Enhancing Public Safety's Real-Time Weapon Detection with Deep Learning

by

Mridul Ghosh (201014085)
Emtu Rani Paul (201014056)
Mrinal Kanti Saha Joy (192014031)

Capstone project report (CSE 499) submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science in Computer Science and Engineering

Under the supervision of

Shohag Barman, PhD

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

UNIVERSITY OF LIBERAL ARTS BANGLADESH

SUMMER 2024

DECLARATION

Project Title	Enhancing Public Safety's Real-Time Weapon Detection with Deep Learning
Authors	*Mridul Ghosh, Emtu Rani Paul, Mrinal Kanti Saha Joy*
Student IDs	201014085, 201014056, 192014031
Supervisor	Shohag Barman, PhD

We declare that this capstone project report entitled *Enhancing Public Safety's Real-Time Weapon Detection with Deep Learning* is the result of our own work except as cited in the references. The capstone project report has not been accepted for any degree and is not concurrently submitted in the candidature of any other degree.

Mridul Ghosh
201014085

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Emtu Rani Paul
201014056

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Mrinal Kanti Saha Joy
192014031

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh

Date: November 03, 2024

Department of Computer Science and Engineering
University of Liberal Arts Bangladesh
Mohammadpur, Dhaka - 1207

CERTIFICATE

This is to certify that the capstone project report entitled **Enhancing Public Safety's Real-Time Weapon Detection with Deep Learning**, submitted by **Mridul Ghosh** (Student ID: 201014085) , **Emtu Rani Paul** (Student ID: 201014056) and **Mrinal Kanti Saha Joy** (Student ID: 192014031) are undergraduate students of the **Department of Computer Science and Engineering** has been examined. Upon recommendation by the examination committee, we hereby accord our approval of it as the presented work and submitted report fulfills the requirements for its acceptance in partial fulfillment for the degree of *Bachelor of Science in Computer Science and Engineering*.

Shohag Barman, PhD, Assistant Professor
Dept. of CSE, University of Liberal Arts Bangladesh

Ms. Rubaiya Hafiz, Senior Lecturer
Dept. of CSE, University of Liberal Arts Bangladesh

Jannatul Ferdous Ruma, Lecturer
Dept. of CSE, University of Liberal Arts Bangladesh

Md. Aktaruzzaman Pramanik, Lecturer
Dept. of CSE, University of Liberal Arts Bangladesh

Place: Dhaka
Date: November 03, 2024

ACKNOWLEDGEMENTS

We would like to express our deep and sincere gratitude to our research supervisor, *Shohag Barman, PhD*, for giving us the opportunity to conduct research and providing invaluable guidance throughout this work. His dynamism, vision, sincerity, and motivation have deeply inspired us. He has taught us the methodology to carry out the work and to present the works as clearly as possible. It was a great privilege and honor to work and study under his guidance.

We are greatly indebted to our honorable teachers of the Department of Computer Science and Engineering at the University of Liberal Arts Bangladesh who taught us during the course of our study. Without any doubt, their teaching and guidance have completely transformed us to the persons that we are today.

We are extremely thankful to our parents for their unconditional love, endless prayers and caring, and immense sacrifices for educating and preparing us for our future. We would like to say thanks to our friends and relatives for their kind support and care.

Finally, we would like to thank all the people who have supported us to complete the project work directly or indirectly.

Mridul Ghosh, *Emtu Rani Paul* , *Mrinal Kanti Saha Joy*
University of Liberal Arts Bangladesh
Date: November 03, 2024

Dedicated To
My beloved parents the guiding light of my life

– Mridul Ghosh

Dedicated To
My parents,the source of my inspiration

– Emtu Rani Paul

Dedicated To
In our hearts, your love we'll always keep

– Mrinal Kanti Saha Joy

ABSTRACT

When it comes to maintaining public safety, weapons detection is crucial. Generally, detecting weapons accurately and efficiently in real-time is important to maintain security in various environments, such as airports, public gatherings, schools, colleges, and universities, to prevent potential threats. This study presents advanced deep-learning techniques to address the challenge of weapon detection. We proposed a model designed to detect and classify weapons in images using a hybrid approach. This model uses NASNetMobile for feature extraction and YOLOv8n for object detection. We improve detection accuracy in this model by using hybrid methods. YOLOv8n is recognized for its fast and reliable object detection capabilities at the same time as NASNetMobile is a convolutional neural network (CNN) capable of extracting important features from images. Using YOLOv8n, the proposed hybrid model achieved 99.89% accuracy using the custom model and 96.40% mAP. Comparative analysis with CNN, VGG16, Faster R-CNN, Yolo v3, SSD, Yolo v5, Yolo v4, -CNN, RsNN, Yolov8, and SVM each have accuracies of 93%, 85%, 84.60%, 94%, 82.9%, 20.58mAP%, 89.4%, 73.80%, 96.80%, 96.04% achieves , 91.73%, 93%, 78%, respectively. Our proposed model achieved better accuracy than other models. The combination of NASNetMobile and YOLOv8 addresses the need for accurate and rapid weapon detection, thereby improving security systems.

Keywords: Weapon detection, Security, NASNetMobile, YOLOv8n, CNN, Object detection, Feature Extraction, and Classification.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.0.1 Background Literature

There is a growing interest in technology-based security monitoring and surveillance systems to ensure a risk-free environment and provide global security. In particular, effective surveillance is essential in crowded places such as airports, railway stations, shopping malls, sports stadiums, and tourist spots. The use of weapons accounts for the majority of deaths worldwide, which takes a toll on health, psychological, and economic costs. It is therefore imperative to reduce weapon violence and ensure peace and security. Governments and non-governmental organizations are increasingly using smart surveillance systems that can automatically identify hazards and help take quick action, strengthening security measures.Vieira et al. (2022) This paper discusses the application of deep learning models for real-time weapon detection in CCTV surveillance, aiming to reduce crime rates. Two key models, YOLOv3 and Faster R-CNN with VGG-16, are analyzed for detecting firearms, specifically guns, in public areas. The motivation stems from the growing need for efficient systems to assist human security personnel, who may not always detect criminal activity. Real-time weapon detection can enable prompt action, enhancing public safety. The study addresses the challenge of limited datasets by utilizing both the COCO dataset for non-weapon images (negative cases) and the Internet Movies Firearms Database (IMFDb) for weapon images (positive cases). It focuses on improving detection accuracy through synthetic datasets and advanced deep-learning architectures. The goals of the research are twofold: first, to introduce and compare the performance of the YOLOv3 and Faster R-CNN models, and second, to analyze the differences in their detection results. The paper reveals that while both models perform well, Faster R-CNN with VGG-16 achieves higher

accuracy. The paper concludes that weapon detection systems can significantly improve surveillance efficiency, contributing to crime prevention in public spaces.

1.0.2 Technological Evolution in Weapon Detection

In today's world,many advanced technologies are being used to solve many serious problems.In the modern age, security is always a main concern in every domain, due to a rise in crime rates in crowded events or suspicious, lonely areas. Jain et al. (2020) Modernization of technology is changing the way of detecting weapons day by day. Convolutional Neural Network (CNN) In addition to playing an important role in detecting weapons, CNN is very accurate and fast, and it can detect weapons from an image. In weapon detection, CNN analyzes the characteristics of different types of weapons, thereby accurately identifying and classifying weapons. CNNs are trained using deep learning and large datasets which have revolutionized the field of weapon detection. This system mainly analyzes the information on detected weapons, which greatly helps in security and military operations. The advanced capabilities and speed of CNNs have made weapon detection more efficient and reliable in the age of modern technology, making our daily life safer and at the same time life much easier.

1.0.3 Hybrid Models Precision in Weapon Detection

A hybrid model combines two or more different methods or systems to improve performance across domains to obtain better performance and accuracy and take advantage of their strengths. Hybrid model, which we have used in our projects in combination with deep learning like YOLOv8n and NASNetMobile. YOLOv8n performs classification for object detection and recommends specific areas. On the other hand, NASNetMobile helps feature extraction. The combination of these two methods helps to overcome the challenges of data combing because YOLOv8n, with its lightweight architecture, can quickly and accurately extract features and make accurate observations. Using YOLOv8n, data for specific regions of the image can be efficiently recovered from previous data. As a result, the accuracy of weapon detection is improved. This hybrid model makes real-time security monitoring more effective and reliable.

1.1 Problem statement

Bangladesh's security system is currently facing significant challenges. Especially using camera feeds to detect dangerous weapons in crowded areas and trying to identify criminals. Increasing the threat of criminal and terrorist activities and looking at installing cameras to increase public safety. It can be said that crime will be reduced by this.

Civilian security has become a growing concern in recent times due to the increasing number of criminal activities and terrorist attacks. In such situations, a variety of weapons often pose a significant threat. Which results in the need for efficient detection and the detection process is not accurate. Although surveillance cameras have become ubiquitous in places as diverse as streets, offices, malls, and banks, accurately identifying the type of weapon has become a complex task for security personnel.Talib & Saud (2024).

Because security personnel cannot monitor multiple screens simultaneously. As a result, various accidents are happening around. Therefore we propose the use of advanced hybrid models such as NASNetMobile and YOLOv8n to ensure real-time detection and prevention of potential threats. By integrating these models, we demonstrate that STEM increases the speed and accuracy of weapon identification from camera images, reduces false positives, and reduces human involvement in continuous surveillance. As a result, it is possible to reduce various major accidents. YOLOv8n allows fast feature extraction and makes real-time detection more efficient. This optimized side helps increase overall security and ensures automatic monitoring, helping to quickly and accurately identify potential hazards in public spaces.

1.2 Objectives and Significance of the Study

1.2.1 Aims

The main goal of our research, to develop deep learning methods through real-time monitoring to improve public monitoring systems to identify suspicious weapons in images. We YOLOv8n to automate the detection process, reducing the need for human observation and using the NASNetMobile model. This automation not only makes detection faster and more accurate but is also used to increase security against potential threats. The YOLOv8n model is particularly useful for real-time monitoring of busy areas because it allows fast and precise feature extraction. By evaluating these combined models, we observe how well they can detect dangerous weapons and help develop better security solutions for public safety, and threat reduction.

1.2.2 Objective

- **Develop an Efficient Hybrid Model:** Design and implement a hybrid model integrating YOLOv8n and NASNetMobile for effective weapon detection in image surveillance. This combination increases efficiency accuracy and provides a powerful solution for detecting potential threats in real-time. .
- **Develop an efficient hybrid model:** To design and implement a hybrid model that plays an effective role, in image surveillance. Integrates NASNetMobile and YOLOv8n for weapon detection and the combination shows increased task accuracy and effectiveness, real-time detection of potential threats. Provides, solid solutions and reduces public threats.
- **Improve Weapon Detection Capability:** YOLOv8n advanced detection capabilities as well as NASNetMobile's feature extraction facilities can be used to improve the accuracy and speed of weapon detection and identify potential threats in real time.
- **Reduce Security Risks:** A model has been developed that plays an important role in reducing security threats, making public life easier. Being able to perform realtime and reliable detection of dangerous weapons has made this possible.

- **Improve efficiency:** Greater efficiency in operations is encouraged by providing security personnel with a reliable and advanced tool to effectively manage and detect weapons-related incidents.
- **Enhancing Security Against Threats:** Ultimately, the research aims to provide an efficient and accurate weapon detection system, contribute to public safety, use advanced technology, and improve security against criminals and violent activists by actively activating surveillance systems.
- **Address the limitations of human observation:** ensuring a more reliable security solution and overcoming the challenges associated with manual observation to improve processing speed and help ensure detection accuracy.

1.2.3 Motivation

The motivation behind this study was to prevent the increasing crime involving weapons in public places. Keeping the general public safe from incidents especially involving guns, revolvers, and pistols. For this, we have developed a real-time weapon detection method using NASNetMobile and YOLOv8n models in our application using camera capture. Conventional surveillance systems that rely on humans often miss threats and are slow to operate. By automating the identification process, this model helps to quickly detect and prevent crime, thereby improving public safety. The ultimate goal is to develop an advanced monitoring system capable of immediately detecting the presence of dangerous weapons and responding quickly.

1.3 Scope and Limitation

Scope:
The research is focused on revolutionizing the detection of weapons in populated areas. The system, developed using NASNetMobile and YOLOv8n models, increases the speed and accuracy of weapon detection and real-time weapon detection, reduces reliance on human observers, and helps identify criminals by enabling rapid response. Helps in the accurate identification of dangerous weapons like pistols and knives. This technology is easily flexible and can be used on a large scale in public safety and security systems, which will help increase security by quickly detecting threats. Which will make people's public life easier.

Limitation:
This study is limited the model can only detect weapons in the dataset and not any other size data. Which reduces its performance in detecting other weapons. In the future, the need to increase the size and diversity of the dataset may include a variety of weapon sizes, making the model more efficient and useful. In addition, better quality cameras should be used so that all sizes of weapons can identify them and can identify weapons in real time. This makes small arms detection easier. Thus, by improving datasets and using better quality cameras, it may be possible to increase weapon detection capabilities and lead to better public safety outcomes.

