when used with datasets.

6 Conclusion and Future Work

This study examines the efficacy of YOLOv7 in detecting firearms when combined with transfer learning. We extensively examined the intricacies of object detection across many scenarios, focusing on a solitary category, namely "handgun." In addition, we performed a comparative analysis utilizing two distinct datasets, which elucidated the robustness and adaptability of our proposed approach. The experiments demonstrated that the model's ability to detect weapons in diverse situations is significantly enhanced when YOLOv7 is used with transfer learning. The network acquired a comprehension of fundamental characteristics by employing a pre-trained model on a comprehensive item detection assignment. As a result, accelerated convergence and enhanced performance were observed in this specific gun detection evaluation.

The latest officially released SOTA from Yolo is Yolov7. The unofficial one is Yolo8. The Yolov7 model trained with the COCO dataset (1st experiment in the paper - Yolov7-1) has performed well compared to the standard results. We also need to consider the lack of resources like GPU and limitations with the computer system used, which make a major difference in the output received. The models took 6 hours of constant running on Google Colab, which would not be possible on regular machines or laptops. The main issue this paper tried to tackle was blurriness in the pictures as they were stretched from 416 x 416 pixels to 640 x 640 pixels. Sadly, the issue remains for the models trained for this paper. However, future work is to improve the dataset's quality and work on the model to see how well it works with less blurriness in the images.

The comparison of the two datasets yielded crucial insights into the model's generalization capacity. The model's versatility in managing diverse datasets underscores its relevance in real-world situations characterized by frequent and fluctuating conditions. Furthermore, incorporating transfer learning streamlined the training process and improved convergence, overcoming specific challenges arising from limited dataset sizes. The algorithm showed a remarkable capacity to detect nuanced patterns connected to handguns, underscoring the importance of using preexisting knowledge. The evaluation metrics provide insights into the model's performance across several datasets, encompassing precision, recall, and the F1 score. The reliability and utility of our proposed gun detection methods are confirmed through a thorough review of these measures, combined with qualitative analysis.

While this study provides valuable insights, future research endeavors could explore additional enhancements, such as refining models using datasets specific to a particular domain and integrating more sophisticated transfer learning methods. The continuous advancement of object identification techniques, along with recent discoveries in deep learning, offers good prospects for greatly enhancing the capabilities of gun detection systems. In summary, our research contributes to the growing body of knowledge in object detection by emphasizing the significance of transfer learning in facilitating the use of models like YOLOv7 for accurate and efficient gun identification in various environments.

References

[1] Ashish Ranjan, Sunita Dhavale and Suresh Kumar. (2023), *YOLO Algorithms for Real-Time Fire*.

[2] Bota-Ioana, R.M. and Gavrilas, A. (2023), "Application for threat surveillance using Python and Yolo-V7", *2023 22nd International Symposium INFOTEH-JAHORINA, INFOTEH 2023*, Institute of Electrical and Electronics Engineers Inc., doi: 10.1109/INFOTEH57020.2023.10094153.

[3] Chandan G, Ayush Jain, Harsh Jain, Mohana, Institute of Electrical and Electronics Engineers and RVS College of Engineering & Technology. (2018), *Real Time Object Detection and Tracking Using Deep Learning and OpenCV*.

[4] Dextre, M., Rosas, O., Lazo, J. and Gutiérrez, J.C. (2021), "Gun Detection in Real-Time, using YOLOv5 on Jetson AGX Xavier", *Proceedings - 2021 47th Latin American Computing Conference, CLEI 2021*, Institute of Electrical and Electronics Engineers Inc., doi: 10.1109/CLEI53233.2021.9640100.

[5] Diwan, T., Anirudh, G. and Tembhurne, J. V. (2023), "Object detection using YOLO: challenges, architectural successors, datasets and applications", *Multimedia Tools and Applications*, Springer, Vol. 82 No. 6, pp. 9243–9275, doi: 10.1007/s11042-022-13644-y.

[6] Dugyala, R., Vishnu Vardhan Reddy, M., Tharun Reddy, C. and Vijendar, G. (2023), "Weapon Detection in Surveillance Videos Using YOLOV8 and PELSF-DCNN", *E3S Web of Conferences*, Vol. 391, EDP Sciences, doi: 10.1051/e3sconf/202339101071.