Chapter 2

Literature Review

A study conducted by found that fast R-CNN models evaluated achieved a mean accuracy (mAP) of 85%. This score may not be high enough for all real-world applications, especially in critical security situations where MAP values were 85%, while the AP level ranges from 80% to 91% depending on the item detected. An average real-time detection speed was found to be 11-13 FPS. Both the accuracy and speed of the faster R-CNN model allow it to be recommended for use in public security monitoring systems aimed at detecting potentially dangerous objects.Both the accuracy and speed of the faster R-CNN model allow it to be recommended for use in public security monitoring systems aimed at detecting potentially dangerous objects. Omiotek & Zhunissova (2024) The accuracy of the model's weapon detection may vary in different environments. Also, the performance of the model may be affected by changes in lighting and object angles, which were not tested. Although the dataset is specialized, it is not sufficient for coverage of potentially dangerous objects and situations encountered in real life. In another study by Talib & Saud (2024) explores the field of weapon detection emphasizing its crucial role in ensuring public safety and security.Technology is needed to prevent attacks without human supervision. So an automatic weapon detection using deep learning is designed. which detects the presence of weapons using neural networks and provides an optimized solution for detection. The paper introduces the YOLOv7 network model is a promising and solution aimed at improving both detection accuracy overall efficiency. Through an examination of related studies the paper identifies existing challenges and gaps in current systems supporting the need for a more robust , efficient approach to weapon detection. Ensembling techniques were implemented in YOLOv5 to increase their speed and performance. The models achieved the highest score of 78% with an estimated speed of 8.1 ms. However the

faster R-CNN models achieved the highest AP of 89%. Using its high quality model will give good results.A study by Chunchwar et al. (n.d.) proposes a model that points out that security camera and video surveillance systems usually Depend on manual observation to detect risky situations, which often leads to late detection of incidents due to a Deficiency of the security Workforce. This Forms an increased risk to public safety, especially when crime rates related to guns and dangerous weapons are increasing every year. As a solution to this problem, the authors propose a low-cost intelligence-based solution that will assist in real-time weapon detection and surveillance video analysis, and public safety. This will enhance and help in the early detection of crimes. Precision, recall, and mean average precision (mAP) were used to evaluate the model's performance, where the YOLOv8 model achieved a score of 0.78 mAP, indicating its success in weapon detection. In 2023 Vijayakumar et al. (2023) proposed Several important points in weapon identification are mentioned in this paper. Previous research has focused mainly on gun and knife detection, but there is still a lack of a Consistent weapons dataset. Also, differences in labeling and pre-processing methods between different algorithms make the datasets inconsistent with each other. To solve these problems, the proposed system tried to detect different types of weapons such as axes, guns, knives, and pistols in a custom dataset using Faster R-CNN and YOLOv4 algorithm. The YOLOv4 model achieved an mAP score of 96.04%, indicating its high performance.In 2022, address the challenges of detecting dangerous objects using CCTV systems. It emphasizes the issues associated with relying on human operators to monitor multiple screens continuously, which often results in missed threats and delayed responses. The paper references prior research by Grega et al., who introduced a method for the automatic detection of dangerous situations using image processing and machine learning techniques.Tram et al. (2023) The paper evaluates the performance of different models such as YOLOv3, VGG-16 and R-CNN. This model Working with a comprehensive dataset, the model underwent rigorous training and verification and reached the precision range of a minimum of 90.38% to a maximum of 95.83%. Foreshadowing the model's sensitivity, the recall metric led the way of 94.39%. The class with the highest F1-Score, which is a composite score of precision and recall, was 5, with a value of 93.95%. Their research examines different techniques for detecting objects in image scenes.Suryavanshi et al. (2024). Their research examines different techniques for detecting objects in image scenes. They mention the work of Liang et al. (2019), who developed a hybrid model combining CNN and RsNN architectures. This model was effective in identifying objects and showcasing their

relationships using the PASCAL VOC 2012 and SYSU datasets, which include detailed scene descriptions. The study also references Li et al. (2019), though specific details about their findings are not provided. The paper highlights the accuracy of various object detection models, revealing that the Faster R-CNN model excels with accuracy rates of 97% for gun detection, 93% for knives, and 90% for identifying individuals. Furthermore, it reports average accuracy rates for other models achieved 78%, YOLO reached 92%, and Faster R-CNN maintained an average of 94%. These results illustrate the effectiveness of different models in tackling object detection tasks across multiple categories.Ruprah & Shrivastava (2022). In another study, it examines the limitations of current object detection methods in digital image processing. It points out that earlier techniques like part-based models were mainly suitable for stable environments, such as airports and train stations. Although later methods like neural networks, CNN, and SVM were introduced, they struggled in dynamic settings. This review emphasizes the necessity for improved methods to increase detection accuracy and performance in real-time situations. The paper notes that combining YOLO v3 with Faster R-CNN achieves a mean Average Accuracy (mAP) of 85%, while YOLO v3 alone reaches an accuracy of 82.9%. Maddileti et al. (2022). In this study, Ahmed et al. (2022) discusses various methods based on deep learning models in weapon detection. This work conducted a comparative analysis of advanced models in weapon detection. In particular, the Scaled-YOLOv4 model gives better results than the previous YOLOv4 model, which is able to improve frames per second (FPS) and mean average precision (mAP). The model performed well in all weapon categories and enabled fast detection in real time. It achieved 97.5% accuracy on ImageNet and IMFDB datasets, giving better results than existing methods in weapon detection. Convolutional neural networks can be used efficiently for object detection. In this paper, specifically, two convolutional neural network (CNN) architectures – fast R-CNN with VGG16 and YOLOv3, are used for such weapon detection. In this paper YOLOv3 is 96.26%, fast R-CNN with VGG-16 is 93%, SVM classifier is 92.6%.The aim of this paper is to present and compare the performance of two models for gun detection in any given scenario. The results of YOLOv3 quickly outperformed R-CNN with VGG16. The ultimate goal of this paper is to accurately identify guns in an image which in turn can help crime investigations and provide immediate results.Das & Tomar (2022) This study addresses the issue of different datasets used, which affects the identification accuracy and overall efficiency of the models. In particular, the YOLO and RCNN models achieved 33 and 34.2 MAP scores, respectively, indicating accuracy,

but the performance of these models may yield different results in other challenging environments. The SSD model's relatively low 20.58 MAP score indicates that it is more complex or less effective. Besides, dynamic situation. Future studies may require larger and more diverse datasets to overcome these limitations. This paper optimizes the detection of commonly used weapons like AK47s, hand revolvers, pistols, combat knives, grenades, etc. in live video footage. An efficient and quick way to determine weapon identification.Varshney et al. (2021) This study addresses the issue of existing research on object detection algorithms, noting that only a few have been designed specifically for weapon detection. It discusses the use of neural networks with area-based descriptors and sliding window techniques for identifying firearms, focusing on reducing false alarms by identifying specific areas of interest. Among the models tested, YOLOv4 performed best with a mean average precision of 91.73%, Other models like VGG, Can, and FasterRCNN also had varied accuracy, ranging between 59% and 93%. Narejo et al. (2021) In this study, we focused on weapon detection using CCTV footage, by testing the evolution of object detection algorithms. It discusses the limitations of traditional techniques such as histogram of oriented gradients (HOG) and sliding window methods, which struggle with real-time weapon detection. The shift to deep learning models like CNN, YOLO, fast R-CNN has been emphasized for their increased accuracy and automatic feature extraction. Dataset quality and diversity are cited as important factors. The paper also introduces confusion classes to reduce false positives and negatives, reporting the accuracy of various models, with Yolov4 achieving a maximum of 99%, underscoring progress in real-time detection. Bhatti et al. (2021a) This study shows that testing includes real-time weapon detection using live video and images, and an SMS alert feature to notify supervisors when a handgun is detected in the system. These results emphasize the efficiency of MobileNetV3 in both accuracy and real-time performance. The MobileNetV3 model achieves 96% accuracy for handgun detection, MobileNetV2-SSD achieves 89% accuracy and GoogleNet achieves 87% accuracy. In addition, MobileNetV3-SSDLite had a feature extractor performance score of 0.942, which was higher than MobileNetV2-SSD's 0.816.Ghazal et al. (2020) This study examines previous methods of weapon detection, particularly small object detection in existing datasets, while also highlighting the lack of realistic CCTV weapon image datasets. The aim of the paper is to improve weapons detection, small arms detection and enhance the existing dataset. It reports improvements in accuracy (3.91%) and F1 scores (2.25%) compared to previous studies, with a 0.80-point boost in average accuracy (AP) for

small objects when combining artificial and real images. González et al. (2020) This research is focused on improving surveillance through real-time detection of fire and handguns using Convolutional Neural Networks (CNN). The proposed deep learning model, based on the YOLOv3 algorithm, processes video frames to detect anomalies such as fire and firearms, generating alerts for authorities. Critical requirements for timely response to incidents such as wildfires, industrial explosions and gun violence are addressed. Tested on datasets such as IMFDB, UGR, and FireNet, it provided 89.3%, 82.6%, and 86.5% accuracy, proving effective for both indoor and outdoor deployments.Mehta et al. (2020) The paper acknowledges the limitations of current surveillance systems in quickly detecting suspicious activities like armed robbery from CCTV footage. It emphasizes the need for automated weapon detection without human intervention, focusing on intra-class detection to identify specific firearm types. highlighting that deep learning, especially YOLOv3 and Faster R-CNN, outperforms traditional methods in both speed and accuracy. YOLOv3 excels in real-time detection, while Faster R-CNN offers higher accuracy but slower speeds. Despite achieving 89.15% accuracy, the small dataset size limits the feasibility of real-time detection for broader applications.Yadav et al. (2023)

2.0.1 Literature Analysis

Table 2.1: Literature Analysis

Reference	YOLO	SVM	VGG	CNN	SSD	Faster R-CNN	RsNN	MobileNet	GoogleNet	R-CNN
Omiotek & Zhunissova (2024)						✓				
Suryavanshi et al. (2024)		✓		✓						✓
Talib & Saud (2024)	✓									
Akhila & Ahmed (2024)	✓									
Akhila & Ahmed (2024)	✓									
Tram et al. (2023)	✓				✓	✓				
Vijayakumar et al. (2023)	✓				✓		✓			
Yadav et al. (2023)	✓					✓				
Das & Tomar (2022)	✓				✓		✓			
Ruprah & Shrivastava (2022)	✓			✓		✓	✓			
Maddileti et al. (2022)		✓		✓		✓				
Ahmed et al. (2022)	✓									
Varshney et al. (2021)	✓				✓					
Narejo et al. (2021)	✓		✓			✓				
Bhatti et al. (2021a)	✓			✓				✓		
Ghazal et al. (2020)					✓			✓	✓	
González et al. (2020)						✓	✓			
Mehta et al. (2020)	✓			✓						

2.0.2 Performance Evaluation Criterion

When evaluating the performance of a weapon detection system, several criteria can be considered to ensure that the model is functioning effectively and reliably. Common performance evaluation criteria specific to our project using NASNetMobile for feature extraction and YOLOv8 for object detection:

i. Mean Average Precision (mAP)
The average of the precision scores at different IoU thresholds is a comprehensive metric for object detection models. It provides a single performance measure across multiple classes and IoU thresholds.

ii. Confusion Matrix
A table is used to describe the performance of a classification model by displaying the true positives, true negatives, false positives, and false negatives. Helps visualize how well the model is performing across different classes.

iii. ROC Curve: A graphical representation of the true positive rate (sensitivity) against the false positive rate (1-specificity) at various threshold settings.
iv. AUC (Area Under Curve): Area Under Curve Can represent the degree of under curve or measure of separability. This helps determine how many classes the model will be able to distinguish between.

v. Inference Time
Estimation time takes a long time for the model to process a single input image, but for real-time applications, time is very important, ensuring that the system works very quickly, takes less time, and can perform accurate weapon detection.

The combination of mAP, confusion matrix, and inference time offers a comprehensive evaluation framework for the weapon detection system. These metrics ensure that the model performs well in terms of correctness and safety-critical factors like minimizing false alarms and ensuring timely detection. This combination of metrics provides a balanced view of the system's strengths and weaknesses, ensuring that it meets the high safety and reliability standards in weapon detection applications.

Chapter 3

Design and Development

3.1 Project Management and Financial Activities

Project Management:
Our project is carefully planned to develop a weapon detection a web application. In the early stages we focus on setting clear goal for the project such as building system that can accurately and weapons in real-time detection . We form dedicated team under the guidance and assign specific roles to each member to ensure smooth progress. In the planning phase we create a detailed plan that outlines each tasks timeline . We also identify risks such as the potential for misidentification or technical challenges. To reduce these risks we come up with strategies to manage them.That's why we have tried to maintain the project all the time and give a correct outline. We select NASNetMobile and YOLOv8n because they detect weapons effectively in real-world situations base. In the operational phase we focus on gathering and preparing information. This ensures that our dataset is of high quality which is important for model training. We start building our hybrid model using NASNetMobile and YOLOv8n for accurate detection in weapon . We continuously monitor the project to ensure we address Any schedule risks and maintain data quality throughout the development process when reach the final stage .we test the weapon detection system to see how well it works. We review and document any lessons learned during the project to improve in future work. We plan to carefully develop a weapon detection web application first in the project. In the first phase of planning we create a detailed plan that outlines each tasks timeline and what resources the execution phase we focus on collecting and preparing data. This ensures that our dataset is of high quality, which is critical for training the model. We begin developing our hybrid model using NASNetMobile and YOLOv8n to

achieve accurate detection. We continuously monitor the project to ensure we stay on schedule, address any risks, and maintain data quality throughout the development process. When we reach the final phase, we test the weapon detection system to see how well it performs. We review and document any lessons learned during the project to improve future work. That focuses on setting clear goals for the project, such as creating a system that can accurately detect weapons in real-time while preventing risks. We identify potentially dangerous areas, such as a crowded area where weapons detection or technical challenges may arise. To reduce these risks, we come up with strategies to manage them. We select the NASNetMobile and YOLOv8n models, as they can effectively detect weapons in real-world situations and provide accurate results. At the operational level, we focus on gathering and preparing information. This confirms how high quality our dataset is, how much data my model will be able to run on. which is very important for training a model. We plan to build our hybrid model using NASNetMobile and YOLOv8n for accurate detection. We continuously monitor the project to ensure that we are on schedule, address any risks and maintain data quality throughout the development process. When we reach the final stage, we test the weapon detection system to see how well it can produce results. We review and document any lessons learned during the project to improve future work and plan to refine the work accordingly.

Financial Management:
Keeping finances in mind, we plan a clear budget at the planning stage. We allocate costs to key parts of the project, such as building the model, all costs for team members and testing the model. Throughout the project, we closely monitor costs and provide regular updates to staff. So that all costs, technology and other expenses for the team members can be run properly. We also set aside funds for unexpected courses. After the project is completed, we review all costs to ensure that everything is in order and that we have managed the budget effectively, and try to stick to a good plan. This organized approach to both project and financial management helps ensure the success of our weapons detection system.

3.1.1 Project Overview

The aim of this project is to build a web application that uses Convolutional Neural Network (CNN) for automatic weapon detection, providing accurate results. Focuses on identifying guns and knives in pictures. We collected datasets of weapon

images in different scenarios to effectively train the CNN model. It also helps to identify the required features of the system, ensuring correct identification. images through multiple layers of processes in real-time, even in challenging environments. This project presents a novel approach for real-time violence detection that harnesses the power of deep learning through the integration of a Convolutional Neural Network (CNN) for weapon detection.More et al. (2024) Through automatic weapon detection, our application can increase security in public spaces, reduce human error, and detect and detect potential threats faster, ultimately leading to improved public safety.

3.1.2 Timeline

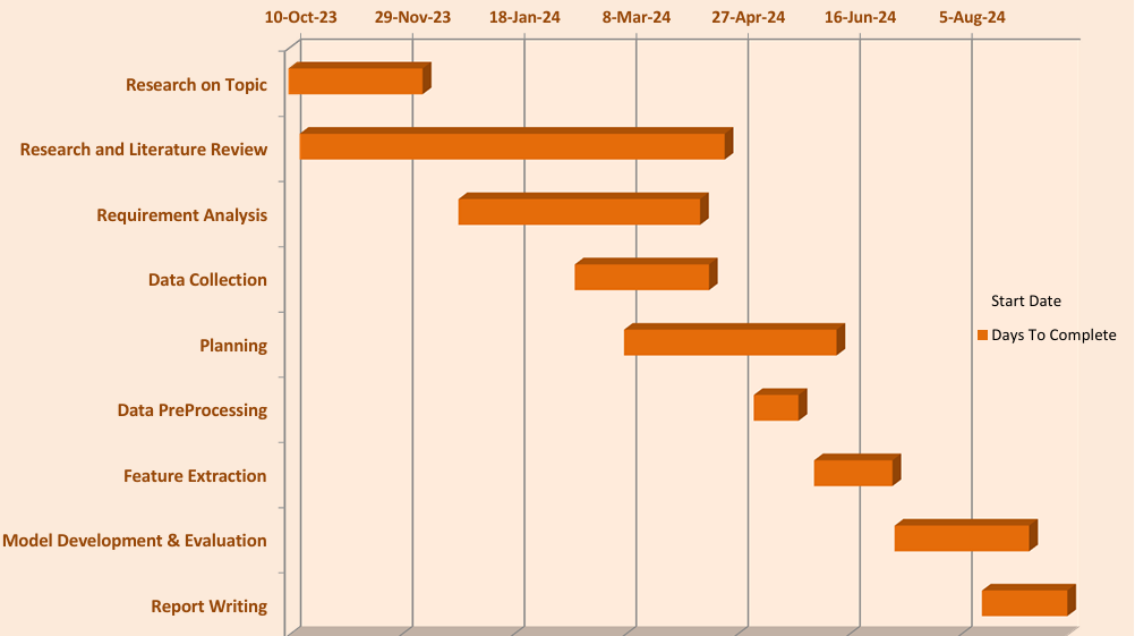

Figure 3.1: Project Timeline for Weapon Detection

We have planned a careful project timeline to ensure that the project is successfully completed within the estimated time frame. It started with our project field selection, which took place from October 2023 to November 2023. Then, we researched that specific topic, and finalized our project concept in the research and proposal section It lasted for a long time, which was followed by delving into extensive research and proposal development from the first day itself alongside the

project field selection. We dedicated a substantial amount of time, from the end of October 2023 to March 2024, to collecting and preprocessing datasets essential for our project. We decided to utilize open-source datasets as well. Model selection and training, which is a crucial aspect of the project, were carried out from March 2024 to April 2024. This was followed by implementing a hybrid model and conducting a comparative study with our system. The model development and evaluation were designed and developed in August 2024. The back-end coding, ensuring accuracy and reliability, and rigorous testing and validation took a few more weeks. Then, we dedicated our time to comprehensive report writing.

3.1.3 Financial Report Project Budget

Table 3.1: Budget Summary

Discretion	Budget
Hardware Costs	155,400
Transportation Cost	2,500
Miscellaneous Cost	500

Table 3.2: Total cost of Hardware

Hardware Description	Unit Price	Quantity	Amount
Intel Core i5 (6th generation or newer) or AMD Ryzen 5	25,000	1	25,000
NVIDIA GPU with CUDA support (e.g., GTX 1050 Ti, 1650, or equivalent)	40,000	1	40,000
SSD with at least 256 GB for faster read-/write times	4,000	1	4,000
Motherboard Gigabyte Z590 VISION G 11th Gen	12,000	1	12,000
cooling	1,500	1	1,500
Power Supply Corsair RM650 650Watt 80 Plus Gold	9,200	1	9,200
Peripherals	50,000	1	50,000
8-12 GB of RAM	8,000	1	8,000
Miscellaneous	500	1	500
Total Budget			155,400

3.2 Adopted Tools and techniques

3.2.1 Hardware Requirements

For small-scale testing and development with moderate-speed detection, the following hardware should suffice:

- CPU: Intel Core i5 (6th generation or newer) or AMD Ryzen 5
- GPU: NVIDIA GPU with CUDA support (e.g., GTX 1050 Ti, 1650, or equivalent)
- Storage: 12 TB
- RAM: 8-12 GB of RAM
- Storage: SSD with at least 256 GB for faster read/write times
- Operating System: Windows 10/11, Ubuntu 18.04/20.04, or macOS (with TensorFlow support)
- Internet connectivity: High-speed internet for web application access
- Display: Monitor or screen for accessing the user interface
- Input devices: Keyboard and mouse or equivalent input methods

3.2.2 Software Requirements

The following software components are essential for the system's optimal functioning:

1. **Operating System:** Windows 10: Suitable for various applications.

2. **Programming Language:**

- **Python 3.x:**Essential for the entire project.
- **Libraries:**TensorFlow, Keras, PyTorch, Ultralytics YOLOv8.

3. **Frameworks and Libraries:**

- **TensorFlow 2.x:**Required for using NasNetMobile as the backbone for feature extraction.

- **Keras:**Keras is integrated into TensorFlow, and it's used to work with pre-trained models like NasNetMobile.
- **YOLOv8 (Ultralytics):**For object detection.
- **PyTorch (Optional):** YOLOv8 can also be used with PyTorch, depending on your preference.

4. Data Handling Libraries:

- **Pandas:**For managing datasets and annotations.
- **NumPy:**For numerical operations with arrays.

5. Computer Vision Libraries:

- **OpenCV:** For image and video processing.

6. Visualization Tools:

- **Matplotlib:** To create visualizations like pie charts, bar charts, and bounding boxes.

7. Deep Learning Utilities:

- **CUDA:** If using NVIDIA GPUs, you'll need the CUDA Toolkit (compatible with your TensorFlow/PyTorch version).
- **cuDNN:** NVIDIA's Deep Neural Network library.
- **NVIDIA Drivers:** Ensure the latest drivers are installed for your GPU.

8. Development Environment:

- **Jupyter Notebook:** For easy experimentation and iteration.
- **PyCharm:** IDEs for coding and managing the project efficiently.

Chapter 4

Methodology

4.1 Methodology

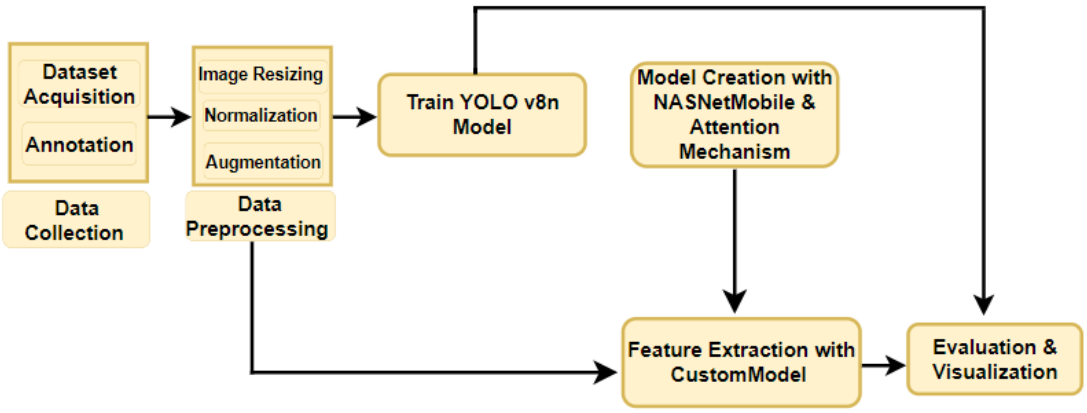

Figure 4.1: Weapon Detection Model Workflow

4.1.1 Data Preprocessing

We have collected data from different fields from various parts of Bangladesh.We collected data from utilized open-source datasets.After data preprocessing and augmentation, our dataset comprised 5,002 images.The dataset was divided into training and testing sets,with 80% allocated for training and 20% for testing. The three classes in the dataset were Gun,Knife, and no Weapon.Through data preprocessing, we ensure the quality,diversity, and reliability of our dataset.The preprocessing were executed to optimize the images for subsequent model training and evaluation.

4.1.2 Image Resizing

Our weapon detection dataset was curated from various sources, including online repositories. We Emphasized images that clearly show various weapons, ensuring the dataset's Importance and effectiveness for our research objectives. This method allows deep learning models to extract the presence of weapons. Clear visual examples of datasets are Significant in enhancing model performance in real-world applications.

4.1.3 Normalization

After resizing the image, we fixed the pixel values of each image by a standard value of [0, 1]. By dividing the pixel values by 255 to extract the maximum intensity value, we have effectively converted the pixel values to a normal range scale. By ensuring consistent pixel value representation, we can accurately detect weapon presence and improve model capabilities, resulting in more reliable performance in a variety of real-world scenarios. This normalization helps increase the convergence of deep learning models when training light variations and helps reduce lighting effects.

4.1.4 Imbalanced Data

In our dataset, we encountered class imbalance where one class had significantly more data samples than the other. We have employed data preprocessing techniques to address this imbalance and avoid this imbalance by focusing on the majority class during model training.

4.1.5 Under Sampling

In our dataset, we encountered class imbalance, where one class had significantly more data samples than the other. To address this imbalance and prevent bias towards the majority class during model training, we employed under-sampling.

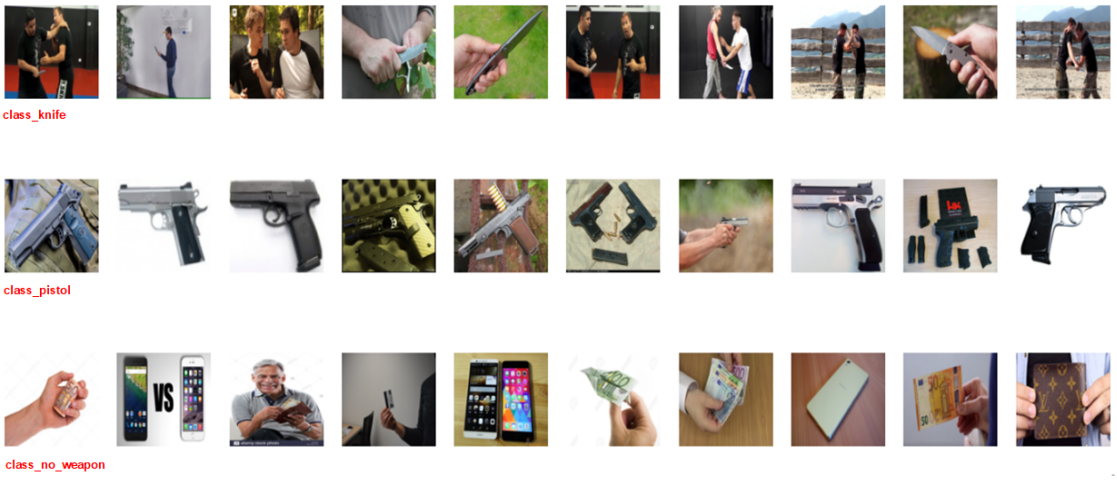

Figure 4.2: Sample from the Weapon Detection Dataset

4.1.6 Data Augmentation

To enhance the diversity of our weapon detection dataset and improve model generalization, we implemented various augmentation techniques, such as rotation, horizontal and vertical flipping, and zooming. Utilizing the ImageDataGenerator class from the Keras library, we dynamically augmented images during the model training process, creating variations that simulate real-world conditions. Our total training images were 3863. We have increased it through data augmentation to 5002 images.

4.1.7 Train-Test Split

We partitioned the preprocessed dataset into distinct training and testing subsets using the train test split function from the sci-kit-learn library. The subsets Adopting an 80:20 split ratio, 80% of the data was allocated for model training, while the remaining 20% was reserved for model evaluation. There were 5600 images in the training. This strategic partitioning strategy ensured a robust evaluation of model performance on unseen data while preserving a substantial training dataset for effective model learning.

4.1.8 Data visualization

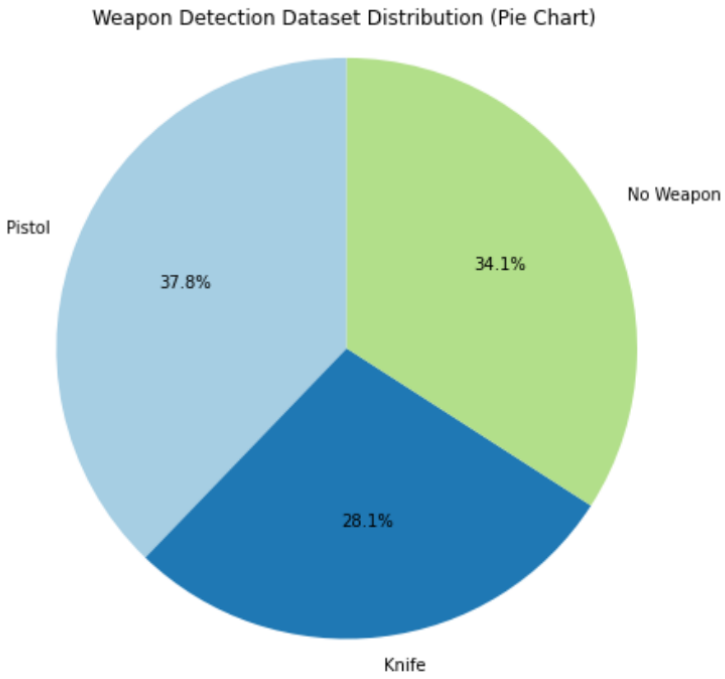

Figure 4.3: Weapon Detection Dataset Distribution (Pie Chart)

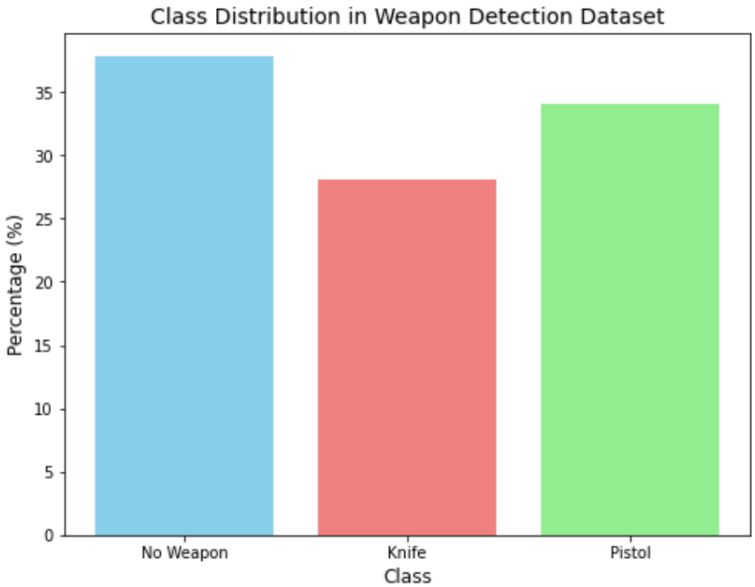

Figure 4.4: Weapon Detection Dataset Distribution (Bar Chart)