[7] Fan, J., Lee, J.H., Jung, I.S. and Lee, Y.K. (2021), "Improvement of Object Detection Based on Faster R-CNN and YOLO", *2021 36th International Technical Conference on Circuits/Systems, Computers and Communications, ITC-CSCC 2021*, Institute of Electrical and Electronics Engineers Inc., doi: 10.1109/ITC-CSCC52171.2021.9501480.

[8] Fathima Safa, K.U. and Suguna, M. (2023), "Subduing Crime and Threat in Real-Time by Detecting Weapons Using Yolov8", *Proceedings of the International Conference on Circuit Power and Computing Technologies, ICCPCT 2023*, Institute of Electrical and Electronics Engineers Inc., pp. 864–868, doi: 10.1109/ICCPCT58313.2023.10245146.

[9] Hashmi, T.S.S., Haq, N.U., Fraz, M.M. and Shahzad, M. (2021), "Application of Deep Learning for Weapons Detection in Surveillance Videos", *2021 International Conference on Digital Futures and Transformative Technologies, ICoDT2 2021*, Institute of Electrical and Electronics Engineers Inc., doi: 10.1109/ICoDT252288.2021.9441523.

[10] Karp, A. (2018), *Civilian-Held Firearms Numbers 1 Estimating GloBal Civilian-HElD FirEarms NumBErs*.

[11] Lin, Q., Ye, G., Wang, J. and Liu, H. (2022), "RoboFlow: a Data-centric Workflow Management System for Developing AI-enhanced Robots", in Faust, A., Hsu, D. and Neumann, G. (Eds.), *Proceedings of the 5th Conference on Robot Learning*, Vol. 164, PMLR, pp. 1789–1794.

[12] Narejo, S., Pandey, B., Esenarro Vargas, D., Rodriguez, C. and Anjum, M.R. (2021), "Weapon Detection Using YOLO V3 for Smart Surveillance System", *Mathematical Problems in Engineering*, Hindawi Limited, Vol. 2021, doi: 10.1155/2021/9975700.

[13] Olmos, R., Tabik, S. and Herrera, F. (2018), "Automatic handgun detection alarm in videos using deep learning", *Neurocomputing*, Elsevier B.V., Vol. 275, pp. 66–72, doi: 10.1016/j.neucom.2017.05.012.

[14] Pullakandam, M., Loya, K., Salota, P., Yanamala, R.M.R. and Javvaji, P.K. (2023), "Weapon Object Detection Using Quantized YOLOv8", *5th International Conference on Energy, Power, and Environment: Towards Flexible Green Energy Technologies, ICEPE 2023*, Institute of Electrical and Electronics Engineers Inc., doi: 10.1109/ICEPE57949.2023.10201506.

[15] Uganya, G., Shadrach, F.D., Sudha, I., Krishnammal, P.M., Lakshmanan, V. and Nandhini, T.J. (2023), "Crime Scene Object Detection from Surveillance Video by using Tiny YOLO Algorithm", *Proceedings - 2023 3rd International Conference on Pervasive Computing and Social Networking, ICPCSN 2023*, Institute of Electrical and Electronics Engineers Inc., pp. 654–659, doi: 10.1109/ICPCSN58827.2023.00114.

[16] Velastin, S.A., Boghossian, B.A. and Vicencio-Silva, M.A. (2006), "A motion-based image processing system for detecting potentially dangerous situations in underground railway stations", *Transportation Research Part C: Emerging Technologies*, Elsevier Ltd, Vol. 14 No. 2, pp. 96–113, doi: 10.1016/j.trc.2006.05.006.

[17] Wang, G., Ding, H., Duan, M., Pu, Y., Yang, Z. and Li, H. (2022), "Fighting against terrorism: A real-time CCTV autonomous weapons detection based on improved YOLO v4", *Digital Signal Processing: A Review Journal*, Elsevier Inc., Vol. 132, doi: 10.1016/j.dsp.2022.103790.

[18] Warsi, F.A., Abdullah, M., Husen, M.N., Yahya, M., Khan, S. and Jawaid, N. (2019), *Gun Detection System Using YOLOv3*.