4.2 Model Selection

In our pursuit to build a dependable weapon detection system, we carefully explored various model architectures. Initially, we started with basic convolutional neural networks (CNN) for detecting weapons, such as gun images. As we progressed, we embraced advanced models like NASNetMobile and YOLOv8n to improve feature extraction, ensuring more detailed and accurate detection of weapons in complex situations. To improve the system's capability of weaponising things, we also combined these feature extraction techniques with categorisation algorithms.This method greatly improved our system's accuracy and dependability under practical circumstances.By developing our models and the detection procedure over time, we created a reliable and effective weapon detection system.

i. VGG16 (Visual Geometry Group 16): A kind of CNN called VGG16 has 16 layers and is used for classifying images.In order to extract elements like edges, forms, and textures, it analyses a picture through layers.Upon completion, the image is categorized into pre-established groups.

ii. YOLO (You Only Look Once): An object detection approach called YOLO looks at the entire picture in just one pass.The picture is divided into grids, and each grid is looked at for items and their positions.Yolo is quick and has real-time object detection capabilities.

iii. SVM (Support Vector Machine): SVM is a machine learning model used for classification tasks. It finds the best boundary (line or plane) that separates data into two categories. It classifies new data based on which side of the boundary it falls.

iv. VGG (Visual Geometry Group) : VGG is a family of CNN architectures with different depths (e.g., VGG11, VGG16, VGG19). These models pass images through many layers to extract increasingly detailed features. The deeper the network, the more complex patterns it can learn.

v. CNN (Convolutional Neural Network): CNN is designed to recognize patterns in images, such as edges, textures, and objects. It uses filters to scan different parts of the image and extract important features. These features are then used to classify the image.

vi. SSD (Single Shot MultiBox Detector): SSD is an object detection model that processes an image in one step and predicts objects at multiple scales. It uses predefined boxes to detect objects of different sizes and shapes. SSD is fast and can detect several objects at once.

vii. Faster R-CNN (Faster Region-based Convolutional Neural Network): Fast R-CNN is a two-step object. which weapon helps in identification. First, it locates potential regions and then classifies the objects where the objects are located and helps in achieving accuracy..

viii. RsNN (Recurrent Spiking Neural Network): RSNN is a type of neural network that explores how biological neurons communicate through spikes. Inquiries are motivated by results. It processes information over time and proceeds to handle sequential data. Recurrent connections allow it to store and process information from previous inputs. This makes it useful for tasks such as motion analysis.

4.2.1 Basic CNN Model

We specifically used a basic Convolutional Neural Network (CNN) architecture for gun detection. One of the main applications of computer vision is to deal with various problems in anomaly detection and monitoring using Convolutional Neural Networks (CNN). Due to the increasing demand in security and protection of personal property, the demand and deployment of video surveillance systems can recognize and interpret the scene.NARASIMMAN et al. (2022) It also helps to understand the concepts required by CNN in detecting firearms in images. With this approach, we can derive optimized models for feature extraction and classification, real-time object detection, and efficient performance. Accurate identification of guns, knives, and weapons. This basic understanding enabled us to improve our model. This makes it possible to identify guns, knives, and other weapons in any situation.

4.2.2 NasNetMobile for Feature Extraction

We used NASNetMobile to develop a weapon detection system focused on feature extraction. NASNetMobile detects important things like the size, texture and shape of weapons. After analyzing the images, it helps the system to further classify and effectively identify different types of guns. NASNetMobile ensures that the detection system can work properly in different situations. NASNetMobile's ability to focus on these key features enables the system to quickly and reliably identify weapons in real-time surveillance or security footage. Besides, it is also tested at different evenings in key places like airports, schools and colleges, where accurate identification is possible. Advanced features of the model ensure that the system can operate in different environments, detect weapons efficiently, and improve overall security.

4.2.3 Selection of YOLOv8 Nano and NasNetMobile

After extensive testing and thorough performance evaluation, we found that we started with a basic CNN that NASNetMobile effectively classifies weapons and captures detailed features. On the other hand, YOLOv8 is capable of real-time object detection. This combination can improve the accuracy of our detection model, real time image object detect and weapon detection By integrating these advanced models, we developed a system capable of handling complex weapon detection. This provides insights into how the process can improve performance using multiple architectures, helps create a predictable weapon, and determines whether the identified system is suitable for real-world applications.

4.2.4 Hybrid model

After careful experiments and model evaluation we find that the integration of NASNetMobile for feature extraction and classification and YOLOv8n deep learning for image recognition has made realtime weapon detection possible for public safety. Which gave us promising results. Its combination showed superior performance in terms of accuracy. Besides, it gave accurate results at the right time. This is why we are interested in working with NASNetMobile and YOLOv8n.

4.2.5 Custom Model Architecture

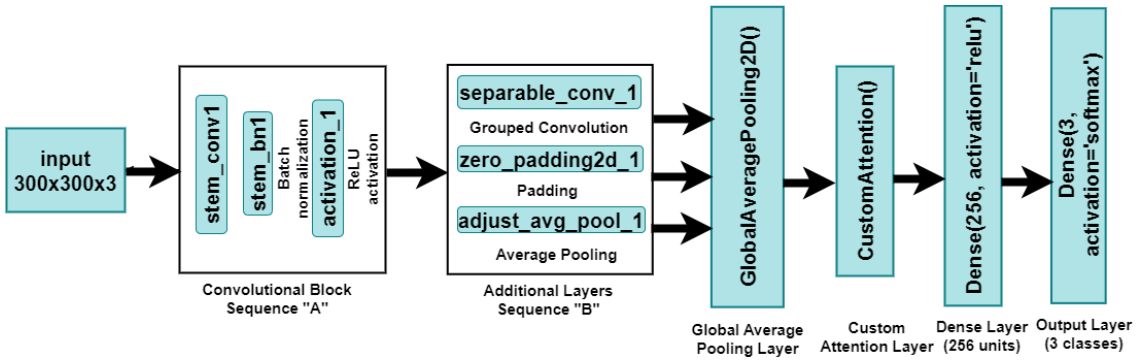

Figure 4.5: Custom Model Architecture for Weapon Detection

First, we are collecting data by data acquisition and data annotation. Then we did data preprocessing by image resizing, normalization, and augmentation. Then dataset is run with the pre-trained YoLo v8n model. Then created a custom model through NASNetMobile and attention mechanisms. Through this, we are running the dataset again after feature extraction. Then we train Yolov8 model results feature extraction with custom model results, and then we are doing evaluation visualization.

We can see in the methodology that we made a custom model with net mobile and attention mechanism. We have given an input size of 300x300 pixels with 3 color channels (RGB). Then convolution block sequence, which is the default in NASNetMobile. The Additional Layers Sequence ”B” consists of grouped convolution padding and average pooling to refine and reduce the spatial dimensions of the input features.

Stage i	Operator $\hat{F}_i$	Resolution $\hat{H}_i$ x $\hat{W}_i$	#Channels $\hat{C}_i$	#Layers $\hat{L}_i$
1	Conv2D + Stem Block	150 x 150	32	1
2	Separable Conv2D + Reduction Cell	75 x 75	44	2
3	Normal Cell Block	75 x 75	88	2
4	Normal Cell Block	38 x 38	176	2
5	Reduction Cell Block	19 x 19	352	2
6	Normal Cell Block	19 x 19	704	2
7	Unfrozen NASNet Layers	19 x 19	704	10
8	Global Average Pooling	1 x 1	704	1
9	Custom Attention Layer	1 x 1	704	1
10	Dense Layer (ReLU)	1 x 1	256	1
11	Dense Layer (Softmax)	1 x 1	3	1

Figure 4.6: Custom Model Architecture for Weapon Detection table

4.2.6 NasNetMobile for Classification

In our endeavor to devise an effective weapon detection system, the integration of NasNetMobile as a classification model proved essential. NasNetMobile, designed for efficiency and accuracy, excels in handling complex image data. Because of this, it may be used in real-time.The model's capacity to extract significant information from photos improves classification performance and makes it possible to accurately identify different kinds of weapons.We adjusted NasNetMobile on certain datasets by utilizing transfer learning, guaranteeing strong detection capabilities.

4.2.7 Transfer Learning

We used our weapon identification dataset to refine a pre-trained NASNetMobile model using transfer learning.The model effectively made use of high-level feature representations acquired from a variety of pictures by starting it with weights that had already been trained on the Image Net dataset.To fine-tune the model to our particular goal while preserving the insights from Image Net,we had to update the parameters of the final layers.This may be expressed mathematically as minimizing the loss function L by varying the model parameters.

4.2.8 Essential Elements of NasNetMobile Classification

i. Linear and Non-linear Classification: The design of NasNetMobile facilitates efficient categorisation by utilising deep learning methods to identify both linear and non-linear patterns in visual data.The model can categorise different sorts of weapons based on the extracted characteristics thanks to this capacity, which allows it to recognise complicated relationships within high-dimensional feature spaces.

ii. Feature Extraction Efficiency: NasNetMobile False's capacity to effectively extract rich information from photos is one of its primary advantages.Effective weapon detection in real-time situations is made possible by the lightweight design,which guarantees quick processing without sacrificing accuracy.This effectiveness is essential in settings where rapid decision-making is required.

iii. Transfer Learning for Enhanced Performance: NasNetMobile may be improved on certain weapon detection datasets by applying transfer learning, which will increase classification accuracy.By using this method, the model may adjust to the distinct features of numerous weapon types, enhancing detection capabilities in a range of environments.

iv. Robustness to Variations: The architecture of the model encourages resilience to changes in backdrop, lighting, and angle.NasNetMobile lowers the chance of misclassification by becoming skilled at identifying weapons in a variety of scenarios by training on a variety of datasets.

v. Handling High-Dimensional Data: Detecting weapons frequently involves looking at high-dimensional feature representations in pictures.By skillfully handling this complexity, NasNetMobile makes sure that subtle aspects of weaponry are recorded for well-informed categorization choices.

vi. Regularization Techniques: NasNetMobile improves generalization to unknown data by including regularisation techniques to prevent overfitting. Because of its flexibility,the learned characteristics are guaranteed to be applicable in a variety of scenarios, producing more trustworthy detection results.

The integration of NasNetMobile in our weapon detection framework highlights our commitment to utilizing advanced methodologies for addressing security challenges. Its capability to extract meaningful features, handle complex patterns, and maintain robustness significantly improves the accuracy and reliability of our system, paving the way for advancements in public safety and security applications.

4.2.9 Model Training

We set different learning rates during model training.We monitored the model's performance on a separate validation dataset.The training process was halted if the validation loss did not improve for a certain number of epochs.We adjusted the dropout rate to achieve better results.

Batch Size and Epochs
We experimented with different batch sizes and epochs to find the optimal configuration for training the model.A batch size of 64 was chosen to balance computational efficiency and model convergence.The model was trained for 100 epochs to ensure convergence and capture complex patterns in the data.[Train Accuracy: 99.68%, Test Accuracy: 97.29%]

4.2.10 Feature Extraction Visualization

In the feature extraction phase of our weapon detection project,t-distributed stochastic neighbor embedding(t-SNE) was utilized to visualize the extracted features in a two-dimensional space.The plot below illustrates the distribution of these features, with each point representing an image transformed into a high-dimensional feature vector.The colors denote different weapon classes,demonstrating the separability of the extracted features based on weapon categories.This visualization allows us to assess how effectively the model distinguishes between various weapon types,providing insights into the feature representation and highlighting areas where the model can be improved.

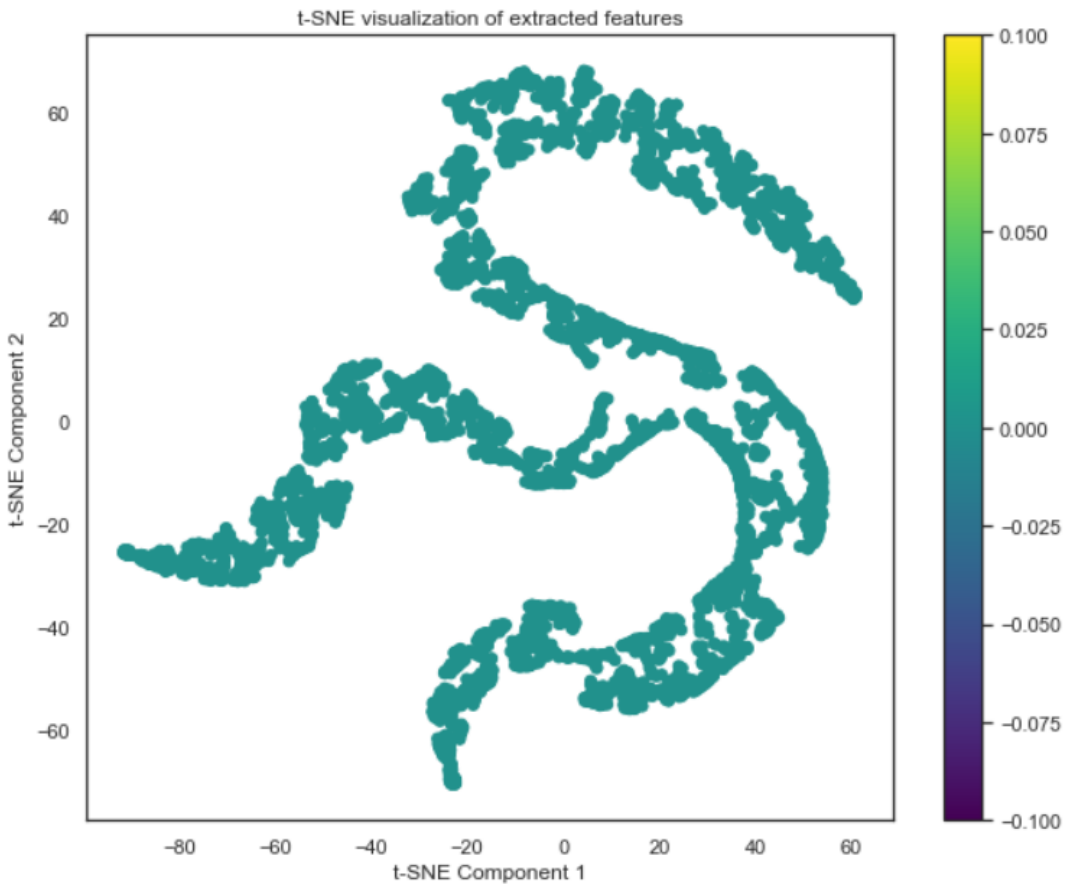

Figure 4.7: Feature Extraction Visualization in Weapon Detection

4.2.11 Feature Correlation Matrix

In weapon detection using the NasNetMobile model,feature correlation metrics play a vital role in assessing the relationships between various extracted features.This analysis helps in recognizing redundant features and highlights those that significantly impact weapon classification.Visualization techniques, such as heatmaps,illustrate these correlations clearly,guiding the feature selection process.Ultimately, ensuring effective and accurate weapon detection across diverse scenarios and conditions.

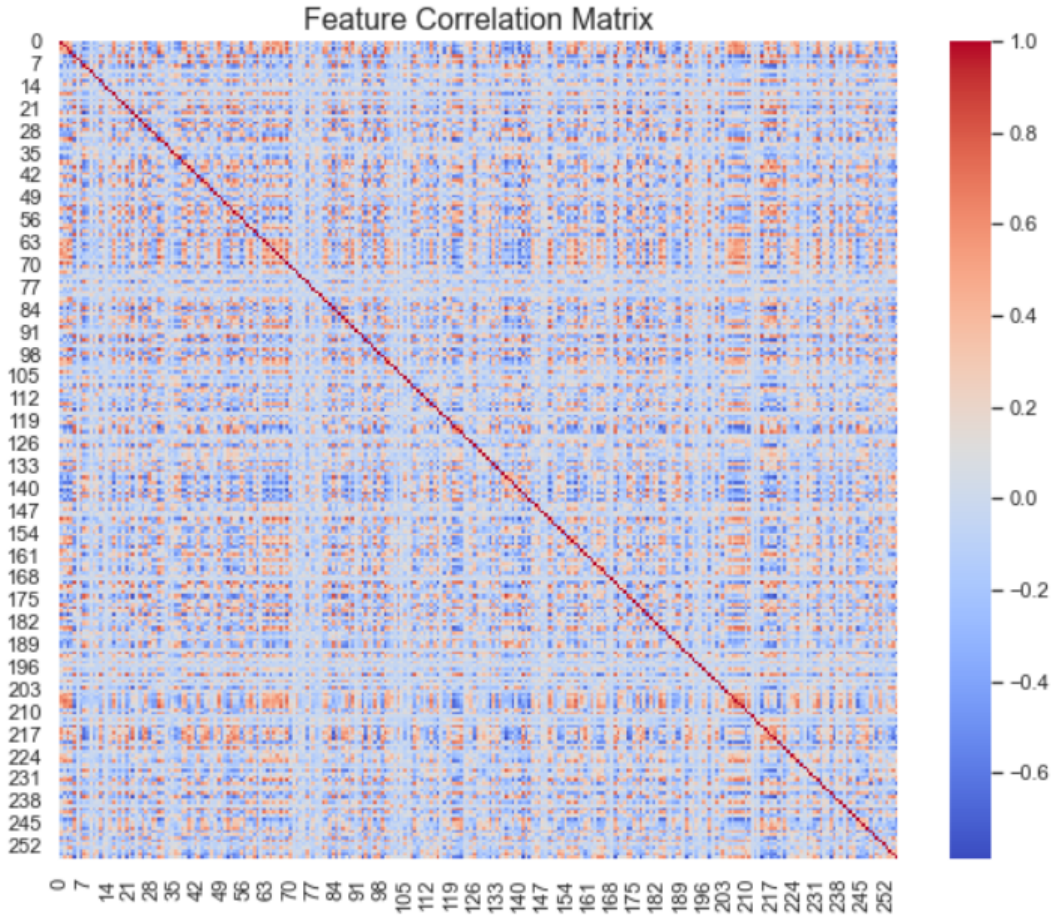

Figure 4.8: Feature Correlation Matrix for Weapon Detection

4.2.12 Key Features of YOLO v8n Object Detection

- **Fast detection:** YOLOv8n quickly finds and identifies objects in images.This speed makes it perfect for areas like security cameras,where quick responses are crucial.
- **Accurate Results:** The model is built to be very accurate when spotting objects.It minimizes mistakes,ensuring that it recognizes the correct items.
- **Detects Various Sizes:** YOLOv8n can detect both small and large objects in the same picture.This ability to handle different sizes is helpful in many situations.
- **Flexible Design:** Unlike older models that rely on fixed boxes,YOLOv8n uses a more adaptable method.This allows it to better fit various shapes and sizes of objects.
- **Advanced Learning Methods:** YOLOv8n employs modern training techniques to learn effectively from images.This makes it more dependable in identifying objects in different conditions.
- **Easy to Use:** Developers find it simple to work with YOLOv8n.It comes with clear guides and tools, making integration into projects hassle-free.
- **Wide Range of Uses:** YOLOv8n is versatile and can be applied in various fields,such as retail,traffic management,and safety. Its ability to identify different objects makes it useful in many industries.
- **Improved result refinement:** After detecting objects,YOLOv8n has smart ways to improve the output.This means it can filter out unnecessary detections for better-quality results.Narejo et al. (2021)

4.3 Implementation

4.3.1 Import necessary libraries and modules

```
import os
import cv2
import random
import numpy as np
import pandas as pd
import seaborn as sns
import tensorflow as tf
import matplotlib.pyplot as plt
import csv
import xml.etree.ElementTree as ET
```

In this code is an initial setup section, importing necessary libraries and modules for our weapon detection classification project.Now we breakdown of the code:

- **import os:**This imports a module named os which provides functions for interacting with the operating system, like navigating directories.
- **import cv2:**This imports a library called OpenCV, which is used for computer vision tasks like image processing and object detection.
- **import random:**This imports a module named random, which provides functions for generating random numbers and performing random selections.
- **import numpy as np:**This imports a numerical computing library called NumPy and aliases it as np for easier use in code.
- **import pandas as pd:**This imports a data manipulation library called Pandas and aliases it as pd for easier use in code.
- **import seaborn as sns:**This imports a data visualization library called Seaborn,which is used for creating statistical graphics.
- **import tensorflow as tf:** This imports TensorFlow,an open-source machine learning framework developed by Google, used for building and training machine learning models.
- **import matplotlib.pyplot as plt:**This imports a plotting library called Matplotlib,often used for creating visualizations in Python.

- **import csv:**The csv module provides functionality to read from and write to CSV (Comma Separated Values)files,which are commonly used for storing tabular data.
- **import xml.etree.ElementTree as ET:**ElementTree is a lightweight XML parser and API for parsing and creating XML data.It's part of Python's standard library under the xml module.

```
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing.image
import ImageDataGenerator
from tensorflow.keras.applications import NASNetMobile
from tensorflow.keras.layers
import Dense, Input, GlobalAveragePooling2D, Layer
from tensorflow.keras.optimizers import Adam
from sklearn.model_selection import train_test_split
from sklearn.metrics
import confusion_matrix, ConfusionMatrixDisplay
from xml.dom import minidom
from ultralytics import YOLO
from tensorflow.keras.callbacks
import EarlyStopping, ModelCheckpoint
```

This code section continues the setup by importing additional modules and libraries

- **from tensorflow.keras.models import Model:** This imports the Model class from the Keras API within TensorFlow, which is used for creating neural network models.
- **from keras.preprocessing.image import ImageDataGenerator:**This imports a class called ImageDataGenerator from the Keras library,used for generating batches of tensor image data with real-time data augmentation.
- **from tensorflow.keras.applications import NASNetMobile:**Keras Applications are pre-trained deep learning models ready for fine-tuning and feature extraction. NASNetMobile is one such model optimized for mobile and embedded devices.
- **from tensorflow.keras.layers import Dense, Input, GlobalAveragePooling2D, Layer:**Keras layers are the building blocks of neural networks.Each layer transforms input data in a specific way to extract features and learn patterns.

- **from tensorflow.keras.optimizers import Adam:**Adam(Adaptive Moment Estimation)is an optimization algorithm that adjusts the learning rate for each parameter dynamically.
- **from sklearn.model_selection import train_test_split:** train_test_split is a utility function that splits datasets into random train and test subsets.
- **from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay:**These tools from scikit-learn help in evaluating the performance of classification models by visualizing true vs. predicted labels.
- **from xml.dom import minidom:**minidom (Minimal DOM Implementation)is a lightweight XML parser that allows for the parsing and manipulation of XML documents.
- **from ultralytics import YOLO:**Ultralytics YOLO is an implementation of the "You Only Look Once" (YOLO) real-time object detection system.It provides tools for training, evaluating,and deploying YOLO models.
- **from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint:**Callbacks are utilities in Keras that allow you to monitor and control the training process.EarlyStopping and ModelCheckpoint are two commonly used callbacks.

4.3.2 Data Augmentation

```
train_datagen = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest'
)
```

Data augmentation is a powerful strategy to enhance our model's performance by artificially increasing the diversity and size of our training dataset. By carefully selecting and applying appropriate augmentation techniques, we can improve our model's ability to generalize to new, unseen data, ultimately leading to more robust and accurate predictions. Our ImageDataGenerator setup with parameters like rotation_range, width_shift_range, height_shift_range, shear_range, zoom_range, and horizontal_flip provides a solid foundation for augmenting our image data. However, when dealing with tasks involving bounding boxes, integrating augmentation libraries like Albumentations can offer more comprehensive and seamless handling of both image and annotation transformations.

```
img_dir='C:\\WeaponDetection\\images\\train'
train_image=os.listdir(img_train)
img_dir='C:\\WeaponDetection\\images\\val'
test_image=os.listdir(img_test)
```

Data Directory The variable DATA DIR (Data) specifies the root directory for the dataset named Data. Training and Testing Data Directories: (TRAIN DATA DIR) and (TEST DATA DIR) is created by joining DATA DIR with the train subdirectory. We use separate directories to ensure that our model is evaluated on unseen data, helping to assess its generalization performance.

4.3.3 Data Processing

```
def image_generator(image_dir,
        csv_file, batch_size=16, img_size=(300, 300)):
        with open(csv_file) as csvfile:
            rows = list(csv.reader(csvfile))[1:]
    while True:
        random.shuffle(rows)
        for i in range(0, len(rows), batch_size):
            batch_rows = rows[i:i + batch_size]
            img_batch, box_batch, label_batch = [], [], []

            for row in batch_rows:
                img_path = row[0]
                full_path = os.path.join(image_dir, img_path)
                img = cv2.imread(full_path)

                if img is not None:
                    img, box = preprocess_image(img,
                    [float(row[4]), float(row[5]),
                    float(row[6]), float(row[7])], img_size)

                    img = train_datagen.random_transform(img)
                    img_batch.append(img)
                    box_batch.append(box)
                    label_batch.append(int(row[3]))

            yield np.array(img_batch),
            np.array(box_batch), np.array(label_batch)

def preprocess_image(img, box, img_size=(300, 300)):
    img = cv2.resize(img, img_size).astype("float32") / 255.0
    box = [b / img_size[0] for b in box]
    return img, box
```

This code block seems to be part of a process to prepare training data for a deep learning model.Now we breakdown of what it does:

- The image_generator()function is designed to continuously provide batches of images,their normalized bounding boxes,and corresponding labels to a machine learning model.This generator loops through the dataset,shuffles the data for each epoch,and preprocesses images and bounding boxes.

- The preprocess_image()function handles the resizing and normalization of the image and bounding box,ensuring that all input images are uniformly processed and ready for training.

This code setup is commonly used in object detection tasks,where models are trained to predict bounding boxes and classify objects within images.The generator allows efficient memory management and ensures that data is processed in real-time during training.

4.3.4 Model

4.3.5 Custom Model with NASNetMobile

```
input_shape = (300, 300, 3)
base_model = NASNetMobile(weights='imagenet',
include_top=False, input_shape=input_shape)

for layer in base_model.layers[-10:]:
    layer.trainable = True

x = base_model.output
x = GlobalAveragePooling2D()(x)

# Use the custom attention layer
attention_output = CustomAttention()(x)

x = Dense(256, activation='relu')(attention_output)
x = Dense(3, activation='softmax')(x)

feature_extraction_model = Model(inputs=base_model.input,
outputs=x)
```

- Transfer Learning:Utilizing pre-trained models accelerates training and improves performance, especially with limited datasets.
- Fine-Tuning:Unfreezing specific layers allows the model to adapt pre-trained features to the new task without overfitting.
- Attention Mechanisms:Enhance feature representations by allowing the model to emphasize important regions within the input data.

- Custom Classification Head:Tailored fully connected layers convert extracted features into actionable predictions.

4.3.6 Pre-Trained YOLOv8 nano Model

```
model = YOLO('yolov8n.pt')
```

This specifies the YOLOv8 Nano model,a smaller,faster version optimized for resource-constrained environments like mobile or edge devices.The .pt file is a pre-trained model weight file,typically trained on a large dataset like COCO,which allows for fast deployment.This model can be used for tasks such as:

- Object detection (detecting objects in images or video)
- Instance segmentation (detecting objects with precise boundaries)
- Image classification (identifying what objects are present)

4.4 Evaluate the solution

4.4.1 Evaluation Metrics

Evaluation metrics explain the performance of the model.They are used to assess the performance and effectiveness of a model or algorithm in solving a particular task.Evaluation metrics can show the distinctions in performance of different machine learning and deep learning models due to their ability to differentiate between model outputs.These metrics are important as they offer insights into a model's performance quality,its precision in prediction,and its alignment with the particular objectives of a given task. The choice of evaluation metrics varies according to the specific machine-learning problem. The evaluation metrics commonly used to assess the performance of a model for image classification are precision,recall,F1 score,confusion matrix,and accuracy.

4.4.2 Precision Score

Precision plays an important role in binary classification tasks,as it assesses the accuracy of positive predictions made by a model.It is the ratio of correctly predicted positive samples to the total optimistic predictions generated by the model.Idakwo et al. (2024)

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (I)$$

4.4.3 Recall Score

Recall serves as a performance metric employed in binary classification tasks.Its purpose is to gauge a model's capacity to accurately pinpoint positive instances.It is the ratio of true positive to the total actual positives.It measures the model's ability to capture all of the positive classes. Idakwo et al. (2024)

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (II)$$

A notable Recall value suggests the model's adeptness at comprehensively capturing positive instances.This attribute proves particularly significant in scenarios featuring imbalanced datasets or situations where the cost of overlooking positive instances is substantial. A higher Recall score is desirable because it indicates the model is good at identifying positive cases.However,it's essential to note that a high Recall may coincide with a lower Recall and Precision often entails a trade-off.

4.4.4 F1 Score

The F1 score combines precision and recall into a single value.It is the harmonic mean of precision and recall that provides a balanced measure of a model's performance. Idakwo et al. (2024)

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (III)$$

It is a useful metric for assessing the effectiveness of classification models,especially in situations featuring imbalanced datasets where one class might significantly outweigh the others.As it combines precision and recall,it offers a well-rounded evaluation of a model's proficiency in recognizing positive samples while accurately reducing false positives and false negatives.It is the measure of a model's accuracy that takes into account both the model's ability to identify relevant data and the ability to include irrelevant data.In this context, precision stands as the proportion of true positive predictions relative to the total predicted positive instances, signifying the accuracy of positive predictions.On the other hand,recall is the proportion of correctly identified positive cases.It gives more weight to the lower of the two values.The F1-score,ranges between 0 and 1,with a higher score indicating a better performance.

4.4.5 Accuracy Score

Accuracy stands as a foundational evaluation metric for gauging the performance of classification models.It is the ratio of accurate predictions made by the model relative to the total number of predictions.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (IV)$$

The higher the accuracy, the better the model is performing. In order for the accuracy to not be misleading, the dataset needs to be balanced.

4.4.6 Confusion Matrix

The confusion matrix is a tabular representation of actual target and model's predictions including True Positive,True Negative.This evaluation metric is used to evaluate the performance of classification models.

		Predicted	
		Negative **(N)** -	Positive **(P)** +
Actual	Negative -	True Negative **(TN)**	**False Positive (FP) Type I Error**
	Positive +	**False Negative (FN) Type II Error**	True Positive **(TP)**

Figure 4.9: Confusion Matrix Analysis for Weapon Detection

Chapter 5

Result Analysis

5.1 YOLOv8n Model Visualization

5.1.1 Classification of Labeling

Visualize various statistical aspects of a dataset used in a classification or object detection task related to knives, pistols, and no_weapon.

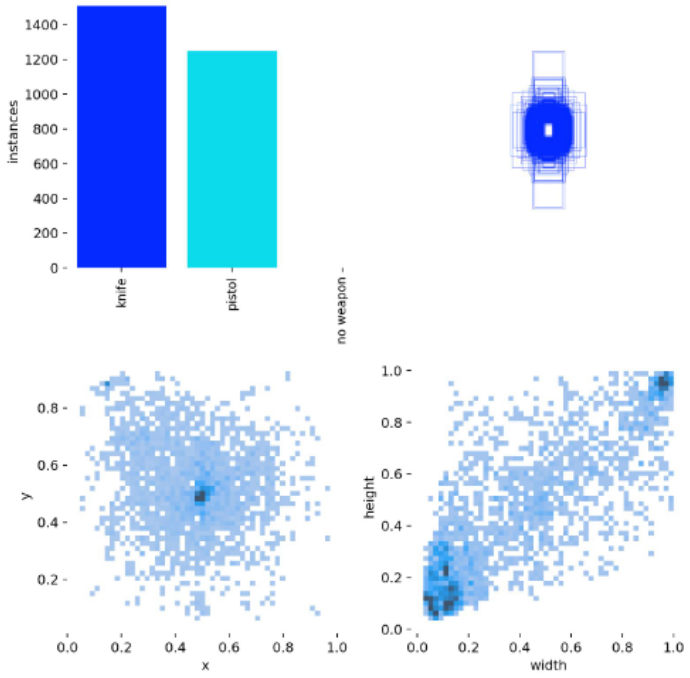

Figure 5.1: Classification of Labeling for Weapon Detection

Top-Left Bar Chart:
The chart indicates that there are more samples labeled as ”knife” than ”pistol” in the dataset, with around 1400 instances of knives and slightly fewer instances of pistols.

Top-Right Overlapping Bounding Boxes: The high overlap suggests that the bounding boxes for different objects (or the same object across different samples) tend to be clustered or centered around a similar region in the image.

Bottom-Left Scatter Plot (X vs. Y coordinates):
The scatter plot shows the distribution of the bounding box center points (possibly the centroid) for all detected objects. The points appear to be scattered with some clusters, indicating that objects are frequently detected in certain regions of the images.

Bottom-Right Scatter Plot (Width vs.Height):
This plot visualizes the relationship between the width and height of the bounding boxes. The clustering in specific regions suggests that certain sizes of bounding boxes (height and width combinations) are more common in the dataset. It shows that many of the objects are smaller in height and width (toward the lower-left region of the plot), while some have larger dimensions (upper right).

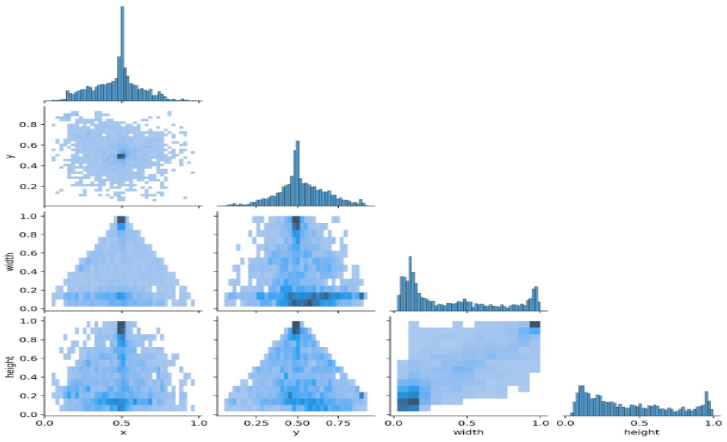

Figure 5.2: Label Classification Correlation Matrix for Weapon Detection

5.2 Model Performance for YOLO

The YOLOv8n model demonstrated better performance in weapon detection for the specified classes—knife, pistol, and no-weapon. The model achieved an overall accuracy of 96.40% on the test set, affirming its ability to generalize well to unseen data.

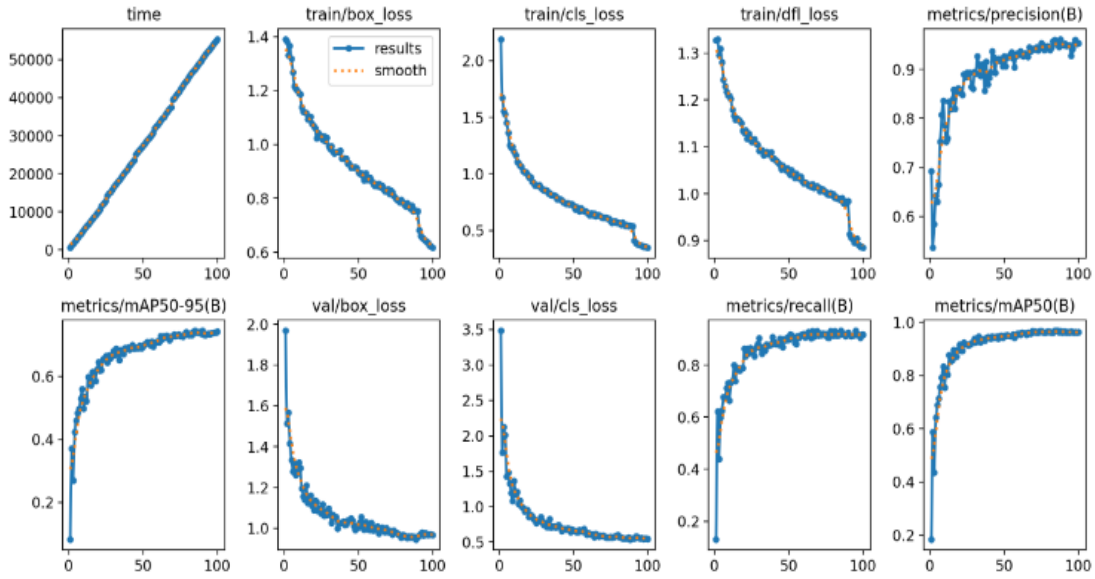

Figure 5.3: YOLOv8n Training and Validation Accuracy and Loss Curves

This graph shows the training progress of a YOLOv8n model with different metrics and losses plotted across training over 100 epochs(typically 0 to 100 on the x-axis): **Top row:** The progression of training time across epochs.The Bounding box loss during training(lower is better),Classification loss during training(how well the model classifies objects,lower is better), Distribution focal loss(for box regression),lower is better,Precision(positive predictive value) in bounding box predictions is higher is better.
Bottom row: Mean Average Precision(mAP) between 50% and 95% IoU thresholds for object detection,higher is better. Validation bounding box loss,measures generalization,validation classification loss,recall, indicating how well true positives are detected; higher is better,mAP at 50% IoU threshold, measuring detection accuracy at a common threshold(higher is better).

5.2.1 Confusion Matrix for YOLOv8n Object Detection

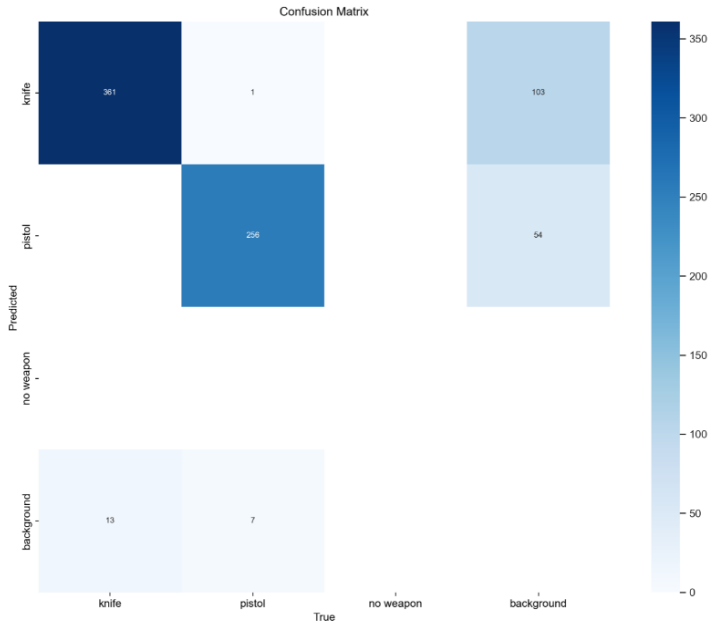

Figure 5.4: Confusion Matrix for YOLOv8n Object Detection

The confusion matrix shows the model has classified different categories(knife, pistol, no weapon,and background).Each cell represents the number of instances for which the actual label(true class) is on the x-axis and the predicted label is on the y-axis.Now we describe the matrix given below:
True Knife Predictions: The model predicted knife 361 times correctly(top-left cell).It incorrectly predicted knife as pistol 1 time (top-right of the first row).
True Pistol Predictions: The model predicted pistol correctly 256 times(middle-left cell).It incorrectly predicted pistol as knife 103 times(middle-right of the first row).
True "No Weapon" Predictions: It was correctly classified 54 times.There are significant misclassifications(such as knife, pistol predictions).
True Background Predictions: It predicted the background correctly 7 times but confused it for a weapon(knife and pistol) in 13 and 7 instances, respectively.
The color gradient helps visualize the magnitude of predictions for each class,with darker colors indicating higher values.

5.2.2 Normalized Confusion Matrix for YOLOv8n Object Detection

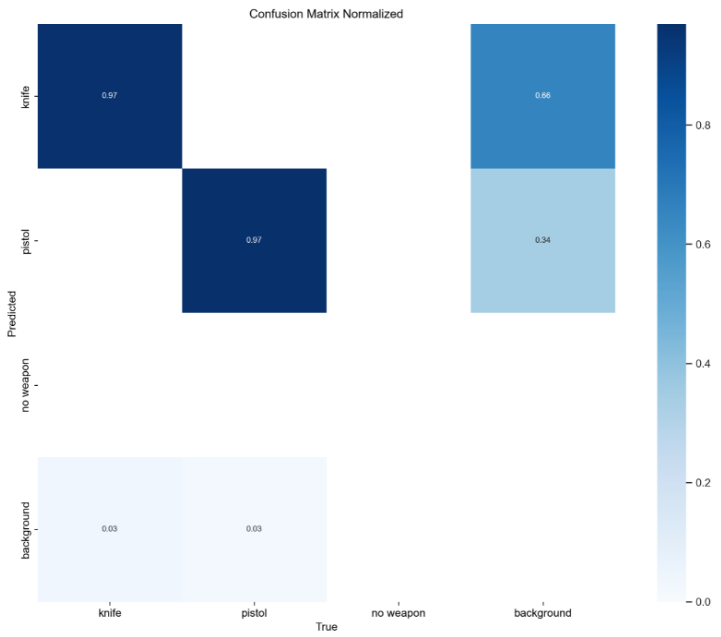

Figure 5.5: Normalized Confusion Matrix for YOLOv8n Object Detection

By normalizing the raw confusion matrix data, we can compare the frequency of accurate and wrong predictions for various groups.Model performance on imbalanced datasets may be assessed with the use of this normalized confusion matrix.The predicted accuracy is now explained as follows:
Knife prediction accuracy: The model correctly predicted the knife 97% of the time (0.97), showing that it learned to recognize the knife well.
Pistol prediction accuracy: The model correctly predicted pistols 97% of the time (0.97).
"No weapon" prediction accuracy: The model shows a significant drop here, with 66% correct predictions (0.66), and 34% incorrect predictions (0.34).
Background prediction accuracy: This appears to be the model's weakest point, with only 3% accuracy in predicting the background as the actual background (0.03), when predicting arms on background images.

5.2.3 P_curve, R_curve, PR_curve and F1-Confidence_Curve

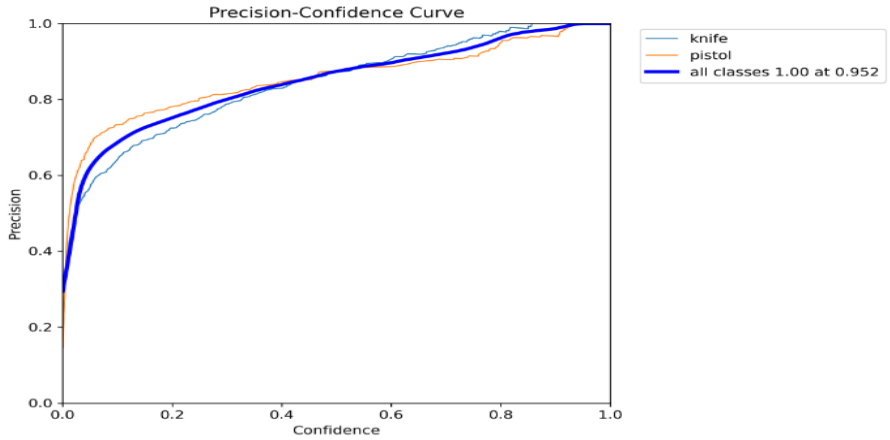

Figure 5.6: Precision-Confidence Curve for YOLOv8n Detection Performance

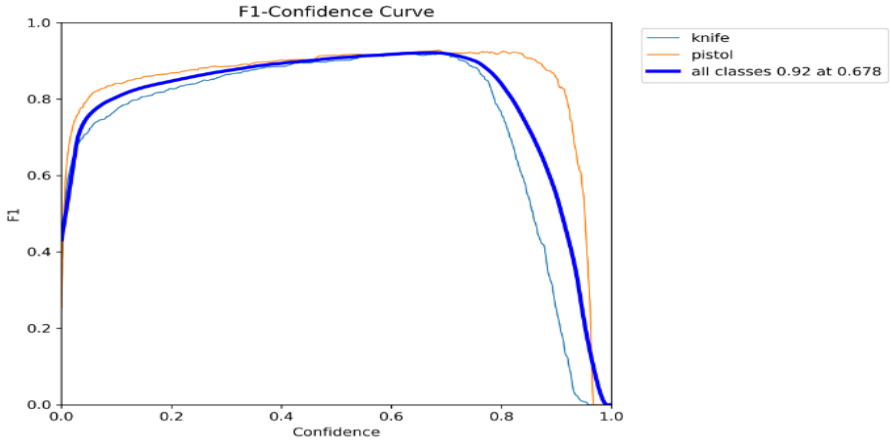

Figure 5.7: F1-Confidence Curve for YOLOv8n

5.2.4 Training Batch Output VS Testing Batch Output

Figure 5.8: Training Batch Visualization for YOLOv8n

Figure 5.9: Testing Batch Visualization for YOLOv8n

5.3 Model Performance for Custom Model with NASNetMobile

The Custom model demonstrated better performance in weapon detection for the specified classes—knife, pistol, and no-weapon. The model achieved an overall accuracy of 99.89% on the test set, affirming its ability to generalize well to unseen data.

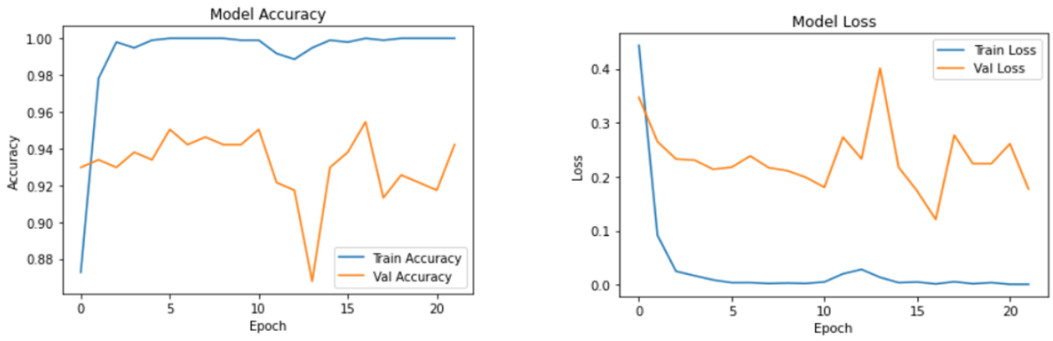

Figure 5.10: Accuracy and Loss Curves for Custom Model Training

This graph shows the custom model accuracy and loss based on training and validation data. The blue line represents the result of training data, and the orange line represents the result of validation data. The custom model accuracy is nearly 99.90%, which is based on classification. We can see the training accuracy or loss and validation accuracy and loss proportionally increasing and decreasing.

5.3.1 Confusion Matrix for Custom Model

The confusion matrix visually represents the model's predictions against actual labels. It helps in identifying patterns of misclassification and understanding which classes might be more challenging for the model. The heatmap shows strong diagonal elements, indicating accurate predictions, while off-diagonal elements highlight misclassifications.

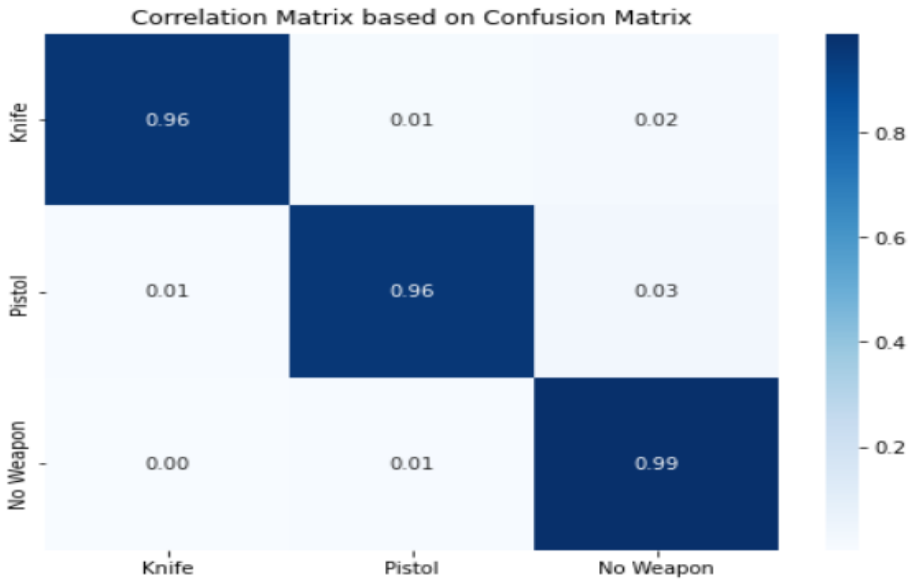

Figure 5.11: CorrelationMatrix based on ConfusionMatrix

The graph displays a correlation matrix for a classification problem with three classes: ”Knife,” ”Pistol,” and ”No Weapon.” It is based on a confusion matrix.

Correct classifications:
Knife (0.96): The model predicted the class ”Knife” correctly 96% of the time.
Pistol (0.96): The model predicted ”Pistol” correctly 96% of the time.
No Weapon (0.99): The model predicted ”No Weapon” correctly 99% of the time.

Misclassifications:
Knife vs. Pistol: The model almost always mixes up a knife and a pistol, as indicated by the 0.01 correlation between these two classes.
Knife vs. No Weapon: The correlation is 0.02; this means that sometimes the model interprets ”Knife” incorrectly as ”No Weapon.”
Pistol vs. Knife: With a 0.01 correlation, the model seldom ever confuses a knife for a handgun.
Pistol vs. No Weapon: The correlation coefficient is 0.03, indicating that ”Pistol” and ”No Weapon” are somewhat more frequently confused than other class pairings. The correlation between ”No Weapon” and ”Knife/Pistol” is very low (0.00 or 0.01), indicating that ”No Weapon” is rarely mistaken for ”Knife” or ”Pistol.”

5.4 After Merging Evaluation of Custom Model and YOLOv8n

5.4.1 Confidence Score Analysis for Model Predictions

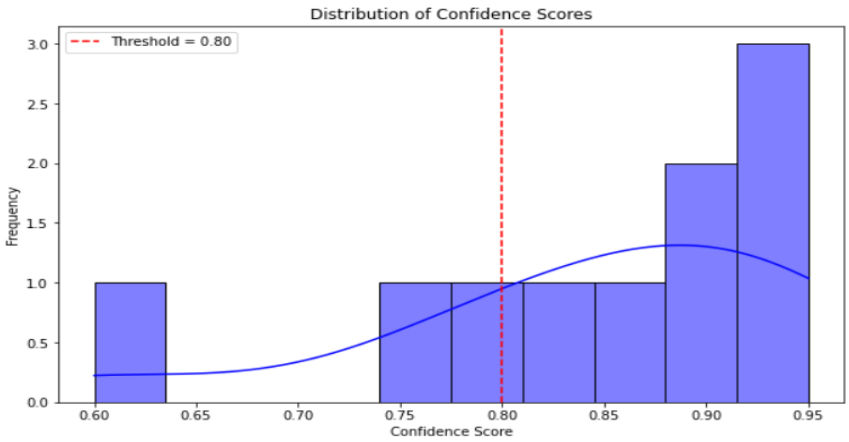

Figure 5.12: Confidence Score Analysis for Model Predictions

The distribution of the confidence ratings is seen in this graph.The frequency of prediction at each confidence score level and the confidence score, which ranges from 0.60 to 0.95, are shown on the X and Y axes.The threshold value of 0.80 is indicated by the red dashed line.The fact that nearly every predicted has a confidence score higher than 0.80 indicates that the model is fairly convinced about these predictions.By selecting a threshold value of 0.80, less dependable forecasts are weeded out by considering only those with a confidence score more than likely as well.

5.4.2 True and False Positives Analysis

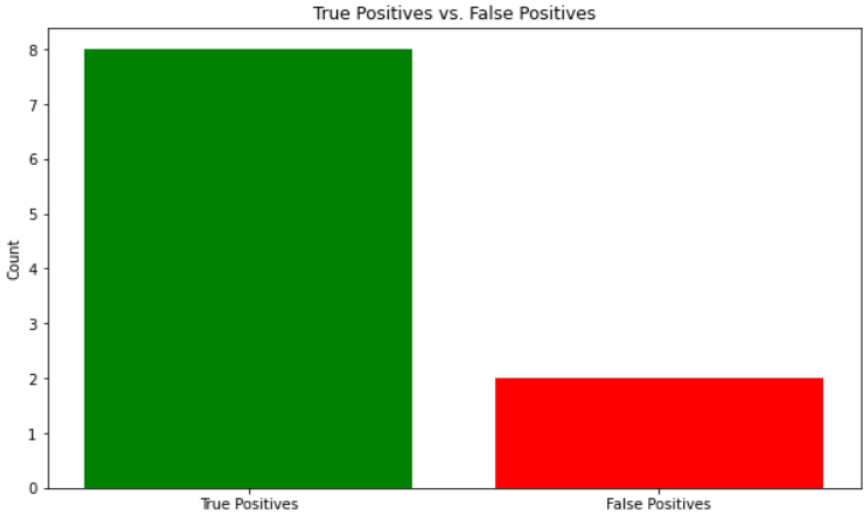

Figure 5.13: True and False Positives Analysis

This graph shows how many times the system correctly identified the object (knife or pistol) versus how many times it made a wrong prediction.

5.4.3 Final Confusion Matrix for Model Evaluation

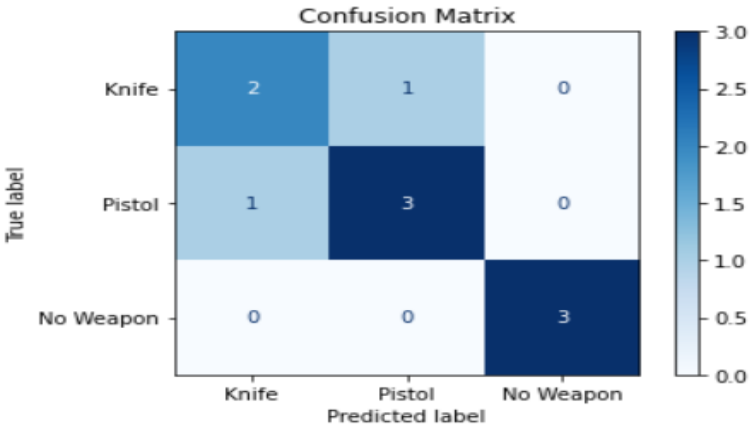

Figure 5.14: Final Confusion Matrix for Model Evaluation

The model's performance in classifying three categories—knife, pistol, and no weapon—is displayed in this matrix.The true labels, or Y-axis, display the real categories.The categories that the model predicts are displayed on the X-axis (Predicted Label).Two knives were found, however one was mistakenly identified as a handgun.Three pistols were found, however one was mistakenly identified as a knife.Three examples of No Weapon that are free of misclassification.

5.4.4 ROC Curves for multi-class classification

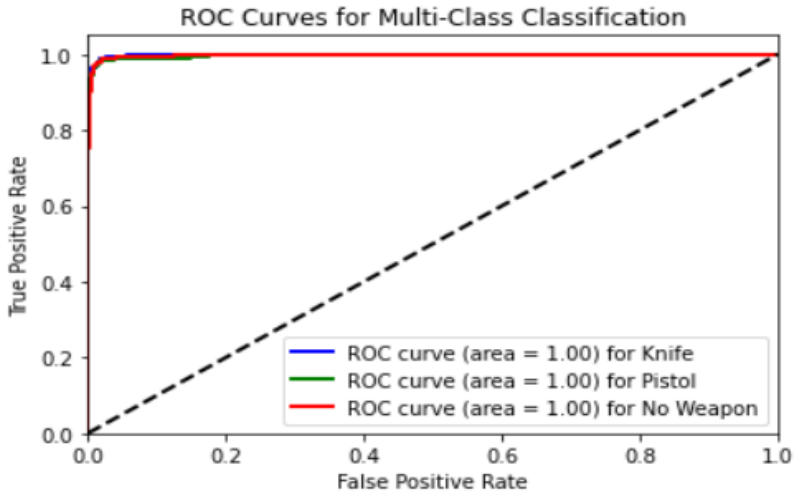

Figure 5.15: ROC Curves for multi-class classification

This graph shows the ROC Curves for multi-class classification Where the result for three classes included a knife, the ROC area result is 1.00, which means the knife class is perfectly classified, 1.00 for pistol, and 1.00 for No weapon. It means it perfectly classified all classes 100% accurately. Because we know the result ¡ 0.50 is misclassified, the result = 0.50 means it is averagely classified, and the result = 1.00 means perfectly classified.

5.4.5 Detected Objects in Image Output

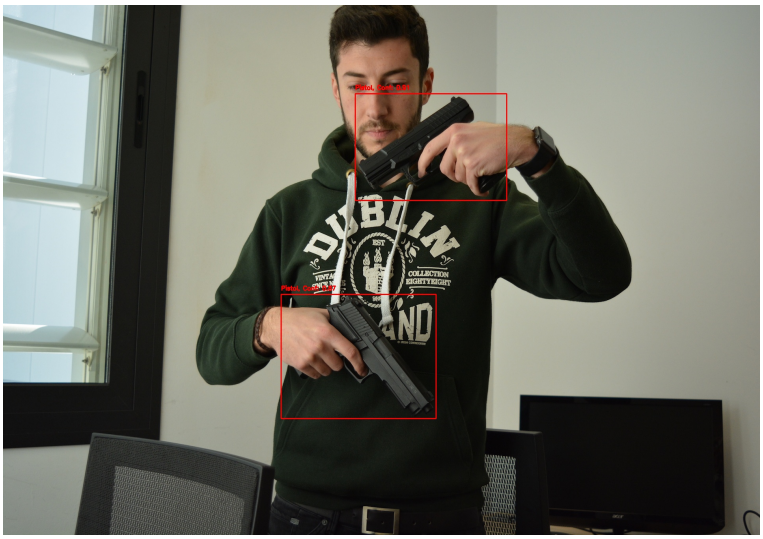

Figure 5.16: Detected Objects in Image Output

YOLOv8n model provides detection confidence (shown in the image with 0.88 and 0.91). Custom NASNetMobile model provides classification confidence (shown in the logs with 0.99 and 0.94). After merging both models, Yolov8n gives a 0.88 and 0.91 confidence score for two pistols in the image. On the contrary, our custom model gives 0.99 and 0.94 confidence scores for two pistols in the image.

5.4.6 Result of Comparison analysis

Table 5.1: Result of Comparison Table for Weapon Detection

Authors and Works	Accuracy (%)	F1 Score (%)	Precision (%)	Recall (%)
Custom Model + YOLOv8n	99.89	96.48	96.43	95.43
Akhila & Ahmed (2024) - YOLOv5	-	78.00	80.52	76.10
Ruprah & Shrivastava (2022) - YOLOv3	-	93	92	95
Vijayakumar et al. (2023) - YOLOv4	-	82.8	90.2	78
Das & Tomar (2022) - YOLO + RCNN	96.26	-	-	-
Kaya et al. (2021)	98.40	92.89	99.28	95.97
Galab et al. (2021)	96.95	98.42	100	96.80
Gelana and yadav(2019)	97.78	96.76	99.45	94.21

Comparison of Different Research Models Applied for Object Detection (On the Basis of Performance metrics like Accuracy, F1 Score, Precision, and Recall) Every row presents the work of other authors, what they do, and therefore what happens. But still, let's take an example of Custom Model+YOLOv8n—Accuracy Accuracy (99.89%), F1 Score, Precision, and Recall come around 96.48% and 95.43%, respectively. So, this specific combination, which uses YOLOv8n and the relevant custom weights, gives a performance that has minimum false positive negative results. With the metric achieving: F1 score Requirements: YOLOv5 = 78% and In this case,Akhila & Ahmed (2024) This is not a very good result: if the model can detect/recognize objects, it makes another mistake rare — It will completely confuse them with others. Ruprah & Shrivastava (2022) claimed an F1 score of 93%, precision of 92%, and recall of 95% using a YOLOv3 model. A classic model with reasonable scores, especially in precision-recall trade-offs. In contrast,Vijayakumar et al. (2023) YOLOv4-based detection had an 82.8% F1 Score, Attribute-level Precision of 90.2 %, and a Recall performance of 78 %. While the shown score is highly accurate, there are only 1/5 objects, which means that a lot of false negatives would be present

in this model. Das & Tomar (2022) YOLO + RCNN, 96.26% Accuracy While this offers a positive view from a performance perspective, it is hard to conclude the bare utility of the system as other metrics like F1 score are missing. Kaya et al. (2021) got an accuracy of 98.40%, an F1 score of 92.89%, a precision of 99.28%, and a recall of 95.97%. Even though it does have high precision performance (most of the detections found are correct), The work by Galab et al. (2021) impressed with an Accuracy of 96.95%, a very high F1 Score of 98.42%, Precision of 100%, and Recall of 96.80%. Perfect precision means no false positives, and combined with high recall, it reflects excellent overall performance. Lastly, (Gelana & Yadav (2019)) achieved an accuracy of 97.78%, an F1 score of 96.76%, a precision of 99.45%, and a recall of 94.21%. The model excels with high precision and balanced recall, resulting in solid accuracy and F1 Score.

Chapter 6

Conclusions

6.0.1 Societal, health, safety, legal and cultural aspects:

The development of a weapon detection system utilizing NASNetMobile for feature extraction and as the backbone is a complex task that involves various societal, health, safety, legal, and cultural aspects. These aspects include improving public safety, fostering trust and acceptance, the risk of discrimination, ensuring equity in deployment, health risks from excessive reliance on technology, enhancing safety measures, complying with privacy laws, protecting data, accountability, ethical and legal responsibilities, regulating AI and surveillance, ensuring consent and transparency, cultural sensitivity, considering religious and ethnic factors, public perceptions of surveillance, educational impacts, and ethical considerations.Bhatti et al. (2021b)

Implementing weapon detection systems in public areas can significantly enhance safety by detecting dangerous weapons early. However, public trust and acceptance depend on how these systems are implemented; transparency and unbiased data training are essential. There are concerns about discriminatory profiling, particularly in regions where specific ethnic or cultural groups might be unfairly targeted.NARASIMMAN et al. (2022).

Equitable implementation of weapon detection technology in various social functions is crucial to prevent growing disparities. Excessive reliance on technology without appropriate human intervention can be provided. Creates a false sense of security and slows response times in emergency situations. Real-time alerting and quick response are essential to minimize damage in critical situations. Legal

considerations include compliance with privacy laws, ensuring data protection and accountability, ethical and legal liability, and public transparency regarding the operations of the maintained system.

This addition not only increases the comprehensiveness of our model but also helps address an important aspect of safety and security concerns. Through extensive testing and fine-tuning, we look forward to building lighter and faster models. By doing so, we aim not only to improve the performance and accuracy of our model but also to extend its applicability to a variety of real-world scenarios. More et al. (2024)

6.0.2 Impact on Environment and Sustainability:

Image classification using NASNetMobile for feature extraction and YOLOv8 for object detection in the weapon detection system resulted in relatively low environmental impact compared to resource-intensive models. These models can strike a balance between accuracy and computational efficiency, resulting in lower energy consumption. Their lightweight design reduces reliance on high-end GPUs and minimizes cooling requirements.

Both NASNetMobile and YOLOv8 are optimized for limited devices, enabling deployment on smaller devices that consume less power. This results in lower initial hardware production costs and reduced energy use over time, which also helps decrease carbon emissions related to network activity and supports the sustainability of mobile-first solutions.

The system's sustainability is enhanced by its ability to operate on low-power devices, decreasing the need for large data centers and extending the lifespan of the hardware. Moreover, it can function on affordable, energy-efficient devices, which lowers the necessity for frequent hardware upgrades. Due to public datasets providing insufficient information on our scope of weapon detection in images, we present an image-based deep learning approach for small arms object detection aimed at pistols and knives.Hnoohom et al. (2022)

Training costs are minimized because these smaller, efficient models use less energy during the tr aining process. Additionally, utilizing pre-trained models reduces environmental impact by skipping or shortening the training phase.

Incorporating renewable energy sources can further improve the sustainability of weapon detection systems. Low-power models like NASNetMobile paired with YOLOv8n are well-suited for deployment in remote locations , ensuring a minimal environmental footprint. By integrating these systems with renewable energy solutions, their environmental and sustainability benefits can be significantly enhanced.

6.0.3 Ethical and professional principles:

Weapon detection systems need to be designed to eliminate bias and discrimination. This means training models on varied datasets to ensure inclusivity and tackle any algorithmic bias. Safety and security are major concerns for today's modern world. For a country to be economically strong, it helps to attract investors and ensure a safe environment for the people. Bhatti et al. (2021b) The system should be fair and not favor any specific demographic group, and there should be clear accountability for the decisions made by the system.

- Transparency and interpretability are essential: the system must make sense and provide reasons for its decisions, especially when identifying potential threats. Emphasis on privacy and data protection should be imperative. This means ensuring compliance with privacy laws and obtaining informed consent by only collecting and processing necessary information.
- Safety and security are very important aspects of developing an application that uses different models to identify weapons. It plays an important role in reducing false positives and false negatives to avoid panic, accidents, or unnecessary interventions. Strong cyber security measures should be in place to protect against unauthorized access, hacking, or data breaches.
- Human rights must be respected and privacy must be protected. A balance must be struck between freedom from discrimination. The system should target individuals or groups based on protected characteristics.

- Ethical use of technology helps to increase security by improving a surveillance-busy environment and creating a safe environment. Weapons detection systems should only be used in situations where there are large gatherings of people, in less sensitive areas, where detection is required later.
- Professional responsibilities include due diligence and validation. This technology is constantly improving and helping in various fields. Ensures informed decision-making and ethical practice s by engaging employees and practicing ethica l leadership.

6.0.4 Brief Summary:

The system utilizes real-time static images to identify potential threats, such as weapons. It employs the NASNetMobile model for feature extraction, creating detailed feature maps that highlight important patterns. These feature maps are then processed through the backbone of the system, improving its accuracy. The system uses bounding box predictions to determine the weapon's location and categorizes it based on established classifications. If a weapon is detected with a confidence score above a set limit, alerts are Activated. Additionally, the system records details about the detected weapon's location, type, and other relevant information for further examination. It can be readjusted or trained to improve its performance.

6.0.5 Future works:

In future developments, the weapon detection system can be significantly improved to improve the performance of security personnel. This includes requiring threat database extensions with classes to cover a wide variety of weapons and scenarios and improving detection accuracy. Aggregates real-time environmental data to assist in timely threat assessment. This enables the creation of a mobile application for easy access, so security teams can access and monitor alerts directly from their smartphones. Additionally, features for user feedback and community engagement can be included.

1. Improvement of Detection Accuracy:

- More powerful models such as Convolutional Neural Network (CNN) or Vision Transformer (VIT) use high-quality models to increase detection power and work with datasets to enhance datasets.
- to improve decision-making and reduce false positives, avoid false news, communicate relevant information such as behavior analysis and include environmental factors.

2. Real-Time Performance Enhancement:

- Model Optimization: Using techniques such as quantization and trimming to streamline the model. This makes the model more efficient, makes it faster to edge device scenarios, and allows better results.
- Edge computing: Using the edge like smart cameras or drones enables real-time detection without relying on centralized cloud systems. Real-time detection is easy and accurate in a smart way. Apply weapon detection to devices like smart cameras or drones.

3. Robustness Against Diverse Conditions:

- Handling occlusion and variation: in different light conditions, weapon size, shape, angle, or partially hidden weapon detection model. All types of data should have the ability to increase.
- Adversarial Defense: Implementing methods to protect the system from adversary attacks. Where slight changes in the image can fool the detector and Can detect any image.

4. Improved Dataset Curation:

- Diverse and Larger Datasets: Create larger, more diverse, imformatic datasets with labeled videos and images of different types of weapons in different scenarios, shapes and sizes and environments to improve model training where detection of all types of weapons is possible.
- Synthetic Data Generation: Rare or challenging weapon detection scenarios require the use of techniques such as Generative Adversarial Networks (GAN) to generate synthetic data that enrich the dataset without the limitations of real-world data.

References

Ahmed, S., Bhatti, M. T., Khan, M. G., Lövström, B., & Shahid, M. (2022). Development and optimization of deep learning models for weapon detection in surveillance videos. *Applied Sciences*, *12*(12), 5772.

Akhila, K., & Ahmed, K. R. (2024). Real time deep learning weapon detection techniques for mitigating lone wolf attacks. *arXiv preprint arXiv:2405.14148*.

Bhatti, M. T., Khan, M. G., Aslam, M., & Fiaz, M. J. (2021a). Weapon detection in real-time cctv videos using deep learning. *IEEE Access*, *9*, 34366-34382. doi: 10.1109/ACCESS.2021.3059170

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

Chunchwar, P., Shelare, U., Nagpure, A., Patil, R., Dhole, D., & Shete, R. M. (n.d.). Real time weapon detection using yolov8 and alert mechanism.

Das, P. A. K., & Tomar, D. S. (2022). Convolutional neural networks based weapon detection: a comparative study. In *Fourteenth international conference on machine vision (icmv 2021)* (Vol. 12084, pp. 351–359).

Galab, M. K., Taha, A., & Zayed, H. H. (2021). Adaptive technique for brightness enhancement of automated knife detection in surveillance video with deep learning. *Arabian Journal for Science and Engineering*, *46*(4), 4049–4058.

Gelana, F., & Yadav, A. (2019). Firearm detection from surveillance cameras using image processing and machine learning techniques. In *Smart innovations in communication and computational sciences: Proceedings of icsiccs-2018* (pp. 25–34).

Ghazal, M., Waisi, N., & Abdullah, N. (2020). The detection of handguns from live-video in real-time based on deep learning. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *18*(6), 3026–3032.

González, J. L. S., Zaccaro, C., Álvarez-García, J. A., Morillo, L. M. S., & Caparrini, F. S. (2020). Real-time gun detection in cctv: An open problem. *Neural networks*, *132*, 297–308.

Hnoohom, N., Chotivatunyu, P., & Jitpattanakul, A. (2022). Acf: an armed cctv footage dataset for enhancing weapon detection. *Sensors*, *22*(19), 7158.

Idakwo, M. A., Yoro, R. E., Achimugu, P., & Achimugu, O. (2024). An improved weapons detection and classification system. *Journal of Network and Innovative Computing*, *12*, 10–10.

Jain, H., Vikram, A., Mohana, Kashyap, A., & Jain, A. (2020). Weapon detection using artificial intelligence and deep learning for security applications. In *2020 international conference on electronics and sustainable communication systems (icesc)* (p. 193-198). doi: 10.1109/ICESC48915.2020.9155832

Kaya, V., Tuncer, S., & Baran, A. (2021). Detection and classification of different weapon types using deep learning. *Applied Sciences*, *11*(16), 7535.

Maddileti, T., Sirisha, J., Srinivas, R., Saikumar, K., et al. (2022). Pseudo trained yolo r_cnn model for weapon detection with a real-time kaggle dataset. *International Journal of Integrated Engineering*, *14*(7), 131–145.

Mehta, P., Kumar, A., & Bhattacharjee, S. (2020). 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).

More, P., Patil, S., & Pattanshetti, T. (2024). Real-time violence and weapon detection and alert system.

NARASIMMAN, D. S., SATHVIKA, N., NIKETHA, P., KIRAN, R. S., & SREE, P. U. (2022). Weapon detection using artificial intelligence and deep learning for security applications.

Narejo, S., Pandey, B., Esenarro Vargas, D., Rodriguez, C., & Anjum, M. R. (2021). Weapon detection using yolo v3 for smart surveillance system. *Mathematical Problems in Engineering*, *2021*(1), 9975700.

Omiotek, Z., & Zhunissova, U. (2024). Dangerous items detection in surveillance camera images using faster r-cnn..

Ruprah, T. S., & Shrivastava, H. (2022). Comparison of various model of deep learning in weapons detection. *Journal of Pharmaceutical Negative Results*, 5759–5766.

Suryavanshi, A., Mehta, S., Bordoloi, D., Manwal, M., & Jain, V. (2024). Innovative defense mechanisms: Multi-class weapon detection with hybrid cnn-svm intelligence. In *2024 2nd world conference on communication & computing (wconf)* (pp. 1–6).

Talib, M., & Saud, J. H. (2024). A multi-weapon detection using deep learning. iraqi journal of information and communication technology. In *2024 iraqi journal of information and communications technology(ijict)* (pp. 11–22). doi: 10.31987/ijict.7.1.242

Tram, N. T. K., Son, D. T., & Võ Thái, A. (2023). Weapon detection using deep learning. In *Proceedings of the 12th international symposium on information and communication technology* (pp. 101–109).

Varshney, P., Tyagi, N. K. H., Lohia, A. K., & Girdhar, P. (2021). A deep learning based approach to detect suspicious weapons. *Proceedings http://ceur-ws. org ISSN*, *1613*, 0073.

Vieira, J. C., Sartori, A., Stefenon, S. F., Perez, F. L., De Jesus, G. S., & Leithardt, V. R. Q. (2022). Low-cost cnn for automatic violence recognition on embedded system. *IEEE Access*, *10*, 25190–25202.

Vijayakumar, K., Pradeep, K., Balasundaram, A., & Dhande, A. (2023). R-cnn and yolov4 based deep learning model for intelligent detection of weaponries in real-time video. *Mathematical Biosciences and Engineering*, *20*(12), 21611–21625.

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